MITSUBISHI

三菱可程式控制器

MELSEC Q 系列　MELSEC L 系列

MELSEC-Q/L
編程手冊
公共指令篇

QSERIES
LSERIES

● 安全注意事項 ●

（使用之前請務必閱讀）

在使用 MELSEC-Q 系列、MELSEC-L 系列可程式控制器之前，應仔細閱讀各產品附帶的手冊以及附帶手冊中介紹的關聯手冊，同時在充分注意安全的前提下正確地操作。

請妥善保管產品附帶手冊以備需要時閱讀，並應將本手冊交給最終用戶。

● 關於產品的應用 ●

(1) 使用三菱可程式控制器時，請符合以下條件：
即使可程式控制器出現問題或故障時，也不會導致重大事故。並且在設備外部以系統性規劃，當發生問題或故障時的備份或故障安全防護功能。

(2) 三菱可程式控制器是以一般工業等用途爲對象，設計和製造的泛用產品。因此，三菱可程式控制器不可用於以下設備、系統等特殊用途。
如果用於以下特殊用途時，對於三菱可程式控制器的品質、性能、安全等所有相關責任（包括，但不限定於債務未履行責任、瑕疵擔保責任、品質保證責任、違法行爲責任、製造物責任），三菱電機將不負責。
· 各電力公司的核能發電廠以及其他發電廠等，對公衆有較大影響的用途。
· 各鐵路公司及公家機關等，對於三菱電機有特別的品質保證體制之架構要求的用途。
· 航空宇宙、醫療、鐵路、焚燒、燃料裝置、乘載移動設備、載人運輸裝置、娛樂設備、安全設備等，預測對性命、人身、財産有較大影響的用途。
但是，即使是上述對象，只要有具體的限定用途，沒有特殊的品質（超出一般規格的品質等）要求之條件下，經過三菱電機的判斷依然可以使用三菱可程式控制器，詳細情形請洽詢當地三菱電機代表窗口。

修訂記錄

※本手冊號在封底的左下角。

印刷日期	※手冊編號	修改內容
2012 年 04 月	SH(NA)-081065CHT-A	第一版

日文手冊原稿： SH(NA)-080804-L

本手冊不授予工業產權或任何其他類型的權利，也不授予任何專利許可。三菱電機對由於使用了本手冊中的內容而引起的涉及工業產權的任何問題不承擔責任。

© 2012 MITSUBISHI ELECTRIC CORPORATION

前言

本手冊“MELSEC-Q/L 編程手冊（公共指令篇）”介紹進行 QCPU、LCPU 編程時需要的公共指令有關內容。

公共指令是指，除 QJ71C24N、QJ71E71-100 等智慧功能模組專用指令、PID 控制專用指令、SFC 專用指令、ST 專用指令、套接字通信功能專用指令、觸發記錄指令、LCPU 的定位功能專用指令・計數器功能專用指令以外的指令。

在使用之前應熟讀本手冊及關聯手冊，在充分瞭解 Q 系列、L 系列可程式控制器的功能・性能的基礎上正確地使用本產品。

將本手冊中介紹的程式示例引用到實際系統中時、應充分驗證物件系統中不會有控制方面的問題。

■物件 CPU 模組

CPU 模組	型號
基本型 QCPU	Q00JCPU、Q00CPU、Q01CPU
高性能型 QCPU	Q02CPU、Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU
過程 CPU	Q02PHCPU、Q06PHCPU、Q12PHCPU、Q25PHCPU
冗餘 CPU	Q12PRHCPU、Q25PRHCPU
通用型 QCPU	Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q10UDHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU、Q03UDECPU、Q04UDEHCPU、Q06UDEHCPU、Q10UDEHCPU、Q13UDEHCPU、Q20UDEHCPU，Q26UDEHCPU、Q50UDEHCPU、Q100UDEHCPU
LCPU	L02CPU、L26CPU-BT、L02CPU-P、L26CPU-PBT

目錄

安全注意事項 1
關於產品的應用 2
修訂記錄 3
前言 4
目錄 5
手冊體系 17

1 概要 19

1.1 相關編程手冊 19
1.2 本手冊中使用的總稱 / 略稱 23

2 指令一覽表 25

2.1 指令分類 25
2.2 指令一覽表的閱讀方法 26
2.3 順控程式指令 28
2.3.1 觸點指令 28
2.3.2 連接指令 29
2.3.3 連接指令 30
2.3.4 移位元指令 30
2.3.5 主控指令 31
2.3.6 結束指令 31
2.3.7 其他指令 31
2.4 基本指令 32
2.4.1 比較運算指令 32
2.4.2 算術運算指令 38
2.4.3 資料轉換指令 42
2.4.4 資料傳送指令 44
2.4.5 程式分支指令 46
2.4.6 程式執行控制指令 47
2.4.7 I/O 刷新指令 47
2.4.8 其他使用方便的指令 47
2.5 應用指令 49
2.5.1 邏輯運算指令 49
2.5.2 旋轉指令 51
2.5.3 移位元指令 52
2.5.4 位元處理指令 53
2.5.5 資料處理指令 54
2.5.6 結構化指令 57
2.5.7 資料表操作指令 59
2.5.8 緩衝記憶體訪問指令 59
2.5.9 顯示指令 60
2.5.10 調試・故障診斷指令 60
2.5.11 字串處理指令 61
2.5.12 特殊函數指令 64
2.5.13 資料控制指令 67
2.5.14 切換指令 69
2.5.15 時鐘指令 69

2.5.16 擴展時鐘指令 72
2.5.17 程式控制指令 73
2.5.18 其他指令 73
2.6 資料鏈接用指令 76
2.6.1 網路刷新指令 76
2.6.2 路由資訊的讀取 / 登錄指令 76
2.7 多 CPU 專用指令 77
2.7.1 至自站 CPU 共用記憶體的寫入指令 77
2.7.2 對其他站 CPU 共用記憶體的讀取指令 77
2.8 多 CPU 高速通信專用指令 78
2.8.1 多 CPU 高速通信專用指令 78
2.9 冗餘系統指令 (用於冗餘 CPU) 78
2.9.1 冗餘系統指令 (用於冗餘 CPU) 78

3 指令構成 79

3.1 指令構成 79
3.2 資料的指定方法 80
3.2.1 使用位元資料時 80
3.2.2 使用字 (16 位元) 資料時 81
3.2.3 使用雙字資料 (32 位元) 時 82
3.2.4 使用實數資料時 85
3.2.5 使用字串資料時 87
3.3 變址修飾 88
3.4 間接指定 96
3.5 縮短指令處理時間 98
3.5.1 子集處理 98
3.5.2 使用通用運算寄存器 (Z) 的運算處理 (只對於通用型 QCPU、LCPU) 99
3.6 編程注意事項 100
3.7 指令執行條件 105
3.8 計算步數 106
3.9 使用同一軟元件的 OUT 指令、SET/RST 指令或 PLS/PLF 指令時的動作 111
3.10 使用文件寄存器時的注意事項 115

4 如何閱讀指令 118

5 順控程式指令 120

5.1 觸點指令 120
5.1.1 LD、LDI 運算開始 120
AND、ANI 串列連接 120
OR、ORI 並行連接 120
5.1.2 LDP、LDF 脈衝運算開始 122
ANDP、ANDF 脈衝串列連接 122
ORP、ORF 脈衝並行連接 122

5.1.3 LDPI、LDFI 脈衝否運算開始 124
ANDPI、ANDFI 脈衝否串列連接 124
ORPI、ORFI 脈衝否並行連接 124
5.2 連接指令 126
5.2.1 ANB 梯形圖塊串列連接 126
ORB 梯形圖塊並行連接 126
5.2.2 MPS 運算結果入棧 127
MRD 運算結果讀取 127
MPP 運算結果退棧 127
5.2.3 INV 運算結果取反 130
5.2.4 MEP、MEF 運算結果脈衝化 131
5.2.5 EGP、EGF 變址繼電器運算結果的脈衝化 132
5.3 輸出指令 134
5.3.1 OUT 輸出指令（除計時器、計數器、報警器以外）....134
5.3.2 OUT T 低速計時器 136
OUTH T 高速計時器 136
OUT ST 低速累計計時器 136
OUTH ST 高速累計計時器 136
5.3.3 OUT C 計數器 139
5.3.4 OUT F 報警器輸出 140
5.3.5 SET 軟元件的設置（報警器除外）....142
5.3.6 RST 軟元件的重定（報警器除外）....143
5.3.7 SET F 報警器的設置 145
RST F 報警器的重定 145
5.3.8 PLS 上升沿輸出 146
PLF 下降沿輸出 146
5.3.9 FF 位軟元件輸出取反 148
5.3.10 DELTA、DELTAP 直接輸出的脈衝化 149
5.4 移位元指令 151
5.4.1 SFT、SFTP 位軟元件移位 151
5.5 主控指令 153
5.5.1 MC 主控的設置 153
MCR 主控的重定 153
5.6 結束指令 157
5.6.1 FEND 主程式的結束 157
5.6.2 END 順控程式的結束 159
5.7 其他指令 161
5.7.1 STOP 順控程式停止 161
5.7.2 NOP、NOPLF、PAGE n 無處理 162

6 基本指令 166

6.1 比較運算指令 166
6.1.1 =、< >、>、<=、<、>= 166
6.1.2 D=、D< >、D>、D<=、 167
D<、D>=

6.1.3 E=、E< >、E>、E<=、E<、E>= 浮點數據比較 （單精確度）.....169
6.1.4 ED=、ED< >、ED>、ED<=、ED<、ED>= 浮點數據比較（雙精度）..... 171
6.1.5 $=、$< >、$>、$<=、$<、$>= 字串資料比較.....173
6.1.6 BKCMP □、BKCMP □ P BIN16 位元塊資料比較.....175
6.1.7 DBKCMP □、DBKCMP □ P BIN32 位元塊資料比較.....178
6.2 算術運算指令 181
6.2.1 +、+P、-、-P BIN16 位加法和減法運算.....181
6.2.2 D+、D+P、D-、D-P BIN32 位加法和減法運算.....184
6.2.3 *、*P、/、/P BIN16 位乘法和除法運算.....187
6.2.4 D*、D*P、D/、D/P BIN32 位乘法和除法運算.....188
6.2.5 B+、B+P、B-、B-P BCD4 位加法和減法運算.....190
6.2.6 DB+、DB+P、DB-、DB-P BCD8 位加法和減法運算.....193
6.2.7 B*、B*P、B/、B/P BCD4 乘法和除法運算.....196
6.2.8 DB*、DB*P、DB/、DB/P BCD8 乘法和除法運算.....198
6.2.9 E+、E+P、E-、E-P 浮點數據的加法和減法運算（單精確度）.....200
6.2.10 ED+、ED+P、ED-、ED-P 浮點數據的加法和減法運算（雙精度）.....204
6.2.11 E*、E*P、E/、E/P 浮點數據的乘法和除法運算（單精確度）.....208
6.2.12 ED*、ED*P、ED/、ED/P 浮點數據的乘法和除法運算（ 雙精度）.....210
6.2.13 BK+、BK+P、BK-、BK-P BIN16 位資料塊加法和減法運算.....212
6.2.14 DBK+、DBK+P、DBK-、DBK-P BIN32 位資料塊加法和減法運算.....214
6.2.15 $+、$+P 字串的合併 217
6.2.16 INC、INCP 16 位元 BIN 資料的遞增運算.....219
DEC、DECP 16 位元 BIN 資料遞減運算.....219
6.2.17 DINC、DINCP 32 位元 BIN 資料的遞增運算.....221
DDEC、DDECP 32 位元 BIN 資料遞減運算.....221
6.3 資料轉換指令 223
6.3.1 BCD、BCDP 從 BIN 資料到 BCD4 位元資料的轉換 223
DBCD、DBCDP 從 BIN 資料到 BCD8 位元資料的轉換 223
6.3.2 BIN、BINP 從 BCD4 位元資料到 BIN 資料的轉換 224
DBIN、DBINP 從 BCD8 位元資料到 BIN 資料的轉換 224
6.3.3 FLT、FLTP 從 BIN16 位元資料到浮點數據的轉換（單精確度）.....226
DFLT、DFLTP 從 BIN32 位元資料到浮點數據的轉換（單精確度）.....226
6.3.4 FLTD、FLTDP 從 BIN16 位元資料到浮點數據的轉換（雙精度）.....228
DFLTD、DFLTDP 從 BIN32 位元資料到浮點數據的轉換（雙精度）.....228
6.3.5 INT、INTP 從浮點數據到 BIN16 位元資料的轉換（單精確度）.....229
DINT、DINTP 從浮點數據到 BIN32 位元資料的轉換（單精確度）.....229

6.3.6 INTD、INTDP 從浮點數據到 BIN16 位元資料的轉換（雙精度）.....231
DINTD、DINTDP 從浮點數據到 BIN32 位元資料的轉換（雙精度）.....231
6.3.7 DBL、DBLP 從 BIN16 位元資料到 BIN32 位元資料的轉換 233
6.3.8 WORD、WORDP 從 BIN32 位元資料到 BIN16 位元資料的轉換 234
6.3.9 GRY、GRYP 從 BIN16 位元資料到格雷碼的轉換 235
DGRY、DGRYP 從 BIN32 位元資料到格雷碼的轉換 235
6.3.10 GBIN、GBINP 從格雷碼到 BIN16 位元資料的轉換 236
DGBIN、DGBINP 從格雷碼到 BIN32 位元資料的轉換 236
6.3.11 NEG、NEGP BIN16 位元資料的 2 進制補數（符號取反）.....237
DNEG、DNEGP BIN32 位元資料的 2 進制補數（符號取反）.....237
6.3.12 ENEG、ENEGP 浮點數據的符號取反（單精確度）.....239
6.3.13 EDNEG、EDNEGP 浮點數據的符號取反（雙精度）.....240
6.3.14 BKBCD、BKBCDP 從塊 BIN16 位元資料到塊 BCD4 位元資料的轉換 241
6.3.15 BKBIN、BKBINP 從塊 BCD4 位元資料到塊 BIN16 位元資料的轉換 242
6.3.16 ECON、ECONP 單精確度→雙精度轉換 244
6.3.17 EDCON、EDCONP 雙精度→單精確度轉換 245
6.4 資料傳送指令 246
6.4.1 MOV、MOVP 16 位元資料傳送.....246
DMOV、DMOVP 32 位元資料傳送.....246
6.4.2 EMOV、EMOVP 浮點數據傳送（單精確度）.....247
6.4.3 EDMOV、EDMOVP 浮點數據傳送（雙精度）.....248
6.4.4 $MOV、$MOVP 字串傳送 249
6.4.5 CML、CMLP 16 位元資料否定傳送.....251
DCML、DCMLP 32 位元資料否定傳送.....251
6.4.6 BMOV、BMOVP 塊 16 位元資料傳送 254
6.4.7 FMOV、FMOVP 相同 16 位資料塊傳送 257
6.4.8 DFMOV、DFMOVP 相同 32 位資料塊傳送 259
6.4.9 XCH、XCHP 16 位元資料交換.....261
DXCH、DXCHP 32 位元資料交換.....261
6.4.10 BXCH、BXCHP 塊 16 位元資料交換 262
6.4.11 SWAP、SWAPP 高位元組和低位元組交換 264
6.5 程式分支指令 265
6.5.1 CJ、SCJ、JMP 程式分支指令 265
6.5.2 GOEND 跳轉至 END.....267
6.6 程式執行控制指令 269
6.6.1 DI 中斷禁止指令 269
EI 中斷允許指令 269
IMASK 中斷程式遮罩 269
6.6.2 IRET 從中斷程式的恢復 274
6.7 I/O 刷新指令 275
6.7.1 RFS、RFSP I/O 刷新指令.....275

6.8 其他方便的指令 277
6.8.1 UDCNT1 單相輸入加法 / 減法計數器 277
6.8.2 UDCNT2 兩相輸入加法 / 減法計數器 279
6.8.3 TTMR 教學計時器 281
6.8.4 STMR 特殊功能計時器 282
6.8.5 ROTC 旋轉台就近控制 284
6.8.6 RAMP 斜坡信號 286
6.8.7 SPD 脈衝密度測定 288
6.8.8 PLSY 恒定周期脈衝輸出 289
6.8.9 PWM 脈衝寬度調製 290
6.8.10 MTR 矩陣輸入 292

7 應用指令 294

7.1 邏輯運算指令 294
7.1.1 WAND、WANDP 16 位元資料的邏輯積 295
DAND、DANDP 32 位元資料的邏輯積 295
7.1.2 BKAND、BKANDP 塊邏輯積 299
7.1.3 WOR、WORP 16 位元資料的邏輯和 301
DOR、DORP 32 位元資料的邏輯和 301
7.1.4 BKOR、BKORP 塊邏輯和 305
7.1.5 WXOR、WXORP 16 位元資料排他邏輯和 307
DXOR、DXORP 32 位元資料排他邏輯和 307
7.1.6 BKXOR、BKXORP 塊排他邏輯和 311
7.1.7 WXNR、WXNRP 16 位元資料否定排他邏輯和 313
DXNR、DXNRP 32 位元資料否定排他邏輯和 313
7.1.8 BKXNR、BKXNRP 塊否定排他邏輯和 317
7.2 旋轉指令 319
7.2.1 ROR、RORP、RCR、RCRP 16 位元資料的右旋轉 319
7.2.2 ROL、ROLP、RCL、RCLP 16 位元資料的左旋轉 321
7.2.3 DROR、DRORP、DRCR、DRCRP 32 位元資料的右旋轉 324
7.2.4 DROL、DROLP、DRCL、DRCLP 32 位元資料的左旋轉 326
7.3 移位元指令 328
7.3.1 SFR、SFRP 16 位元資料的 n 位元右移 328
SFL、SFLP 16 位元資料的 n 位元左移 328
7.3.2 BSFR、BSFRP n 位元資料的 1 位元右移 330
BSFL、BSFLP n 位元資料的 1 位元左移 330
7.3.3 SFTBR、SFTBRP n 位元資料的 n 位元右移 332
SFTBL、SFTBLP n 位元資料的 n 位元左移 332
7.3.4 DSFR、DSFRP n 字資料的 1 字右移 334
DSFL、DSFLP n 字資料的 1 字左移 334
7.3.5 SFTWR、SFTWRP n 字資料的 n 字右移 336
SFTWL、SFTWLP n 字資料的 n 字左移 336
7.4 位元處理指令 338
7.4.1 BSET、BSETP 字軟元件的位設置 338
BRST、BRSTP 字軟元件的位重定 338

7.4.2 TEST、TESTP、DTEST、DTESTP 位測試 339
7.4.3 BKRST、BKRSTP 位軟元件的批量重定 341
7.5 資料處理指令 343
7.5.1 SER、SERP 16 位元資料搜索 343
DSER、DSERP 32 位元資料搜索 343
7.5.2 SUM、SUMP 16 位元資料的位元檢查 345
DSUM、DSUMP 32 位元資料的位元檢查 345
7.5.3 DECO、DECOP 8 位元到 256 位元的解碼 347
7.5.4 ENCO、ENCOP 將 256 位元編碼爲 8 位元 348
7.5.5 SEG、SEGP 7 段解碼 350
7.5.6 DIS、DISP 16 位元資料的 4 位元分離 352
7.5.7 UNI、UNIP 16 位元資料的 4 位元合併 353
7.5.8 NDIS、NDISP 任意資料的位元分離 354
NUNI、NUNIP 任意資料的位元合併 354
7.5.9 WTOB、WTOBP 以位元組爲單位的資料分離 358
BTOW、BTOWP 以位元組爲單位的資料合併 358
7.5.10 MAX、MAXP 16 位元資料的最大值搜索 361
DMAX、DMAXP 32 位元資料的最大值搜索 361
7.5.11 MIN、MINP 16 位元資料的最小值查找 362
DMIN、DMINP 32 位元資料的最小值查找 362
7.5.12 SORT 16 位元資料的排序 364
DSORT 32 位元資料的排序 364
7.5.13 WSUM、WSUMP 16 位元資料的合計值計算 367
7.5.14 DWSUM、DWSUMP 32 位元資料的合計值計算 368
7.5.15 MEAN、MEANP 16 位元資料的平均值計算 369
DMEAN、DMEANP 32 位元資料的平均值計算 369
7.6 結構化指令 371
7.6.1 FOR、NEXT FOR 至 NEXT 指令迴圈 371
7.6.2 BREAK、BREAKP FOR ～ NEXT 指令迴圈的強制結束 372
7.6.3 CALL、CALLP 副程式調用 374
7.6.4 RET 從副程式返回 377
7.6.5 FCALL、FCALLP 副程式輸出 OFF 調用 378
7.6.6 ECALL、ECALLP 程式文件之間的副程式調用 382
7.6.7 EFCALL、EFCALLP 程式文件之間的副程式輸出 OFF 調用
OFF 調用 387
7.6.8 XCALL 副程式調用 390
7.6.9 COM 刷新指令 395
7.6.10 COM 選擇刷新 397
7.6.11 CCOM、CCOMP 選擇刷新 400
7.6.12 IX、IXEND 整個梯形圖的變址修飾 400
7.6.13 IXDEV、IXSET 整個梯形圖的變址修飾中修飾值的指定 403
7.7 資料表操作指令 406
7.7.1 FIFW、FIFWP 將資料寫入資料表 406
7.7.2 FIFR、FIFRP 從表格中讀取最舊的資料 407
7.7.3 FPOP、FPOPP 從資料表中讀取最新資料 409

7.7.4 FDEL、FDELP 資料表的資料刪除 411
FINS、FINSP 資料表的資料插入 411
7.8 緩衝記憶體訪問指令 413
7.8.1 FROM、FROMP 從智慧功能模組中讀取 1 字資料 413
DFRO、DFROP 從智慧功能模組中讀取 2 字資料 413
7.8.2 TO、TOP 將 1 字資料寫入智慧功能模組 415
DTO、DTOP 將 2 字資料寫入智慧功能模組 415
7.9 顯示指令 418
7.9.1 PR ASCII 碼列印指令 418
7.9.2 PRC 注釋列印指令 421
7.9.3 LEDR 出錯顯示或報警器重定指令 423
7.10 調試和故障診斷指令 426
7.10.1 CHKST、CHK 特殊格式故障檢查 426
7.10.2 CHKCIR、CHKEND 改變檢查指令的檢查格式 429
7.11 字串處理指令 433
7.11.1 BINDA、BINDAP 從 BIN16 位到 10 進制 ASCII 碼的轉換 433
DBINDA、DBINDAP 從 BIN32 位到 10 進制 ASCII 碼的轉換 433
7.11.2 BINHA、BINHAP 從 BIN16 位元資料到 16 進制 ASCII 碼的轉換 435
DBINHA、DBINHAP 從 BIN32 位元資料到 16 進制 ASCII 碼的轉換 435
7.11.3 BCDDA、BCDDAP 從 BCD4 位元資料到 10 進制 ASCII 碼的轉換 438
DBCDDA、DBCDDAP 從 BCD8 位元資料到 10 進制 ASCII 碼的轉換 438
7.11.4 DABIN、DABINP 從 10 進制 ASCII 碼到 BIN16 位元資料的轉換 440
DDABIN、DDABINP 從 10 進制 ASCII 碼到 BIN32 位元資料的轉換 440
7.11.5 HABIN、HABINP 從 16 進制 ASCII 到 BIN16 位元資料的轉換 443
DHABIN、DHABINP 從 16 進制 ASCII 到 BIN32 位元資料的轉換 443
7.11.6 DABCD、DABCDP 從 10 進制 ASCII 碼到 BCD4 位元資料的轉換 444
DDABCD、DDABCDP 從 10 進制 ASCII 碼到 BCD8 位元資料的轉換 444
7.11.7 COMRD、COMRDP 讀取軟元件注釋資料 447
7.11.8 LEN、LENP 字串長度檢測 449
7.11.9 STR、STRP 從 BIN16 位元到字串的轉換 450
DSTR、DSTRP 從 BIN32 位元到字串的轉換 450
7.11.10 VAL、VALP 從字串到 BIN16 位元資料的轉換 454
DVAL、DVALP 從字串到 BIN32 位元資料的轉換 454
7.11.11 ESTR、ESTRP 從浮點數到字串的轉換 458
7.11.12 EVAL、EVALP 從字串到浮點數的轉換 463
7.11.13 ASC、ASCP 從 16 進制 BIN 至 ASCII 碼的轉換 466
7.11.14 HEX、HEXP 從 ASCII 碼到 16 進制 BIN 的轉換 468
7.11.15 RIGHT、RIGHTP 從字串的右側提取資料 470
LEFT、LEFTP 從字串的左側提取資料 470
7.11.16 MIDR、MIDRP 字串的任意提取 472
MIDW、MIDWP 字串的任意置換 472
7.11.17 INSTR、INSTRP 字串搜索 475
7.11.18 STRINS、STRINSP 字串插入 477

7.11.19 STRDEL、STRDELP 字串刪除 479
7.11.20 EMOD、EMODP 從浮點數到 BCD 的分解 480
7.11.21 EREXP、EREXPP 從 BCD 格式資料到浮點數 482
7.12 特殊函數指令 484
7.12.1 SIN、SINP 浮點數的 SIN 運算（單精確度）..... 484
7.12.2 SIND、SINDP 浮點數的 SIN 運算（雙精度）..... 485
7.12.3 COS、COSP 浮點數的 COS 運算（單精確度）..... 487
7.12.4 COSD、COSDP 浮點數的 COS 運算（雙精度）..... 488
7.12.5 TAN、TANP 浮點數的 TAN 運算（單精確度）..... 490
7.12.6 TAND、TANDP 浮點數的 TAN 運算（雙精度）..... 492
7.12.7 ASIN、ASINP 浮點數的 SIN^{-1} 運算（單精確度）..... 493
7.12.8 ASIND、ASINDP 浮點數的 SIN^{-1} 運算（雙精度）..... 495
7.12.9 ACOS、ACOSP 浮點數的 COS^{-1} 運算（單精確度）..... 496
7.12.10 ACOSD、ACOSP 浮點數的 COS^{-1} 運算（雙精度）..... 498
7.12.11 ATAN、ATANP 浮點數的 TAN^{-1} 運算（單精確度）..... 499
7.12.12 ATAND、ATANDP 浮點數的 TAN^{-1} 運算（雙精度）..... 501
7.12.13 RAD、RADP 從浮點數角度到弧度的轉換（單精確度）..... 502
7.12.14 RADD、RADDP 從浮點數角度到弧度的轉換（雙精度）..... 503
7.12.15 DEG、DEGP 從浮點數弧度到角度的轉換（單精確度）..... 505
7.12.16 DEGD、DEGDP 從浮點數弧度到角度的轉換（雙精度）..... 506
7.12.17 POW、POWP 浮點數的冪運算（單精確度）..... 508
7.12.18 POWD、POWDP 浮點數的冪運算（雙精度）..... 509
7.12.19 SQR、SQRP 浮點數的平方根運算（單精確度）..... 510
7.12.20 SQRD、SQRDP 浮點數的平方根運算（雙精度）..... 512
7.12.21 EXP、EXPP 浮點數的指數運算（單精確度）..... 513
7.12.22 EXPD、EXPDP 浮點數的指數運算（雙精度）..... 515
7.12.23 LOG、LOGP 浮點數的自然對數運算（單精確度）..... 516
7.12.24 LOGD、LOGDP 浮點數的自然對數運算（雙精度）..... 518
7.12.25 LOG10、LOG10P 浮點數的常用對數運算（單精確度）..... 519
7.12.26 LOG10D、LOG10DP 浮點數的常用對數運算（雙精度）..... 520
7.12.27 RND、RNDP 亂數的產生 521
SRND、SRNDP 系列變更 521
7.12.28 BSQR、BSQRP BCD4 位平方根 522
BDSQR、BDSQRP BCD8 位平方根 522
7.12.29 BSIN、BSINP BCD 型 SIN 運算 524
7.12.30 BCOS、BCOSP BCD 型 COS 運算 526
7.12.31 BTAN、BTANP BCD 型 TAN 運算 528
7.12.32 BASIN、BASINP BCD 型 SIN^{-1} 運算 530
7.12.33 BACOS、BACOSP BCD 型 COS^{-1} 運算 532
7.12.34 BATAN、BATANP BCD 型 TAN^{-1} 運算 534
7.13 資料控制指令 536
7.13.1 LIMIT、LIMITP BIN16 位元資料的上下限控制 536
DLIMIT、DLIMITP BIN32 位元資料的上下限控制 536
7.13.2 BAND、BANDP BIN16 位元死區控制 538
DBAND、DBANDP BIN32 位元死區控制 538
7.13.3 ZONE、ZONEP BIN16 位元資料的區域控制 541
DZONE、DZONEP BIN32 位元資料的區域控制 541

7.13.4 SCL、SCLP、DSCL、DSCLP 標度 (點座標資料) 543
7.13.5 SCL2、SCL2P、DSCL2、DSCL2P 標度 (X/Y 座標資料) 546
7.14 文件寄存器切換指令 549
7.14.1 RSET、RSETP 文件寄存器的塊號切換 549
7.14.2 QDRSET、QDRSETP 文件寄存器用文件的設置 550
7.14.3 QCDSET、QCDSETP 注釋用文件的設置 552
7.15 時鐘指令 555
7.15.1 DATERD、DATERDP 時鐘資料的讀取 555
7.15.2 DATEWR、DATEWRP 時鐘資料的寫入 556
7.15.3 DATE+、DATE+P 時鐘資料加法運算 558
7.15.4 DATE-、DATE-P 時鐘資料減法運算 560
7.15.5 SECOND、SECONDP 時間資料轉換 (從小時 / 分鐘 / 秒到秒) 561
7.15.6 HOUR、HOURP 時間資料轉換 (從秒到小時 / 分鐘 / 秒) 563
7.15.7 DT=、DT<>、DT>、DT<=、DT<、DT>= 日期比較 564
7.15.8 TM=、TM<>、TM>、TM<=、TM<、TM>= 時間比較 567
7.16 擴展時鐘指令 571
7.16.1 S.DATERD、SP.DATERD 擴展時鐘資料的讀取 571
7.16.2 S.DATE+、SP.DATE+ 擴展時鐘資料的加法運算 573
7.16.3 S.DATE-、SP.DATE- 擴展時鐘資料減法運算 575
7.17 程式控制指令 578
7.17.1 PSTOP、PSTOPP 程式待機指令 579
7.17.2 POFF、POFFP 程式輸出 OFF 待機指令 580
7.17.3 PSCAN、PSCANP 程式掃描執行登錄指令 581
7.17.4 PLOW、PLOWP 程式低速執行登錄指令 582
7.17.5 PCHK 程式執行狀態檢查指令 583
7.18 其他指令 585
7.18.1 WDT、WDTP 看門狗計時器重定 585
7.18.2 DUTY 定時脈衝發生 586
7.18.3 TIMCHK 時間檢查指令 587
7.18.4 ZRRDB、ZRRDBP 文件寄存器的直接 1 位元組讀取 588
7.18.5 ZRWRB、ZRWRBP 文件寄存器的直接 1 位元組寫入 589
7.18.6 ADRSET、ADRSETP 間接地址讀取 591
7.18.7 KEY 鍵盤的數位鍵輸入 592
7.18.8 ZPUSH、ZPUSHP 變址寄存器的批量保存 596
ZPOP、ZPOPP 變址寄存器的批量恢復 596
7.18.9 UNIRD、UNIRDP 模組資訊讀取 598
7.18.10 TYPERD、TYPERDP 模組型號讀取 602
7.18.11 TRACE 追蹤設置 606
TRACER 追蹤重定 606
7.18.12 SP.FWRITE 寫資料到指定的文件 607
7.18.13 SP.FREAD 從指定文件中讀取資料 617
7.18.14 SP.DEVST 向標準 ROM 中寫入資料 628
7.18.15 S.DEVLD、SP.DEVLD 從標準 ROM 中讀取資料 629
7.18.16 PLOADP 通過存儲卡的程式裝載 631

7.18.17 PUNLOADP 從程式記憶體中卸載程式 633
7.18.18 PSWAPP 裝載 + 卸載 634
7.18.19 RBMOV、RBMOVP 文件寄存器的高速塊傳送 636
7.18.20 UMSG 用戶資訊 640

8 資料鏈接指令 644

8.1 網路刷新指令 644
8.1.1 S.ZCOM、SP.ZCOM 對指定模組的刷新指令 644
8.2 路由資訊的讀取 / 寫入 648
8.2.1 S.RTREAD、SP.RTREAD 路由資訊的讀取 648
8.2.2 S.RTWRITE、SP.RTWRITE 路由資訊的登錄 649

9 多 CPU 專用指令 651

9.1 寫入自站 CPU 共用記憶體 651
9.1.1 S.TO、SP.TO 寫入到自站 CPU 共用記憶體 652
9.1.2 TO、TOP、DTO、DTOP 寫入到自站 CPU 共用記憶體 655
9.2 從其他站 CPU 共用記憶體中讀取資料 659
9.2.1 FROM、FROMP、DFRO、DFROP 從其他站 CPU 共用記憶體中讀取資料 660

10 多 CPU 高速通信專用指令 665

10.1 概要 665
10.2 D.DDWR、DP.DDWR 至其他站的軟元件寫入 675
10.3 D.DDSRD、DP.DDRD 從其他站讀取軟元件 679

11 冗餘系統指令 (用於冗餘 CPU) 683

11.1 SP.CONTSW 系統切換 683

附錄 686

附錄 1 運算處理時間 686
附錄 1.1 運算處理時間的思路 686
附錄 1.2 基本型 QCPU 的運算處理時間 687
附錄 1.3 高性能型 QCPU/ 過程 CPU/ 冗餘 CPU 的運算處理時間 701
附錄 1.4 通用型 QCPU 的運算處理時間 723
附錄 1.4.1 子集指令處理時間 723
附錄 1.4.2 子集指令以外的指令處理時間 735
附錄 1.5 LCPU 的運算處理時間.....772
附錄 1.5.1 子集指令處理時間 772
附錄 1.5.2 子集指令以外的指令處理時間一覽表 777
附錄 2 CPU 的性能比較..... 793
附錄 2.1 QCPU/LCPU 與 AnNCPU/AnACPU/AnUCPU 的比較.....793
附錄 2.1.1 可用的軟元件 793
附錄 2.1.2 I/O 控制方式 794
附錄 2.1.3 可用於指令的資料 794

附錄 2.1.4 計時器的比較 795
附錄 2.1.5 計數器的比較 796
附錄 2.1.6 顯示指令的比較 796
附錄 2.1.7 指定格式被變更的指令 (AnACPU/AnUCPU 的專用指令除外) 797
附錄 2.1.8 AnACPU/AnUCPU 的專用指令 798
附錄 3 應用程式示例 799
附錄 3.1執行 X^n、運算的程式的思路 799

術語索引 800

指令索引 805

質保 809

手冊體系

在基本手冊中介紹基本的規格、功能、使用方法有關內容。
其他手冊介紹相應的 CPU 模組及功能。
請根據需要參考下表訂購各手冊。

“ “ CPU 模組 ” 欄中所示的編號及 CPU 模組的對應如下所示：

編號	CPU 模組
1)	基本型 QCPU
2)	高性能型 QCPU
3)	過程 CPU
4)	冗餘 CPU
5)	通用型 QCPU
6)	LCPU

●：基本手冊　○：使用相應 CPU 模組 / 功能時參閱。

手冊名稱 < 手冊編號 >	記載內容	CPU 模組					
		1)	2)	3)	4)	5)	6)
■用戶手冊							
QCPU 用戶手冊 （硬體設計 / 維護點檢篇） <SH(NA)-080501CHN>	介紹 CPU 模組、電源模組、基板、擴展電纜及存儲卡等的硬體規格及系統維護 · 點檢、故障排除、出錯代碼等。	●	●	●	●	●	
QnUCPU 用戶手冊 （功能解說 / 程式基礎篇） <SH(NA)-080812CHN>	介紹創建程式所需的功能、編程方法和軟元件等。					●	
Qn(H)/QnPH/QnPRHCPU 用戶手冊 （功能解說 / 程式基礎篇） <SH(NA)-080808ENG>	介紹創建程式所需的功能、編程方法和軟元件等。	●	●	●	●		
QnUCPU 用戶手冊 （內置乙太網板通信篇） <SH(NA)-080813CHN>	介紹 CPU 內置乙太網板通信的功能有關內容。					○	
MELSEC-L CPU 用戶手冊 （硬體設計 / 維護點檢篇） <SH(NA)-080943CHN>	介紹 CPU 模組、電源模組、分支模組、擴展模組、存儲卡等的硬體規格及系統維護、點檢、故障排除、出錯代碼等。						●
MELSEC-L CPU 用戶手冊 （功能解說 / 程式基礎篇 <SH(NA)-080942CHN>	介紹創建程式所需的功能、編程方法和軟元件等。						●
MELSEC-L CPU 用戶手冊 （內置 I/O 功能篇） <SH(NA)-080945CHN>	介紹 CPU 內置 I/O 功能相關內容。						○
MELSEC-L CPU 用戶手冊 （內置乙太網功能篇） <SH(NA)-080944CHN>	介紹 CPU 內置乙太網板通信功能相關內容。						○
MELSEC-L CPU 用戶手冊 （資料記錄功能篇） <SH(NA)-080946CHN>	介紹 CPU 模組的資料記錄功能相關內容。						○

手冊名稱 < 手冊編號 >	記載內容	CPU 模組					
		1)	2)	3)	4)	5)	6)
■編程手冊							
MELSEC-Q/L 編程手冊 (公共指令篇) <SH(NA)-081065CHT>	介紹順控程式指令、基本指令以及應用指令等的使用方法。	●	●	●	●	●	●
MELSEC-Q/L/QnA 編程手冊 (SFC 篇) <SH(NA)-080283CHN>	介紹順控程式指令、基本指令以及應用指令等的使用方法。	○	○	○	○	○	○
MELSEC-Q/L 編程手冊 (MELSAP-L 篇) <SH(NA)-080973CHN>	介紹 MELSAP-L 格式的 SFC 程式的創建所需的編程方法、規格、功能等。	○	○	○	○	○	○
MELSEC-Q/L 編程手冊 (結構化文本篇) <SH(NA)-081064CHT>	介紹結構化文本語言的編程方法。	○	○	○	○	○	○
MELSEC-Q/L/QnA 編程手冊 (PID 控制指令篇) <SH(NA)-080240C>	介紹用於執行 PID 控制的專用指令。	○	○		○	○	○
QnPHCPU/QnPRHCPU 編程手冊 (程序控制指令篇) <SH(NA)-080449CHN>	介紹用於執行程序控制的專用指令。			○	○		

關聯手冊

手冊名稱 < 手冊編號 >	記載內容
CC-Link IE 控制網路參考手冊 <SH(NA)-080710CHN>	介紹 CC-Link IE 控制網路的控制網路規格、投運前的設置及步驟、參數設置、編程及故障排除等。
MELSEC-Q CC-Link IE 現場網路主站・本地站模組用戶手冊 <SH(NA)-081023CHN>	介紹 MELSEC-Q CC-Link IE 現場網路規格、投運前的設置及步驟、參數設置、編程及故障排除等。
MELSEC-L CC-Link IE 現場網路主站 / 本地站模組用戶手冊 <SH(NA)-081026CHN>	介紹 CC-Link IE 現場網路的現場網路的規格、投運前的設置及步驟、參數設置、編程及故障排除等。
Q 系列 MELSECNET/H 網路系統參考手冊 (可程式控制器網路篇) <SH(NA)-080289C>	介紹 MELSECNET/H 網路系統的可程式控制器網路的規格、投運前的設置及步驟、參數設置、編程及故障排除等。
Q 系列 MELSECNET/H 網路系統參考手冊 (遠端 I/O 網路篇) <SH(NA)-080290C>	介紹 MELSECNET/H 網路系統的遠端 I/O 網路的規格、投運前的設置及步驟、參數設置、編程及故障排除等。
MELSECNET、MELSECNET/B 資料鏈接系統參考手冊 <SH(NA)-080206C>	介紹 MELSECNET(II) 和 MELSECNET/B 的概要、規格和各部位名稱及設置等。
Q 系列乙太網介面模組用戶手冊 (應用篇) <SH(NA)-080285C>	介紹乙太網模組的電子郵件功能、可程式控制器 CPU 的狀態監視、經由 LSECNET/H 或 MELSECNET/10 網路的通信功能、通過資料鏈接用指令的通信功能、文件傳輸 (FTP 伺服器) 等功能。

第 1 章 概要

本手冊介紹進行 QCPU、LCPU 編程時所需的公共指令有關內容。

公共指令是指，除 QJ71C24N、QJ71E71-100 等智慧功能模組專用指令、PID 控制專用指令、SFC 專用指令、ST 專用指令、套接字通信功能專用指令、LCPU 的觸發記錄指令・LCPU 的定位功能專用指令・計數器功能專用指令以外的指令。

1.1 相關編程手冊

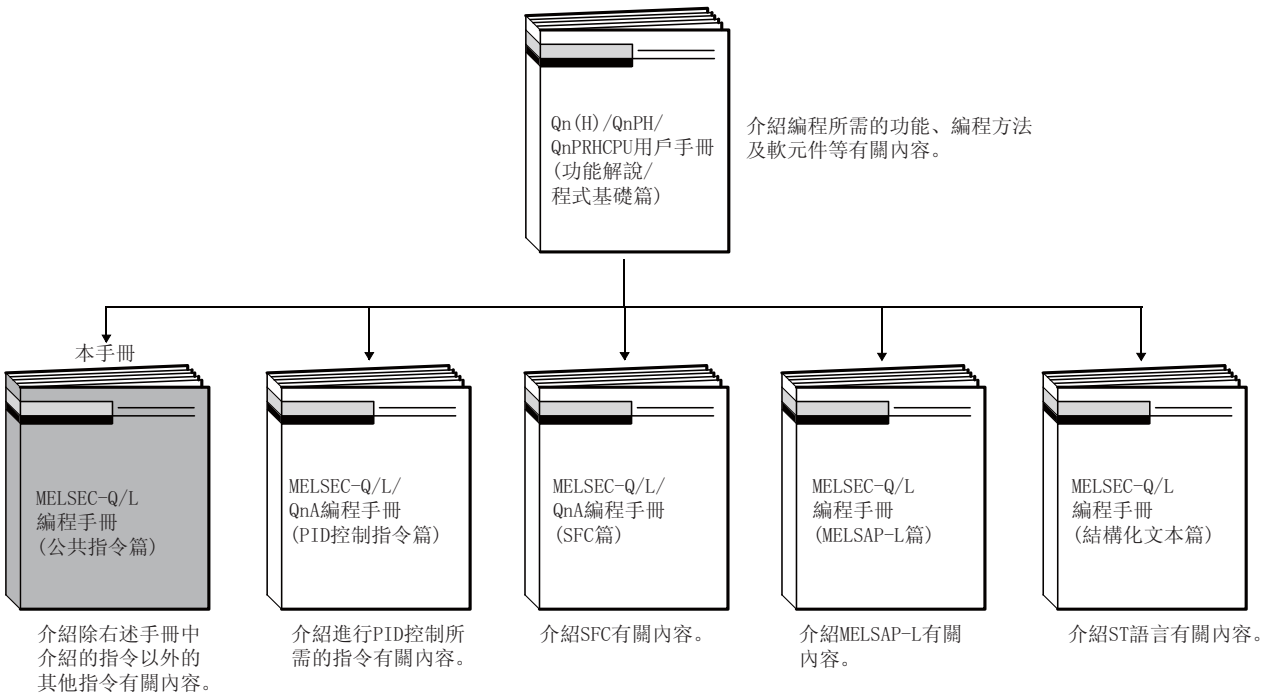
在閱讀本手冊之前，關於 CPU 模組編程所必需的功能、編程方法、軟元件等，請參閱下列手冊：

・QnUCPU 用戶手冊（功能解說 / 程式基礎篇）

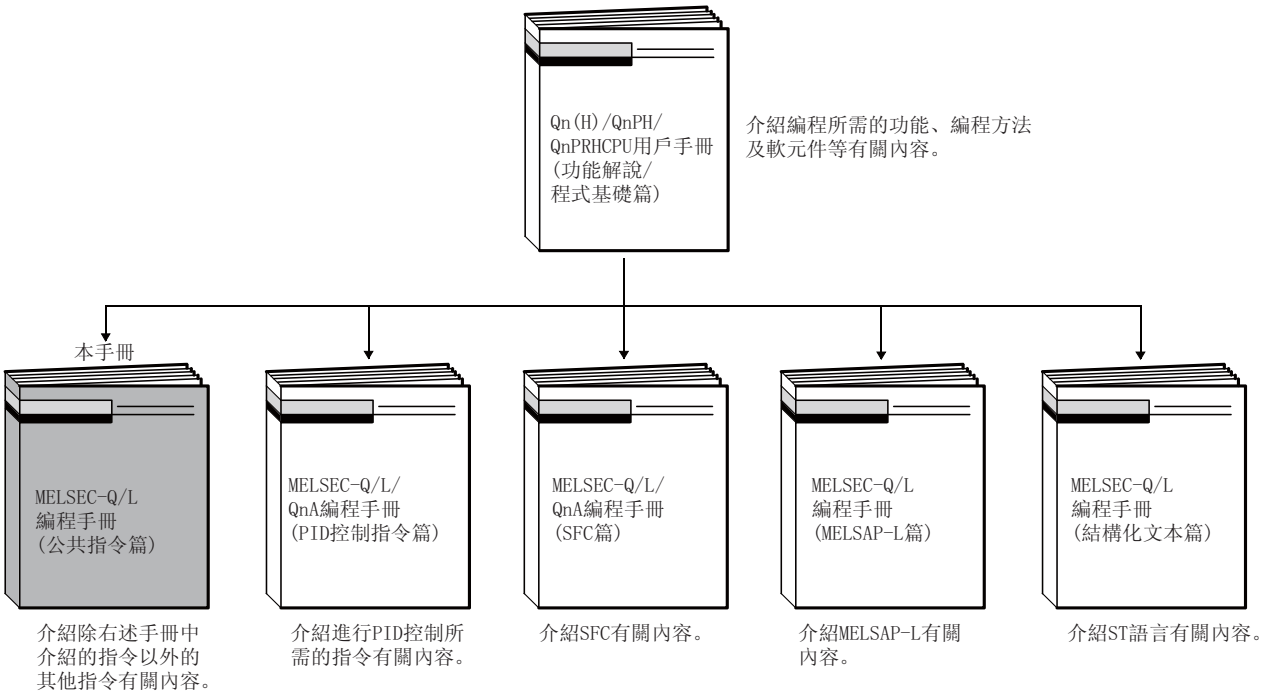
・Qn(H)/QnPH/QnPRHCPU 編程手冊（功能解說 / 程式基礎篇）

・MELSEC-L CPU 用戶手冊（功能解說 / 程式基礎篇）

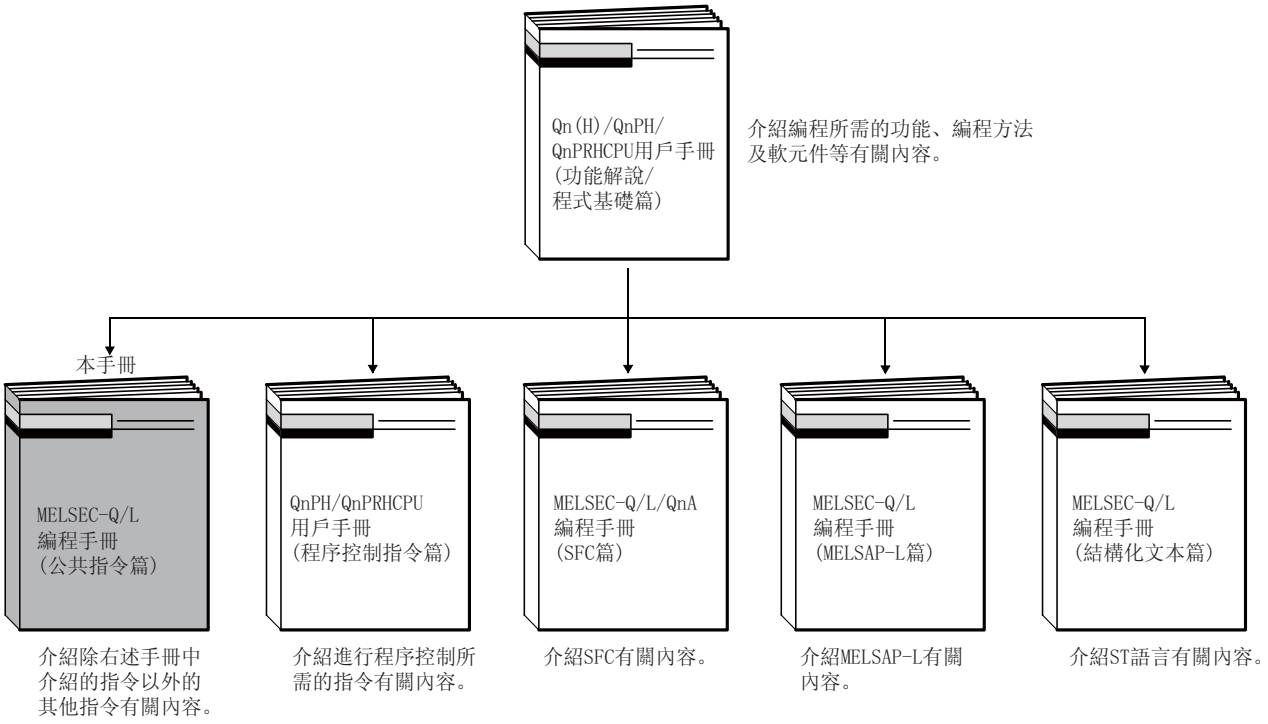
(1) 使用基本型 QCPU 時



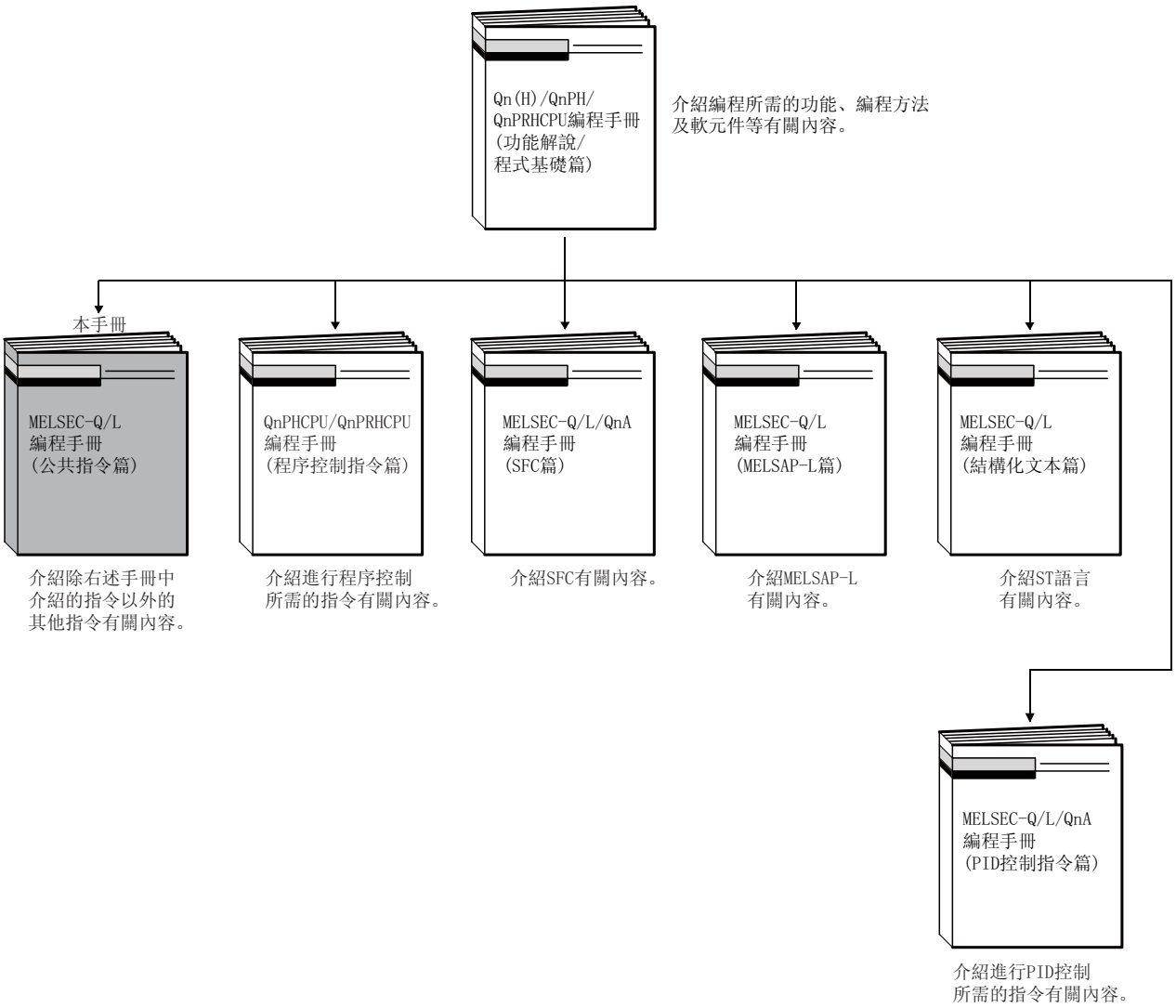
(2) 使用高性能型 QCPU 時



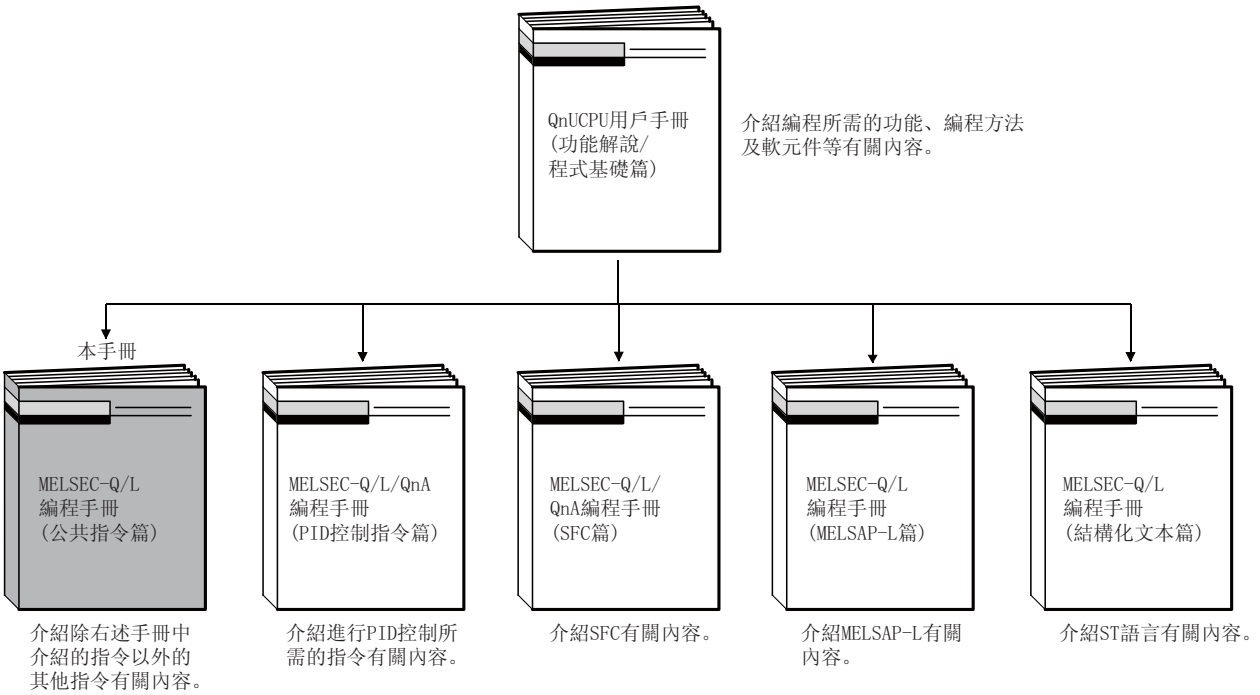
(3) 使用過程 CPU 時



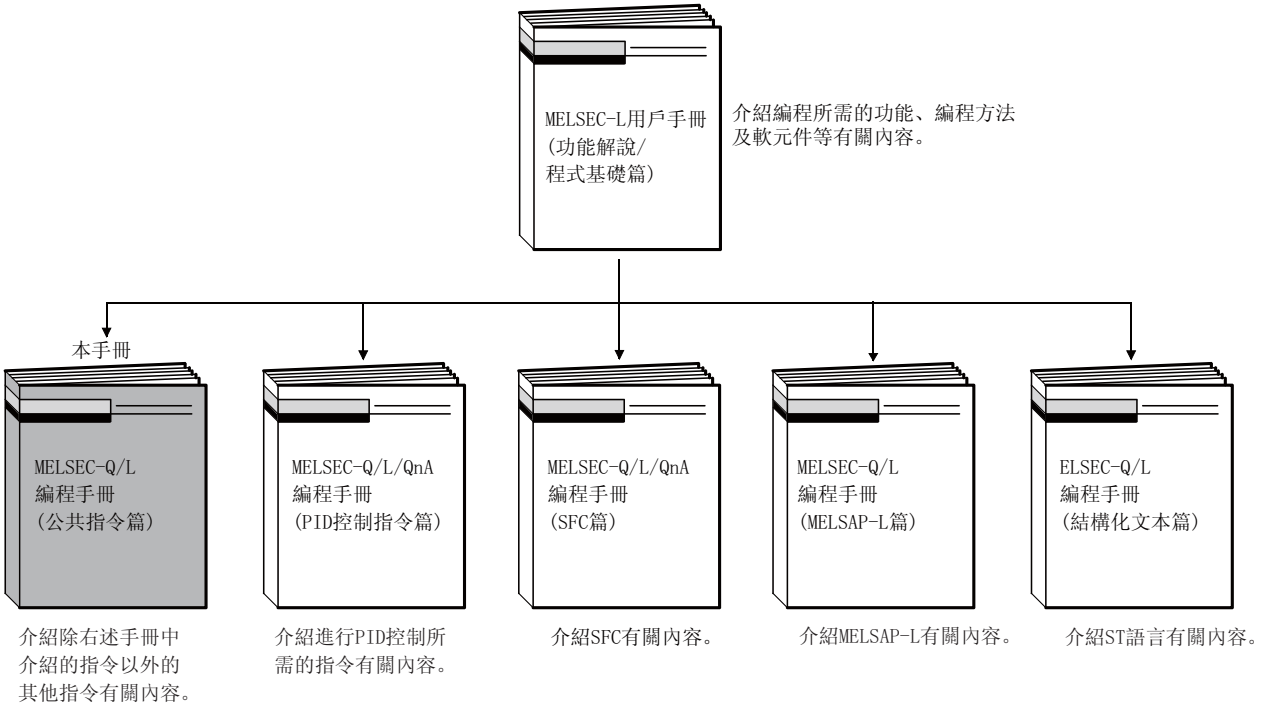
(4) 使用冗餘 CPU 時



(5) 使用通用型 QCPU 時



(6) 使用 LCPU 時


MELSEC-L用戶手冊
(功能解說/
程式基礎篇)
介紹編程所需的功能、編程方法
及軟元件等有關內容。
本手冊
MELSEC-Q/L
編程手冊
(公共指令篇)
介紹除右述手冊中
介紹的指令以外的
其他指令有關內容。
MELSEC-Q/L/QnA
編程手冊
(PID控制指令篇)
介紹進行PID控制所
需的指令有關內容。
MELSEC-Q/L/QnA
編程手冊
(SFC篇)
介紹SFC有關內容。
MELSEC-Q/L
編程手冊
(MELSAP-L篇)
介紹MELSAP-L有關內容。
ELSEC-Q/L
編程手冊
(結構化文本篇)
介紹ST語言有關內容。

1.2 本手冊中使用的總稱 / 略稱

本手冊中除特別注明以外，將使用如下表所示的總稱和略稱來介紹 Q 系列 CPU 模組的有關內容。

* □表示多個型號及版本等的總稱時的可變部分。

總稱 / 略稱	總稱 / 略稱的內容
■系列名稱	
Q 系列	三菱通用可程式控制器 MELSEC-Q 系列的略稱。
L 系列	三菱通用可编程控制器 MELSEC-L 系列的略稱。
■ CPU 模組的類型名稱	
CPU 模組	基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU、通用型 QCPU、LCPU 的總稱。
基本型 QCPU	Q00JCPU、Q00CPU 和 Q01CPU 的總稱。
高性能型 QCPU	Q02CPU、Q02HCPU、Q06HCPU、Q12HCPU 和 Q25HCPU 的總稱。
過程 CPU	Q02PHCPU、Q06PHCPU、Q12PHCPU 和 Q25PHCPU 的總稱。
冗餘 CPU	Q12PRHCPU 和 Q25PRHCPU 的總稱。
通用型 QCPU	Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q10UDHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU、Q03UDECPU、Q04UDEHCPU、Q06UDEHCPU、Q10UDEHCPU、Q13UDEHCPU、Q20UDEHCPU、Q26UDEHCPU、Q50UDEHCPU、Q100UDEHCPU 的總稱。
■ CPU 模組的型號	
QnCPU	Q00JCPU、Q00CPU、Q01CPU 和 Q02CPU 的總稱。
QnHCPU	Q02HCPU、Q06HCPU、Q12HCPU 和 Q25HCPU 的總稱。
QnPHCPU	Q02PHCPU、Q06PHCPU、Q12PHCPU、Q25PHCPU 的總稱。
QnPRHCPU	Q12PRHCPU 和 Q25PRHCPU 的總稱。
QnUCPU	Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q10UDHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU、Q03UDECPU、Q04UDEHCPU、Q06UDEHCPU、Q10UDEHCPU、Q13UDEHCPU、Q20UDEHCPU、Q26UDEHCPU、Q50UDEHCPU、Q100UDEHCPU 的總稱。
QnU(D)(H)CPU	Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q10UDHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU 的總稱。
QnUD(H)CPU	Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q10UDHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU 的總稱。
QnUDE(H)CPU	Q03UDECPU、Q04UDEHCPU、Q06UDEHCPU、Q10UDEHCPU、Q13UDEHCPU、Q20UDEHCPU、Q26UDEHCPU、Q50UDEHCPU、Q100UDEHCPU 的總稱。
LCPU	L02CPU、L26CPU-BT 、L02CPU-P、L26CPU-PBT 的總稱。

（續）

總稱 / 略稱	總稱 / 略稱的內容
■其他	
編程工具	GX Developer、GX Works2 的總稱。
GX Developer	Q、L 系列相容的產品型號 SW □ D5C-GPPW 的總稱產品名。 □表示軟體版本。 關於可適用於各 CPU 模組的 GX Developer 的版本，請參閱所使用的 CPU 用戶手冊（硬體設計 / 維護點檢篇）的“ 系統配置 ”。
GX Works2	Q、L 系列相容的產品型號 SW □ DNC-GXW2 的總稱產品名。 □表示軟體版本。 關於可適用於各 CPU 模組的 GX Works2 的版本，請參閱所使用的 CPU 用戶手冊（硬體設計 / 維護點檢篇）的“ 系統配置 ”。
CC-Link IE	CC-Link IE 控制網路、CC-Link IE 現場網路的總稱。
MELSECNET/H	MELSECNET/H 網路系統的略稱。
MELSECNET/10	MELSECNET/10 網路系統的略稱。
MELSECNET(II、/B)	MELSECNET 和 MELSECNET/B 資料鏈接系統的略稱。
乙太網	乙太網網路系統的略稱。
CC-Link	Control & Communication Link 的略稱。
智慧功能模組軟元件	智慧功能模組軟元件和特殊功能模組軟元件的總稱。
Q3 □ DB	可安裝 CPU 模組 (Q00JCPU 除外)、Q 系列電源模組、Q 系列輸入輸出模組、智慧功能模組的 Q35DB、Q38DB、Q312DB 型多 CPU 間高速主基板的總稱。
Q5 □ B	可安裝 Q 系列輸入輸出模組、智慧功能模組的 Q52B、Q55B 型擴展基板的總稱。
Q6 □ B	可安裝 Q 系列電源模組、Q 系列輸入輸出模組、智慧功能模組的 Q63B、Q65B、Q68B、Q612B 型擴展基板的總稱。
Q6 □ WRB	可安裝冗餘電源模組、Q 系列輸入輸出模組、智慧功能模組的 Q65WRB 型冗餘擴展基板的別稱。
QA1S6 □ B	可安裝 AnS 系列電源模組、AnS 系列輸入輸出模組、特殊功能模組的 QA1S65B、QA1S68B 型擴展基板的總稱。
QA6 □ B	可安裝 A 系列電源模組、A 系列輸入輸出模組、特殊功能模組的 QA65B、QA68B 型擴展基板的總稱。
A5 □ B	可安裝 A 系列輸入輸出模組、特殊功能模組的無需電源型的 A52B、A55B、A58B 型擴展基板的總稱。
A6 □ B	可安裝 A 系列輸入輸出模組、特殊功能模組的 A62B、A65B、A68B 型擴展基板的總稱。
QA6ADP	QA6ADP 型 QA 轉換適配器模組的略稱。
QA6ADP+A5 □ B/A6 □ B	安裝了 QA6ADP 的 A 大型擴展基板的略稱。

第 2 章 指令一覽表

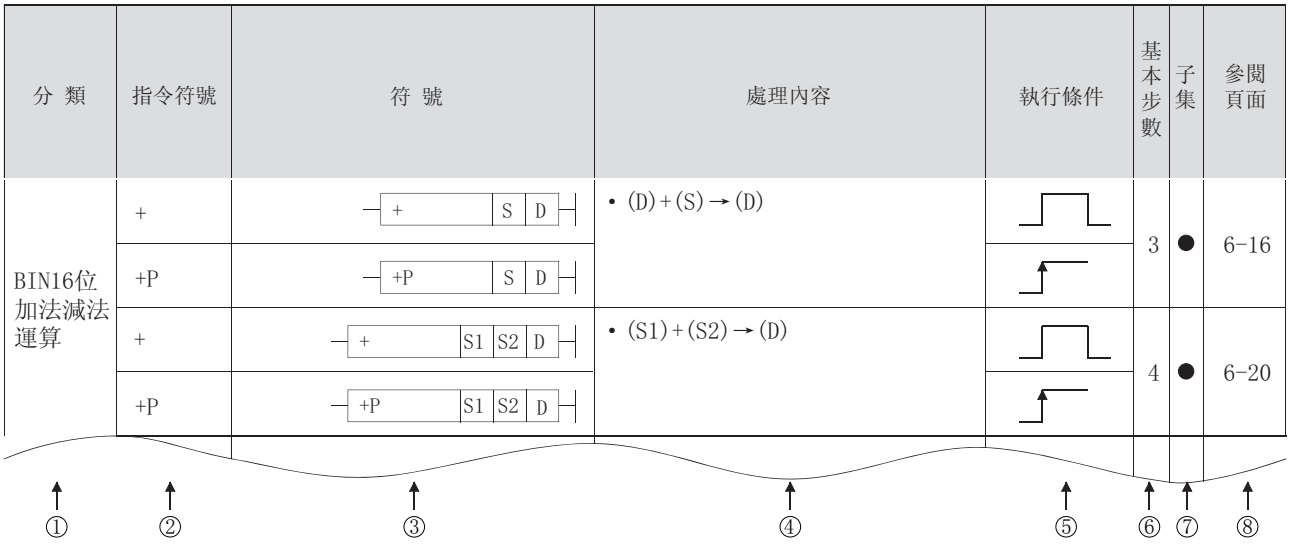
2.1 指令分類

CPU 模組指令的主要類型包括順控程式指令、基本指令、應用指令、資料鏈接指令、多 CPU 高速專用指令、多 CPU 高速通信專用指令和冗餘系統用指令。指令的分類如下表所示。

指令分類		內容	參閱章節
順控程式指令	觸點指令	運算開始、串列連接、並行連接。	120 頁第 5 章
	連接指令	梯形圖塊的連接、運算結果的記憶・讀取、運算結果的脈衝化。	
	輸出指令	位元軟元件的輸出、脈衝輸出、輸出取反。	
	移位元指令	位元軟元件的移位	
	主控制指令	主控制	
	結束指令	結束程式	
	其他指令	程式停止、無處理等未列入上述分類中的指令。	
基本指令	比較運算指令	=、＞、＜等的比較。	166 頁第 6 章
	算術運算指令	BIN、BCD 加法、減法、乘法或除法。	
	BCD↔BIN 轉換指令	將 BCD 轉換成 BIN 以及將 BIN 轉換成 BCD。	
	資料轉移指令	傳送指定的資料。	
	程式分支指令	程式跳轉	
	程式的執行控制指令	中斷程式的允許 / 禁止	
	I/O 刷新指令	部分刷新的執行	
	其他使用方便的指令	用於以下目的的指令： 計數器增加 / 減小、示教計時器、特殊功能計時器、旋轉台的就近控制等指令。	
應用指令	邏輯運算指令	邏輯和、邏輯積等的邏輯運算。	294 頁第 7 章
	旋轉指令	指定資料的旋轉	
	移位元指令	指定資料的移位元	
	位元處理指令	位元設置 / 重定、位測試、位軟元件的批量重定。	
	資料處理指令	16 位元資料的查找、解碼和編碼等資料處理。	
	結構化指令	重復運算、副程式調用、梯形圖單位的變址修飾。	
	表操作指令	資料表的讀 / 寫	
	緩衝記憶體訪問指令	智慧功能模組的資料讀 / 寫	
	顯示指令	ASCII 碼的列印、字元的 LED 顯示等。	
	調試・故障診斷指令	檢查、狀態鎖存、採樣跟蹤。	
	字串處理指令	BIN/BCD 與 ASCII 之間的轉換、BIN 與字串之間的轉換、浮點數據與字串之間的轉換、字串處理等。	
	特殊函數指令	三角函數、角度和弧度之間的轉換、指數運算、自然對數、常用對數、方根運算。	
	資料控制指令	上下限控制、死區控制、區域控制、標度。	
	切換指令	文件寄存器的塊號切換、文件寄存器 / 注釋文件的指定。	
	時鐘指令	年、月、日、時、分、秒和星期的讀 / 寫；時、分、秒的加減法運算；時、分、秒→秒的變換；年、月、日的比較；時、分、秒的比較。	
	擴展時鐘指令	年、月、日、時、分、秒、1/1000 秒和星期的讀取；時、分、秒、1/1000 秒的加減法運算。	
	程式控制用指令	用於切換程式的執行條件的指令	
	其他指令	其他未列入上述分類的指令，如看門狗計時器重定指令和定時時鐘指令等。	
資料鏈接用指令	鏈結刷新用指令	指定網路的刷新	644 頁第 8 章
	路由資訊讀取 / 寫入指令	路由資訊的讀取 / 登錄	
多 CPU 專用指令	多 CPU 專用指令	至自站 CPU 共用記憶體的寫入，從其他站 CPU 共用記憶體的讀取。	651 頁第 9 章
多 CPU 高速通信專用指令	多 CPU 軟元件寫入 / 讀取指令	至其他站的軟元件寫入、從其他站的軟元件讀取。	665 頁第 10 章
冗餘系統指令	用於冗餘 CPU 的指令	系統切換	683 頁第 11 章

2.2 指令一覽表的閱讀方法

28 頁 2.3 節至 78 頁 2.9 節的指令一覽表的形式如下所示：



說明

1) ..... 將指令按用途進行分類。

2) ..... 表示程式中使用的指令符號。

指令符號是以 16 位元指令爲基準。

32 位元指令、僅在 OFF → ON 的上升沿時執行的指令、實數指令、字串指令時的情況如下所示：

・32 位元指令 ..... 在指令的起始附加 D。

示例 + → D+

+：16位元指令　D+：32位元指令

・僅在 OFF → ON 的上升沿時執行的指令 ..... 在指令的末尾附加 P。

示例 + → +P

+：ON狀態下執行的指令　+P：僅在OFF→ON的上升沿時執行的指令

・實數指令 ..... 在指令的起始附加 E。

示例 + → E+

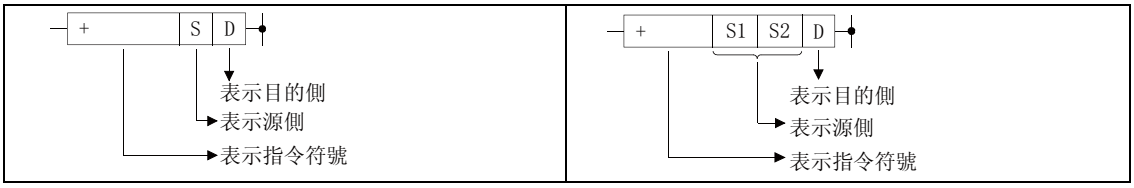
E+：實數指令

・字串指令 ..... 在指令的起始附加 $。

示例 + → $+

$+：字串指令

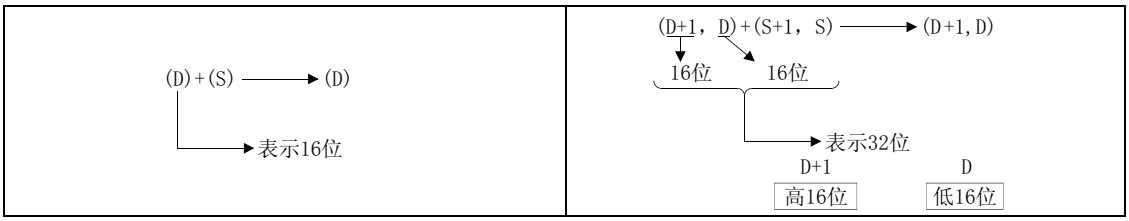
3) .....表示在梯形圖上的符號圖。



目的 (destination) ..... 表示運算後的資料去向。

源 (source) ..... 存儲運算前的資料。

4) .....表示各指令的處理內容。



5) .....各指令的執行條件的詳細內容如下所示：

符號	執行條件
未記入	是常時執行的指令，與指令的前條件的 ON/OFF 無關，一直執行。 在前條件爲 OFF 的情況下，該指令執行 OFF 處理。
	是在 ON 狀態下執行的指令，僅在指令的前條件爲 ON 的期間執行該指令。前條件爲 OFF 時，不執行該指令，不進行處理。
	是在 ON 時僅執行 1 次的指令，僅在指令的前條件的上升沿時 (OFF → ON) 執行指令。 以後即使條件變爲 ON 也不執行該指令，不進行處理。
	是在 OFF 狀態下執行的指令，僅在指令的前條件爲 OFF 的期間執行該指令。前條件爲 ON 時，不執行該指令，不進行處理。
	是在 OFF 時僅執行 1 次的指令，僅在指令的前條件的下降沿時 (ON → OFF) 執行指令。 以後即使條件變爲 OFF 也不執行該指令，不進行處理。

6) .....表示各指令的基本步數。

關於步數的內容，請參閱 106 頁 3.8 節。

7) .....●符號表示包含有可進行子集處理的指令。

關於子集處理的詳細內容，請參閱 98 頁 3.5 節。

8) .....表示說明各指令的頁面。

2.3 順控程式指令

2.3.1 觸點指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
觸點	LD		· 邏輯運算開始 (a 觸點邏輯運算開始)		*1	●	120 頁
	LDI		· 邏輯否運算開始 (b 觸點邏輯運算開始)				
	AND		· 邏輯積 (a 觸點串列連接)				
	ANI		· 邏輯積否定 (b 觸點串列連接)				
	OR		· 邏輯和 (a 觸點並行連接)				
	ORI		· 邏輯和否定 (b 觸點並行連接)				
	LDP		· 上升脈衝運算開始		*1	●	122 頁
	LDF		· 下降脈衝運算開始				
	ANDP		· 上升脈衝串列連接				
	ANDF		· 下降脈衝串列連接				
	ORP		· 上升脈衝並行連接				
	ORF		· 下降脈衝並行連接				
	LDPI		· 上升脈衝否定運算開始		3*2	●	124 頁
	LDFI		· 下降脈衝否定運算開始		3*2		
	ANDPI		· 上升脈衝否定串列連接		4*2		
	ANDFI		· 下降脈衝否定串列連接		4*2		
	ORPI		· 上升脈衝否定並行連接		4*2		
	ORFI		· 下降脈衝否定並行連接		4*2		

*1: 步數根據所使用的軟元件而有所不同。

使用軟元件	步數
使用內部軟元件、文件寄存器 (R0 至 32767) 時	1
使用直接訪問輸入 (DX) 時	2
使用除上述以外的軟元件時	3

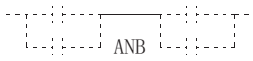
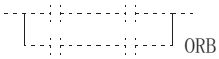
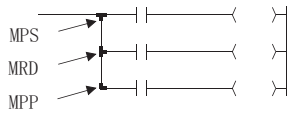
*2: 步數根據所使用的軟元件以及 CPU 模組類型而有所不同。

使用軟元件	步數
使用內部軟元件、文件寄存器 (R0 至 32767) 時	1
使用直接訪問輸入 (DX) 時	1
使用除上述以外的軟元件時	3

步數根據所使用的軟元件而有所不同。

使用軟元件	步數
使用內部軟元件、文件寄存器 (R0 至 32767) 時	基本步數
使用連號訪問方式文件寄存器 (ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)、多 CPU 共用軟元件 ((U3En\G10000 ～) 時	基本步數 +1
使用直接訪問輸入 (DX) 時	基本步數 +1
使用除上述以外的軟元件時	基本步數 +2

2.3.2 連接指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
連接	ANB		· 邏輯塊之間的 AND(邏輯塊之間的串列連接)		1	-	126 頁
	ORB		· 邏輯塊之間的 OR(邏輯塊之間的串列連接)				
	MPS		· 運算結果的記憶		1	-	127 頁
	MRD		· 通過 MPS 記憶的運算結果的讀取				
	MPP		· 通過 MPS 記憶的運算結果的讀取及重定				
	INV		· 運算結果的取反		1	-	130 頁
	MEP		· 運算結果上升脈衝化		1	-	131 頁
	MEF		· 運算結果下降脈衝化				
	EGP		· 運算結果上升脈衝化 (通過 Vn 記憶)		1	-	132 頁
	EGF		· 運算結果下降脈衝化 (通過 Vn 記憶)		*1		

*1: 步數根據所使用的軟元件以及 CPU 模組類型而有所不同。

CPU 類型	基本步數
高性能型 QCPU 過程 CPU 冗餘 CPU 通用型 QCPU LCPU	1
基本型 QCPU	2

2.3.3 連接指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
輸出	OUT	—( )—	· 軟元件的輸出		*1	-	134 頁 136 頁 139 頁 140 頁
	SET	SET D	· 軟元件的設置	*2	*1	-	142 頁 145 頁
	RST	RST D	· 軟元件的重定	*2	*1	-	143 頁 145 頁
	PLS	PLS D	· 輸入信號的上升沿時產生 1 個程式周期的脈衝。		2	-	146 頁
	PLF	PLF D	· 輸入信號的下降沿時產生 1 個程式周期的脈衝。				
	FF	FF D	· 軟元件輸出的取反		2	-	148 頁
	DELTA	DELTA D	· 直接輸出的脈衝化		2	-	149 頁
	DELTAP	DELTAP D					

*1: 步數根據所使用的軟元件而有所不同。
關於步數的內容，請參閱各指令的參閱頁面。

*2: 僅在使用報警器 (F) 時變爲 。

2.3.4 移位元指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
移位	SFT	SFT D	· 軟元件的 1 位移動		2	-	151 頁
	SFTP	SFTP D					

2.3.5 主控指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
主控	MC	MC n D	· 主控開始		2	-	153 頁
	MCR	MCR n	· 主控解除		1		

2.3.6 結束指令

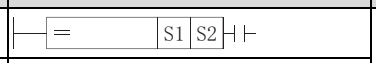
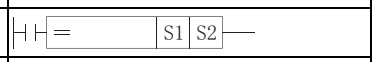
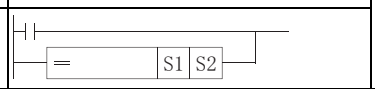
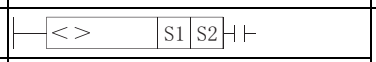
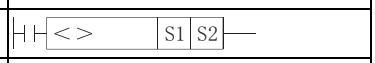
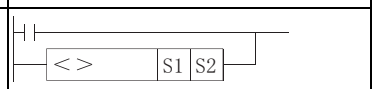
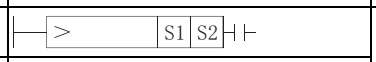
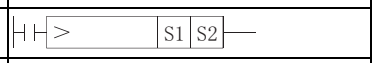
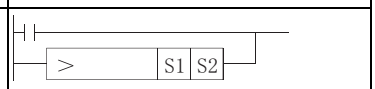
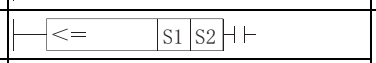
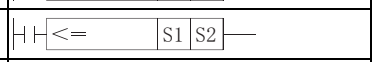
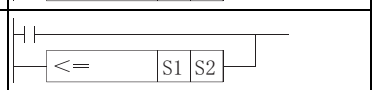
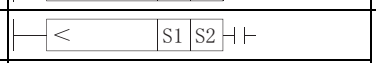
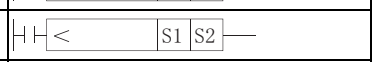
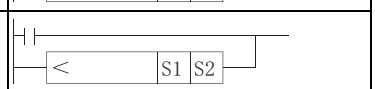
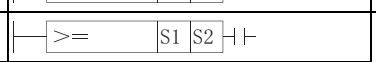
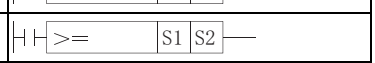
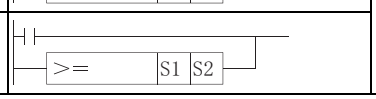
分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
結束	FEND	FEND	· 結束主程序		1	-	157 頁
	END	END	· 結束順控程式				159 頁

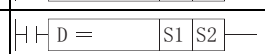
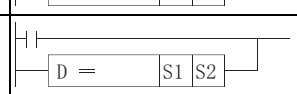
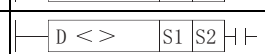
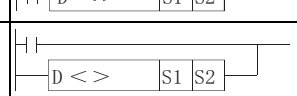
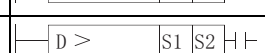
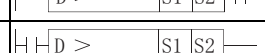
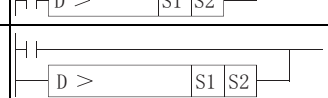
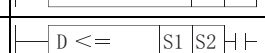
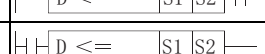
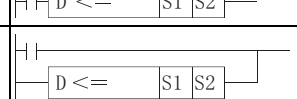
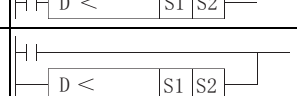
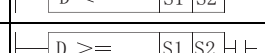
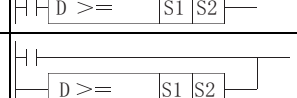
2.3.7 其他指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
停止	STOP	STOP	· 輸入條件成立後，停止順控程式的運算。 · 如果將 RUN/STOP(鍵) 開關再次置於 RUN，將執行順控程式。		1	-	161 頁
無處理	NOP	------	· 無處理 (用于程序的檫除或者空格)		1	-	162 頁
	NOPLF	NOPLF	· 無處理 (用於程式的檫除或者空格)				
	PAGE	PAGE n	· 無處理 (將後面的程式作爲第 n 頁的 0 步開始進行管理)				

2.4 基本指令

2.4.1 比較運算指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
BIN16 位元資料比較	LD=		· 當 (S1)=(S2) 時處於導通狀態 · 當 (S1) ≠ (S2) 時處於不導通狀態		3	●	166 頁
	AND=						
	OR=						
	LD<>		· 當 (S1) ≠ (S2) 時處於導通狀態 · 當 (S1)=(S2) 時處於不導通狀態		3	●	
	AND<>						
	OR<>						
	LD>		· 當 (S1)>(S2) 時處於導通狀態 · 當 (S1) ≤ (S2) 時處於不導通狀態		3	●	
	AND>						
	OR>						
	LD<=		· 當 (S1) ≤ (S2) 時處於導通狀態 · 當 (S1)>(S2) 時處於不導通狀態		3	●	
	AND<=						
	OR<=						
	LD<		· 當 (S1)<(S2) 時處於導通狀態 · 當 (S1) ≥ (S2) 時處於不導通狀態		3	●	
	AND<						
	OR<						
	LD>=		· 當 (S1) ≥ (S2) 時處於導通狀態 · 當 (S1)<(S2) 時處於不導通狀態		3	●	
	AND>=						
	OR>=						

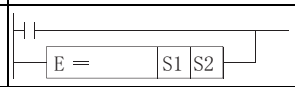
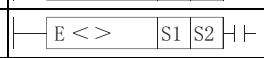
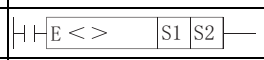
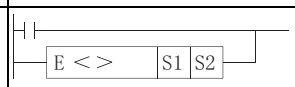
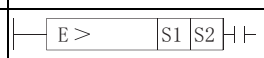
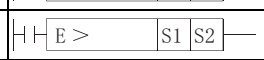
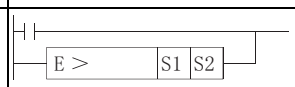
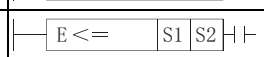
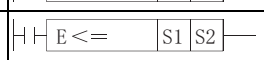
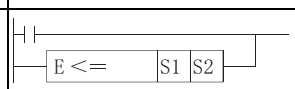
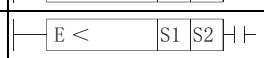
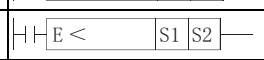
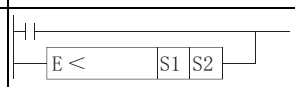
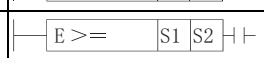
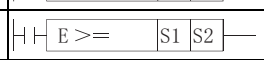
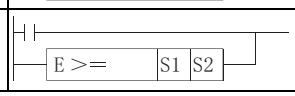
分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
BIN32 位元資料比較	LDD=		· 當 (S1+1,S1)=(S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1) ≠ (S2+1,S2) 時處於不導通狀態		*1	●	167 頁
	ANDD=						
	ORD=						
	LDD<>		· 當 (S1+1,S1) ≠ (S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1)=(S2+1,S2) 時處於不導通狀態		*1	●	
	ANDD<>						
	ORD<>						
	LDD>		· 當 (S1+1,S1)>(S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1) ≤ (S2+1,S2) 時處於不導通狀態		*1	●	
	ANDD>						
	ORD>						
	LDD<=		· 當 (S1+1,S1) ≤ (S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1)>(S2+1,S2) 時處於不導通狀態		*1	●	
	ANDD<=						
	ORD<=						
	LDD<		· 當 (S1+1,S1)<(S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1) ≥ (S2+1,S2) 時處於不導通狀態		*1	●	
	ANDD<						
	ORD<						
	LDD>=		· 當 (S1+1,S1) ≥ (S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1)<(S2+1,S2) 時處於不導通狀態		*1	●	
	ANDD>=						
	ORD>=						

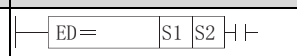
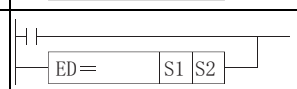
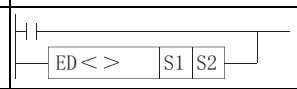
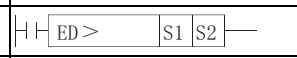
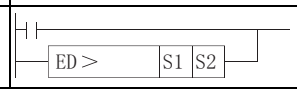
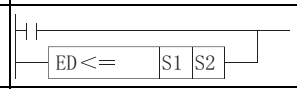
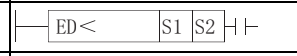
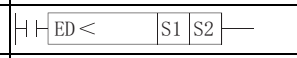
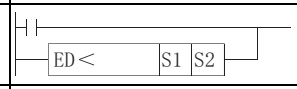
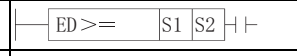
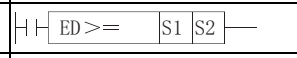
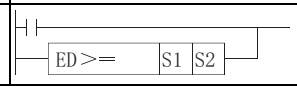
*1: 步數根據所使用的軟元件以及 CPU 類型而有所不同。

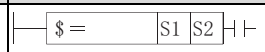
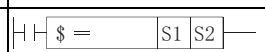
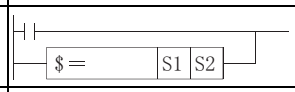
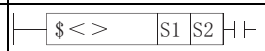
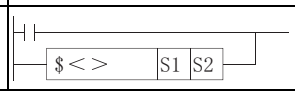
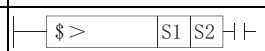
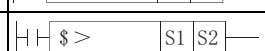
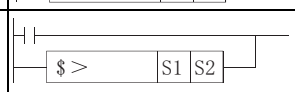
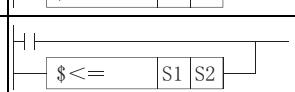
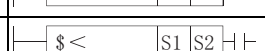
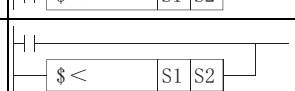
CPU 類型	使用軟元件	步數
高性能型 QCPU 過程 CPU 冗餘 CPU	· 字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） · 位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K8，無變址修飾。 · 常數： 無限制	5 注 1)
	使用除上述以外的軟元件時。	3 注 2)
基本型 QCPU 通用型 QCPU LCPU	可使用的所有軟元件	3 注 2)

注 1) 在高性能型 QCPU、過程 CPU、冗餘 CPU 的情況下，步數將會增加，但處理速度將會變快。

注 2) 根據 106 頁 3.8 節的條件步數有可能會增加。

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
浮點數據比較（單精確度）	LDE=		· 當 (S1+1,S1)=(S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1) ≠ (S2+1,S2) 時處於不導通狀態		3	-	169 頁
	ANDE=						
	ORE=						
	LDE<>		· 當 (S1+1,S1) ≠ (S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1)=(S2+1,S2) 時處於不導通狀態		3	-	
	ANDE<>						
	ORE<>						
	LDE>		· 當 (S1+1,S1)>(S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1) ≤ (S2+1,S2) 時處於不導通狀態		3	-	
	ANDE>						
	ORE>						
	LDE<=		· 當 (S1+1,S1) ≤ (S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1)>(S2+1,S2) 時處於不導通狀態		3	-	
	ANDE<=						
	ORE<=						
	LDE<		· 當 (S1+1,S1)<(S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1) ≥ (S2+1,S2) 時處於不導通狀態		3	-	
	ANDE<						
	ORE<						
	LDE>=		· 當 (S1+1,S1) ≥ (S2+1,S2) 時處於導通狀態 · 當 (S1+1,S1)<(S2+1,S2) 時處於不導通狀態		3	-	
	ANDE>=						
	ORE>=						

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
浮點數據比較（雙精度）	LDED=		· 當 (S1+3,S1+2,S1+1,S1)=(S2+3, S2+2,S2+1,S2) 時處於導通狀態 · 當 (S1+3,S1+2,S1+1,S1) ≠ (S2+3, S2+2,S2+1,S2) 時處於不導通狀態		3	-	171 頁
	ANDED=						
	ORED=						
	LDED<>		· 當 (S1+3,S1+2,S1+1,S1) ≠ (S2+3, S2+2,S2+1,S2) 時處於導通狀態 · 當 (S1+3,S1+2,S1+1,S1)=(S2+3, S2+2,S2+1,S2) 時處於不導通狀態		3	-	
	ANDED<>						
	ORED<>						
	LDED>		· 當 (S1+3,S1+2,S1+1,S1)>(S2+3, S2+2,S2+1,S2) 時處於導通狀態 · 當 (S1+3,S1+2,S1+1,S1) ≤ (S2+3, S2+2,S2+1,S2) 時處於不導通狀態		3	-	
	ANDED>						
	ORED>						
	LDED<=		· 當 (S1+3,S1+2,S1+1,S1) ≤ (S2+3, S2+2,S2+1,S2) 時處於導通狀態 · 當 (S1+3,S1+2,S1+1,S1)>(S2+3, S2+2,S2+1,S2) 時處於不導通狀態		3	-	
	ANDED<=						
	ORED<=						
	LDED<		· 當 (S1+3,S1+2,S1+1,S1)<(S2+3, S2+2,S2+1,S2) 時處於導通狀態 · 當 (S1+3,S1+2,S1+1,S1) ≥ (S2+3, S2+2,S2+1,S2) 時處於不導通狀態		3	-	
	ANDED<						
	ORED<						
	LDED>=		· 當 (S1+3,S1+2,S1+1,S1) ≥ (S2+3, S2+2,S2+1,S2) 時處於導通狀態 · 當 (S1+3,S1+2,S1+1,S1)<(S2+3, S2+2,S2+1,S2) 時處於不導通狀態		3	-	
	ANDED>=						
	ORED>=						

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
字串資料比較	LD$=		· 對字串 S1 和字串 S2 進行逐字比較。*2 · 當 (字串 S1)=(字串 S2) 時處於導通狀態 · 當 (字串 S1) ≠ (字串 S2) 時處於不導通狀態		3	-	173 頁
	AND$=						
	OR$=						
	LD$<>		· 對字串 S1 和字串 S2 進行逐字比較。*2 · 當 (字串 S1) ≠ (字串 S2) 時處於導通狀態 · 當 (字串 S1)=(字串 S2) 時處於不導通狀態		3	-	
	AND$<>						
	OR$<>						
	LD$>		· 對字串 S1 和字串 S2 進行逐字比較。*2 · 當 (字串 S1)>(字串 S2) 時處於導通狀態 · 當 (字串 S1) ≤ (字串 S2) 時處於不導通狀態		3	-	
	AND$>						
	OR$>						
	LD$<=		· 對字串 S1 和字串 S2 進行逐字比較。*2 · 當 (字串 S1) ≤ (字串 S2) 時處於導通狀態 · 當 (字串 S1)>(字串 S2) 時處於不導通狀態		3	-	
	AND$<=						
	OR$<=						
	LD$<		· 對字串 S1 和字串 S2 進行逐字比較。*2 · 當 (字串 S1)<(字串 S2) 時處於導通狀態 · 當 (字串 S1) ≥ (字串 S2) 時處於不導通狀態		3	-	
	AND$<						
	OR$<						
	LD$>=		· 對字串 S1 和字串 S2 進行逐字比較。*2 · 當 (字串 S1) ≥ (字串 S2) 時處於導通狀態 · 當 (字串 S1)<(字串 S2) 時處於不導通狀態		3	-	
	AND$>=						
	OR$>=						

*2: 進行字串比較時的比較條件如下所示：

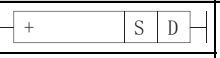
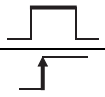
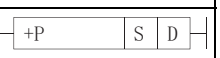
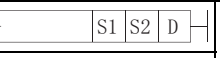
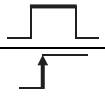
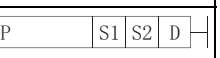
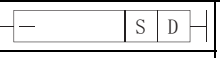
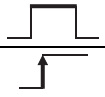
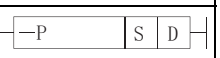
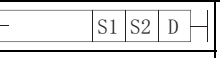
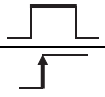
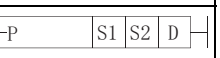
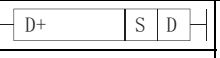
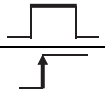
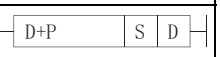
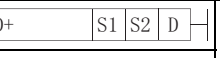
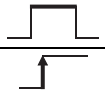
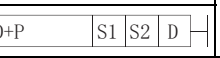
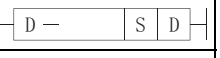
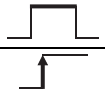
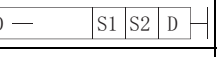
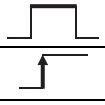
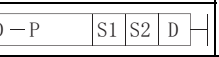
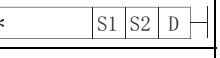
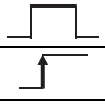
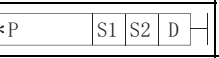
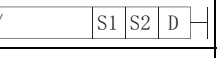
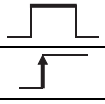
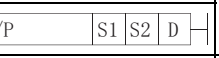
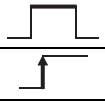
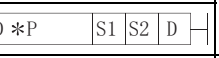
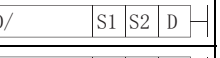
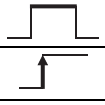
· 一致的條件 ：字串中的所有字元都一致時。

· 較大字串的條件 ：如果字串不同，取字元碼較大的字串。如果字串的長度不同，取較長的字串。

· 較小字串的條件 ：如果字串不同，取字元碼較小的字串。如果字串的長度不同，取較短的字串。

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
BIN16 位元塊資料比較	BKCMP=	BKCMP＝ S1 S2 D n	・將從 S1 開始的 n 點數據與從 S2 開始的 n 點數據以 BIN16 位元資料進行比較，並且將比較結果存儲到從 (D) 中指定的位軟元件開始的 n 點中。		5	-	175 頁
	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					
BIN32 位元塊資料比較	DBKCMP=	DBKCMP＝ S1 S2 D n	・將從 S1 中指定的軟元件開始的 n 點 BIN 32 位元資料或者常數與從 S2 中指定的軟元件開始的 n 點 BIN 32 位元資料進行比較，並且將運算結果存儲到 (D) 中指定的軟元件的後面。		5	-	178 頁
	DBKCMP<>	DBKCMP＜＞ S1 S2 D n					
	DBKCMP>	DBKCMP＞ S1 S2 D n					
	DBKCMP<=	DBKCMP＜＝ S1 S2 D n					
	DBKCMP<	DBKCMP＜ S1 S2 D n					
	DBKCMP>=	DBKCMP＞＝ S1 S2 D n					
	DBKCMP=P	DBKCMP＝P S1 S2 D n					
	DBKCMP<>P	DBKCMP＜＞P S1 S2 D n					
	DBKCMP>P	DBKCMP＞P S1 S2 D n					
	DBKCMP<=P	DBKCMP＜＝P S1 S2 D n					
	DBKCMP<P	DBKCMP＜P S1 S2 D n					
	DBKCMP>=P	DBKCMP＞＝P S1 S2 D n					

2.4.2 算術運算指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
BIN16 位加減運算	+		· (D)+(S) → (D)		3	●	181 頁
	+P						
	+		· (S1)+(S2) → (D)		4	●	182 頁
	+P						
	-		· (D)-(S) → (D)		3	●	181 頁
	-P						
	-		· (S1)-(S2) → (D)		4	●	182 頁
	-P						
BIN32 位加減運算	D+		· (D+1,D)+(S+1,S) → (D+1,D)		*1	●	184 頁
	D+P						
	D+		· (S1+1,S1)+(S2+1,S2) → (D+1,D)		*2	●	185 頁
	D+P						
	D-		· (D+1,D)-(S+1,S) → (D+1,D)		*1	●	184 頁
	D-P						
	D-		· (S1+1,S1)-(S2+1,S2) → (D+1,D)		*2	●	185 頁
	D-P						
BIN16 位乘除運算	*		· (S1) × (S2) → (D+1,D)		*3	●	187 頁
	*P						
	/		· (S1) ÷ (S2) 商 (D), 餘數 (D+1)		4 *4	●	
	/P						
BIN32 位乘除運算	D*		· (S1+1,S1) × (S2+1,S2) → (D+3,D+2,D+1,D)		4 *4	●	188 頁
	D*P						
	D/		· (S1+1,S1) ÷ (S2+1,S2) 商 (D+1,D), 餘數 (D+3,D+2)		4 *4	●	
	D/P						

*1: 步數根據所使用的軟元件以及 CPU 類型而有所不同。

CPU 類型	使用軟元件	步數
高性能型 QCPU 過程 CPU 冗餘 CPU	・字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） ・位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K8，無變址修飾。 ・常數： 無限制	5 注 1)
	使用除上述以外的軟元件時。	3 注 2)
基本型 QCPU 通用型 QCPU LCPU	可使用的所有軟元件	3 注 2)

注 1）在高性能型 QCPU、過程 CPU、冗餘 CPU 的情況下，步數將會增加，但處理速度將會變快。
注 2）根據 106 頁 3.8 節的條件步數有可能會增加。

*2: 步數根據所使用的軟元件以及 CPU 類型而有所不同。

CPU 類型	使用軟元件	步數
高性能型 QCPU 過程 CPU 冗餘 CPU	・字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） ・位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K8，無變址修飾。 ・常數： 無限制	6 注 1)
	使用除上述以外的軟元件時。	4 注 2)
基本型 QCPU	可使用的所有軟元件	4 注 2)
通用型 QCPU LCPU		3 注 2)

注 1）在高性能型 QCPU、過程 CPU、冗餘 CPU 的情況下，步數將會增加，但處理速度將會變快。
注 2）根據 106 頁 3.8 節的條件步數有可能會增加。

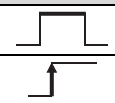
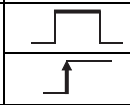
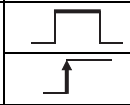
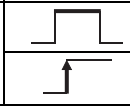
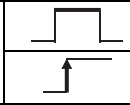
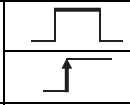
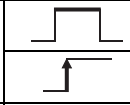
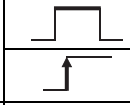
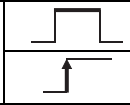
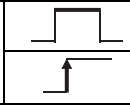
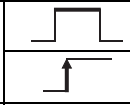
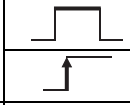
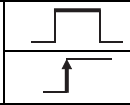
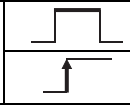
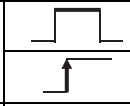
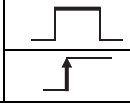
*3: 步數根據所使用的軟元件以及 CPU 類型而有所不同。

CPU 類型	使用軟元件	步數
QCPU、LCPU	・字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） ・位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K8，無變址修飾。 ・常數： 無限制	3
	使用除上述以外的軟元件時。	4 注 1)

注 1）根據 106 頁 3.8 節的條件步數有可能會增加。

*4: 只有在通用型 QCPU、LCPU 時基本步數爲 3。

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
BCD4 位數加減運算	B+	B+ S D	・(D)+(S) → (D)	↑	3	●	190 頁
	B+P	B+P S D		↑			
	B+	B+ S1 S2 D	・(S1)+(S2) → (D)	↑	4	-	192 頁
	B+P	B+P S1 S2 D		↑			
	B-	B− S D	・(D)-(S) → (D)	↑	3	●	190 頁
	B-P	B−P S D		↑			
	B-	B− S1 S2 D	・(S1)-(S2) → (D)	↑	4	-	192 頁
	B-P	B−P S1 S2 D		↑			

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
BCD8 位數加減運算	DB+	DB+ S D	· (D+1,D)+(S+1,S) → (D+1,D)		3	-	193 頁
	DB+P	DB+P S D					
	DB+	DB+ S1 S2 D	· (S1+1,S1)+(S2+1,S2) → (D+1,D)		4	-	195 頁
	DB+P	DB+P S1 S2 D					
	DB-	DB － S D	· (D+1,D)-(S+1,S) → (D+1,D)		3	-	193 頁
	DB-P	DB －P S D					
	DB-	DB － S1 S2 D	· (S1+1,S1)-(S2+1,S2) → (D+1,D)		4	-	195 頁
	DB-P	DB －P S1 S2 D					
BCD4 位數乘除運算	B*	B＊ S1 S2 D	· (S1) × (S2) → (D+1,D)		4	●	196 頁
	B*P	B＊P S1 S2 D					
	B/	B/ S1 S2 D	· (S1) ÷ (S2) 商 (D), 餘數 (D+1)		4	●	
	B/P	B/P S1 S2 D					
BCD8 位數乘除運算	DB*	DB ＊ S1 S2 D	· (S1+1,S1) × (S2+1,S2) → (D+3,D+2,D+1,D)		4	-	198 頁
	DB*P	DB ＊P S1 S2 D					
	DB/	DB/ S1 S2 D	· (S1+1,S1) ÷ (S2+1,S2) →商 (D+1,D), 餘數 (D+3,D+2)		4	●	
	DB/P	DB/P S1 S2 D					
浮點數據加減運算（單精確度）	E+	E+ S D	· (D+1,D)+(S+1,S) → (D+1,D)		3	● *6	200 頁
	E+P	E+P S D					
	E+	E+ S1 S2 D	· (S1+1,S1)+(S2+1,S2) → (D+1,D)		4 *5	● *6	202 頁
	E+P	E+P S1 S2 D					
	E-	E － S D	· (D+1,D)-(S+1,S) → (D+1,D)		3	● *6	200 頁
	E-P	E －P S D					
	E-	E － S1 S2 D	· (S1+1,S1)-(S2+1,S2) → (D+1,D)		4 *5	● *6	202 頁
	E-P	E －P S1 S2 D					
浮點數據加減運算（雙精度）	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	●	204 頁
	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	●	206 頁
	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	●	204 頁
	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	●	206 頁
	ED-P	ED－P S1 S2 D					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
浮點數據加減運算(單精確度)	E*	E* S1 S2 D	· (S1+1,S1) × (S2+1,S2) → (D+1,D)		3	● *6	208 頁
	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					
浮點數據加減運算(雙精度)	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	●	210 頁
	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	●	
	ED/P	ED/P S1 S2 D					
BIN16 位資料塊加減運算	BK+	BK+ S1 S2 D n	· 將從 (S1) 開始的 n 點 BIN16 位元資料與從 (S2) 開始的 n 點數據進行批量加法。		5	-	212 頁
	BK+P	BK+P S1 S2 D n					
	BK-	BK − S1 S2 D n	· 將從 (S1) 開始的 n 點 BIN16 位元資料與從 (S2) 開始的 n 點數據進行批量減法。		5	-	
	BK-P	BK −P S1 S2 D n					
BIN32 位資料塊加減運算	DBK+	DBK+ S1 S2 D n	· 將從 (S1) 中指定的軟元件開始的 n 點 BIN32 位元資料或常數與從 (S2) 中指定的軟元件開始 · 的 n 點 BIN32 位元資料進行加法運算後，將結果存儲到 (D) 中指定的軟元件的後面。		5	-	214 頁
	DBK+P	DBK+P S1 S2 D n					
	DBK-	DBK − S1 S2 D n	· 將從 (S1) 中指定的軟元件開始的 n 點 BIN32 位元資料或常數與從 (S2) 中指定的軟元件開始的 n 點 BIN32 位元資料進行減法運算後，將結果存儲到 (D) 中指定的軟元件的後面。		5	-	
	DBK-P	DBK −P S1 S2 D n					
字串資料合併	$+	$+ S D	· 將 (S) 中指定的字串與 (D) 中指定的字串相連接，並存儲到 (D) 的後面。		3	-	217 頁
	$+P	$+P S D					
	$+	$+ S1 S2 D	· 將 (S2) 中指定的字串與 (S1) 中指定的字串相連接，並存儲到 (D) 的後面。		4	-	218 頁
	$+P	$+P S1 S2 D					

*5: 只有在使用通用型 QCPU、LCPU 時，基本步數爲 3。
*6: 只有在使用通用型 QCPU、LCPU 時，子集才會有效。

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
BIN 資料遞增、遞減	INC	INC D	· (D)+1 → (D)		2	●	219 頁
	INCP	INCP D					
	DINC	DINC D	· (D+1,D)+1 → (D+1,D)		*7	●	221 頁
	DINCP	DINCP D					
	DEC	DEC D	· (D)-1 → (D)		2	●	219 頁
	DECP	DECP D					
	DDEC	DDEC D	· (D+1,D)-1 → (D+1,D)		*7	●	221 頁
	DDECP	DDECP D					

*7: 步數根據所使用的軟元件以及 CPU 類型而有所不同。

CPU 類型	使用軟元件	步數
高性能型 QCPU 過程 CPU 冗餘 CPU	·字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） ·位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K8，無變址修飾。 ·常數： 無限制	3 注 1)
	使用除上述以外的軟元件時。	2 注 2)
基本型 QCPU 通用型 QCPU LCPU	可使用的所有軟元件	2 注 2)

注 1）在高性能型 QCPU、過程 CPU、冗餘 CPU 的情況下，步數將會增加，但處理速度將會變快。
注 2）根據 106 頁 3.8 節的條件步數有可能會增加。

2.4.3 資料轉換指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
BCD 轉換	BCD	BCD S D	• (S) 轉換爲BCD → (D) BIN(0～9999)		3 *1	●	223 頁
	BCDP	BCDP S D					
	DBCD	DBCD S D	• (S+1, S) 轉換爲BCD → (D+1, D) BIN(0～99999999)		3 *1	●	
	DBCDP	DBCDP S D					
BIN 轉換	BIN	BIN S D	• (S) BIN轉換 → (D) BCD(0～9999)		3 *1	●	224 頁
	BINP	BINP S D					
	DBIN	DBIN S D	• (S+1, S) 轉換爲BIN → (D+1, D) BCD(0～99999999)		3 *1	●	
	DBINP	DBINP S D					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
BIN ↓ 浮點數轉換 (單精確度)	FLT	FLT S D	轉換爲實數 (S) → (D+1, D) BIN(−32768～32767)		3 *1	● *2	226 頁
	FLTP	FLTP S D					
	DFLT	DFLT S D	轉換爲實數 (S+1, S) → (D+1, D) BIN(−2147483648～2147483647)		3 *1	● *2	
	DFLTP	DFLTP S D					
BIN ↓ 浮點數轉換 (雙精度)	FLTD	FLTD S D	轉換爲實數 (S) → (D+3, D+2, D+1, D) BIN(−32768～32767)		4	●	228 頁
	FLTDP	FLTDP S D					
	DFLTD	DFLTD S D	轉換爲實數 (S+1, S) → (D+3, D+2, D+1, D) BIN(−2147483648～2147483647)		4	●	
	DFLTDP	DFLTDP S D					
浮點數 (單精確度) ↓ BIN 轉換	INT	INT S D	轉換爲BIN (S+1, S) → (D) 實數(−32768～32767)		3 *1	● *2	229 頁
	INTP	INTP S D					
	DINT	DINT S D	轉換爲BIN (S+1, S) → (D+1, D) 實數(−2147483648～2147483647)		3 *1	● *2	
	DINTP	DINTP S D					
浮點數 (雙精度) ↓ BIN 轉換	INTD	INTD S D	轉換爲BIN (S+3, S+2, S+1, S) → (D) 實數(−32768～32767)		3	●	231 頁
	INTDP	INTDP S D					
	DINTD	DINTD S D	轉換爲BIN (S+3, S+2, S+1, S) → (D+1, D) 實數(−2147483648～2147483647)		3	●	
	DINTDP	DINTDP S D					
BIN 16 位 ↕ 32 位轉換	DBL	DBL S D	轉換 (S) → (D+1, D) BIN(−32768～32767)		3	-	233 頁
	DBLP	DBLP S D					
	WORD	WORD S D	轉換 (S+1, S) → (D) BIN(−32768～32767)		3	-	234 頁
	WORDP	WORDP S D					
BIN ↓ 格雷碼轉換	GRY	GRY S D	轉換爲格雷碼 (S) → (D) BIN(−32768～32767)		3	-	235 頁
	GRYP	GRYP S D					
	DGRY	DGRY S D	轉換爲格雷碼 (S+1, S) → (D+1, D) BIN(−2147483648～2147483647)		3	-	
	DGRYP	DGRYP S D					
格雷码 ↓ BIN 轉換	GBIN	GBIN S D	轉換爲BIN資料 (S) → (D) 格雷碼(−32768～32767)		3	-	236 頁
	GBINP	GBINP S D					
	DGBIN	DGBIN S D	轉換爲BIN資料 (S+1, S) → (D+1, D) 格雷碼(−2147483648～2147483647)		3	-	
	DGBINP	DGBINP S D					

*1: 只有在使用通用型 QCPU、LCPU 時，基本步數爲 2。

*2: 只有在使用通用型 QCPU、LCPU 時，子集才會有效。

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
2 的補數（符號取反）	NEG	NEG D	• (D) → (D) BIN資料		2	-	237 頁
	NEGP	NEGP D					
	DNEG	DNEG D	• (D+1, D) → (D+1, D) BIN資料		2	-	
	DNEGP	DNEGP D					
	ENEG	ENEG D	• (D+1, D) → (D+1, D) 實數資料		2	-	239 頁
	ENEGP	ENEGP D					
	EDNEG	EDNEG D	• (D+3, D+2, D+1, D) → (D+3, D+2, D+1, D) 實數資料		3	-	240 頁
	EDNEGP	EDNEGP D					
塊轉換	BKBCD	BKBCD S D n	· 將從 (S) 開始的 n 點的 BIN 資料批量轉換爲 BCD 資料，並存儲到 (D) 的後面。		4	-	241 頁
	BKBCDP	BKBCDP S D n					
	BKBIN	BKBIN S D n	· 將從 (S) 開始的 n 點的 BCD 資料批量轉換爲 BIN 資料，並存儲到 (D) 的後面。		4	-	242 頁
	BKBINP	BKBINP S D n					
浮點數 單精確度 ↓ 雙精度	ECON	ECON S D	• (S+1, S) 轉換爲雙精度 → (D+3, D+2, D+1, D) 32位浮點型實數		3	-	244 頁
	ECONP	ECONP S D					
浮點數 雙精度 ↓ 單精確度	EDCON	EDCON S D	• (S+3, S+2, S+1, S) 轉換爲單精確度 → (D+1, D) 64位浮點型實數		3	-	245 頁
	EDCONP	EDCONP S D					

2.4.4 資料傳送指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
16 位元資料傳送	MOV	MOV S D	• (S) → (D)		*1	●	246 頁
	MOVP	MOVP S D					
32 位元資料傳送	DMOV	DMOV S D	• (S+1, S) → (D+1, D)		*2	●	
	DMOVP	DMOVP S D					
浮點數據傳送（單精確度）	EMOV	EMOV S D	• (S+1, S) → (D+1, D) 實數資料		*2	● *3	247 頁
	EMOVP	EMOVP S D					
浮點數據傳送（雙精度）	EDMOV	EDMOV S D	• (S+3, S+2, S+1, S) → (D+3, D+2, D+1, D) 實數資料		2	● *3	248 頁
	EDMOVP	EDMOVP S D					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
字串資料傳送	$MOV	$MOV S D	·將 (S) 中指定的字串傳送到 (D) 中指定的軟元件的後面。		3	-	249 頁
	$MOVP	$MOVP S D					
16 位元資料否定傳送	CML	CML S D	• $\overline{(S)}$ → (D		*1	●	251 頁
	CMLP	CMLP S D					
32 位元資料否定傳送	DCML	DCML S D	• $\overline{(S+1,S)}$ → (D+1,D		*2	●	
	DCMLP	DCMLP S D					
塊傳送	BMOV	BMOV S D n	(S) → (D) n		4	●	254 頁
	BMOVP	BMOVP S D n					
同一 16 位資料塊傳送	FMOV	FMOV S D n	(S) → (D) n		4	●	257 頁
	FMOVP	FMOVP S D n					
同一 32 位資料塊傳送	DFMOV	DFMOV S D n	(S+1,S) → (D+1,D) n		4	●	259 頁
	DFMOVP	DFMOVP S D n					
16 位元資料轉換	XCH	XCH D1 D2	• (D1) ↔ (D2)		3	●	261 頁
	XCHP	XCHP D1 D2					
32 位元資料轉換	DXCH	DXCH D1 D2	• (D1+1,D1) ↔ (D2+1,D2)		3	●	
	DXCHP	DXCHP D1 D2					
塊資料轉換	BXCH	BXCH S D n	(S) ↔ (D) n		4	-	262 頁
	BXCHP	BXCHP S D n					
高低位元組轉換	SWAP	SWAP D	(S) b15 ～ b8 b7 ～ b0 [8位][8位] → (D) b15 ～ b8 b7 ～ b0 [8位][8位]		3	-	264 頁
	SWAPP	SWAPP D					

*1: 步數根據所使用的軟元件以及 CPU 類型而有所不同。

CPU 類型	使用軟元件	步數
QCPU LCPU	·字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） ·位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K4，無變址修飾。 ·常數： 無限制	2
	使用除上述以外的軟元件時。	3 注 1)

注 1）根據 106 頁 3.8 節的條件步數有可能會增加。

*2: 步數根據所使用的軟元件以及 CPU 類型而有所不同。

CPU 類型	使用軟元件	步數
高性能型 QCPU 過程 CPU 冗餘 CPU	・字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） ・位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K8，無變址修飾。 ・常數： 無限制	3
	使用除上述以外的軟元件時。	3 注 1)
基本型 QCPU	・字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） ・位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K8，無變址修飾。 ・常數： 無限制 （使用上述軟元件 + 常數時，步數變爲 3。）	2
	使用除上述以外的軟元件時。	3 注 1)
通用型 QCPU LCPU	可使用的所有軟元件	2 注 1)

注 1）根據 106 頁 3.8 節的條件步數有可能會增加。

*3: 步數根據所使用的軟元件以及 CPU 類型而有所不同。

CPU 類型	使用軟元件	步數
QCPU LCPU	・字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） ・位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K4，無變址修飾。 ・常數： 無限制	2
	使用除上述以外的軟元件時。	3 注 1)

注 1）根據 106 頁 3.8 節的條件步數有可能會增加。

2.4.5 程式分支指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
跳轉	CJ	CJ Pn	・當輸入條件成立時跳轉到 Pn。		2	●	265 頁
	SCJ	SCJ Pn	・從輸入條件成立後的下一個掃描跳轉到 Pn。		2	●	
	JMP	JMP Pn	・無條件地跳轉到 Pn。		2	●	
	GOEND	GOEND	・當輸入條件成立時，跳轉到 END 指令。		1	-	267 頁

2.4.6 程式執行控制指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
中斷禁止	DI	DI	· 禁止執行中斷程式。		1	-	269 頁
中斷允許	EI	EI	· 解除中斷程式的執行禁止。		1	-	
中斷禁止允許設置	IMASK	IMASK S	· 對各中斷程式進行中斷禁止 / 允許設置。		2	-	
返回	IRET	IRET	· 從中斷程式返回至順控程式。		1	-	274 頁

2.4.7 I/O 刷新指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
I/O 刷新	RFS	RFS S n	· 在掃描過程中對相應的輸入輸出部分進行刷新。		3	-	275 頁
	RFSP	RFSP S n					

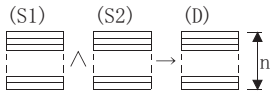
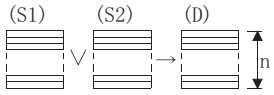
2.4.8 其他使用方便的指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
遞增 / 遞減計數器	UDCNT1	UDCNT1 S D n	(S)+0 (S)+1 遞增 遞減 遞增 Cn當前值 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 -1 -2 -3 -2 -1 0 Cn觸點		4	-	277 頁
	UDCNT2	UDCNT2 S D n	(S)+0 (S)+1 Cn當前值 0 1 2 3 4 5 4 3 2 1 0 -1 Cn觸點		4	-	279 頁
教學計時器	TTMR	TTMR D n	• (TTMR爲ON的時間)×n ⟶ (D) n=0:1, n=1:10, n=2:100		3	-	281 頁

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
特殊計時器	STMR	STMR S n D	· 根據 STMR 指令的輸入條件的 ON/OFF 狀態，對從 (D) 中指定的位軟元件開始的 4 點執行以下動作： (D)+0：OFF 延遲計時器輸出 (D)+1：OFF 後單次觸發計時器輸出 (D)+2：ON 後單次觸發計時器輸出 (D)+3：ON 延遲計時器 +OFF 延遲計時器		3	-	282 頁
就近控制	ROTC	ROTC S n1 n2 D	· 在 n1 分割的旋轉臺上，從停止位置按照最短路徑旋轉到 (S+1) 中指定的位置。		5	-	284 頁
斜坡信號	RAMP	RAMP n1 n2 D1 n3 D2	· 將 D1 中指定的軟元件資料通過 n3 次掃描使其從 n1 變爲 n2。		6	-	286 頁
脈衝密度	SPD	SPD S n D	· 將 (S) 中指定的軟元件的脈衝輸入在 n 中指定的時間內計數，並將計數結果存儲到 (D) 中指定的軟元件中。		4	-	288 頁
恒定周期脈衝輸出	PLSY	PLSY n1 n2 D	將 n1 中指定的頻率的脈衝以 n2 中指定的次數輸出到 (D) 中指定的輸出編號 (Y) 中。		4	-	289 頁
脈衝寬度調製	PWM	PWM n1 n2 D	將 n1 中指定的 ON 時間及 n2 中指定的周期的脈衝輸出到 (D) 中指定的輸出編號 (Y) 中。		4	-	290 頁
矩陣輸入	MTR	MTR S D1 D2 n	· 從 (S) 中指定的軟元件開始，依次讀取 16 點 × n 列的資料，並將其存儲到 (D2) 中指定的軟元件後面。		5	-	292 頁

2.5 應用指令

2.5.1 邏輯運算指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
邏輯積	WAND	WAND S D	• (D) ∧ (S) → (D)		3	●	295 頁
	WANDP	WANDP S D					
	WAND	WAND S1 S2 D	• (S1) ∧ (S2) → (D)		4 *1	●	296 頁
	WANDP	WANDP S1 S2 D					
	DAND	DAND S D	• (D+1, D) ∧ (S+1, S) → (D+1, D)		*2	●	295 頁
	DANDP	DANDP S D					
	DAND	DAND S1 S2 D	• (S1+1, S1) ∧ (S2+1, S2) → (D+1, D)		*3	●	296 頁
	DANDP	DANDP S1 S2 D					
	BKAND	BKAND S1 S2 D n			5	-	299 頁
	BKANDP	BKANDP S1 S2 D n					
邏輯和	WOR	WOR S D	• (D) ∨ (S) → (D)		3	●	301 頁
	WORP	WORP S D					
	WOR	WOR S1 S2 D	• (S1) ∨ (S2) → (D)		4 *1	●	303 頁
	WORP	WORP S1 S2 D					
	DOR	DOR S D	• (D+1, D) ∨ (S+1, S) → (D+1, D)		*2	●	301 頁
	DORP	DORP S D					
	DOR	DOR S1 S2 D	• (S1+1, S1) ∨ (S2+1, S2) → (D+1, D)		*3	●	303 頁
	DORP	DORP S1 S2 D					
	BKOR	BKOR S1 S2 D n			5	-	305 頁
	BKORP	BKORP S1 S2 D n					
排他邏輯和	WXOR	WXOR S D	• (D) ⩒ (S) → (D)		3	●	307 頁
	WXORP	WXORP S D					
	WXOR	WXOR S1 S2 D	• (S1) ⩒ (S2) → (D)		4 *1	●	309 頁
	WXORP	WXORP S1 S2 D					
	DXOR	DXOR S D	• (D+1, D) ⩒ (S+1, S) → (D+1, D)		*2	●	307 頁
	DXORP	DXORP S D					
	DXOR	DXOR S1 S2 D	• (S1+1, S1) ⩒ (S2+1, S2) → (D+1, D)		*3	●	309 頁
	DXORP	DXORP S1 S2 D					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
排他邏輯和	BKXOR	BKXOR S1 S2 D n	(S1) (S2) (D) n		5	-	311 頁
	BKXORP	BKXORP S1 S2 D n					
否定排他邏輯和	WXNR	WXNR S D	• $\overline{(D) \veebar (S)} \rightarrow (D)$		3	●	313 頁
	WXNRP	WXNRP S D					
	WXNR	WXNR S1 S2 D	• $\overline{(S1) \veebar (S2)} \rightarrow (D)$		4 *1	●	315 頁
	WXNRP	WXNRP S1 S2 D					
	DXNR	DXNR S D	• $\overline{(D+1, D) \veebar (S+1, S)} \rightarrow (D+1, D)$		*2	●	313 頁
	DXNRP	DXNRP S D					
	DXNR	DXNR S1 S2 D	• $\overline{(S1+1, S1) \veebar (S2+1, S2)} \rightarrow (D+1, D)$		*3	●	315 頁
	DXNRP	DXNRP S1 S2 D					
	BKXNR	BKXNR S1 S2 D n	$\overline{(S1)}$ $\overline{(S2)}$ (D) n		5	-	317 頁
	BKXNRP	BKXNRP S1 S2 D n					

*1: 只有在使用通用型 QCPU、LCPU 時，基本步數爲 3。

*2: 步數根據所使用的軟元件以及 CPU 模組類型而有所不同。

CPU 類型	使用軟元件	步數
高性能型 QCPU 過程 CPU 冗餘 CPU	・字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） ・位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K8，無變址修飾。 ・常數： 無限制	5 注 1)
	使用除上述以外的軟元件時。	3 注 2)
基本型 QCPU 通用型 QCPU LCPU	可使用的所有軟元件	3 注 2)

注 1) 在高性能型 QCPU、過程 CPU、冗餘 CPU 的情況下，步數將會增加，但處理速度將會變快。

注 2) 根據 106 頁 3.8 節的條件步數有可能會增加。

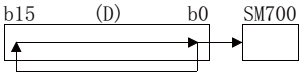
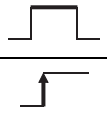
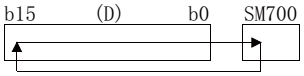
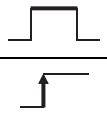
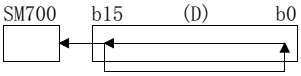
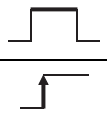
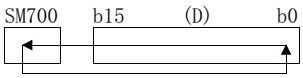
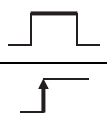
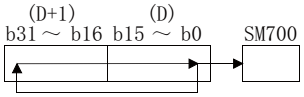
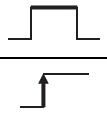
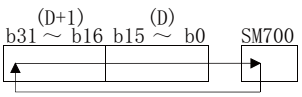
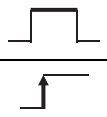
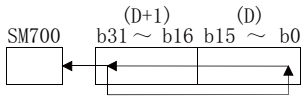
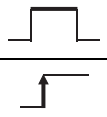
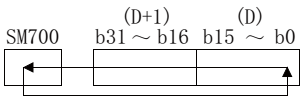
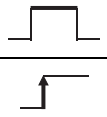
*3: 步數根據所使用的軟元件以及 CPU 模組類型而有所不同。

CPU 類型	使用軟元件	步數
高性能型 QCPU 過程 CPU 冗餘 CPU	・字軟元件： 內部軟元件（但是文件寄存器 ZR 除外） ・位軟元件： 軟元件號爲 16 的倍數，位數指定爲 K8，無變址修飾。 ・常數： 無限制	6 注 1)
	使用除上述以外的軟元件時。	4 注 2)
基本型 QCPU 通用型 QCPU LCPU	可使用的所有軟元件	4 注 2)
		3 注 2)

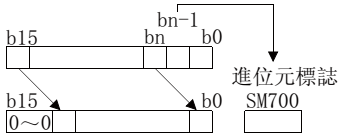
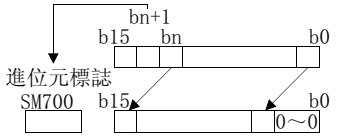
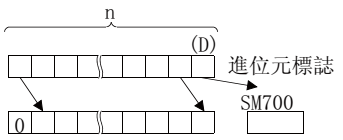
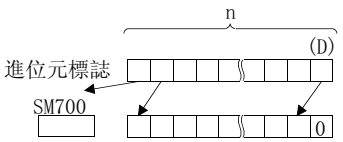
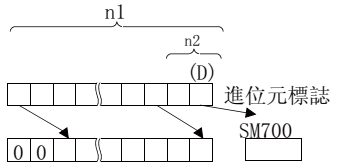
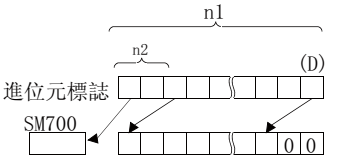
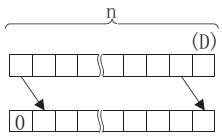
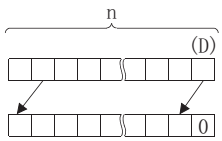
注 1) 在高性能型 QCPU、過程 CPU、冗餘 CPU 的情況下，步數將會增加，但處理速度將會變快。

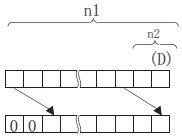
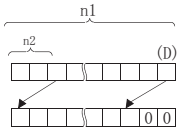
注 2) 根據 106 頁 3.8 節的條件步數有可能會增加。

2.5.2 旋轉指令

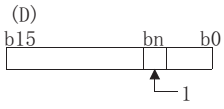
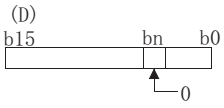
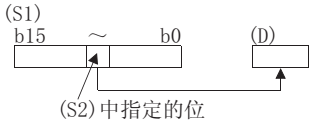
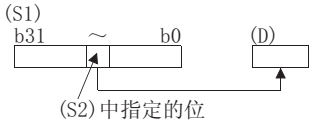
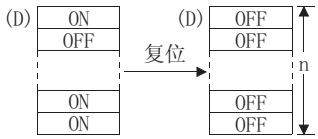
分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
16 位元資料的右旋轉	ROR	ROR D n	 向右旋轉n位 進位元標誌		3	●	319 頁
	RORP	RORP D n					
	RCR	RCR D n	 向右旋轉n位 進位元標誌		3	●	
	RCRP	RCRP D n					
16 位元資料的左旋轉	ROL	ROL D n	 進位元標誌 向左旋轉n位		3	●	321 頁
	ROLP	ROLP D n					
	RCL	RCL D n	 進位元標誌 向左旋轉n位		3	●	
	RCLP	RCLP D n					
32 位元資料的右旋轉	DROR	DROR D n	 向右旋轉n位 進位元標誌		3	●	324 頁
	DRORP	DRORP D n					
	DRCR	DRCR D n	 向右旋轉n位 進位元標誌		3	●	
	DRCRP	DRCRP D n					
32 位元資料的左旋轉	DROL	DROL D n	 進位元標誌 向左旋轉n位		3	●	326 頁
	DROLP	DROLP D n					
	DRCL	DRCL D n	 進位元標誌 向左旋轉n位		3	●	
	DRCLP	DRCLP D n					

2.5.3 移位元指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
16 位元資料的 n 位移位元	SFR	SFR D n			3	●	328 頁
	SFRP	SFRP D n					
	SFL	SFL D n			3	●	
	SFLP	SFLP D n					
n 位元資料的 1 位移位元	BSFR	BSFR D n			3	-	330 頁
	BSFRP	BSFRP D n					
	BSFL	BSFL D n			3	-	
	BSFLP	BSFLP D n					
n 位元資料的 n 位移位元	SFTBR	SFTBR D n1 n2			4	-	332 頁
	SFTBRP	SFTRP D n1 n2					
	SFTBL	SFTBL D n1 n2			4	-	
	SFTBLP	SFTBLP D n1 n2					
n 字資料的 1 字移位元	DSFR	DSFR D n			3	●	334 頁
	DSFRP	DSFRP D n					
	DSFL	DSFL D n			3	●	
	DSFLP	DSFLP D n					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
n 字資料的 n 字移位元	SFTWR	SFTWR D n1 n2			4	-	336 頁
	SFTWRP	SFTWRP D n1 n2					
	SFTWL	SFTWL D n1 n2			4	-	
	SFTWLP	SFTWLP D n1 n2					

2.5.4 位元處理指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
位設置 / 重定	BSET	BSET D n			3	●	338 頁
	BSETP	BSETP D n					
	BRST	BRST D n			3	●	
	BRSTP	BRSTP D n					
位測試	TEST	TEST S1 S2 D			4	-	339 頁
	TESTP	TESTP S1 S2 D					
	DTEST	DTEST S1 S2 D			4	-	
	DTESTP	DTESTP S1 S2 D					
位軟元件批量重定	BKRST	BKRST D n			3	-	341 頁
	BKRSTP	BKRSTP D n					

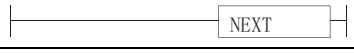
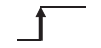
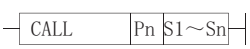
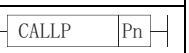
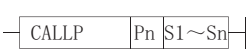
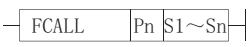
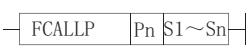
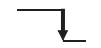
2.5.5 資料處理指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
16 位元 / 32 位元資料查找	SER	SER S1 S2 D n	(S1) (S2) n (D) :匹配號 (D+1) :匹配個數		5	-	343 頁
	SERP	SERP S1 S2 D n					
	DSER	DSER S1 S2 D n	32位 (S1) (S2) n (D) :匹配號 (D+1) :匹配個數		5	-	
	DSERP	DSERP S1 S2 D n					
位檢查	SUM	SUM S D	(S) b15 b0 (D) :1的個數		3	●	345 頁
	SUMP	SUMP S D					
	DSUM	DSUM S D	(S+1) (S) (D) :1的個數		3	●	
	DSUMP	DSUMP S D					
解碼	DECO	DECO S D n	8→256解碼 (S) 解碼 (D) n 2^n位		4	-	347 頁
	DECOP	DECOP S D n					
編碼	ENCO	ENCO S D n	256→8編碼 (S) 2^n位 編碼 (D) n		4	-	348 頁
	ENCOP	ENCOP S D n					
7 段解碼	SEG	SEG S D	b3～b0 (S) (D) 7SEG		3	●	350 頁
	SEGP	SEGP S D					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
分離・合併	DIS	DIS S D n	・將 (S) 中指定的 16 位元資料以 4 位元爲單位進行分離後，存儲到從 (D) 開始的 n 點的低 4 位中。(n ≦ 4)		4	-	352 頁
	DISP	DISP S D n					
	UNI	UNI S D n	・對 (S) 中指定的軟元件開始的 n 點的低 4 位元資料進行合併後，存儲到 (D) 中指定的軟元件中。(n ≦ 4)		4	-	353 頁
	UNIP	UNIP S D n					
	NDIS	NDIS S1 D S2	・將 (S1) 中指定的軟元件後面的資料按 (S2) 中指定的位進行分離後，依次存儲到 (D) 中指定的軟元件後面。		4	-	354 頁
	NDISP	NDISP S1 D S2					
	NUNI	NUNI S1 D S2	・將 (S1) 中指定的軟元件後面的資料與 (S2) 後面指定的各個位進行合併後，依次存儲到 (D) 中指定的軟元件後面。				
	NUNIP	NUNIP S1 D S2					
	WTOB	WTOB S D n	・將 (S) 中指定的軟元件開始的 n 點 16 位元資料以 8 位元爲單位進行分解後，依次存儲到 (D) 中指定的軟元件後面。		4	-	358 頁
	WTOBP	WTOBP S D n					
	BTOW	BTOW S D n	・將 (S) 中指定的軟元件開始的 n 點 16 位元資料的低 8 位元合併爲 16 位元後，依次存儲到 (D) 中指定的軟元件後面。				
	BTOWP	BTOWP S D n					
搜索	MAX	MAX S D n	・將 (S) 中指定的軟元件開始的 n 點的資料以 16 位元爲單位進行搜索，將最大值存儲到 (D) 中指定的軟元件中。		4	-	361 頁
	MAXP	MAXP S D n					
	MIN	MIN S D n	・將 (S) 中指定的軟元件開始的 n 點的資料以 16 位元爲單位進行搜索，將最小值存儲到 (D) 中指定的軟元件中。				362 頁
	MINP	MINP S D n					
	DMAX	DMAX S D n	・將 (S) 中指定的軟元件開始的 2 × n 點的資料以 32 位元爲單位進行搜索，將最大值存儲到 (D) 中指定的軟元件中。		4	-	361 頁
	DMAXP	DMAXP S D n					
	DMIN	DMIN S D n	・將 (S) 中指定的軟元件開始的 2 × n 點的資料以 32 位元爲單位進行搜索，將最小值存儲到 (D) 中指定的軟元件中。				362 頁
	DMINP	DMINP S D n					
排序	SORT	SORT S1 n S2 D1 D2 ・S2：1次執行的比較數 ・D1：排序結束後變爲ON的軟元件 ・D2：系統用	・將 (S1) 中指定的軟元件開始的 n 點的資料以 16 位元爲單位進行排序。 [需要 n × (n-1)/2 次掃描]		6	-	364 頁
	DSORT	DSORT S1 n S2 D1 D2 ・S2：1次執行的比較數 ・D1：排序結束後變爲ON的軟元件 ・D2：系統用	・將 (S1) 中指定的軟元件開始的 2 × n 點的資料以 32 位元爲單位進行排序。 [需要 n × (n-1)/2 次掃描]				

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
合計值計算	WSUM	WSUM S D n	· 將 (S) 中指定的軟元件開始的 n 點的 16 位元 BIN 資料全部進行加法運算後，存儲到 (D) 中指定的軟元件中。		4	-	367 頁
	WSUMP	WSUMP S D n					
	DWSUM	DWSUM S D n	· 將 (S) 中指定的軟元件開始的 n 點的 32 位元 BIN 資料全部進行加法運算後，存儲到 (D) 中指定的軟元件中。				368 頁
	DWSUMP	DWSUMP S D n					
平均值計算	MEAN	MEAN S D n	· 對 (S) 中指定的軟元件開始的 n 點的 (以 16 位元爲單位) 進行平均值計算後，存儲到 (D) 中指定的軟元件中。		4	-	369 頁
	MEANP	MEANP S D n					
	DMEAN	DMEAN S D n	· 對 (S) 中指定的軟元件開始的 n 點的 (以 32 位元爲單位) 進行平均值計算後，存儲到 (D) 中指定的軟元件中。				
	DMEANP	DMEANP S D n					

2.5.6 結構化指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
重復	FOR		· 在 FOR ～ NEXT 之間執行 n 次。		2	-	371 頁
	NEXT				1	-	
	BREAK		· 強制結束 FOR ～ NEXT 之間的執行後，跳轉到指標 Pn 處。		3	-	372 頁
	BREAKP						
副程式調用	CALL	 	· 當輸入條件成立時執行副程式 Pn。 (S1 ～ Sn 是至副程式的變數。n ≤ 5)		*1 2 + n	● *3	374 頁
	CALLP	 					
	RET		· 從副程式處返回。		1	-	377 頁
	FCALL	 	· 當輸入條件不成立時不執行副程式 Pn。 (S1 ～ Sn 是至副程式的變數。n ≤ 5)		*1 2 + n	-	378 頁
	FCALLP	 					
	ECALL	 *:檔案名	· 當輸入條件成立時執行指定程式的副程式 Pn。 (S1 ～ Sn 是至副程式的變數。n ≤ 5)		*2 3 + n	-	382 頁
	ECALLP	 *:檔案名					

*1: n 表示至副程式的變數數。
*2: n 表示至副程式的變數數與程式名的合計。
程式名的步數變爲（程式中的字元數÷ 2) 步（小數點以下被進位）
*3: 只有在使用通用型 QCPU、LCPU 時，子集才會有效。

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
副程式調用	EFCALL	EFCALL ＊ Pn EFCALL ＊ Pn S1～Sn ＊:檔案名	・當輸入條件不成立時不執行指定程式的副程式 Pn。 (S1 ～ Sn 是至副程式的變數。n ≤ 5)	⊓	*2 3 + n	-	387 頁
	EFCALLP	EFCALLP ＊ Pn EFCALLP ＊ Pn S1～Sn ＊:檔案名		↓			
	XCALL	XCALL Pn S1～Sn	・當輸入條件成立時執行副程式 Pn。 ・當輸入條件不成立時不執行副程式 Pn。 (S1 ～ Sn 是至副程式的變數。n ≤ 5)	⊓	*1 2 + n	-	390 頁
選擇刷新	COM	COM	・執行智慧功能模組的自動刷新、鏈結刷新的自動刷新及通信處理。		1	-	395 頁
			・執行智慧功能模組的自動刷新、鏈結刷新、CPU 共用記憶體的自動刷新及通信處理。		1	-	397 頁
	CCOM	CCOM	・通過輸入條件成立執行智慧功能模組的自動刷新、CPU 共用記憶體的自動刷新及通信處理。	⊓	1	-	400 頁
	CCOMP	CCOMP		↑	1	-	
固定變址修飾	IX	IX S 軟元件修飾梯形圖	・對軟元件修飾梯形圖中使用的各軟元件進行變址修飾。。		2	-	400 頁
	IXEND	IXEND			1	-	
	IXDEV	IXDEV	・將 IX ～ IXEND 之間用於進行變址修飾的修飾值存儲到 D 中指定的軟元件的後面。		1	-	403 頁
	IXSET	IXSET S D 修飾值的指定			3	-	

*4: n 表示至副程式的引數的個數。

*5: n 表示至副程式的引數個數與程式名的合計。
程式名的步數變爲（程式中的字元數 ÷ 2）步（小數點以下被進位）

2.5.7 資料表操作指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
資料表處理	FIFW	FIFW S D	(S) (D) 指標 指標+1 指標+1的軟元件		3	-	406 頁
	FIFWP	FIFWP S D					
	FIFR	FIFR S D	(S) 指標 指標−1 (D)		3	-	407 頁
	FIFRP	FIFRP S D					
	FPOP	FPOP S D	(S) 指標 指標−1 (D) 指標+1的軟元件		3	-	409 頁
	FPOPP	FPOPP S D					
	FDEL	FDEL S D n	(S) 指標 指標−1 (D) n中指定		4	-	411 頁
	FDELP	FDELP S D n					
	FINS	FINS S D n	(S) (D) 指標 指標+1 n中指定		4	-	411 頁
	FINSP	FINSP S D n					

2.5.8 緩衝記憶體訪問指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
資料讀取	FROM	FROM n1 n2 D n3	· 從智慧功能模組以 16 位元爲單位進行資料讀取。		5	-	413 頁
	FROMP	FROMP n1 n2 D n3					
	DFRO	DFRO n1 n2 D n3	· 從智慧功能模組以 32 位元爲單位進行資料讀取。		5	-	
	DFROP	DFROP n1 n2 D n3					
資料寫入	TO	TO n1 n2 S n3	· 從智慧功能模組以 16 位元爲單位進行資料寫入。		5	-	415 頁
	TOP	TOP n1 n2 S n3					
	DTO	DTO n1 n2 S n3	· 從智慧功能模組以 32 位元爲單位進行資料寫入。		5	-	
	DTOP	DTOP n1 n2 S n3					

2.5.9 顯示指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
ASCII 列印	PR	＊SM701爲OFF時 PR S D	· 將從 (S) 中指定的軟元件開始至 00H 爲止的 ASCII 碼輸出到輸出模組中。	↑	3	-	418 頁
	PR	＊SM701爲ON時 PR S D	· 從 (S) 中指定的軟元件開始，將 8 點 (16 個字元) 的 ASCII 碼輸出到輸出模組中。				
	PRC	PRC S D	· 將 (S) 中指定的軟元件的注釋轉換爲 ASCII 碼後，輸出到輸出模組中。				421 頁
重定	LEDR	LEDR	· 進行報警器的重定以及 LED 顯示器的顯示重定。	↑	1	-	423 頁

2.5.10 調試 · 故障診斷指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
檢查	CHKST	CHKST	· 當執行 CHKST 時執行 CHK 指令。 · 當 CHKST 未執行時，跳轉至 CHK 指令的下一步。		1	-	426 頁
	CHK	CHK 檢查條件	· 正常時→ SM80：OFF ； SD80：0 · 異常時→ SM80：ON ； SD80：故障號				
	CHKCIR	CHKCIR	· 通過 CHK 指令檢查的梯形圖模式的變更開始。		1	-	429 頁
	CHKEND	CHKEND	· 通過 CHK 指令檢查的梯形圖模式的變更結束。				

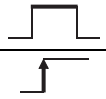
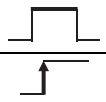
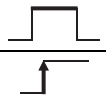
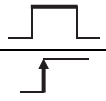
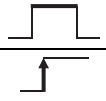
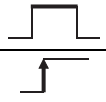
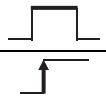
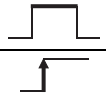
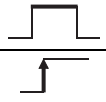
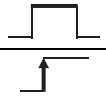
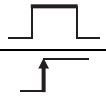
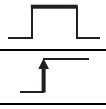
2.5.11 字串處理指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
BIN ↓ 10 進制 ASCII	BINDA	BINDA S D	· 將 (S) 中指定的 1 字 BIN 值轉換爲 5 位數 10 進制 ASCII 值後，存儲到 (D) 中指定的字軟元件中。		3	-	433 頁
	BINDAP	BINDAP S D					
	DBINDA	DBINDA S D	· 將 (S) 中指定的 2 字 BIN 值轉換爲 10 位數 10 進制 ASCII 值後，存儲到 (D) 中指定的字軟元件中。		3	-	
	DBINDAP	DBINDAP S D					
BIN ↓ 16 進制 ASCII	BINHA	BINHA S D	· 將 (S) 中指定的 1 字 BIN 值轉換爲 4 位數 16 進制 ASCII 值後，存儲到 (D) 中指定的字軟元件中。		3	-	435 頁
	BINHAP	BINHAP S D					
	DBINHA	DBINHA S D	· 將 (S) 中指定的 2 字 BIN 值轉換爲 8 位數 16 進制 ASCII 值後，存儲到 (D) 中指定的字軟元件中。		3	-	
	DBINHAP	DBINHAP S D					
BCD ↓ 10 進制 ASCII	BCDDA	BCDDA S D	· 將 (S) 中指定的 1 字 BCD 值轉換爲 4 位數 10 進制 ASCII 值後，存儲到 (D) 中指定的字軟元件號後面。		3	-	438 頁
	BCDDAP	BCDDAP S D					
	DBCDDA	DBCDDA S D	· 將 (S) 中指定的 2 字 BCD 值轉換爲 8 位數 10 進制 ASCII 值後，存儲到 (D) 中指定的字軟元件號後面。		3	-	
	DBCDDAP	DBCDDAP S D					
100 進制 ASCII ↓ BIN	DABIN	DABIN S D	· 將 (S) 中指定的 5 位數 10 進制 ASCII 值轉換爲 1 字 BIN 值後，存儲到 (D) 中指定的字軟元件號後面。		3	-	440 頁
	DABINP	DABINP S D					
	DDABIN	DDABIN S D	· 將 (S) 中指定的 10 位數 10 進制 ASCII 值轉換爲 2 字 BIN 值後，存儲到 (D) 中指定的字軟元件號後面。		3	-	
	DDABINP	DDABINP S D					
16 進制 ASCII ↓ BIN	HABIN	HABIN S D	· 將 (S) 中指定的 4 位數 16 進制 ASCII 值轉換爲 1 字 BIN 值後，存儲到 (D) 中指定的字軟元件號後面。		3	-	443 頁
	HABINP	HABINP S D					
	DHABIN	DHABIN S D	· 將 (S) 中指定的 8 位數 16 進制 ASCII 值轉換爲 2 字 BIN 值後，存儲到 (D) 中指定的字軟元件號後面。		3	-	
	DHABINP	DHABINP S D					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
10 進制 ASCII ↓ BCD	DABCD	DABCD S D	· 將 (S) 中指定的 4 位數 10 進制 ASCII 值轉換爲 1 字 BCD 值後，存儲到 (D) 中指定的字軟元件號後面。		3	-	444 頁
	DABCDP	DABCDP S D					
	DDABCD	DDABCD S D	· 將 (S) 中指定的 8 位數 10 進制 ASCII 值轉換爲 2 字 BCD 值後，存儲到 (D) 中指定的字軟元件號後面。		3	-	
	DDABCDP	DDABCDP S D					
軟元件注釋的讀取	COMRD	COMRD S D	· 將 (S) 中指定的軟元件的注釋資料存儲到 (D) 中指定的軟元件中。		3	-	447 頁
	COMRDP	COMRDP S D					
字串的長度檢測	LEN	LEN S D	· 將 (S) 中指定的軟元件中存儲的字串資料的長度 (字元數) 存儲到 (D) 中指定的軟元件中。		3	-	449 頁
	LENP	LENP S D					
BIN ↓ 10 進制字串	STR	STR S1 S2 D	· 將 (S2) 中指定的 1 字 BIN 值轉換爲 (S1) 中指定的總位元數和小數部分位元數的 10 進制字串後，存儲到 (D) 中指定的軟元件中。		4	-	450 頁
	STRP	STRP S1 S2 D					
	DSTR	DSTR S1 S2 D	· 將 (S2) 中指定的 2 字 BIN 值轉換爲 (S1) 中指定的總位元數和小數部分位元數的 10 進制字串後，存儲到 (D) 中指定的軟元件中。		4	-	
	DSTRP	DSTRP S1 S2 D					
10 進制字串 ↓ BIN	VAL	VAL S D1 D2	· 將 (S) 中指定的包含有小數點的字串轉換爲 1 字 BIN 值及小數部分位元數後，存儲到 (D1)、(D2) 中指定的軟元件中。		4	-	454 頁
	VALP	VALP S D1 D2					
	DVAL	DVAL S D1 D2	· 將 (S) 中指定的包含有小數點的字串轉換爲 2 字 BIN 值及小數部分位元數後，存儲到 (D1)、(D2) 中指定的軟元件中。		4	-	
	DVALP	DVALP S D1 D2					
浮點數 ↓ 字串	ESTR	ESTR S1 S2 D	· 將 (S) 中指定的 32 位浮點數據轉換爲字串後，存儲到 (D) 中指定的軟元件中。		4	-	458 頁
	ESTRP	ESTRP S1 S2 D					
字串 ↓ 浮點數	EVAL	EVAL S D	· 將 (S) 中指定的字串轉換爲 32 位元浮點數據後，存儲到 (D) 中指定的軟元件中。		3	-	463 頁
	EVALP	EVALP S D					
16 進制 BIN ↓ ASCII	ASC	ASC S D n	· 將 (S) 中指定的軟元件號後面的 1 字 BIN 值轉換爲 16 進制 ASCII 後，以 n 中指定的字元數存儲到 (D) 中指定的字軟元件號後面。		4	-	466 頁
	ASCP	ASCP S D n					
ASCII ↓ 16 進制 BIN	HEX	HEX S D n	· 將 (S) 中指定的字軟元件後面的 16 進制 ASCII 資料以 n 中指定的字元數轉換爲 BIN 值後，存儲到 (D) 中指定的軟元件號後面。		4	-	468 頁
	HEXP	HEXP S D n					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
字串處理	RIGHT	RIGHT S D n	· 將 (S) 中指定的字串的最後字元開始的 n 個字元存儲到 (D) 中指定的軟元件中。		4	-	470 頁
	RIGHTP	RIGHTP S D n					
	LEFT	LEFT S D n	· 將 (S) 中指定的字串的起始字元開始的 n 個字元存儲到 (D) 中指定的軟元件中。				
	LEFTP	LEFTP S D n					
	MIDR	MIDR S1 D S2	· 在 (S1) 中指定的字串中，將從 (S2) 中指定位置開始的指定數量的字元存儲到 (D) 中指定的軟元件中。		4	-	472 頁
	MIDRP	MIDRP S1 D S2					
	MIDW	MIDW S1 D S2	· 將從 (S1) 中指定的字串開始的指定數量的字元，存儲到 (D) 中指定的字串的 (S2) 中指定的位置處。				
	MIDWP	MIDWP S1 D S2					
	INSTR	INSTR S1 S2 D n	· 從 (S2) 中指定的字串的第 n 個字元開始搜索 (S1) 中指定的字串，並將匹配的位置存儲到 (D) 中。		5	-	475 頁
	INSTRP	INSTRP S1 S2 D n					
	STRINS	STRINS S D n	· 將 (S) 中指定的字串資料插入到 (D) 中指定的字串資料的從起始開始的第 (n) 個字元 (插入位置) 處。		4	-	477 頁
	STRINSP	STRINSP S D n					
	STRDEL	STRDEL D n1 n2	· 在 (D) 中指定的字串資料中，從起始算起的第 (n1) 個字元 (刪除開始位置) 開始，刪除 (n2) 中指定的字元數。		4	-	479 頁
	STRDELP	STRDELP D n1 n2					
浮點數 ↓ BCD 分解	EMOD	EMOD S1 S2 D	· 將 (S1) 中指定的 32 位浮點數據按照 (S2) 中指定的小數部分的位元數轉換爲 BCD 資料後，存儲到 (D) 中指定軟元件中。		4	-	480 頁
	EMODP	EMODP S1 S2 D					
BCD ↓ 浮點數	EREXP	EREXP S1 S2 D	· 將 (S1) 中指定的 BCD 資料按照 (S2) 中指定的小數部分的位元數轉換爲 32 位浮點數據後，存儲到 (D) 中指定的軟元件中。		4	-	482 頁
	EREXPP	EREXPP S1 S2 D					

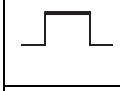
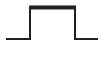
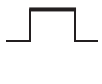
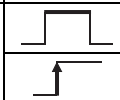
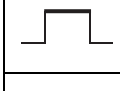
2.5.12 特殊函數指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
三角函數（浮點數單精確度）	SIN	SIN S D	· Sin (S+1 , S) ⟶ (D+1 , D)		3	-	484 頁
	SINP	SINP S D					
	COS	COS S D	·Cos (S+1 , S) ⟶ (D+1 , D)		3	-	487 頁
	COSP	COSP S D					
	TAN	TAN S D	·Tan(S+1 , S) ⟶ (D+1 , D)		3	-	490 頁
	TANP	TANP S D					
	ASIN	ASIN S D	·Sin^{-1}(S+1 , S) ⟶ (D+1 , D)		3	-	493 頁
	ASINP	ASINP S D					
	ACOS	ACOS S D	·Cos^{-1}(S+1 , S) ⟶ (D+1 , D)		3	-	496 頁
	ACOSP	ACOSP S D					
	ATAN	ATAN S D	·Tan^{-1}(S+1 , S) ⟶ (D+1 , D)		3	-	499 頁
	ATANP	ATANP S D					
三角函數（浮點數雙精度）	SIND	SIND S D	·Sin(S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D)		3	-	485 頁
	SINDP	SINDP S D					
	COSD	COSD S D	·Cos(S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D)		3	-	488 頁
	COSDP	COSDP S D					
	TAND	TAND S D	·Tan(S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D)		3	-	492 頁
	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	-	495 頁
	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	-	498 頁
	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	-	501 頁
	ATANDP	ATANDP S D					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
度 ↕ 弧度轉換	RAD	RAD S D	• (S+1, S) ⟶ (D+1, D) 度 → 弧度轉換		3	-	502 頁
	RADP	RADP S D					
	RADD	RADD S D	• (S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D) 度 → 弧度轉換		3	-	503 頁
	RADDP	RADDP S D					
	DEG	DEG S D	• (S+1, S) ⟶ (D+1, D) 弧度 → 度轉換		3	-	505 頁
	DEGP	DEGP S D					
	DEGD	DEGD S D	• (S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D) 弧度 → 度轉換		3	-	506 頁
	DEGDP	DEGDP S D					
冪	POW	POW S1 S2 D	· (S1+1, S1) $^{(S2+1, S2)}$ ⟶ (D+1, D)		4	-	508 頁
	POWP	POWP S1 S2 D					
	POWD	POWD 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	-	509 頁
	POWDP	POWDP S1 S2 D					
平方根	SQR	SQR S D	· $\sqrt{(S+1, S)}$ ⟶ (D+1 , D)		3	-	510 頁
	SQRP	SQRP S D					
	SQRD	SQRD S D	· $\sqrt{(S+3, S+2, S+1, S)}$ ⟶ (D+3, D+2, D+1, D)		3	-	512 頁
	SQRDP	SQRDP S D					
指數運算	EXP	EXP S D	· $e^{(S+1, S)}$ ⟶ (D+1 , D)		3	-	513 頁
	EXPP	EXPP S D					
	EXPD	EXPD S D	· $e^{(S+3, S+2, S+1, S)}$ ⟶ (D+3, D+2, D+1, D)		3	-	515 頁
	EXPDP	EXPDP S D					
自然對數	LOG	LOG S D	· Log_e (S+1 , S) ⟶ (D+1 , D)		3	-	516 頁
	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	-	518 頁
	LOGDP	LOGDP S D					
常用對數	LOG10	LOG10 S D	· log10 (S+1, S) ⟶ (D+1, D)		3	-	519 頁
	LOG10P	LOG10P S D					
	LOG10D	LOG10D S D	· log10 (S+3, S+2S+1, S) ⟶ (D+3, D+2, D+1, D)		3	-	520 頁
	LOG10DP	LOG10DP S D					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
亂數產生	RND	RND D	· 產生一個亂數 (0 ～ 32767) 後，存儲到 (D) 中指定的軟元件中。		2	-	521 頁
	RNDP	RNDP D					
亂數系列變更	SRND	SRND S	· 根據存儲在 (S) 中指定的軟元件中的 16 位元 BIN 資料的內容，對亂數系列進行變更。				
	SRNDP	SRNDP S					
平方根	BSQR	BSQR S D	• $\sqrt{(S)}$ ⟶ (D)+0 整數部分 +1 小數部分		3	-	522 頁
	BSQRP	BSQRP S D					
	BDSQR	BDSQR S D	• $\sqrt{(S+1,\ S)}$ ⟶ (D)+0 整數部分 +1 小數部分		3	-	
	BDSQRP	BDSQRP S D					
三角函數 (BCD 型)	BSIN	BSIN S D	• Sin(S) ⟶ (D)+0 符號 +1 整數部分 +2 小數部分		3	-	524 頁
	BSINP	BSINP S D					
	BCOS	BCOS S D	• Cos(S) ⟶ (D)+0 符號 +1 整數部分 +2 小數部分		3	-	526 頁
	BCOSP	BCOSP S D					
	BTAN	BTAN S D	• Tan(S) ⟶ (D)+0 符號 +1 整數部分 +2 小數部分		3	-	528 頁
	BTANP	BTANP S D					
	BASIN	BASIN S D	• $Sin^{-1}(S)$ ⟶ (D)+0 符號 +1 整數部分 +2 小數部分		3	-	530 頁
	BASINP	BASINP S D					
	BACOS	BACOS S D	• $Cos^{-1}(S)$ ⟶ (D)+0 符號 +1 整數部分 +2 小數部分		3	-	532 頁
	BACOSP	BACOSP S D					
	BATAN	BATAN S D	• $Tan^{-1}(S)$ ⟶ (D)+0 符號 +1 整數部分 +2 小數部分		3	-	534 頁
	BATANP	BATANP S D					

2.5.13 資料控制指令

分類	指令符號	符號	執行條件	執行條件	基本步數	子集	參閱頁面
上下限控制	LIMIT	LIMIT S1 S2 S3 D	· (S3)<(S1) 時 ..... 將 (S1) 的值存儲到 (D) 中 · (S1) ≤ (S3) ≤ (S2) 時 ..... 將 (S3) 的值存儲到 (D) 中 · (S2)<(S3) 時 ..... 將 (S2) 的值存儲到 (D) 中		5	-	536 頁
	LIMITP	LIMITP S1 S2 S3 D					
	DLIMIT	DLIMIT S1 S2 S3 D	· ((S3+1,S3)<(S1+1,S1) 時 ..... 將 (S1+1，S1) 的值存儲到 ((D)+1，(D)) 中 · (S1+1,S1) ≤ (S3+1,S3)<(S2+1,S2) 時 ... 將 (S3+1，S3) 的值存儲到 ((D)+1，(D)) 中 · (S2,S2+1)<(S3,S3+1) 時 ..... 將 (S2+1，S2) 的值存儲到 ((D)+1，(D)) 中		5	-	
	DLIMITP	DLIMITP S1 S2 S3 D					
死區控制	BAND	BAND S1 S2 S3 D	· (S1) ≤ (S3) ≤ (S2) 時 ...0(D) · (S3)<(S1) 時 .....(S3)-(S1) → (D) · (S2)<(S3) 時 .....(S3)-(S2) → (D)		5	-	538 頁
	BANDP	BANDP S1 S2 S3 D					
	DBAND	DBAND S1 S2 S3 D	· (S1+1,S1) ≤ (S3+1,S3) ≤ (S2+1,S2) 時 ...0 → ((D)+1,(D)) · (S3+1,S3)<(S1+1,S1) 時 ...(S3+1,S3)-(S1+1,S1) → ((D)+1,(D)) · (S2+1,S2)<(S3+1,S3) 時 ...(S3+1,S3)-(S2+1,S2) → ((D)+1,(D))		5	-	
	DBANDP	DBANDP S1 S2 S3 D					
區域控制	ZONE	ZONE S1 S2 S3 D	· (S3)=0 時 ...0 → (D) · (S3)>0 時 ...(S3)+(S2) → (D) · (S3)<0 時 ...(S3)(S1) → (D)		5	-	541 頁
	ZONEP	ZONEP S1 S2 S3 D					
	DZONE	DZONE S1 S2 S3 D	· (S3+1,S3)=0 時 ...0 → ((D)+1,(D)) · (S3+1,S3)>0 時 ...(S3+1,S3)+(S2+1,S2) → ((D)+1,(D)) · (S3+1,S3)<0 時 ...(S3+1,S3)+(S1+1,S1) → ((D)+1,(D))		5	-	
	DZONEP	DZONEP S1 S2 S3 D					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
點座標資料	SCL	SCL S1 S2 D	· 對 (S2) 中指定的標度用轉換資料 (16 位元資料單位) 通過 (S1) 中指定的輸入值進行標度，將運算結果存儲到 (D) 中指定的軟元件中。 標度轉換是基於存儲在 (S2) 中指定的軟元件後面的標度用轉換資料進行的。		4	-	543 頁
	SCLP	SCLP S1 S2 D					
	DSCL	DSCL S1 S2 D	· 對 (S2) 中指定的標度用轉換資料 (32 位元資料單位) 通過 (S1) 中指定的輸入值進行標度，將運算結果存儲到 (D) 中指定的軟元件中。 標度轉換是基於存儲在 (S2) 中指定的軟元件後面的標度用轉換資料進行的。		4	-	
	DSCLP	DSCLP S1 S2 D					
X/Y 座標資料	SCL2	SCL2 S1 S2 D	· 對 (S2) 中指定的標度用轉換資料 (16 位元資料單位) 通過 (S1) 中指定的輸入值進行標度，將運算結果存儲到 (D) 中指定的軟元件中。 標度轉換是基於存儲在 (S2) 中指定的軟元件後面的標度用轉換資料進行的。		4	-	546 頁
	SCL2P	SCL2P S1 S2 D					
	DSCL2	DSCL2 S1 S2 D	· 對 (S2) 中指定的標度用轉換資料 (32 位元資料單位) 通過 (S1) 中指定的輸入值進行標度，將運算結果存儲到 (D) 中指定的軟元件中。 標度轉換是基於存儲在 (S2) 中指定的軟元件後面的標度用轉換資料進行的。		4	-	
	DSCL2P	DSCL2P S1 S2 D					

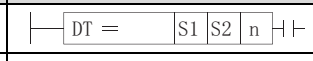
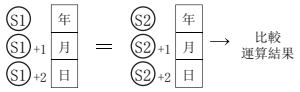
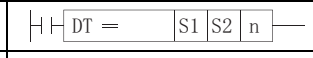
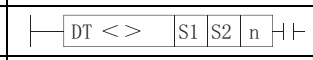
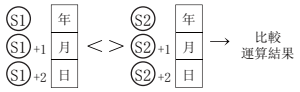
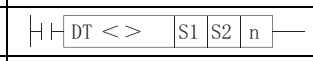
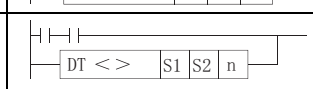
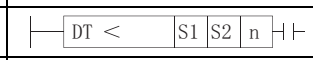
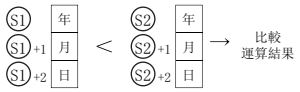
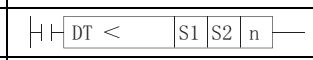
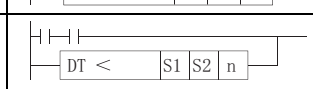
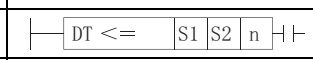
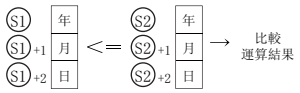
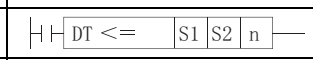
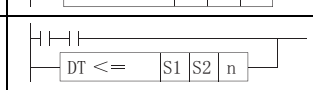
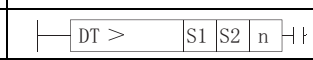
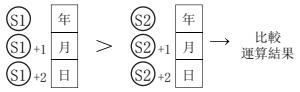
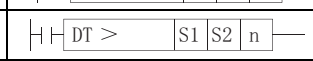
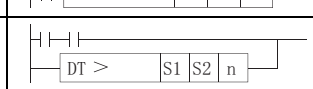
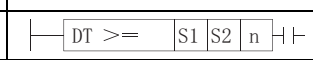
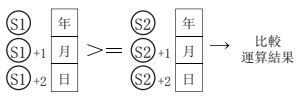
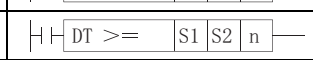
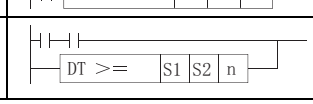
2.5.14 切換指令

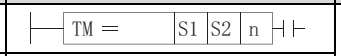
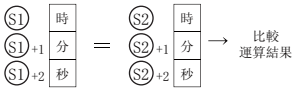
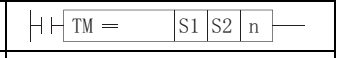
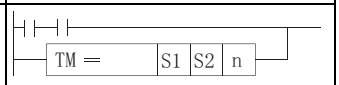
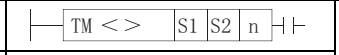
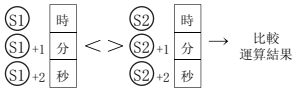
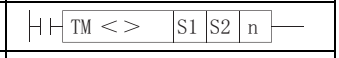
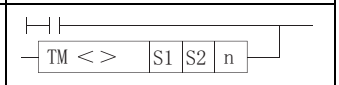
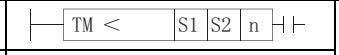
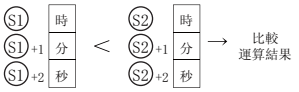
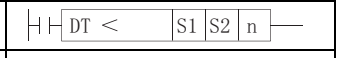
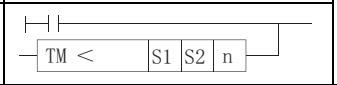
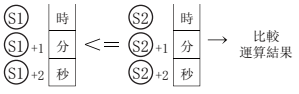
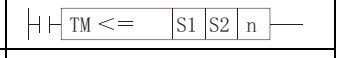
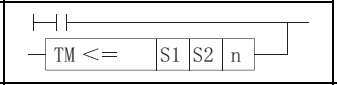
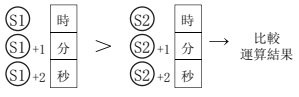
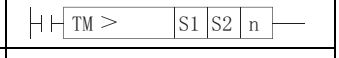
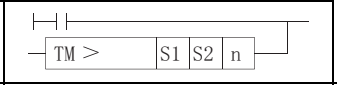
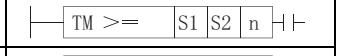
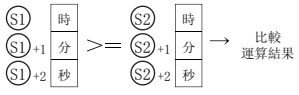
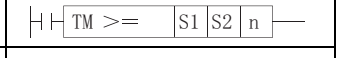
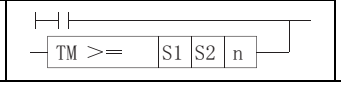
分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
塊號切換	RSET	RSET S	· 將擴展文件寄存器的塊號變更爲 (S) 中指定的編號。		2	-	549 頁
	RSETP	RSETP S					
文件設置	QDRSET	QDRSET 檔案名	· 對作爲文件寄存器使用的檔案名進行設置。		*1 2 + n	-	550 頁
	QDRSETP	QDRSETP 檔案名					
	QCDSET	QCDSET 檔案名	· 對作爲注釋文件使用的檔案名進行設置。。		*1 2 + n	-	552 頁
	QCDSETP	QCDSETP 檔案名					

*1: n 表示 “ 檔案名的字元數 ÷ 2 ” 步。(小數點以下被進位)

2.5.15 時鐘指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
時鐘資料的讀取 / 寫入	DATERD	DATERD D	• (時鐘因數) → (D)+0 年 +1 月 +2 日 +3 時 +4 分 +5 秒 +6 星期		2	-	555 頁
	DATERDP	DATERDP D					
	DATEWR	DATEWR S	• (D)+0 年 → (時鐘因數) +1 月 +2 日 +3 時 +4 分 +5 秒 +6 星期		2	-	556 頁
	DATEWRP	DATEWRP S					
時鐘資料的加減運算	DATE+	DATE+ S1 S2 D	(S1) 時 分 秒 + (S2) 時 分 秒 → (D) 時 分 秒		4	-	558 頁
	DATE+P	DATE+P S1 S2 D					
	DATE-	DATE- S1 S2 D	(S1) 時 分 秒 − (S2) 時 分 秒 → (D) 時 分 秒		4	-	560 頁
	DATE-P	DATE-P S1 S2 D					
時鐘資料的轉換	SECOND	SECOND S D	(S) 時 分 秒 → (D) 秒(低位元) 秒(高位)		3	-	561 頁
	SECONDP	SECONDP S D					
	HOUR	HOUR S D	(S) 秒(低位元) 秒(高位) → (D) 時 分 秒		3	-	563 頁
	HOURP	HOURP S D					

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
日期比較	LDDT=				4	-	564 頁
	ANDDT=						
	ORDT=						
	LDDT<>				4	-	
	ANDDT<>						
	ORDT<>						
	LDDT<				4	-	
	ANDDT<						
	ORDT<						
	LDDT<=				4	-	
	ANDDT<=						
	ORDT<=						
	LDDT>				4	-	
	ANDDT>						
	ORDT>						
	LDDT>=				4	-	
	ANDDT>=						
	ORDT>=						

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
時鐘比較	LDTM=				4	-	567 頁
	ANDTM=						
	ORTM=						
	LDTM<>				4	-	
	ANDTM<>						
	ORTM<>						
	LDTM<				4	-	
	ANDTM<						
	ORTM<						
	LDTM<=				4	-	
	ANDTM<=						
	ORTM<=						
	LDTM>				4	-	
	ANDTM>						
	ORTM>						
	LDTM>=				4	-	
	ANDTM>=						
	ORTM>=						

2.5.16 擴展時鐘指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
擴展時鐘資料的讀取	S.DATERD	S. DATERD D	• (時鐘因數) →(D) +0 年 +1 月 +2 日 +3 時 +4 分 +5 秒 +6 星期 +7 1/1000秒		6	-	571 頁
	SP.DATERD	SP. DATERD D					
擴展時鐘資料的加減運算	S.DATE+	S. DATE+ S1 S2 D	(S1) 時 分 秒 — 1/1000秒 + (S2) 時 分 秒 — 1/1000秒 → (D) 時 分 秒 — 1/1000秒		8	-	573 頁
	SP.DATE+	SP. DATE+ S1 S2 D					
	S.DATE-	S. DATE− S1 S2 D	(S1) 時 分 秒 — 1/1000秒 − (S2) 時 分 秒 — 1/1000秒 → (D) 時 分 秒 — 1/1000秒		8	-	
	SP.DATE-	SP. DATE− S1 S2 D					

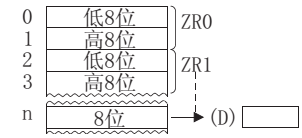
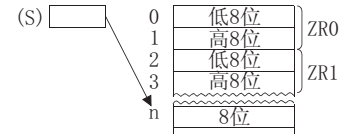
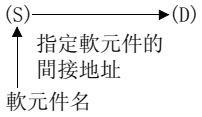
2.5.17 程式控制指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
程式控制指令	PSTOP	PSTOP 檔案名	· 將指定的程式設置爲待機類型。		*1 2 + n	-	579 頁
	PSTOPP	PSTOPP 檔案名					
	POFF	POFF 檔案名	· 將指定程式的 OUT 指令的線圈置於 OFF 後設置爲待機類型。		*1 2 + n	-	580 頁
	POFFP	POFFP 檔案名					
	PSCAN	PSCAN 檔案名	· 將指定程式作爲掃描執行型進行登錄。		*1 2 + n	-	581 頁
	PSCANP	PSCANP 檔案名					
	PLOW	PLOW 檔案名	· 將指定程式作爲低速執行型進行登錄。		*1 2 + n	-	582 頁
	PLOWP	PLOWP 檔案名					
	LDPCHK	PCHK 檔案名	· 指定檔案名的程式處於執行狀態時變爲導通狀態。 · 指定檔案名的程式處於未執行狀態時變爲不導通狀態。		*1 2 + n	-	583 頁
	ANDPCHK	PCHK 檔案名					
	ORPCHK	PCHK 檔案名					

*1: n 表示 (檔案名的字元數÷ 2) 步。(小數點以下被進位)

2.5.18 其他指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
WDT 重定	WDT	WDT	· 在順控程式中對看門狗計時器進行重定。		1	-	585 頁
	WDTP	WDTP					
定時時鐘	DUTY	DUTY n1 n2 D	(D) n1掃描 n2掃描 SM420～SM424, SM430～SM434		4	-	586 頁
時間檢查	TIMCHK	TIMCHK S1 S2 D	· 對輸入條件的 ON 進行計測，如果連續 ON 時間超過所設置的時間，則將 (D) 中指定的軟元件置於 ON。		4	-	587 頁

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
以 1 位元組爲單位進行直接讀取 / 寫入	ZRRDB	ZRRDB n D			3	-	588 頁
	ZRRDBP	ZRRDBP n D					
	ZRWRB	ZRWRB n S			3	-	589 頁
	ZRWRBP	ZRWRBP n S					
	ADRSET	ADRSET S D			3	-	591 頁
	ADRSETP	ADRSETP S D					
通過鍵盤進行的數位鍵入	KEY	KEY S n D1 D2	· 對 (S) 中指定的輸入模組的 8 點的 ASCII 資料進行讀取，轉換爲 16 進制數值後存儲到 (D1) 中指定的軟元件號後面。		5	-	592 頁
變址寄存器的批量保存	ZPUSH	ZPUSH D	· 將變址寄存器的內容保存到 (D) 中指定的軟元件後面。		2	-	596 頁
	ZPUSHP	ZPUSHP D					
變址寄存器的批量恢復	ZPOP	ZPOP D	· 將保存在 (D) 中指定的軟元件後面的資料讀取到變址寄存器中。				
	ZPOPP	ZPOPP D					
模組資訊讀取	UNIRD	UNIRD n1 D n2	· 從 (n) 中指定的起始 I/O 號開始，讀取 (n2) 中指定點數的模組資訊後，存儲在 (D) 中指定的軟元件後面。		4	-	598 頁
	UNIRDP	UNIRDP n1 D n2					
模組型號讀取	TYPERD	TYPERD n D	· 對 (n) 中指定的起始 I/O 號的模組型號進行讀取後，存儲在 (D) 中指定的軟元件後面。		3	-	602 頁
	TYPERDP	TYPERDP n D					
跟蹤設置	TRACE	TRACE	· SM800、SM801、SM802 爲 ON 時按設置的次數，將通過週邊設備設置的跟蹤資料存儲到採樣跟蹤用文件中。		1	-	606 頁
跟蹤重定	TRACER	TRACER	· 對通過 TRACE 指令設置的資料進行重定。		1	-	
向指定文件寫入資料	SP.FWRITE	SP.FWRITE U0 S0 D0 S1 S2 D1	· 對指定的文件進行資料寫入。		11	-	607 頁
從指定文件讀取資料	SP.FREAD	SP.FREAD U0 S0 D0 S1 S2 D1	· 對指定的文件進行資料讀取。		11	-	617 頁
向標準 ROM 寫入資料	SP.DEVST	SP.DEVST n1 S n2 D	· 對標準 ROM 的軟元件資料存儲用文件進行資料寫入。		9	-	628 頁
從標準 ROM 讀取資料	S.DEVLD	S.DEVLD n1 D n2	· 對標準 ROM 的軟元件資料存儲用文件進行資料讀取。		8	-	629 頁
	SP.DEVLD	SP.DEVLD n1 D n2					
從記憶體進行程式裝載	PLOADP	PLOADP S D	· 將存儲在存儲卡、標準記憶體 (除驅動器 0 以外) 中的程式傳送到驅動器 0 中，並置於待機狀態。		3	-	631 頁

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
從程式記憶體中卸載程式	PUNLOADP	PUNLOADP S D	· 將存儲在標準記憶體 (驅動器 0) 中的待機程式從記憶體中刪除。		3	-	633 頁
裝載 + 卸載	PSWAPP	PSWAPP S1 S2 D	· 將存儲在 (S1) 中指定的標準記憶體 (驅動器 0) 中的待機程式從記憶體中刪除，將存儲在 (S2) 中指定的存儲卡、標準記憶體 (除驅動器 0 以外) 中的程式傳送到驅動器 0 中，並置於待機狀態。		4	-	634 頁
文件寄存器高速塊傳送	RBMOV	RBMOV S D n	· 將從 (S) 中指定的軟元件開始的 n 點的 16 位元資料批量傳送到 (D) 中指定的軟元件開始的 n 點區域內。		4	-	636 頁
	RBMOVP	RBMOVP S D n					
用戶資訊	UMSG	UMSG S	· 將指定的字串作爲用戶資訊顯示到顯示模組中。		2	-	640 頁

2.6 資料鏈接用指令

2.6.1 網路刷新指令

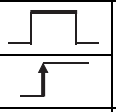
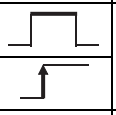
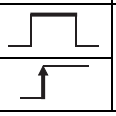
分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
網路刷新	S.ZCOM	S.ZCOM Jn	進行指定網路的刷新處理。		5	-	644 頁
	SP.ZCOM	SP.ZCOM Jn					
	S.ZCOM	S.ZCOM Un					
	SP.ZCOM	SP.ZCOM Un					

2.6.2 路由資訊的讀取 / 登錄指令

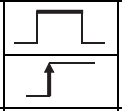
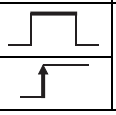
分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
路由資訊的讀取	S.RTREAD	S.RTREAD n D	對路由參數中設置的資料進行讀取。		7	-	648 頁
	SP.RTREAD	SP.RTREAD n D					
路由資訊的登錄	S.RTWRITE	S.RTWRITE n S	將路由資料寫入到路由參數中指定的區域。		8	-	649 頁
	SP.RTWRITE	SP.RTWRITE n S					

2.7 多 CPU 專用指令

2.7.1 至自站 CPU 共用記憶體的寫入指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
自站 CPU 共用記憶體寫入	S.TO	S.TO n1 n2 n3 n4 D	· 將自站的軟元件寫入到自站 CPU 模組的 CPU 共用記憶體中。		5	-	652 頁
	SP.TO	SP.TO n1 n2 n3 n4 D					
	TO	TO n1 n2 S n3	· 將自站的軟元件寫入到自站 CPU 模組的 CPU 共用記憶體中。		5	-	655 頁
	TOP	TOP n1 n2 S n3					
	DTO	DTO n1 n2 S n3	· 將自站的軟元件以 32 位元爲單位寫入到自站 CPU 模組的 CPU 共用記憶體中。		5	-	
	DTOP	DTOP n1 n2 S n3					

2.7.2 對其他站 CPU 共用記憶體的讀取指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
其他站 CPU 共用記憶體讀取	FROM	FROM n1 n2 D n3	· 從其他站 CPU 模組的 CPU 共用記憶體中將軟元件讀取到自站 CPU 中。		5	-	660 頁
	FROMP	FROMP n1 n2 D n3					
	DFRO	DFRO n1 n2 D n3	· 從其他站 CPU 模組的 CPU 共用記憶體中將軟元件以 32 位元爲單位讀取到自站 CPU 中		5	-	
	DFROP	DFROP n1 n2 D n3					

2.8 多 CPU 高速通信專用指令

2.8.1 多 CPU 高速通信專用指令

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
向其他站 CPU 寫入軟元件	D.DDWR	D.DDWR n S1 S2 D1 D2	多 CPU 系統配置時，將自站 CPU 中指定的軟元件(S2)後面的資料，按(S1)+1 中指定的寫入資料點數存儲到其他站 CPU(n1) 的指定軟元件(D1)的後面。	⎍	10	-	675 頁
	DP.DDWR	DP.DDWR n S1 S2 D1 D2		⎍↑	10	-	
從其他站 CPU 讀取軟元件	D.DDRD	D.DDRD n S1 S2 D1 D2	多 CPU 系統配置時，將其他站 CPU(n1) 的指定軟元件(D1)的後面的資料，按(S1)+1 中中指定的讀取資料點數存儲到自站 CPU 中指定的軟元件(S2)的後面。	⎍	10	-	679 頁
	DP.DDRD	DP.DDRD n S1 S2 D1 D2		⎍↑	10	-	

2.9 冗餘系統指令 (用於冗餘 CPU)

2.9.1 冗餘系統指令 (用於冗餘 CPU)

分類	指令符號	符號	處理內容	執行條件	基本步數	子集	參閱頁面
系統切換	SP.CONTSW	SP.CONTSW S D	在執行了 SP.CONTSW 指令的掃描的 END 處理時，進行控制系統與待機系統的切換。	⎍↑	8	-	683 頁

第 3 章　指令構成

3.1　指令構成

多數的 CPU 模組指令可分爲指令部分和軟元件部分。

指令部分和軟元件部分的用途如下所示：

· 指令部分 ..... 表示該指令的功能。

· 軟元件部分 ..... 表示指令中使用的資料。

軟元件部分被分爲源資料、目標資料和軟元件數。

(1) 源資料 (S)

(a) 源資料是用於運算的資料。

(b) 根據各指令中指定的軟元件，其形式如下所示：

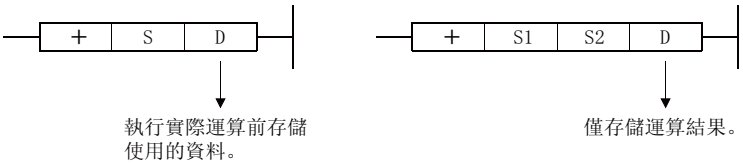
· 常數 ..... 指定運算中使用的數值。
是在創建程式時進行設置，因此在程式執行過程中無法變更。
將常數作爲可變資料使用時，應進行變址修飾。

· 位元軟元件、字軟元件 ..... 指定存儲運算中使用的資料的軟元件。
資料必須存儲在指定的軟元件當中，直到運算開始執行。
在程式執行過程中，通過變更指定軟元件中存儲的資料，可以對該指令中使用的資料進行更改。

(2) 目標資料 (D)

(a) 目標資料中存儲運算後的資料。
但是，根據指令情況，有時運算前目標資料中也需要存儲用於運算的資料。

例 BIN16 位元資料的加法運算時

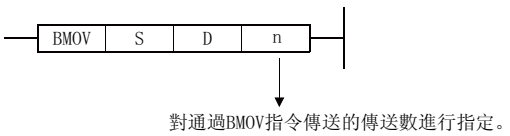


(b) 目標資料中必須設置用於資料存儲的軟元件。

(3) 軟元件數 / 傳送數 (n)

(a) 在軟元件數 / 傳送數中，對使用多個軟元件的指令中所使用的軟元件數 / 傳送數進行指定。

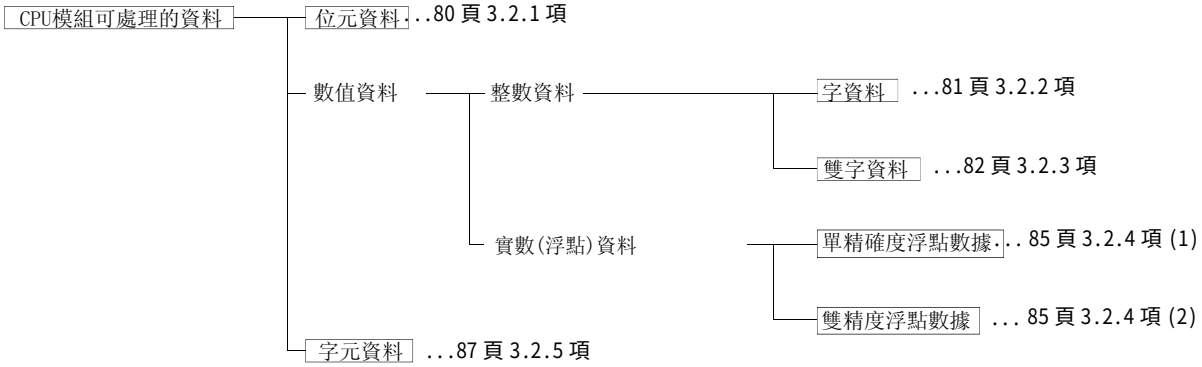
例 使用塊傳送指令時



(b) 軟元件數 / 傳送數的設置範圍爲 0 ～ 32767。
但是，軟元件數 / 傳送數爲 0 時，該指令將被視爲無效。

3.2 資料的指定方法

以下 6 種資料可以用於 CPU 模組指令：



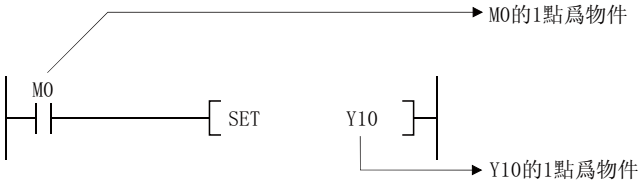
3.2.1 使用位元資料時

位元資料是觸點或線圈等以 1 位元爲單位處理的資料。

“ 位軟元件 ”及“ 位指定字軟元件 ”可以被當作位元資料使用。

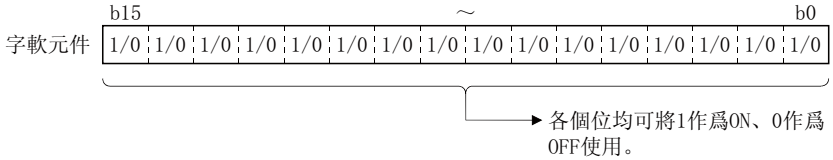
(1) 使用位軟元件時

位元軟元件以 1 點爲單位進行指定。



(2) 使用字軟元件時

(a) 字軟元件通過指定位號，可以將指定位號的 1/0 作爲位元資料使用。

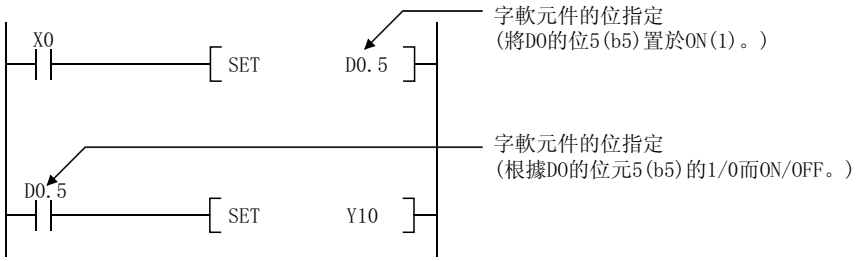


(b) 字軟元件的位指定是通過指定“ **字軟元件**. **位號** ”來完成的。

(位號指定是以 16 進制數進行)。

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

但是，對於計時器 (T)、累計計時器 (ST)、計數器 (C) 或變址寄存器 (Z)，不能進行位指定 (例： 不能指定爲 Z0.0)。



3.2.2 使用字 (16 位元) 資料時

字資料是基本指令和應用指令中使用的 16 位元數值資料。

以下兩種形式的字資料可以在 CPU 模組中使用：

· 10 進制常數 .... K-32768 ～ K32767

· 16 進制常數 .... H0000 ～ HFFFF

字軟元件和進行了位數指定的位軟元件可以作爲字資料使用。

但是，對於直接訪問輸入 (DX) 和直接訪問輸出 (DY)，不能通過位元數指定進行字資料指定。(關於直接訪問輸入和直接訪問輸出，請參閱 CPU 用戶手冊 (功能解說 / 程式基礎篇)。)

(1) 使用位軟元件時

(a) 通過位數指定，位軟元件就可以處理字資料。

位元資料的位元數指定是通過指定 “ 位數 位軟元件的起始號 ” 來完成的。

位數指定以 4 點 (4 位元) 爲單位，可在 K1 ～ K4 的範圍內指定。

(對於鏈結直接軟元件，指定是通過 “ J 網路號 \ 位數 位軟元件的起始號 ” 來完成的。例如將網路號 2 指定爲 X100 ～ X10F 時，變爲 J2\K4X100)。

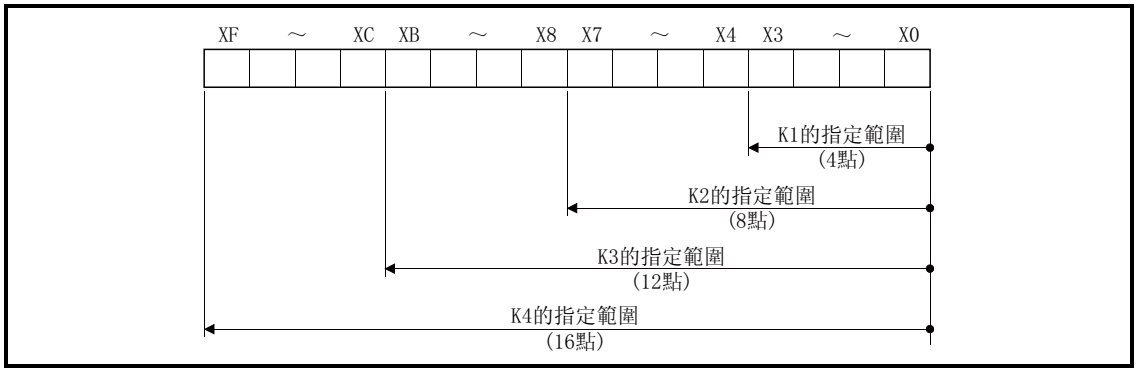
例如，將位數指定爲 X0 時，點數指定的情況如下所示：

· K1X0 ..... X0 ～ X3 的 4 點被指定

· K2X0 ..... X0 ～ X7 的 8 點被指定

· K3X0 ..... X0 ～ XB 的 12 點被指定

· K4X0 ..... X0 ～ XF 的 16 點被指定

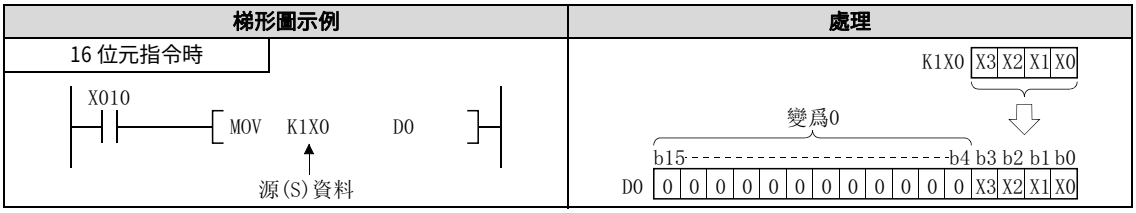


(b) 在源資料 (S) 中已進行了位數指定時，可作爲源資料處理的數值如下表所示。

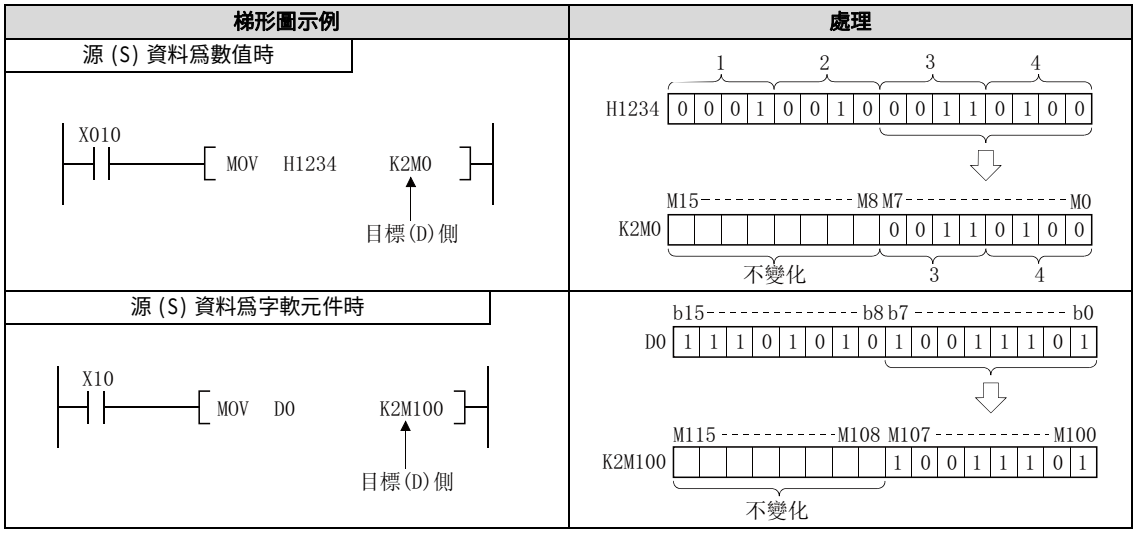
指定位元數	16 位元指令時
K1(4 點)	0 ～ 15
K2(8 點)	0 ～ 255
K3(12 點)	0 ～ 4095
K4(16 點)	-32768 ～ 32767

(c) 目標爲字軟元件時

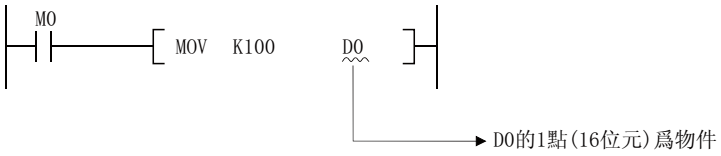
對於目標側的字軟元件，在源資料中已進行了數位指定的位元後面的位元狀態將被存儲爲 0。



(d) 在目標 (D) 中已存在有位數指定時，則指定的點數將被作爲目標使用。
進行了位數指定的點數後面的位元軟元件不發生變化。



(2) 使用字軟元件時
字軟元件是以 1 點 (16 位元) 爲單位進行指定。



要 點

1. 進行位數指定處理時，位軟元件的起始軟元件號可以爲任意數值。
2. 直接訪問輸入輸出 (DX,DY) 不能進行位數指定。

3.2.3 使用雙字資料 (32 位元) 時

雙字資料是基本指令和應用指令中使用的 32 位元數值資料。

CPU 模組可處理的雙字資料有以下 2 種：

· 10 進制常數 ..... K-2147483648 ～ K2147483647

· 16 進制常數 ..... H00000000 ～ HFFFFFFFF

字軟元件以及進行了位數指定的位軟元件可以當作雙字資料使用。

但是，對於直接訪問輸入 (DX) 和直接訪問輸出 (DY)，不能通過位數指定進行雙字資料指定。

(1) 使用位軟元件時

(a) 位軟元件通過位數指定可以處理雙字資料。

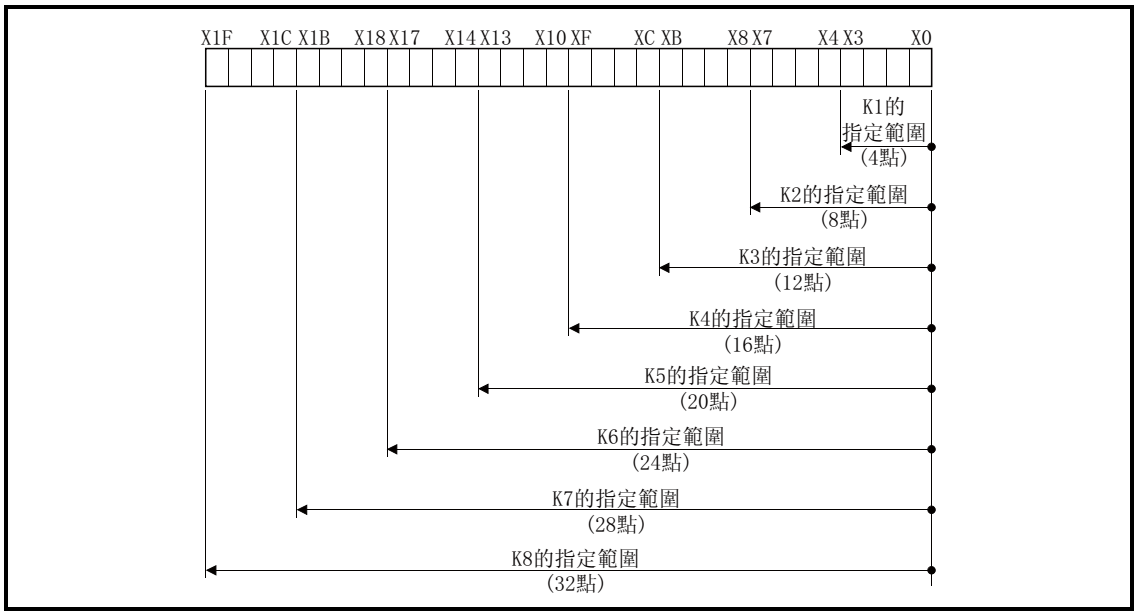
位軟元件的位數指定是通過指定“ 位數 位軟元件的起始號 ”來完成的。

(對於鏈結直接軟元件，指定是通過“ J 網路號 \ 位數 位軟元件的起始號 ”來完成的。) 將網路號 2 指定爲 X100 ～ X11F 時，變爲 J2\K8X100)。

位數指定以 4 點 (4 位元) 爲單位，可在 K1 ～ K8 的範圍內指定。

例如，將位數指定爲 X0 時，點數指定的情況如下所示：

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



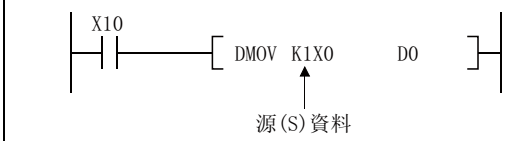
(b) 在源資料 (S) 中已存在有位數指定時，可作爲源資料處理的數值如下表所示。

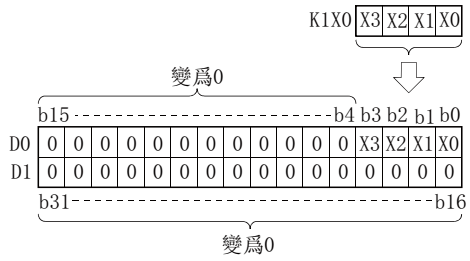
指定位元數	32 位元指令時	指定位元數	32 位元指令時
K1(4 點)	0 ～ 15	K5(20 點)	0 ～ 1048575
K2(8 點)	0 ～ 255	K6(24 點)	0 ～ 16777215
K3(12 點)	0 ～ 4095	K7(28 點)	0 ～ 268435455
K4(16 點)	0 ～ 65535	K8(32 點)	-2147483648 ～ 2147483647

(c) 目標爲字軟元件時

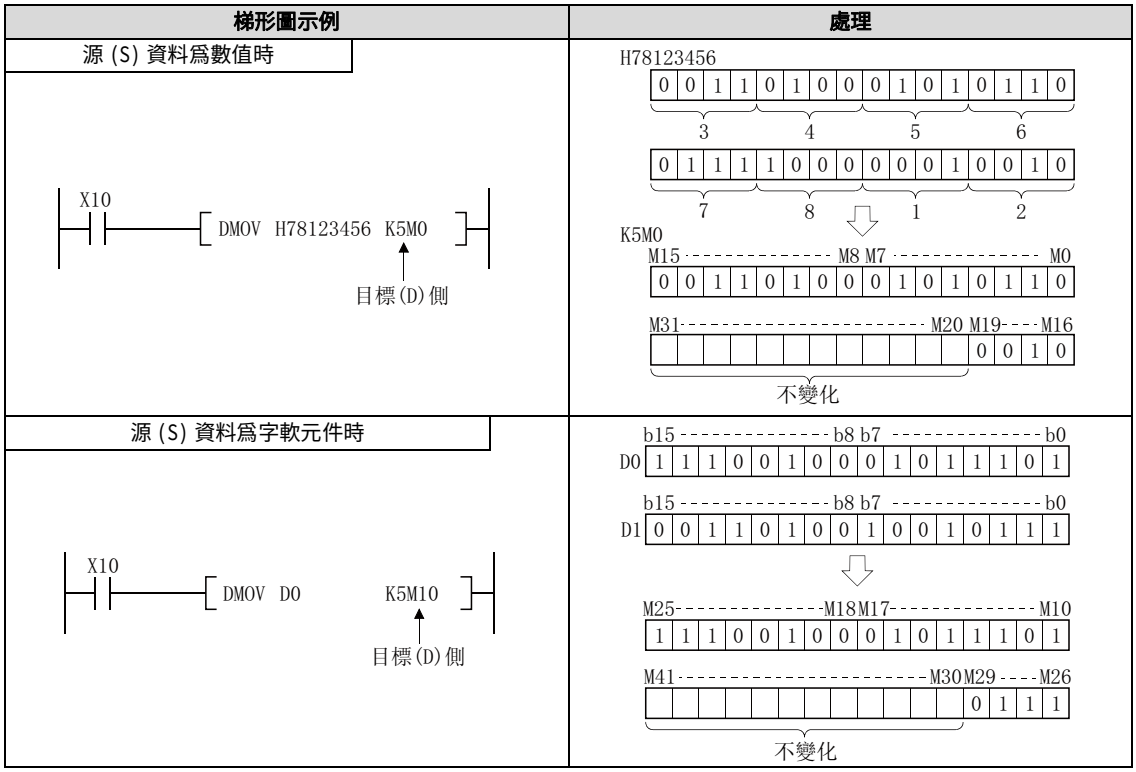
對於目標側的字軟元件，在源資料中已進行了數位指定的位元後面的位元狀態將被存儲爲 0。

梯形圖示例	處理
32 位元指令時	





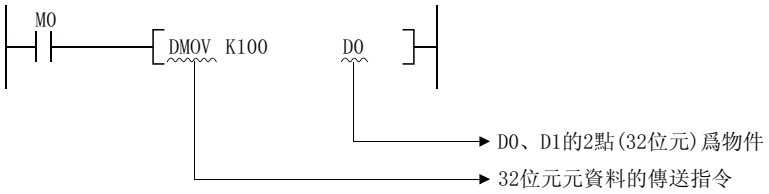
(d) 在目標 (D) 中已存在有位數指定時，則指定的點數將被作爲目標使用。
進行了位數指定的點數後面的位元軟元件不發生變化。



要 點

1. 進行位數指定處理時，位軟元件的起始軟元件號可以爲任意數值。
2. 直接訪問輸入輸出 (DX,DY) 不能進行位數指定。

(2) 使用字軟元件時
字軟元件是被指定爲在低 16 位中使用的軟元件。
在 32 位元指令中，使用 (指定軟元件號) 及 (指定軟元件號 +1)。



3.2.4 使用實數資料時

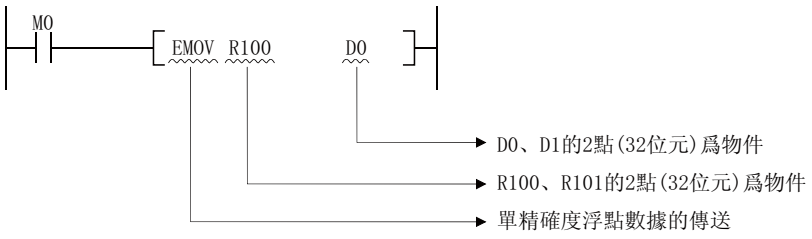
實數資料是用於基本指令和應用指令的浮點數據。

只有字軟元件能夠存儲實數資料。

(1) 單精確度浮點數據

在處理單精確度浮點數據的指令中，指定低 16 位中使用的軟元件。

單精確度浮點數據存儲在（指定軟元件號）及（指定軟元件號 +1) 的 32 位中。



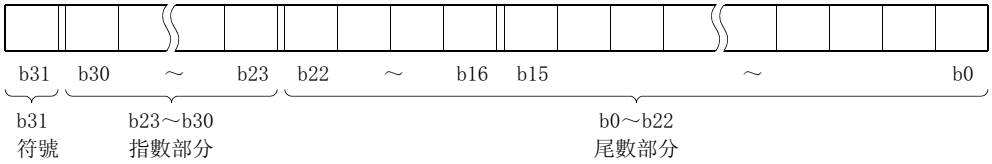
備註

在順控程式中，浮點數據通過 E□指定。

單精確度浮點數據使用 2 個字軟元件並以下列方式表達：

[符號] 1.[尾數部分] × 2 [指數部分]

單精確度浮點數據內部表示時的位構成及含義如下：



· 符號　通過 b13 表示符號。

0: 正

1: 負

· 指數部分　通過 b23 ～ b30 表示 2^n 的 n。

根據 b23 ～ b30 的 BIN 值，n 的值如下所示：

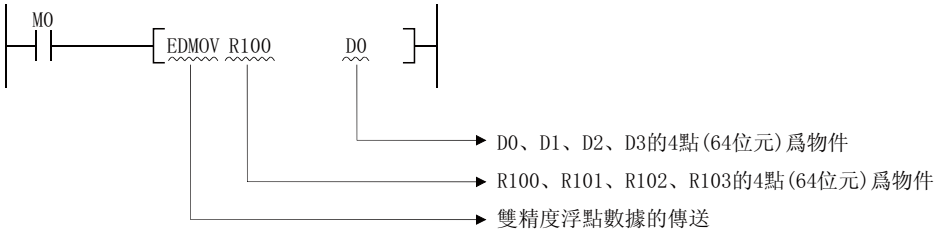
b23～b30	FFH	FEH	FDH			81H	80H	7FH	7EH			02H	01H	00H
n	未使用	127	126			2	1	0	−1			−125	−126	未使用

· 尾數部分　通過 b0 ～ b22 的 23 位表示，在 2 進制數中 1.XXXXXX…表示爲 XXXXXX…的值。

(2) 雙精度浮點數據

在處理雙精度浮點數據的指令中，指定低 16 位中使用的軟元件。

雙精度浮點數據存儲在（指定軟元件號）～（指定軟元件號 +3) 的 64 位中。



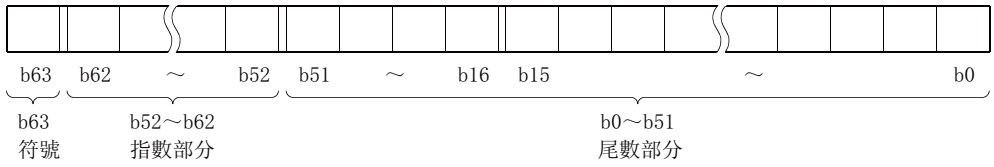
備註

在順控程式中，浮點數據通過 E□指定。

雙精度浮點數據使用 4 個字軟元件並以下列方式表達：

[符號] 1.[尾數部分] × 2 [指數部分]

雙精度浮點數據內部表示時的位構成及含義如下：



· 符號　通過 b63 表示符號。

0：正

1：負

· 指數部分　通過 b52 ～ b62 表示 2^n 的 n。

根據 b52 ～ b62 的 BIN 值，n 的值如下所示：

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

· 尾數部分　通過 b0 ～ b51 的 52 位表示，在 2 進制數中 1.XXXXXX…表示爲 XXXXXX…的值。

要 點

1. 通過編程工具的監視功能可以監視 CPU 模組的浮點數據。
2. 在浮點數據表示 0 時，以下範圍全部變爲 0。
   (a)　單精確度浮點數據時： b0 ～ b31
   (b)　雙精度浮點數據時： b0 ～ b63
3. 浮點數據的設置範圍如下所示。*1
   (a)　單精確度浮點數據時
   -2^{128} ＜軟元件≤ -2^{-126}，0 ，2^{-126} ≤軟元件＜ 2^{128}
   (b)　雙精度浮點數據時
   -2^{1024} ＜軟元件≤ -2^{-1022}，0 ，2^{-1022} ≤軟元件＜ 2^{1024}
4. 不要在浮點數據中指定 0(只有浮點型實數的最高位爲 1 時)。(如果以 0 進行浮點運算，將發生運算錯誤)。
   在下述 CPU 模組中指定了 -0 時，在 CPU 模組內部將會把 -0 轉換爲 0 後再執行浮點運算，因此不會發生運算錯誤。
   · 內部運算設置爲雙精度的高性能型 QCPU*2(浮點運算的內部運算默認爲雙精度。)
   在下述 CPU 模組中指定了 -0 時，因爲處理速度優先，在浮動點運算時原樣不變地使用 -0，因此將發生運算錯誤。
   · 基本型 QCPU*3
   · 內部運算設置爲單精確度的高性能型 QCPU*2
   · 過程 CPU
   · 冗餘 CPU
   · 通用型 QCPU
   · LCPU

*1:　關於實數超出範圍和輸入了特殊值時的動作情況，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

*2:　浮點運算的內部運算的單精確度與雙精度之間的切換是在可程式控制器參數的可程式控制器系統設置中進行。關於浮點運算的單精確度及雙精度有關內容，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

*3:　在序列號的前五位爲 “ 04122 或以後 ” 的基本型 QCPU 中，能夠執行浮點運算。

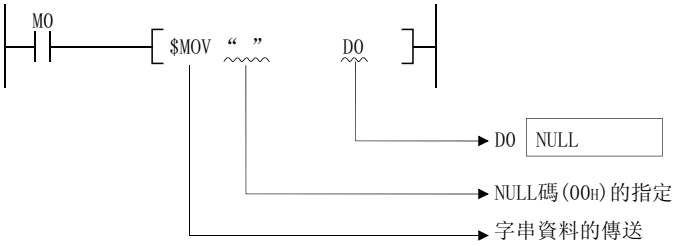
3.2.5 使用字串資料時

字串資料是基本指令和應用指令中使用的字元資料。

它包含從指定字元起至表示字串末尾的 NULL 碼 (00_H) 爲止的所有資料。

(1) 當指定字元爲 NULL 碼時

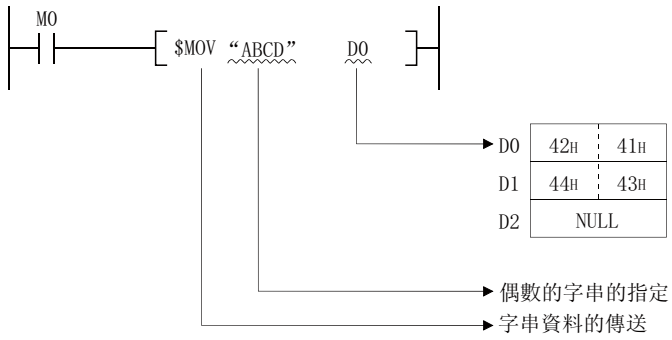
使用 1 個字來存儲 NULL 碼。



(2) 當字元數是偶數時

使用 (字元數 /2+1) 個字存儲字串及 NULL 碼。

例如，如果將 " ABCD " 傳送至 D0 ～，則字串 (ABCD) 將被存儲到 D0 及 D1 中，NULL 碼將被存儲到 D2 中。(NULL 碼將被存儲到最後的 1 個字中。)

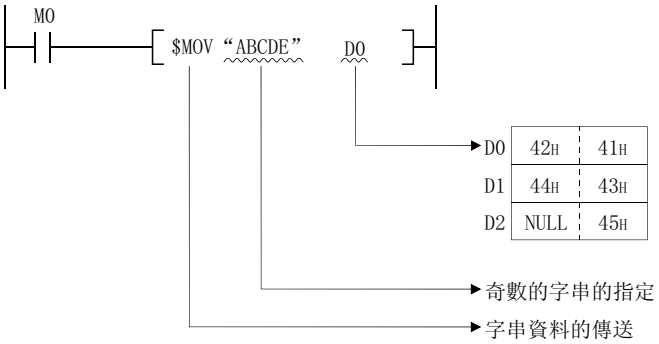


(3) 當字元數是奇數時

使用 (字元數 /2) 個字 (小數部分進位元) 存儲字串及 NULL 碼。

例如，如果將 " ABCDE " 傳送到 D0 ～，則字串 (ABCDE) 及 NULL 碼將被存儲到 D0 ～ D2 中。

(NULL 碼將被存儲到最後 1 個字的高 8 位處。)



3.3 變址修飾

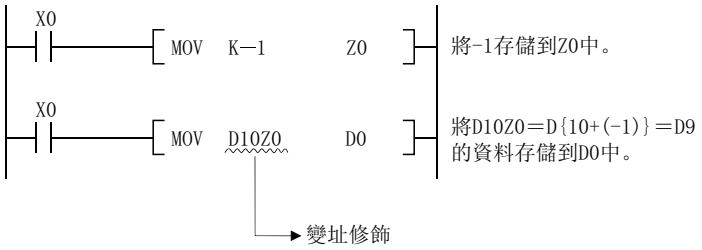
(1) 變址修飾的概要

(a) 變址修飾是通過使用變址寄存器進行的間接設置。
在順控程式中使用變址修飾時，使用的軟元件將變爲 (直接指定的軟元件號)+(變址寄存器的內容)。
例如，指定了 D2Z2 時，如果 Z2 的內容爲 3，則 D(2+3)=D5，D5 成爲指定的軟元件。

(b) 只有在使用通用型 QCPU、LCPU 時，才可以使用 16 位變址寄存器和 32 位變址寄存器進行變址修飾。

(2) 通過 16 位變址寄存器進行的變址修飾

(a) 在 16 位範圍內進行變址修飾時
各變址寄存器均可在 -32768 至 32767 的範圍內進行設置。
變址修飾按照以下方式執行：



(b) 可進行變址修飾的軟元件

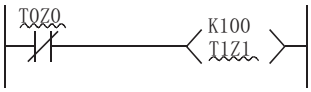
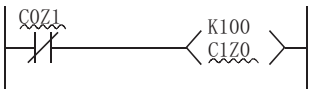
除去下面所列的限制以外，變址修飾可以應用於觸點、線圈、基本指令和應用指令中使用的軟元件。

1) 不能進行變址修飾的軟元件

軟元件	內容
E	浮點數據
$	字串資料
□.□	字軟元件的位指定
FX、FY、FD	功能軟元件
P	作爲標簽的指標
I	作爲標簽的中斷指標
Z	變址寄存器
S	步進繼電器
TR	SFC 傳送軟元件 *1
BL	SFC 塊軟元件 *1

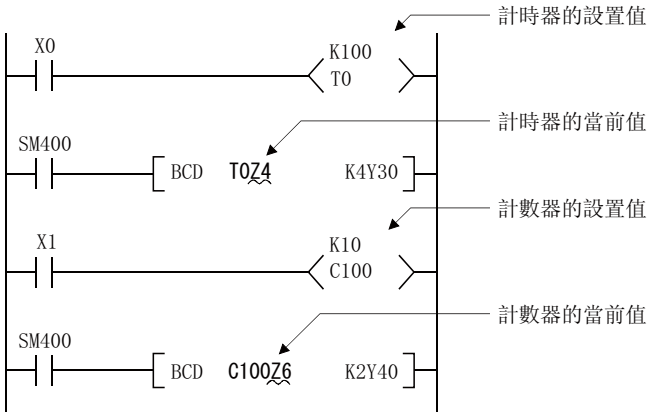
*1: SFC 傳送軟元件和 SFC 塊軟元件是供 SFC 使用的軟元件。
關於如何使用這些軟元件，請參閱以下手冊：
・MELSEC-Q/L/QnA 編程手冊 (SFC 篇)

2) 對使用變址寄存器有限制的軟元件

軟元件	內容	使用示例
T	・只有 Z0 和 Z1 可以用作計時器的觸點和線圈。	
C	・只有 Z0 和 Z1 可以用作計數器的觸點和線圈。	

備 註

對於計時器和計數器的當前值，變址寄存器編號的使用無限制。



(c) 進行了變址修飾時及實際處理軟元件的情況如下所示：

(當 Z0=20，Z1=-5 時)

梯形圖示例	實際處理軟元件
X0 MOV K20 Z0 MOV K－ 5 Z1 X1 MOV K2X50Z0 K1M38Z1	X1 MOV K2X64 K1M33 說明 K2X50Z0·····K2X(50+14)=K2X64 將K20轉換爲16進制數 K1M38Z1·····K1M(38-5)=K1M33
X0 MOV K20 Z0 MOV K－5 Z1 X1 MOV D0Z0 K3Y12FZ1	X1 MOV D20 K3Y12A 說明 D0Z0 ·····D(0+20)=D20 K3Y12FZ1·····K3Y(12F－5)=K3Y12A 16進制數

(3) 通過 32 位變址寄存器進行的變址修飾

(對於通用型 QCPU （Q00UJCPU 除外)、LCPU)

通過 32 位進行變址修飾時的變址寄存器的指定方法可以從以下 2 種中選擇：

· 指定 32 位變址修飾中使用的變址寄存器的範圍。

· 通過 “ ZZ ” 表示指定 32 位變址修飾。

要 點

通過 “ ZZ ” 表示的 32 位變址修飾只能在下述 CPU 模組中使用，此外，關於可使用的編程工具，請參閱編程工具的操作手冊。

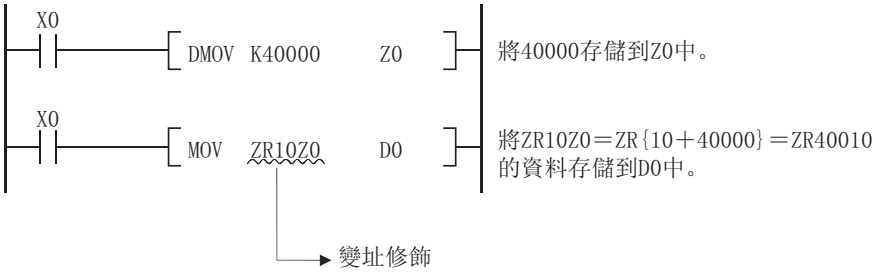
· 序列號的前 5 位數爲 “ 10042 ” 以後的 QnU(D)(H)CPU (Q00UJCPU 除外)

· QnUDE(H)CPU

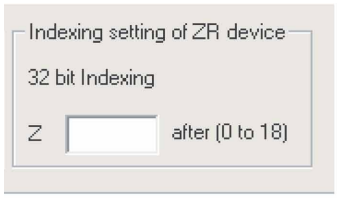
· LCPU

(a) 對使用 32 位變址修飾的變址寄存器的範圍進行指定時

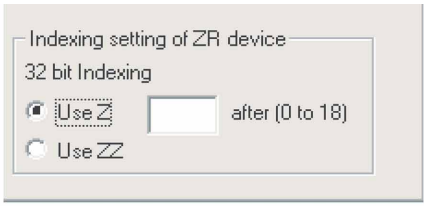
1) 各變址修飾寄存器的設置範圍爲 -2147483648 ～ 2147483647。
變址修飾的情況如下所示：



2) 指定方法
通過 32 位變址寄存器進行變址修飾時，在可程式控制器參數的軟元件設置中，中指定使用的變址寄存器的起始號。



GX Developer版本8.68R以前



GX Developer版本8.68W以後

要 點

當在可程式控制器參數的軟元件設置中，將使用的變址寄存器的起始號進行了更改時，不要只修改參數或者進行可程式控制器寫入。必須是與程式一道進行可程式控制器寫入。
如果進行強制寫入，將會發生 CAN'T EXE. PRG 錯誤。(出錯代碼 :2500)

3) 可進行變址修飾的軟元件
變址修飾只能使用如下所示的軟元件：

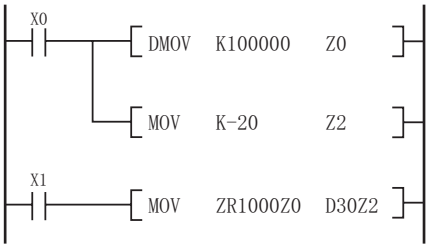
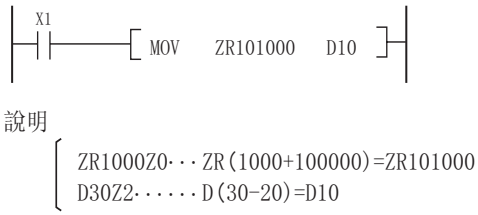
軟元件	內容
ZR	連號訪問方式文件寄存器
D	擴展資料寄存器
W	擴展鏈結寄存器

4) 變址寄存器的使用範圍
下表列出了通過 32 位變址寄存器進行變址修飾時的變址寄存器可用範圍。
在通過 32 位變址寄存器進行變址修飾時，使用指定的變址寄存器 (Zn) 和緊接著的下一個變址寄存器 (Zn+1)，因此請注意防止所使用的變址寄存器重疊。

設置值	使用的變址寄存器	設置值	使用的變址寄存器
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	禁止使用

5) 進行了變址修飾時及實際的處理軟元件如下所示：

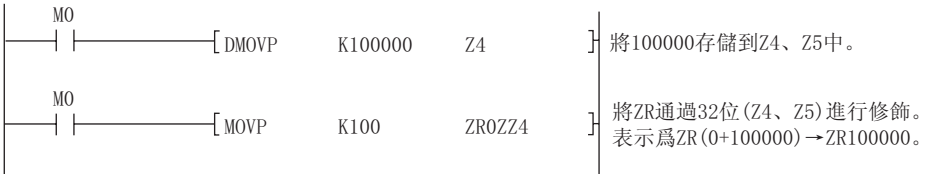
(當 Z0(32 位)=100000，Z2(16 位)=-20 時)

梯形圖示例	實際的處理軟元件
	

(b) 通過 “ ZZ ” 表示指定 32 位變址修飾時

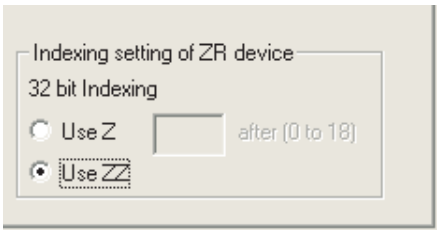
1) 通過表示爲 “ ZR0ZZ4 ” 的 “ ZZ ” 表示對變址修飾進行指定，可以通過任意的變址寄存器進行 32 位變址修飾指定。

通過 “ ZZ ” 表示的 32 位變址修飾的示例如下：



2) 指定方法

通過 “ ZZ ” 表示進行 32 位變址修飾時，在可程式控制器參數的 [軟元件設置] → [文件寄存器擴展設置] 中設置 “ 使用 ZZ ”



3) 可進行變址修飾的軟元件

變址修飾只能使用如下所示的軟元件：

軟元件	內容
ZR	連號訪問方式文件寄存器
D	擴展資料寄存器
W	擴展鏈結寄存器

4) 變址寄存器的使用範圍

下表列出了通過 “ ZZ ” 表示進行 32 位變址修飾時的變址寄存器可用範圍。

指定通過 “ ZZ ” 表示進行 32 位變址修飾時，以 ZRmZZn 的形式進行指定。

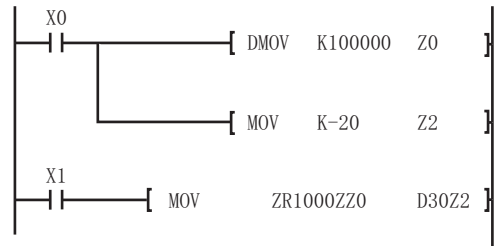
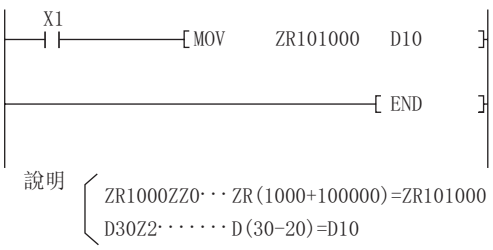
通過指定 ZRmZZn，將 ZRm 的軟元件號以 Zn，Zn+1 的 32 位值進行修飾。

“ ZZ ” 表示 *2	使用的變址寄存器	“ ZZ ” 表示 *2	使用的變址寄存器
□ZZ0	Z0，Z1	□ZZ10	Z10，Z11
□ZZ1	Z1，Z2	□ZZ11	Z11，Z12
□ZZ2	Z2，Z3	□ZZ12	Z12，Z13
□ZZ3	Z3，Z4	□ZZ13	Z13，Z14
□ZZ4	Z4，Z5	□ZZ14	Z14，Z15
□ZZ5	Z5，Z6	□ZZ15	Z15，Z16
□ZZ6	Z6，Z7	□ZZ16	Z16，Z17
□ZZ7	Z7，Z8	□ZZ17	Z17，Z18
□ZZ8	Z8，Z9	□ZZ18	Z18，Z19
□ZZ9	Z9，Z10	□ZZ19	禁止使用

*2： □表示修飾物件的軟元件名 (ZR、D.W)。

5）通過 “ ZZ ” 表示進行了變址修飾時及實際的處理軟元件如下所示：

（當 Z0(32 位)=100000，Z2(16 位)=-20 時)

梯形圖示例	實際的處理軟元件
	

6）可使用 “ ZZ ” 表示的功能

在下表所示的功能中，可以使用通過 “ ZZ ” 表示進行 32 位變址修飾指定。

編號	功能名稱・說明
1	通過程式中的指令進行軟元件指定
2	軟元件登錄監視
3	軟元件測試
4	帶執行條件的軟元件測試
5	監視條件設置
6	採樣跟蹤（跟蹤點（軟元件指定）、跟蹤物件軟元件）
7	資料記錄功能（採集間隔（軟元件指定）、記錄物件資料）

要 點

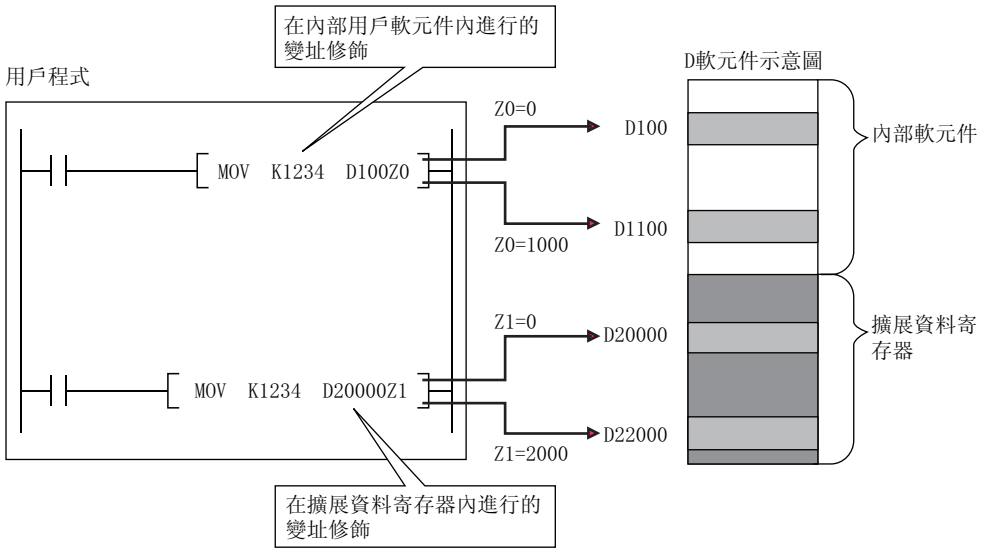
不能將單個的 ZZn 作爲軟元件處理，例如 “ DMOV K100000 ZZ0 ”。 爲了通過 “ ZZ ” 表示進行 32 位變址修飾指定，對變址寄存器進行值的設置時，應對 Zn(Z0 ～ Z19) 進行設置。

在各功能中，不能單獨輸入 ZZn。

(4) 通過擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 進行變址修飾時

（通用型 QCPU(Q00UJCPU 除外)、LCPU)

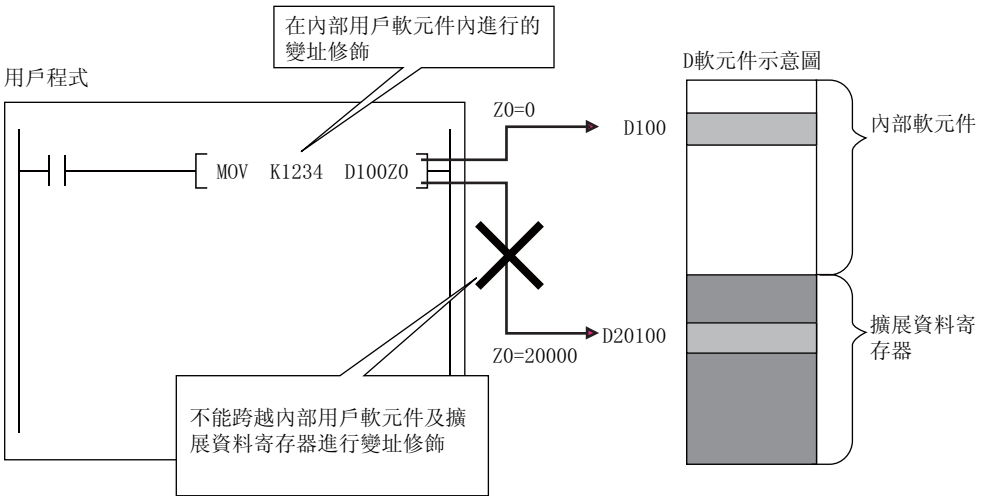
與通過內部用戶軟元件的資料寄存器 (D)、擴展鏈結寄存器 (W) 進行的變址修飾相同，可以在擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 的範圍內通過變址修飾進行軟元件指定。



1) 跨越內部用戶軟元件及擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 的變址修飾

不能跨越內部用戶軟元件及擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 進行變址修飾指定。變址修飾時的軟元件範圍檢查有效的情況下，將變爲出錯狀態。

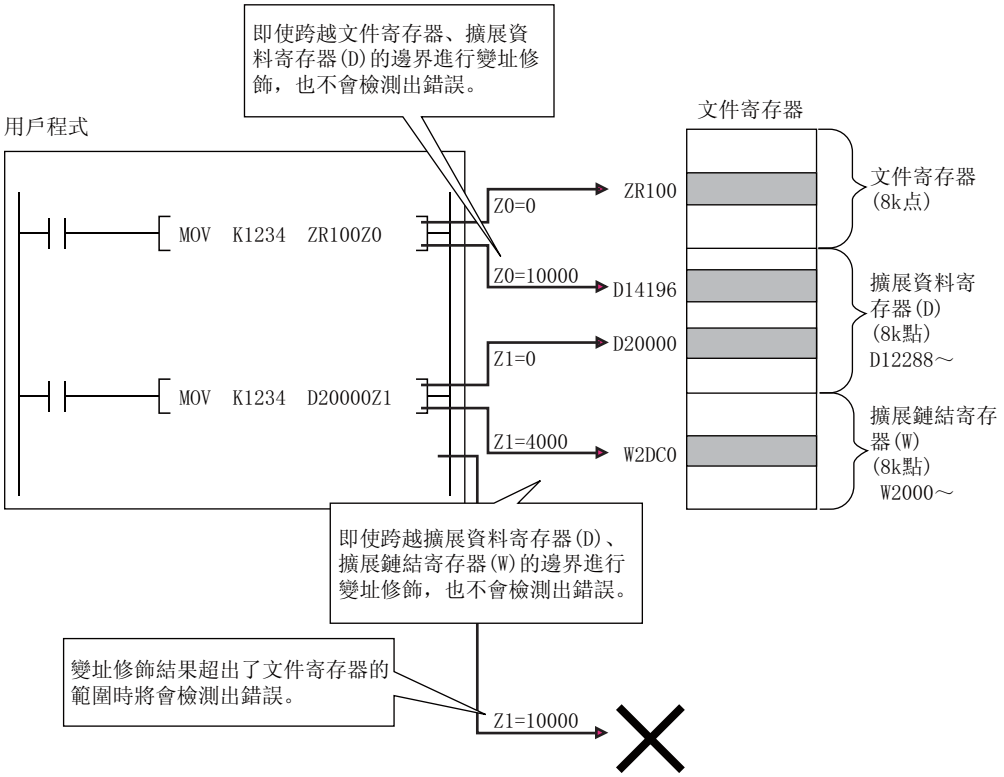
(出錯代碼： 4101)



2) 跨越文件寄存器 (ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 的變址修飾

即使跨越文件寄存器 (ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 進行變址修飾，也不會變爲出錯狀態。

但是，如果文件寄存器 (ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 的變址修飾結果改變了文件寄存器的範圍，將會變爲出錯狀態。(出錯代碼： 4101)

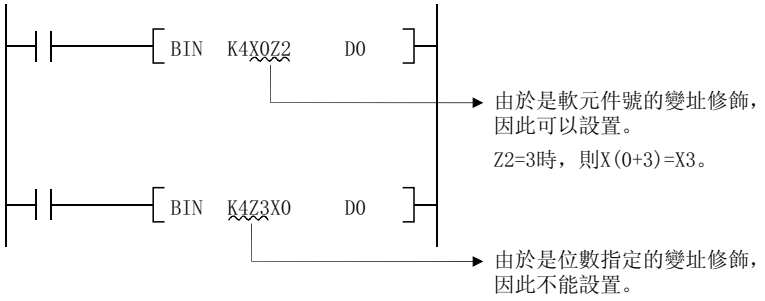


(5) 進行其他的變址修飾時

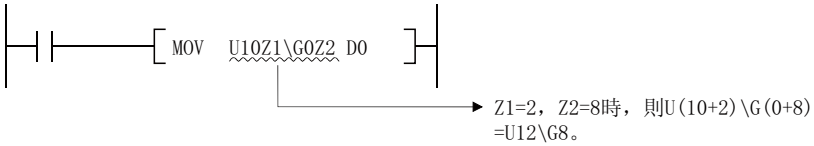
(a) 位元資料

在進行了位數指定時，可以進行軟元件號的變址修飾。

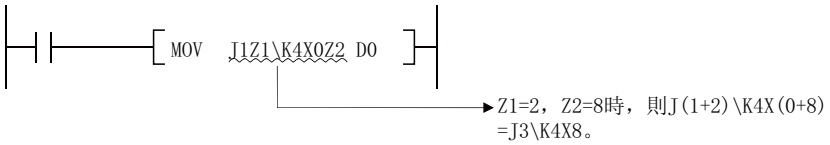
但是，不能進行位數指定的變址修飾。



(b) 在智慧功能模組軟元件 *3 中，對智慧功能模組的起始 I/O 地址號及緩衝記憶體地址均可以進行變址修飾。

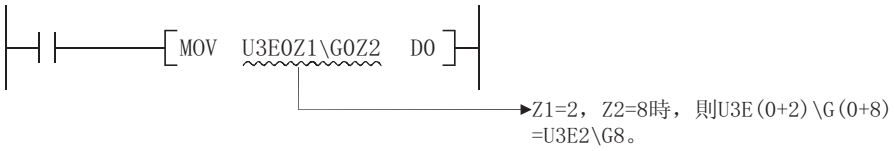


(c) 在鏈結直接軟元件 *3 中，對網路號及軟元件號均可以進行變址修飾。



*3： 關於智慧功能模組軟元件、鏈結直接軟元件，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

(d) 在多 CPU 共用軟元件 *4 中，對 CPU 模組的起始 I/O 位址號和 CPU 共用記憶體位址均可以進行變址修飾。



*4: 關於多 CPU 共用軟元件，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

(e) 通過擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 的 32 位進行變址修飾時

(通用型 QCPU （Q00UJCPU 除外)、LCPU)

進行擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 的變址修飾時，與文件寄存器 (ZR) 的變址修飾相同，可以使用下述 2 種方法進行 32 位的變址修飾：

・指定用於 32 位變址修飾的變址寄存器。

・指定通過 “ ZZ ” 表示的 32 位變址修飾。

要 點

通過 “ ZZ ” 表示的 32 位變址修飾只能在下述 CPU 模組中使用，此外，關於可使用的編程工具，請參閱編程工具的操作手冊。

・ 序列號的前 5 位數爲 “ 10042 ” 以后的 QnU(D)(H)CPU(Q00UJCPU 除外)

・ QnUDE(H)CPU

・ LCPU

(6) 注意事項

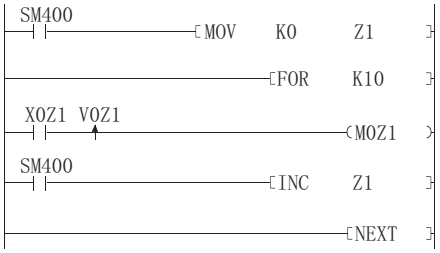
(a) 在 FOR ～ NEXT 指令之間進行變址修飾時

通過在 FOR ～ NEXT 指令之間使用變址繼電器 (V)，可以進行脈衝輸出。

但是，不能進行使用 PLS/PLF/ 脈衝化 (□ P) 指令的脈衝輸出。

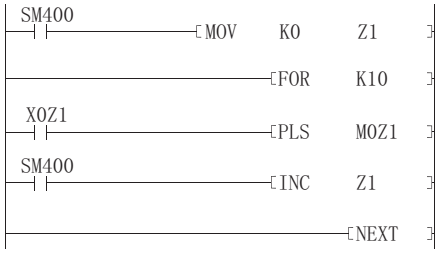
[使用變址繼電器時]

(MOZ1 可以进行正常的脉冲输出。)



[未使用變址繼電器時]

(MOZ1 不能進行正常的脈衝輸出。)



備 註

通過變址繼電器 V0Z1 記憶 X0Z1 的 ON/OFF 資訊。
例如，X0 的 ON/OFF 資訊由 V0 記憶，X1 的 ON/OFF 資訊由 V1 記憶。

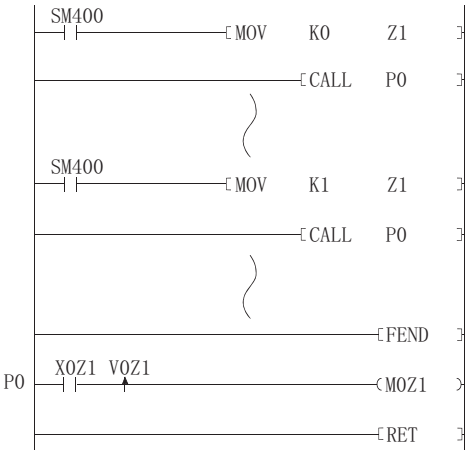
(b) 通過 CALL 指令進行變址修飾時

在 CALL 指令中通過使用變址繼電器 (V)，可以進行脈衝輸出。

但是，不能通過 PLS/PLF/ 脈衝 (□ P) 指令進行脈衝輸出。

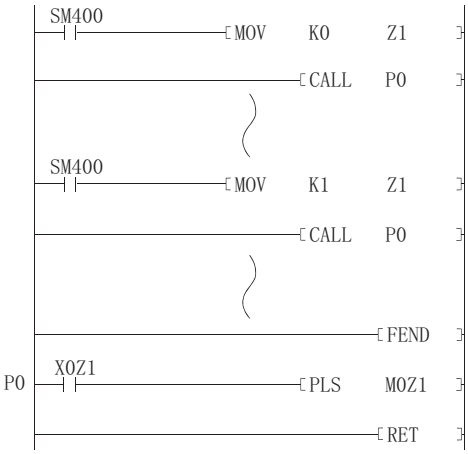
[使用變址繼電器時]

(MOZ1 可以進行正常的脈衝輸出。)



[未使用變址繼電器時]

(MOZ1 不能進行正常的脈衝輸出。)



(c) 變址修飾時的軟元件範圍檢查

1) 使用基本型 QCPU、高性能型 QCPU、過程 CPU、冗余 CPU 時

在進行變址修飾時不進行軟元件範圍檢查。

因此，進行了變址修飾後的結果超出了用戶指定的軟元件範圍時，不會發生錯誤，而是將資料寫入到其他軟元件中。(但是，如果進行了變址修飾後的結果超出了用戶軟元件範圍，被寫入到系統用的軟元件中時，將會變爲出錯狀態。(出錯代碼： 1103))

在進行編程的過程中使用變址修飾時，應特別加以注意。

2) 使用通用型 QCPU、LCPU 時

在進行變址修飾時，進行軟元件範圍檢查。

此外，通過更改可程式控制器參數設置，也可以不進行軟元件範圍檢查。

(d) 16 位 32 位變址修飾範圍的變更

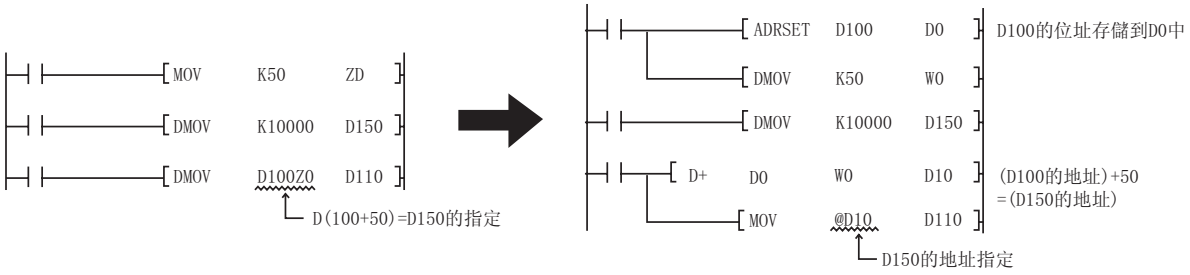
進行 16 位 32 位變址修飾範圍的變更時，應重新確認程式內的變址修飾位置

在 32 位範圍的變址修飾中，使用指定的變址寄存器 (Zn) 和緊接著的下一個變址寄存器 (Zn+1)，因此應注意避免所使用的變址寄存器重疊。

3.4 間接指定

(1) 間接指定

(a) 間接指定是指，通過 2 個字的字軟元件（字軟元件的 2 點）對順控程式中使用的軟元件地址進行指定的方法。
在變址寄存器不足的情況下，應使用這種方法進行變址修飾。



［使用變址寄存器時］　　［使用間接指定時］

(b) 用於進行指定軟元件地址指定的軟元件是通過“@+(字軟元件號)”來指定的。
例如，指定爲 @D100 時，D101、D100 的內容將成爲軟元件地址。

(c) 進行間接指定的軟元件位址可以通過 ADRSET 指令進行確認。
關於 ADRSET 指令，請參閱 591 頁 7.18.6 項。

(2) 可間接指定的軟元件

可進行間接指定的 CPU 模組軟元件如下表所示：

軟元件分類		可否間接指定	間接指定示例
內部用戶軟元件	位軟元件 *1	否	------
	字軟元件 *1	可	・@D100 ・@D100Z2*2
鏈結直接軟元件	位軟元件 *1	否	------
	字軟元件 *1	可 *3	・@J1\W10 ・@J1Z1\W10Z2*2
智慧功能模組軟元件		可 *3	・@U10\G0 ・@U10Z1\G0Z2*2
變址寄存器		否	------
文件寄存器		可	・@R0,@ZR20000 ・@R0Z1,@ZR20000Z1*2
擴展資料寄存器		可	・@D1000 ・@W1000
擴展鏈結寄存器			
嵌套		否	------
指標			------
常數			------
其他	SFC 塊軟元件		------
	SFC 傳送軟元件		
	網路號指定軟元件		
	I/O 號指定軟元件		

*1: 關於軟元件名稱，請參閱所使用的 CPU 模組的用戶手冊（功能解說 / 程式基礎篇）。
*2: 表示通過變址寄存器進行變址修飾時
*3: 可以進行間接指定，但是不能通過 ADRSET 指令進行位址寫入。

(3) 注意事項

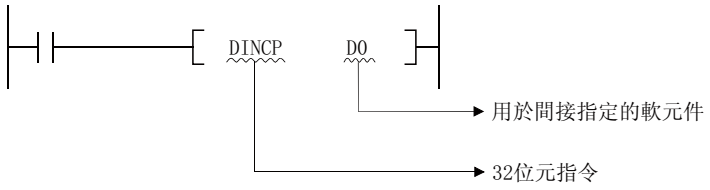
(a) 間接指定的地址

間接指定的地址是通過 2 字進行指定的。

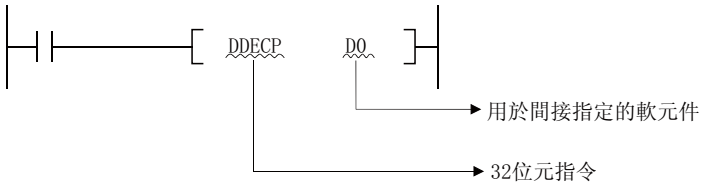
因此，用間接指定替代變址修飾時，應進行 32 位的加減運算。

對存儲在 D1 和 D0 中的間接指定用軟元件地址進行加減運算的梯形圖如下所示。

[間接指定的軟元件地址 +1 時]



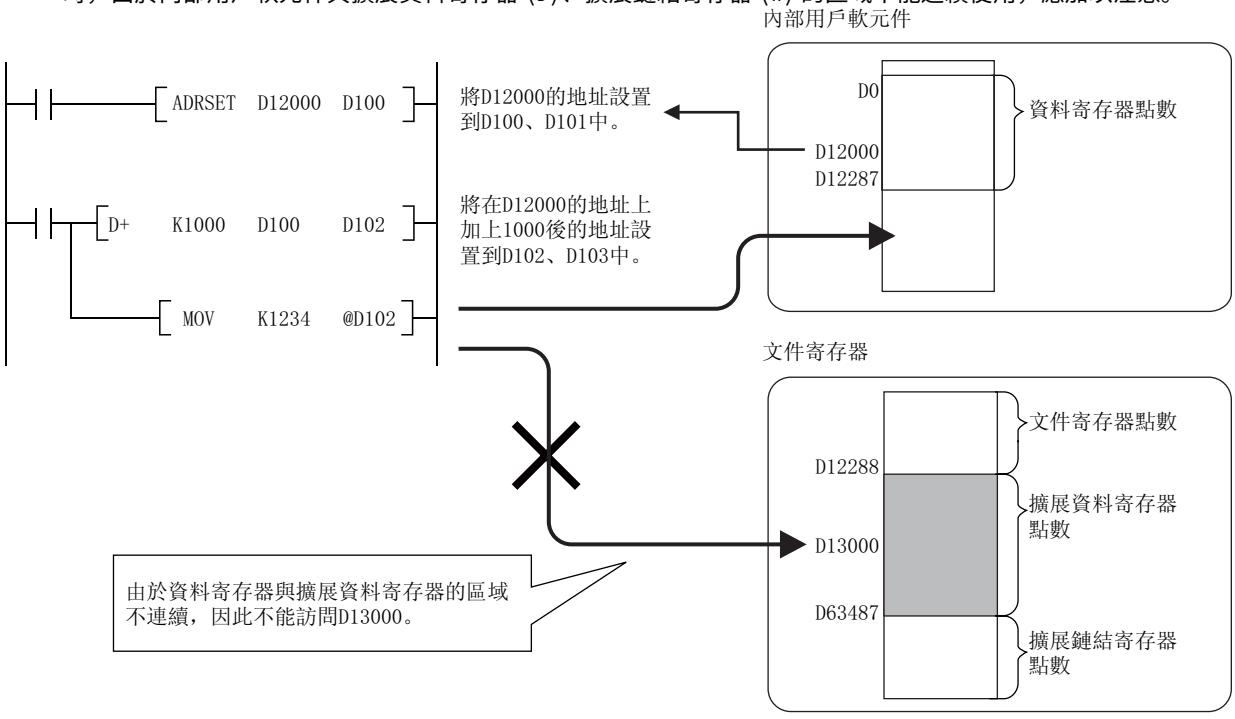
[間接指定的軟元件地址 -1 時]



(b) 擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 的間接指定

擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 可以通過間接位址進行間接指定。

通過內部用戶軟元件的資料寄存器 (D)、鏈結寄存器 (W) 及擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 進行間接指定時，由於內部用戶軟元件與擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 的區域不能連續使用，應加以注意。



3.5 縮短指令處理時間

3.5.1 子集處理

子集處理用來對基本指令及應用指令中使用的軟元件設置限制，目的是爲了提高處理速度。

但是，指令符號等不發生變化。

希望縮短掃描時間時，應在如下所示的條件下執行指令。

(1) 子集處理中各軟元件應滿足的條件

(a) 使用字資料時

軟元件	條件
位軟元件	· 以 16 的倍數指定位軟元件號。 · 位數指定只能指定爲 K4。 · 不進行變址修飾。
字軟元件	· 內部用戶軟元件 · 文件寄存器 (R、ZR*4) · 多 CPU 共用軟元件 *1*2 · 變址寄存器 (Z)/ 通用運算寄存器 (Z)*3
常數	· 無限制

(b) 使用雙字資料時

軟元件	條件
位軟元件	· 以 16 的倍數指定位軟元件號。 · 位數指定只能指定爲 K8。 · 不進行變址修飾。
字軟元件	· 內部用戶軟元件 · 文件寄存器 (R、ZR*4) · 多 CPU 共用軟元件 *1*2 · 變址寄存器 (Z)/ 通用運算寄存器 (Z)*3
常數	· 無限制

(c) 使用位元資料時

軟元件	條件
位軟元件	· 內部用戶軟元件 (可進行變址修飾)
字軟元件	· 內部用戶軟元件的位指定 · 寄存器 (R、ZR*4) 的位指定 · 多 CPU 共用軟元件的位指定 *1*2

*1: 只對應於通用型 QCPU。

*2: 只對多 CPU 高速通信區 (U3En\G10000 ～) 有效。
但是，對 CPU 模組的起始 I/O 位址號進行了變址修飾 (U3EnZn\G10000) 時除外。

*3: 只對應於通用型 QCPU、LCPU。

*4: 只對應於通用型 QCPU(Q00UJCPU 除外)、LCPU

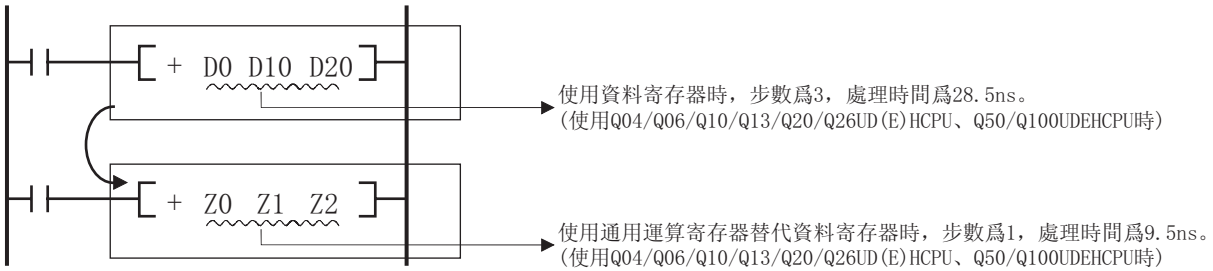
(2) 可進行子集處理的指令

指令分類	指令符號
觸點指令	LD、LDI、AND、ANI、OR、ORI、LDP、LDF、ANDP、ANDF、ORP、ORF、LDPI、ANDPI、ANDFI、ORPI、ORFI
輸出指令	OUT、SET、RST
比較運算指令	·=、<>、<、<=、>、>=、D=、D<>、D<、D<=、D>、D>=
算術運算	·+、-、*、/、INC、DEC、D+、D、D、D/、DINC、DDEC ·B+、B-、B*、B/、E+、E-、E*、E/
資料轉換指令	·BCD、BIN、DBCD、DBIN、FLT、DFLT、INT、DINT
資料傳送指令	·MOV、DMOV、CML、DCML、XCH、DXCH ·FMOV、BMOV、EMOV
程式分支指令	·CJ、SCJ、JMP
邏輯運算	·WAND、DAND、WOR、DOR、WXOR、DXOR、WXNR、DXNR
旋轉指令	·RCL、DRCL、RCR、DRCR、ROL、DROL、ROR、DROR
移位元指令	·SFL、DSFL、SFR、DSFR
資料處理指令	·SUM、SEG
結構化指令	·FOR、CALL

3.5.2 使用通用運算寄存器 (Z) 的運算處理 (只對於通用型 QCPU、LCPU)

使用通用運算寄存器 (Z) 可以縮短運算處理時間。

使用通用運算寄存器 (Z) 的程式示例如下所示：



通過可以進行子集處理的指令，可以縮短運算處理時間。

關於步數的變更，請參閱 106 頁 3.8 節。

關於各指令的運算時間，請參閱 686 頁附錄 1。

要 點

由於通用運算寄存器與變址寄存器是相同的軟元件，因此在進行變址修飾時，不要使通用運算寄存器的軟元件號與變址寄存器的軟元件號重復。

3.6 編程注意事項

在 CPU 模組中執行基本指令和應用指令時，在下列情況下將會發生運算錯誤：

· 發生了各指令的說明頁面中記載的出錯時。

· 使用智慧功能模組軟元件時，指定的 I/O 地址號的位置上未安裝智慧功能模組。

· 使用智慧功能模組軟元件時，指定的緩衝記憶體地址不存在。

· 使用鏈結軟元件時，相應的網路不存在。

· 使用鏈結軟元件時，在指定的 I/O 位址號位置上未安裝網路模組。

· 使用多 CPU 共用軟元件時，在指定 CPU 模組的起始 I/O 位址號的位置上未安裝 CPU 模組。(僅通用型 QCPU(Q00UJCPU 除外))

· 使用多 CPU 共用軟元件時，指定的共用記憶體地址不存在。(僅通用型 QCPU(Q00UJCPU 除外))

· 進行了跨越內部用戶軟元件及擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 的設置 (通用型 QCPU （Q00UJCPU 除外)、LCPU 時)

要 點

未進行文件寄存器設置時，或進行了文件寄存器設置但文件寄存器的文件不存在時，對文件寄存器進行了的讀取 / 寫入操作時的情況如下所示：

(1) 對於高性能型 QCPU、過程 CPU、冗餘 CPU
即使對文件寄存器進行讀取 / 寫入操作，也不會發生錯誤。但是，如果從文件寄存器中讀取，“ 0_H ”將被存儲。

(2) 對於通用型 QCPU、LCPU
如果對文件寄存器執行讀取 / 寫入操作，將會發生 OPERATION ERROR(出錯代碼： 4101)。

(1) 軟元件範圍檢查

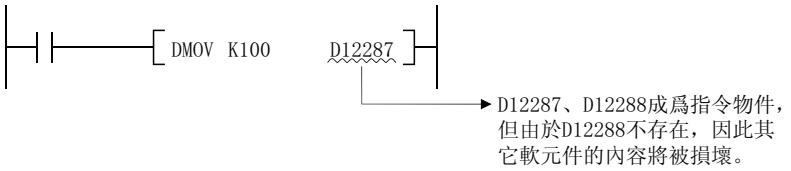
在 CPU 模組中，基本指令和應用指令中使用的軟元件範圍檢查的情況如下所示：

(a) 用於處理固定長度軟元件的指令 (MOV、DMOV 等)

1) 對於基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU

不進行軟元件範圍檢查。如果超出了相應軟元件範圍，資料將被寫入到其他軟元件中。*1

例如，在資料寄存器被分配了 12k 點時，即使超過了 D12287，也不會變爲出錯狀態。



當執行了變址修飾時，也不進行軟元件範圍檢查。

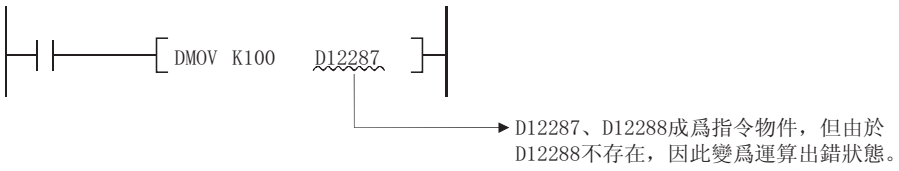
如果變址修飾的執行結果超出了相應的軟元件範圍，資料將被寫入到其他軟元件中。*1

*1: 關於內部軟元件的分配順序，請參閱 100 頁 3.6 節 (1)(c) 字串資料。

2) 對於通用型 QCPU、LCPU

進行軟元件範圍檢查。超出了相應軟元件範圍時，將會變爲運算出錯狀態。

例如，如果資料寄存器被分配了 12k 點時，如果超過了 D12287，將會變爲出錯狀態。



執行了變址修飾時也將進行軟元件範圍檢查。

此外，通過更改可程式控制器參數設置，可以設置爲不進行軟元件範圍檢查。*2

*2: 關於將設置變更爲不進行軟元件範圍檢查的有關內容，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

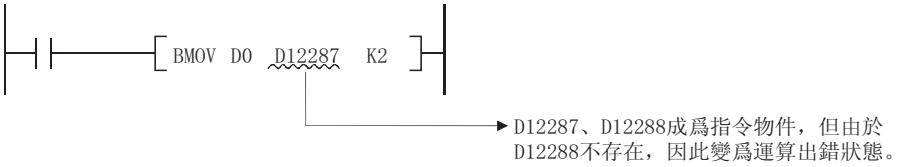
(b) 用於處理可變長度軟元件的指令（指定傳送數量的 BMOV、FMOV 等）

1) 對於基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU

進行軟元件範圍檢查。

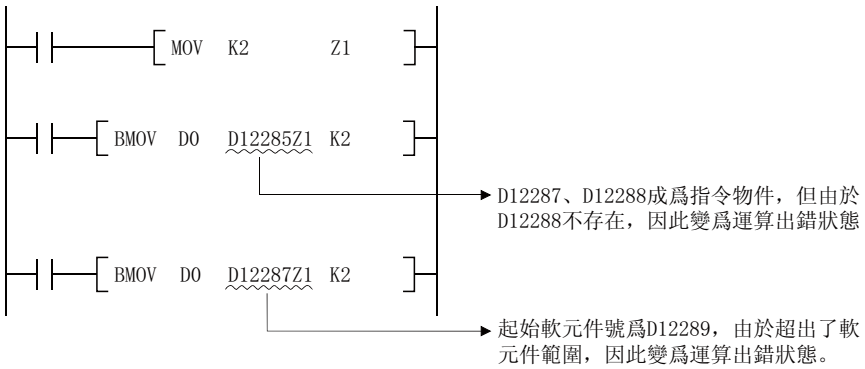
如果超出了相應軟元件範圍，將發生運算錯誤。

例如，資料寄存器被分配了 12k 點時，如果超過了 D12287，將會變爲出錯狀態。



當執行了變址修飾時也將進行軟元件範圍檢查。

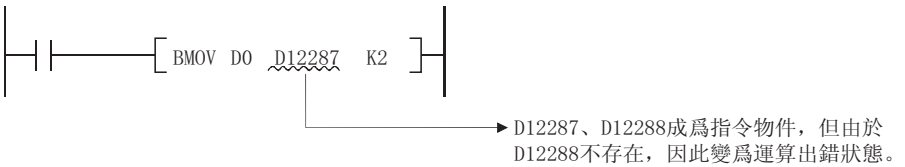
但是，即使由於變址修飾導致起始軟元件號超出了相應軟元件範圍，也不會變爲出錯狀態。



2) 對於通用型 QCPU、LCPU

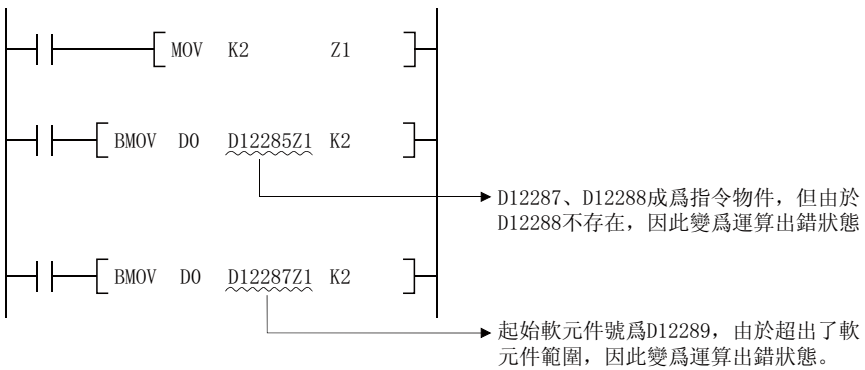
進行軟元件範圍檢查。超出了相應軟元件範圍時，將發生運算錯誤。

例如，當資料寄存器被分配了 12k 點時，如果軟元件號超過了 D12287，將會變爲出錯狀態。



進行了變址修飾時也將進行軟元件範圍檢查。

由於變址修飾的結果導致起始軟元件號超出了相應軟元件範圍時，將變爲出錯狀態。



此外，通過更改可程式控制器參數設置，可以設置爲不進行軟元件範圍檢查。*2

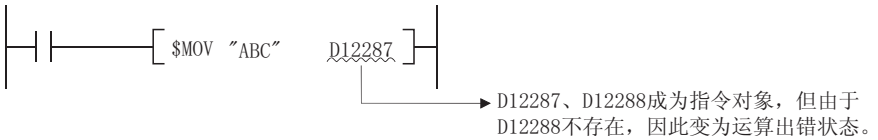
*2: 關於將設置變更爲不進行軟元件範圍檢查的有關內容，請參閱所使用的 CPU 模組的用戶手冊（功能解說／程式基礎篇）。

(c) 字串資料

因爲所有的字串資料長度可變，因此將進行軟元件的範圍檢查。

如果超出了相應的軟元件範圍，將發生運算錯誤。

例如，資料寄存器被分配了 12k 點時，如果超過了 D12287，將發生錯誤。



但是，在基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU 中，當執行了變址修飾時，在起始軟元件號超出了軟元件範圍的情況下也不會發生運算錯誤，而是將訪問其他的軟元件。

在通用型 QCPU、LCPU 中進行了下述訪問時，將會發生錯誤。(出錯代碼： 4101)

1) 由於變址修飾，超出了軟元件的邊界 (區域 A 的範圍) 的訪問

各軟元件的分配順序如下所示：

2) 由於變址修飾，超出了文件寄存器的邊界的訪問

3) 未進行文件寄存器文件設置時，對文件寄存器 (R、ZR) 進行的訪問

4) 對超出了文件寄存器文件的範圍的文件寄存器 (R、ZR) 進行的訪問

此外，通過在可程式控制器參數中設置爲變址修飾時不進行軟元件範圍檢查，則可實現即使進行了上述 1) ～ 4) 的訪問也不會檢測出錯誤。

但是，根據通用型 QCPU 的序列號，其動作情況如下表所示。*2

變址修飾時的軟元件範圍設置	通用型 QCPU 的序列號的前 5 位數		LCPU
	序列號 “ 10021 ” 以前	序列號 “ 10022 ” 以後	
實施	進行上述 1) ～ 4) 的訪問時檢測出錯誤。		
不實施	進行上述 2) ～ 4) 的訪問時檢測出錯誤。	不檢測出錯誤。	

*2: 關於如何將變址修飾時的軟元件範圍檢查設置爲不進行，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

要 點

對於通用型 QCPU、LCPU，不能指定跨越內部用戶軟元件 (SW) 與文件寄存器 (R) 之間的區域的變址修飾。(否則將發生錯誤，出錯代碼： 4101)

備 註

關於內部用戶軟元件的分配的變更，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

(d) 如果通過直接的訪問輸出 (DY) 進行了變址修飾，將進行軟元件範圍檢查。

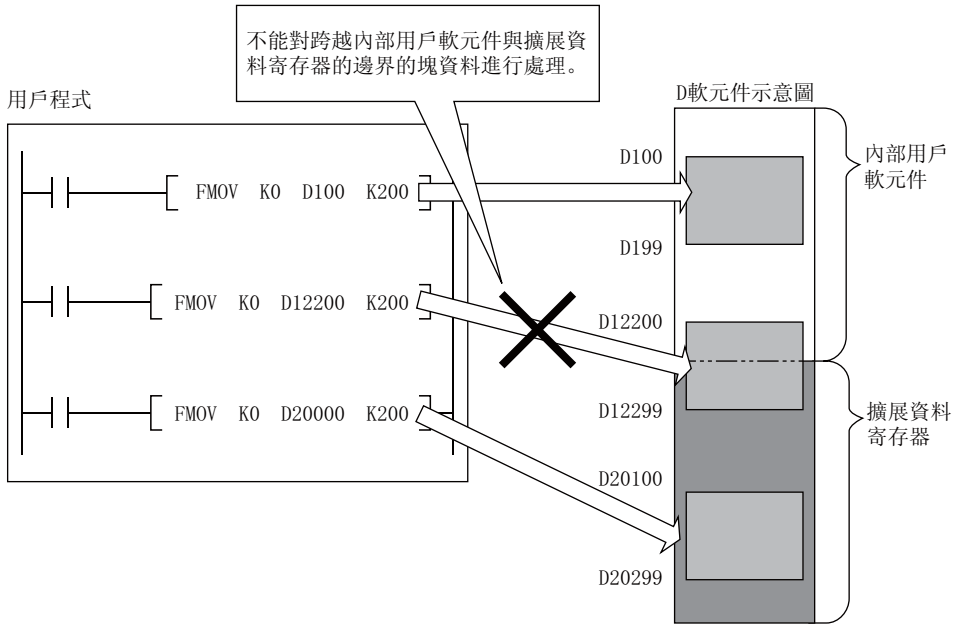
(e) 使用擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 時的注意事項 (除 Q00UJCPU 以外的通用型 QCPU、LCPU)

在下述指定方法中，不能跨越內部用戶軟元件與擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 的邊界進行指定。如果進行了指定，將發生出錯 OPERATION ERROR(出錯代碼： 4101)。

· 變址修飾指定

· 間接指定

· 通過塊資料處理指令進行的指定 *1



*1: 塊資料是指如下所示的資料：

· 通過以 FMOV、BMOV、BK+ 等多字資料爲運算物件的指令所處理的資料

· 通過 SP.FWRITE、SP.FREAD 等指定的 2 字以上資料所構成的控制資料

· 32 位元以上資料格式的資料 (BIN32 位元、實數、軟元件的間接地址)

(2) 軟元件的資料檢查

在 CPU 模組中，對於基本指令和應用指令中使用的軟元件的資料檢查情況如下所示：

(a) 使用 BIN 資料時

即使運算結果上溢或下溢，也不會變爲出錯狀態。

此時進位元標誌也不會爲 ON。

(b) 使用 BCD 資料時

1) 將對各個位進行是否爲 BCD 值 (0 ～ 9) 的檢查。

如果某個位超出了 0 ～ 9 的範圍 (A ～ F)，將會變爲出錯狀態。

2) 即使運算結果上溢或下溢，也不會變爲出錯狀態。

此時進位元標誌同樣也不會爲 ON。

(c) 使用浮點數據時

1) 使用單精確度浮點運算指令時，在運算結果爲如下所示的情況下，將會變爲運算出錯狀態。

1.0×2^{-127} 以下時

1.0×2^{128} 以上時

2) 使用雙精度浮點運算指令時，在運算結果爲如下所示的情況下，將會變爲運算出錯狀態。

1.0×2^{-1023} 以下時

1.0×2^{1024} 以上時

(d) 使用字串資料時

不進行資料檢查。

(3) 至緩衝記憶體的訪問

對緩衝記憶體進行訪問時，建議通過使用了智慧功能模組軟元件 (Un\G0 ～) 的指令進行訪問。

(4) 至多 CPU 共用記憶體的訪問

對多 CPU 共用記憶體進行訪問時，建議通過使用了多 CPU 共用軟元件 (U3En\G10000 ～) 的指令進行訪問。

3.7 指令執行條件

對於 CPU 模組的順控指令、基本指令和應用指令，存在以下 4 種類型的執行條件。

· 常時執行.....執行的指令與軟元件 ON/OFF 狀態無關。

例 LD X0、OUT Y10

· 在 ON 時執行.....在輸入條件爲 ON 的狀態下執行的指令。

例 MOV 指令、FROM 指令

· 在上升沿執行.....只在輸入條件的上升沿 (OFF → ON) 時執行的指令。
PLS 指令、MOVP 指令

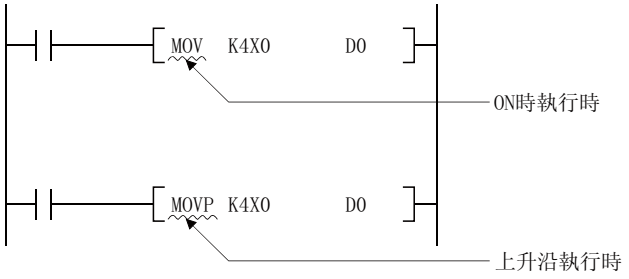
· 在下降沿執行.....只在輸入條件的下降沿 (ON → OFF) 時執行的指令。
PLF 指令

在相當於線圈的基本指令或應用指令中，如果同一指令可以在 “ ON 時執行 ” 或者在 “ 上升沿執行 ”，則在指令名稱後附加一個 “ P ” 用以區別執行條件。

· ON 時執行的指令 指令名稱

· 上升沿執行的指令 指令名稱 +P

在 MOV 指令中，對 ON 時執行或上升沿執行進行指定的方式如下所示：



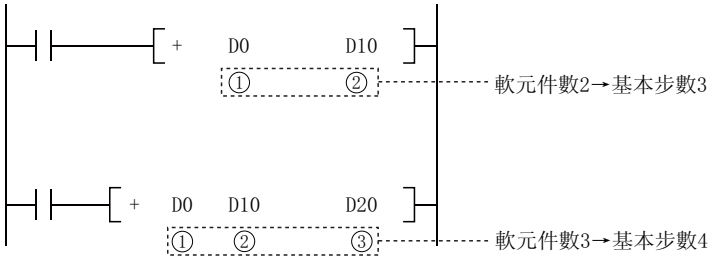
3.8 計算步數

根據使用的軟元件、是否進行了間接設置等，CPU 模組的順控指令、基本指令和應用指令的步數有所不同。

(1) 計算基本步數

基本指令和應用指令的基本步數爲 (軟元件數 +1) 步。

例如，“ + 指令 ”時的情況如下所示：



(2) 步數增加的條件

當使用軟元件的間接指定或增加了步數的軟元件時，步數將會超過基本步數。

(a) 軟元件的間接指定時

當通過 @ 進行了間接指定時，步數在基本步數上增加 1 步。

例如： 當一個 3 步的 MOV 指令被間接指定時 (例： MOV K4X0@D0)，將會增加 1 步而變爲 4 步。

(b) 增加了步數的軟元件 (基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU)

增加了步數的軟元件	增加的步數	示例
智慧功能模組軟元件	1	MOV U4\G10 D0
智慧功能模組軟元件		MOV U3E1\G0 D0
鏈結直接軟元件		MOV J3\B20 D0
變址寄存器		MOV Z0 D0
連號訪問方式文件寄存器		MOV ZR123 D0
32 位常數		DMOV K123 D0
實數常數		EMOV E0.1 D0
字串常數	偶數時： 字元數÷ 2 奇數時： (字元數 +1) ÷ 2	$MOV “ 123 ” D0

(c) 增加了步數的軟元件 (通用型 QCPU(Q00UJCPU 除外)、LCPU)

1) 進行了子集處理的指令

下表所示爲進行了子集處理的指令中的軟元件的步數。

指令符號	增加了步數的軟元件	增加的步數 (指令步數)	基本步數
LD、LDI、AND、ANI、OR、ORI、 LDP、LDF、ANDP、ANDF、ORP、ORF	連號訪問方式文件寄存器、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	1(2)	1
	多 CPU 共用軟元件 *3		
LDPI、LDFI	連號訪問方式文件寄存器、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	1(4)	3
	多 CPU 共用軟元件 *3		
ANDPI、ANDFI、ORPI、ORFI	連號訪問方式文件寄存器、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	1(5)	4
	多 CPU 共用軟元件 *3		
SET	連號訪問方式文件寄存器、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	1(2)	1
	多 CPU 共用軟元件 *3		
OUT	計時器・計數器	3(4)	1
	連號訪問方式文件寄存器、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	1(2)	
	多 CPU 共用軟元件 *3		
RST(位元軟元件)	連號訪問方式文件寄存器、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	1(2)	1
	多 CPU 共用軟元件 *3		
RST(字軟元件)	計時器・計數器 (位 / 字軟元件)	2(4)	2
	連號訪問方式文件寄存器、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	1(3)	
	多 CPU 共用軟元件 *3	1(3)	
LD=、LD<>、LD<、LD<=、LD>、LD>=、 AND=、AND<>、AND<、AND<=、AND>、AND>=、 OR=、OR<>、OR<、OR<=、OR>、OR>=	通用運算寄存器 *2	-1	3
	連號訪問方式文件寄存器、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	1	
	多 CPU 共用軟元件 *3		
LDD=、LDD<>、LDD<、LDD<=、LDD>、LDD>=、 ANDD=、ANDD<>、ANDD<、ANDD<=、ANDD>、 ANDD>=、ORD=、ORD<>、ORD<、ORD<=、 ORD>、ORD>=	通用運算寄存器 *2	-1	3
	連號訪問方式文件寄存器、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	1	
	多 CPU 共用軟元件 *3		
	10 進制常數、16 進制常數、實數常數		
+、-、+P、-P、WAND、WOR、WXOR、WXNR、 WANDP、WORP、WXORP、WXNRP (2 個軟元件)	通用運算寄存器 *2	Ⓓ: -1	3
	連號訪問方式文件寄存器、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	Ⓢ1: 1、Ⓓ: 3	
	多 CPU 共用軟元件 *3		

指令符號	增加了步數的軟元件	增加的步數（指令步數）	基本步數
D+、D-、D+P、D-P、DAND、DOR、DXOR、DXNR、DANDP、DORP、DXORP、DXNRP（2 個軟元件）	通用運算寄存器 *2	Ⓓ：-1	3
	連號訪問方式文件寄存器、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	Ⓢ1：1、Ⓓ：3	
	多 CPU 共用軟元件 *3		
	10 進制常數、16 進制常數、實數常數	Ⓢ1：1	
+、-、+P、-P、WAND、WOR、WXOR、WXNR、WANDP、WORP、WXORP、WXNRP（3 個軟元件）*1	連號訪問方式文件寄存器、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	Ⓢ1、Ⓢ2：1、Ⓓ：2	3
	多 CPU 共用軟元件 *3		
D+、D-、D+P、D-P、DAND、DOR、DXOR、DXNR、DANDP、DORP、DXORP、DXNRP（3 個軟元件）*1	連號訪問方式文件寄存器、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	Ⓢ1、Ⓢ2：1、Ⓓ：2	3
	多 CPU 共用軟元件 *3		
	10 進制常數、16 進制常數、實數常數	Ⓢ1、Ⓢ2：1	
*、*P、/、/P	連號訪問方式文件寄存器、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	Ⓢ1、Ⓢ2：1、Ⓓ：2	3
	多 CPU 共用軟元件 *3		
D*、D*P、D/、D/P、E*、E*P	連號訪問方式文件寄存器、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	Ⓢ1、Ⓢ2：1、Ⓓ：2	3
	多 CPU 共用軟元件 *3		
	10 進制常數、16 進制常數、實數常數	Ⓢ1、Ⓢ2：1	
INC、INCP、DEC、DECP、DINC、DINCP、DDEC、DDECP	變址寄存器 / 通用運算寄存器 *2	-1	2
	連號訪問方式文件寄存器、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	3	
	多 CPU 共用軟元件 *3		
MOV、MOVP	連號訪問方式文件寄存器、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	1	2
	多 CPU 共用軟元件 *3		
DMOV、DMOVP、EMOV、EMOVP	連號訪問方式文件寄存器、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	1	2
	多 CPU 共用軟元件 *3		
	10 進制常數、16 進制常數、實數常數		
BCD、BCDP、BIN、BINP、FLT、FLTP、CML、CMLP	連號訪問方式文件寄存器、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	Ⓢ1：1、Ⓢ2：2	2
	多 CPU 共用軟元件 *3		
DBCD、DBCDP、DBIN、DBINP、INT、INTP、DINT、DINTP、DFLT、DFLTP、DCML、DCMLP	連號訪問方式文件寄存器、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	Ⓢ1：1、Ⓢ2：2	2
	多 CPU 共用軟元件 *3		
	10 進制常數、16 進制常數、實數常數	Ⓢ1：1	

*1: 如果Ⓢ1與Ⓢ2使用相同的軟元件，則基本步數將增加 1 步。

*2: 使用通用運算寄存器時，步數將減少。

*3: 在 LCPU 中不能使用。

在進行了子集處理的指令中，在指令的多處使用了通用運算寄存器時，則步數將減少。下表列出了上述情況下各指令的步數。

指令符號	指定通用運算寄存器的位置	增加的步數（指令步數）	基本步數
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>、 ANDD>=、ORD=、ORD<>、ORD<、ORD<=、 ORD>、ORD>=	ⓈⓈ1與ⓈⓈ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 個軟元件）	ⓈⓈ1與ⒹⒹ	-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 個軟元件） *1	ⓈⓈ1與ⓈⓈ2與ⒹⒹ	-2(1)	3
	ⓈⓈ1或者ⓈⓈ2與ⒹⒹ	-1(2)	
	ⓈⓈ1與ⓈⓈ2 （只有在 ⒹⒹ 中指定了 不增加步數的軟元件時）	± 0(3)	
	ⓈⓈ1與ⓈⓈ2 （只有在 ⒹⒹ 中指定了 連號訪問方式文件寄存器時）	+2(5)	
*、*P、/、/P	ⓈⓈ1與ⓈⓈ2與ⒹⒹ	-2(1)	3
	ⓈⓈ1或者ⓈⓈ2與ⒹⒹ	-1(2)	
D*、D*P、D/、D/P、E*、E*P	ⓈⓈ1與ⓈⓈ2與ⒹⒹ	-2(1)	3
	ⓈⓈ1或者ⓈⓈ2與ⒹⒹ	-1(2)	
	ⓈⓈ1與ⓈⓈ2 （只有在 ⒹⒹ 中指定了 不增加步數的軟元件時）	± 0(3)	
	ⓈⓈ1與ⓈⓈ2 （只有在 ⒹⒹ 中指定了 連號訪問方式文件寄存器時）	+2(5)	
MOV、MOVP、DMOV、DMOVP、EMOV、EMOVP	ⓈⓈ1與ⒹⒹ	-1(1)	2
BCD、BCDP、BIN、BINP、DBCD、DBCDP、 DBIN、DBINP、FLT、FLTP、DFLT、DFLTP、 INT、INTP、DINT、DINTP、CML、CMLP、 DCML、DCMLP	ⓈⓈ1與ⒹⒹ	-1(1)	2

*1: 如果ⓈⓈ1與ⒹⒹ使用相同的軟元件，則基本步數將增加 1 步。

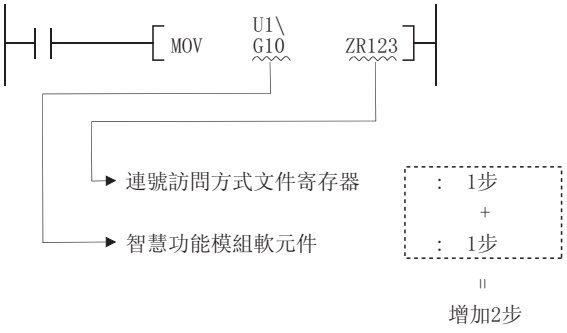
2）未進行子集處理的指令

下表是在未進行子集處理的指令中，增加了步數的軟元件的情況。

增加了步數的軟元件	增加的步數	示例
智慧功能模組軟元件	1	MOV U4\G10 D0
多 CPU 共用軟元件		MOV U3E1\G10000 D0
鏈結直接軟元件		MOV J3\B20 D0
變址寄存器 / 通用運算寄存器		MOV Z0 D0
連號訪問方式文件寄存器		MOV ZR123 D0
32 位常數		DMOV K123 D0
實數常數		EMOV E0.1 D0
字串常數	偶數時： 字元數÷ 2 奇數時：(字元數 +1) ÷ 2	$MOV “ 123 ” D0

(d) 如果在上述 (a) ～ (c) 中的某些條件重疊，則步數將被累加。

例 如果指定了 MOV U1\G10 ZR123，則合計增加 2 步。



3.9 使用同一軟元件的 OUT 指令、SET/RST 指令或 PLS/PLF 指令時的動作

以下介紹在 1 個掃描中多次執行同一軟元件的 OUT 指令、SET/RST 指令或 PLS/PLF 指令時的動作。

(1) 使用同一軟元件的 OUT 指令時

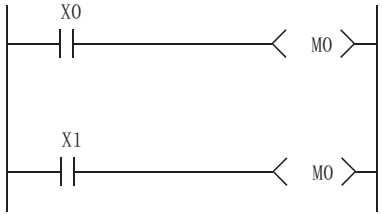
在編程時，應避免在 1 個掃描中多次執行同一軟元件的 OUT 指令。

如果在 1 個掃描多次執行了同一軟元件的 OUT 指令，則在執行各個 OUT 指令時，指定的軟元件都將根據 OUT 指令的運算結果而 ON 或 OFF。

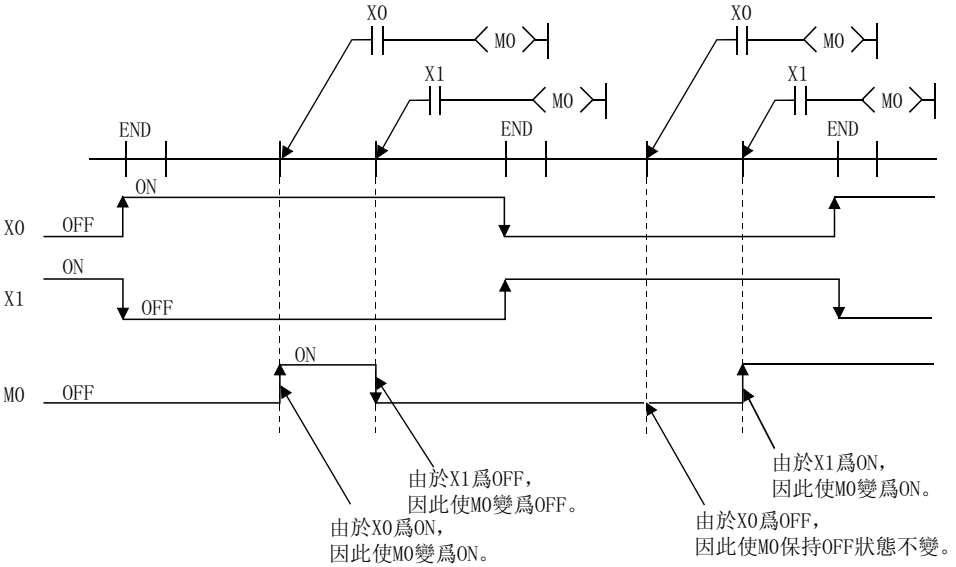
由於各個 OUT 指令的執行都將決定指定軟元件的 ON 或 OFF，這樣在 1 個掃描內軟元件可能會反復變爲 ON 或 OFF。

通過輸入 X0 和 X1 使同一內部繼電器 (M0) 變爲 ON/OFF 的梯形圖的動作如下所示。

[梯形圖]



[時序圖]



對於刷新型 CPU 模組，當通過 OUT 指令指定輸出 (Y) 後，1 個掃描的最後執行的 OUT 指令的 ON/OFF 狀態將被輸出。

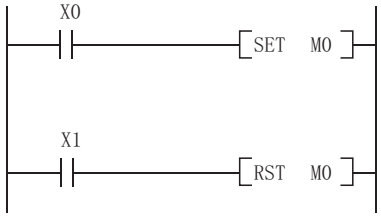
(2) 使用同一軟元件的 SET/RST 指令時

(a) 當 SET 指令爲 ON 時使指定軟元件變爲 ON，此後當執行指令爲 OFF 時，不執行任何動作。
因此，如果在 1 個掃描中對同一個軟元件的 SET 指令執行了多次時，如果某個 SET 指令爲 ON，則指定軟元件將變爲 ON。

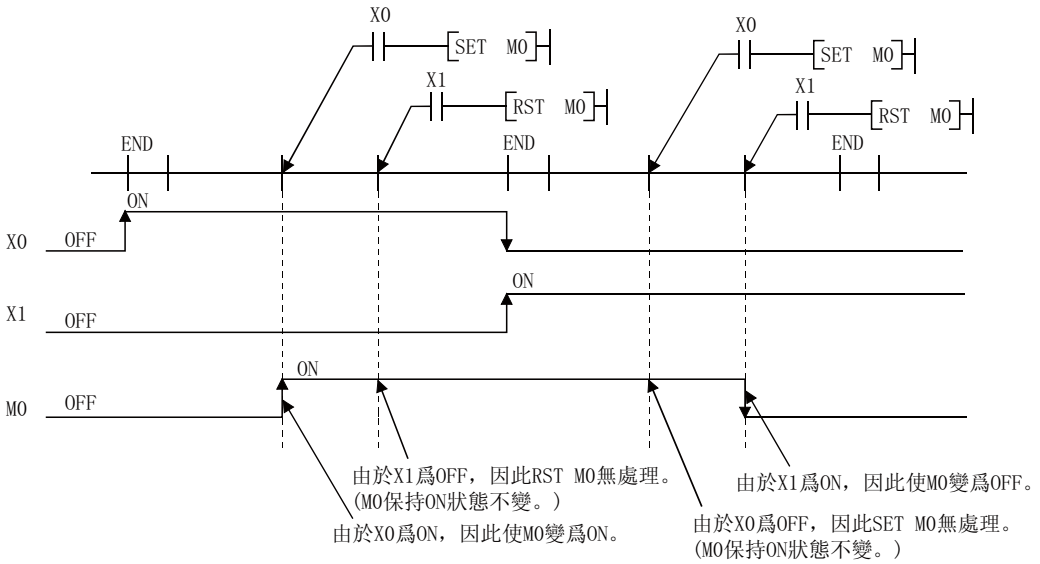
(b) 當 RST 指令爲 ON 時使指定軟元件變爲 OFF，此後當 RST 指令爲 OFF 時，不執行任何動作。
因此，如果在 1 個掃描中對同一個軟元件的 RST 指令執行了多次時，如果某個 RST 指令爲 ON，則指定軟元件將變爲 OFF。

(c) 如果在 1 個掃描中同時存在有同一個軟元件的 SET 指令和 RST 指令，則當 SET 指令爲 ON 時，使指定軟元件變爲 ON，當 RST 指令爲 ON 時，使指定軟元件變爲 OFF。
當 SET 指令和 RST 指令均爲 OFF 時，指定軟元件的 ON/OFF 狀態將不發生變化。

[梯形圖]



[時序圖]



對於刷新型 CPU 模組，當通過 SET/RST 指令指定輸出 (Y) 後，1 個掃描的最後執行的 SET/RST 指令的 ON/OFF 狀態將被輸出。

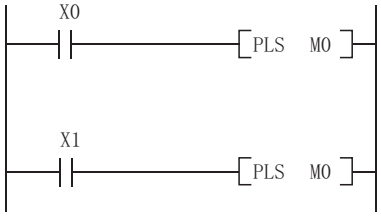
(3) 使用同一軟元件的 PLS 指令時

當 PLS 指令由 OFF → ON 時，使指定軟元件變爲 ON。在除 OFF → ON 以外 (OFF → OFF、ON → ON、ON → OFF) 的情況下，均使指定軟元件變爲 OFF。

當在 1 個掃描中執行了多次同一軟元件的 PLS 指令的情況下，各 PLS 指令由 OFF → ON 時，使指定軟元件變爲 ON。在除 OFF → ON 以外時，均使指定的軟元件變爲 OFF。

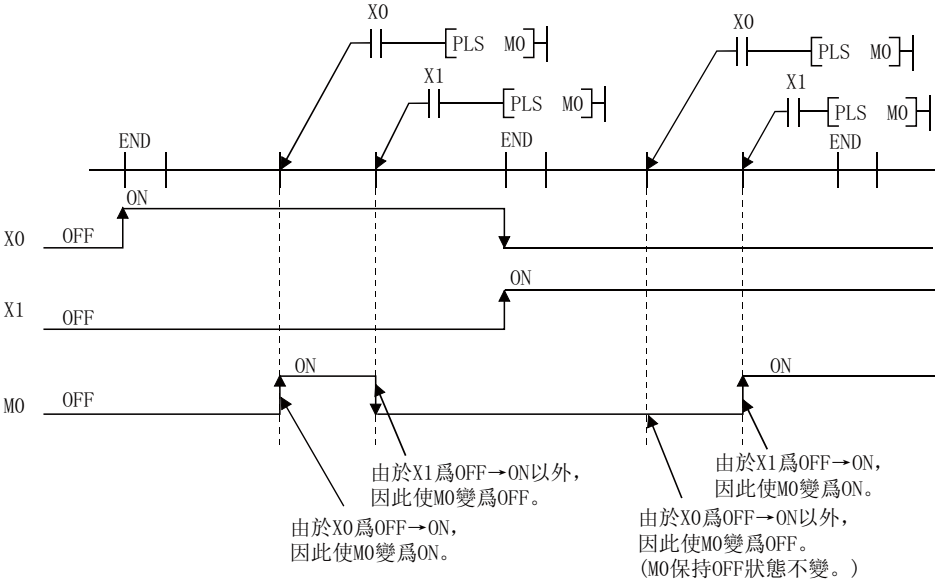
因此，在 1 個掃描中執行了多次同一軟元件的 PLS 指令時，通過 PLS 指令變爲 ON 的軟元件可能在 1 個掃描中不變爲 ON。

[梯形圖]

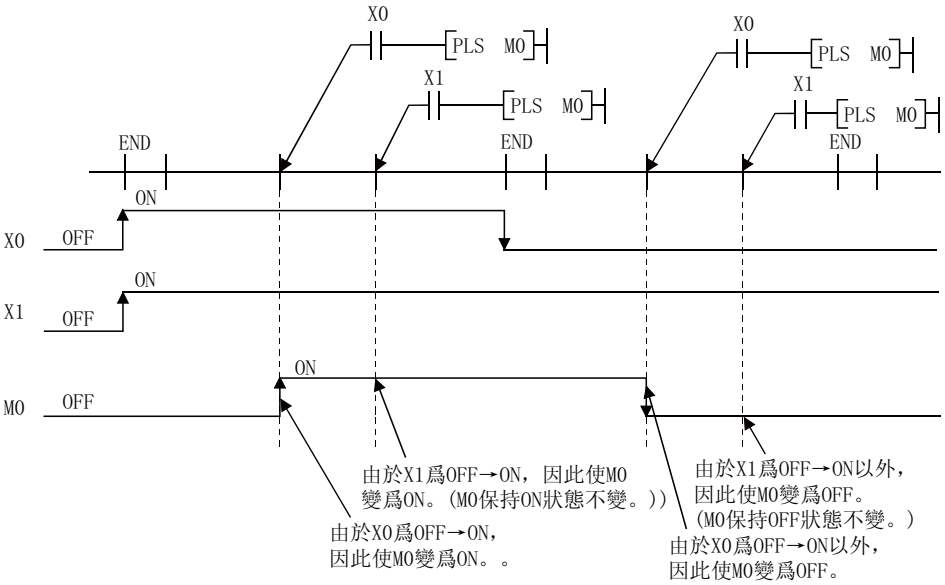


[時序圖]

· X0 與 X1 的 ON/OFF 時機不相同時 (指定軟元件在 1 個掃描內不變爲 ON。)



· X0 與 X1 的 OFF → ON 時機相同時



對於刷新型 CPU 模組，當通過 PLS 指令指定輸出 (Y) 後，1 個掃描的最後執行的 PLS 指令的 ON/OFF 狀態將被輸出。

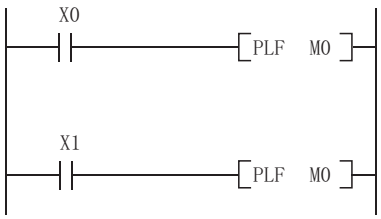
(4) 使用同一軟元件的 PLF 指令時

當 PLF 指令由 ON → OFF 時，使指定軟元件變爲 ON，在除 ON → OFF 以外 (OFF → OFF、OFF → ON、ON → ON) 的情況下，均使指定軟元件變爲 OFF。

當在 1 個掃描中執行了多次同一軟元件的 PLF 指令的情況下，各 PLF 指令由 ON → OFF 時，使指定軟元件變爲 ON。在除 ON → OFF 以外時，均使指定的軟元件變爲 OFF。

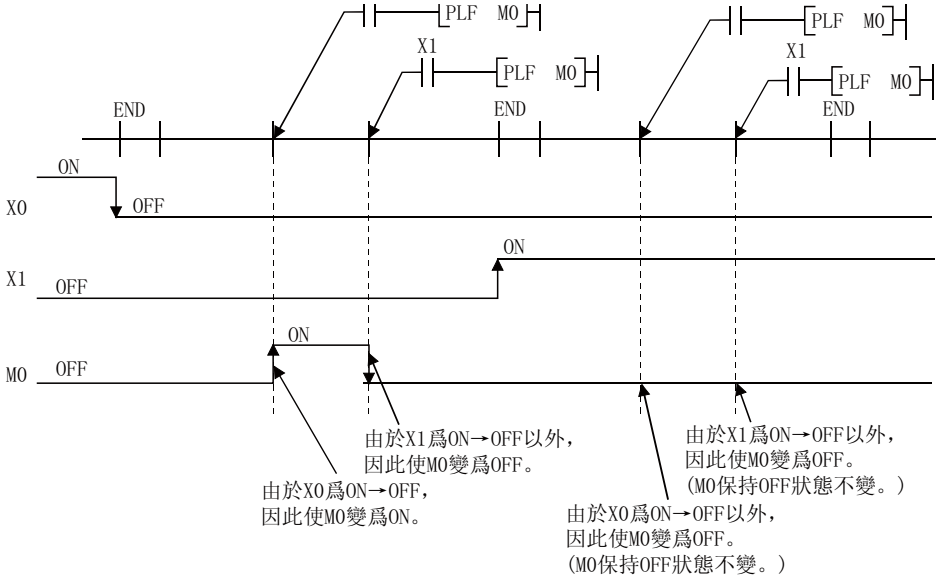
因此，在 1 個掃描中執行了多次同一軟元件的 PLF 指令時，通過 PLF 指令變爲 ON 的軟元件可能在 1 個掃描中不變爲 ON。

[梯形圖]

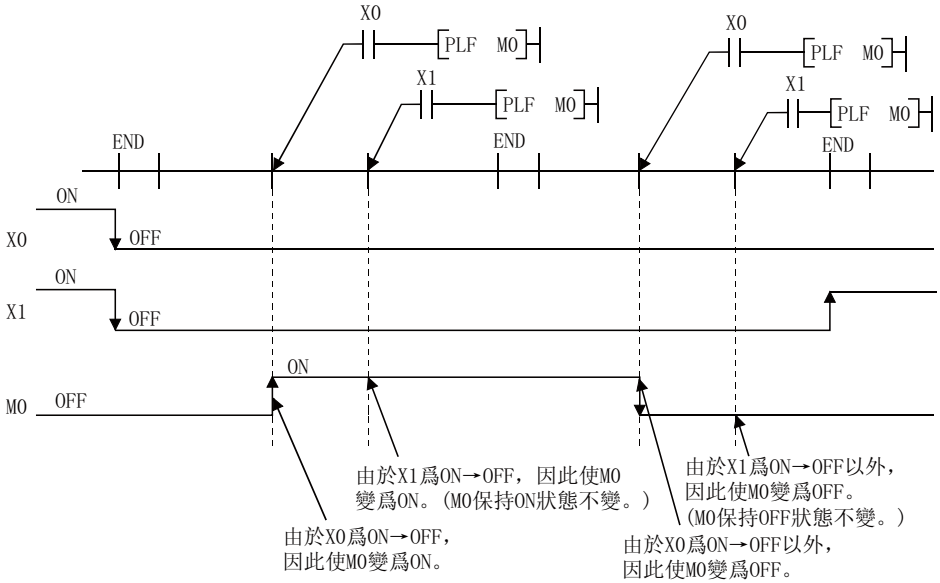


[時序圖]

· X0 與 X1 的 ON/OFF 時機不相同時 (指定軟元件在 1 個掃描內不變爲 ON。)



· X0 與 X1 的 OFF → ON 時機相同時



對於刷新型 CPU 模組，當通過 PLF 指令指定輸出 (Y) 後，1 個掃描的最後執行的 PLS 指令的 ON/OFF 狀態將被輸出。

3.10 使用文件寄存器時的注意事項

本節介紹在 QCPU、LCPU 中使用文件寄存器時的注意事項。

(1) 不能使用文件寄存器的 CPU 模組

Q00JCPU、Q00UJCPU 不能使用文件寄存器。

當使用文件寄存器時，應使用除 Q00JCPU、Q00UJCPU 以外的 CPU 模組。

(2) 所用文件寄存器的設置

如果要使用文件寄存器，需要通過可程式控制器參數或 QDRSET 指令對所用文件寄存器進行設置。

(由於 Q00CPU、Q01CPU 和 LCPU 的可程式控制器參數已被設置爲 “ 使用文件寄存器 ”，因此無需再設置。此外，在 LCPU 中不能使用 QDRSET 指令。)

如果未對所用的文件寄存器進行設置，則在使用了文件寄存器的指令中不能進行正常運算。

要 點

即使未在可程式控制器參數中對所用的文件寄存器進行設置，也可以創建使用了文件寄存器的程式。此外，在除通用型 QCPU、LCPU 以外的模組中，即使將該程式寫入到 CPU 模組中並執行，也不會變爲出錯狀態。
但是，不能對文件寄存器進行正常的資料讀取 / 寫入，應加以注意。
在除通用型 QCPU、LCPU 以外的模組中，如果執行使用了文件寄存器的程式，將會變爲出錯狀態。

(3) 文件寄存器區域的預留

(a) 高性能型 QCPU、過程 CPU、冗餘 CPU、通用型 QCPU(Q00UJCPU 除外)

當使用文件寄存器時，將文件寄存器登錄到標準 RAM/ 存儲卡中，應預留出文件寄存器的區域。

(b) 基本型 QCPU(Q00JCPU 除外)

文件寄存器區域已預先在標準 RAM 中被預留。

用戶無需再對文件寄存器區域進行預留。

(c) LCPU

使用文件寄存器時，將文件寄存器登錄到標準 RAM 中，應預留出文件寄存器的區域。

各 CPU 模組中可使用文件寄存器的記憶體如下表所示。

記憶體	高性能型 QCPU、過程 CPU、冗餘 CPU、通用型 QCPU(Q00UJCPU 除外)	基本型 QCPU (Q00JCPU 除外)LCPU
標準 RAM	○	○
存儲卡 *1 *2	○ *3	×

○ : 可登錄 ; × : 不可登錄

*1: 使用快閃記憶體時，只能從文件寄存器中進行讀取。(不能對快閃記憶體進行寫入。)

*2: 使用 E^2PROM 時，可以通過 PROMWR 指令對 E^2PROM 進行寫入。

*3: 對於 Q00UCPU、Q01UCPU，不能使用存儲卡。

備 註

關於文件寄存器的設置方法及文件寄存器區域的預留方法，請參閱考所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

(4) 文件寄存器編號的指定超過了所登錄的點數時

(a) 對於基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU

即使對超出所登錄點數的編號的文件寄存器進行了讀取 / 寫入，也不會發出出錯資訊信息。

但是，不能對文件寄存器進行正常的資料讀取 / 寫入，應加以注意。

(b) 對於通用型 QCPU、LCPU

對超出所登錄點數的編號的文件寄存器進行了讀取 / 寫入時，將會變爲出錯狀態。

(出錯代碼： 4101)

(5) 文件寄存器的指定方法

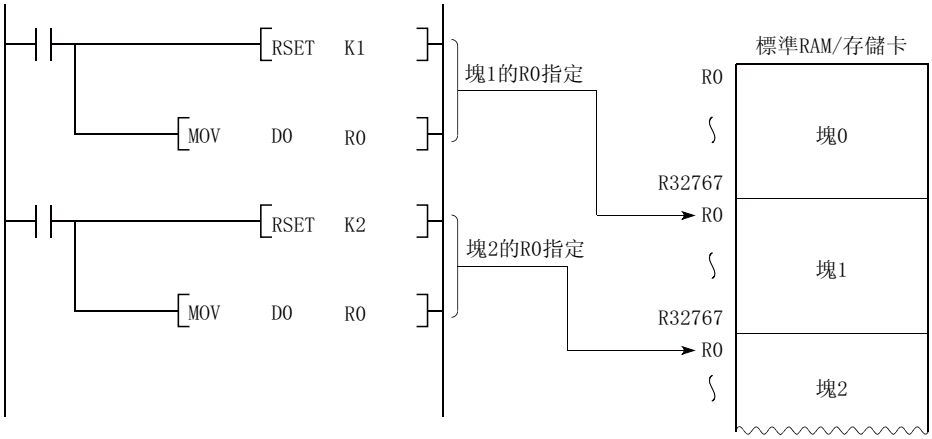
文件寄存器的指定方法有塊切換方式及連號訪問方式這兩種。

(a) 塊切換方式

在塊切換方式中，以 32k 點 (1 個塊) 爲單位對所使用的文件寄存器點數進行分區。

對於 32k 點以上的文件寄存器，通過 RSET 指令對所使用的文件寄存器的塊號進行切換後進行指定。

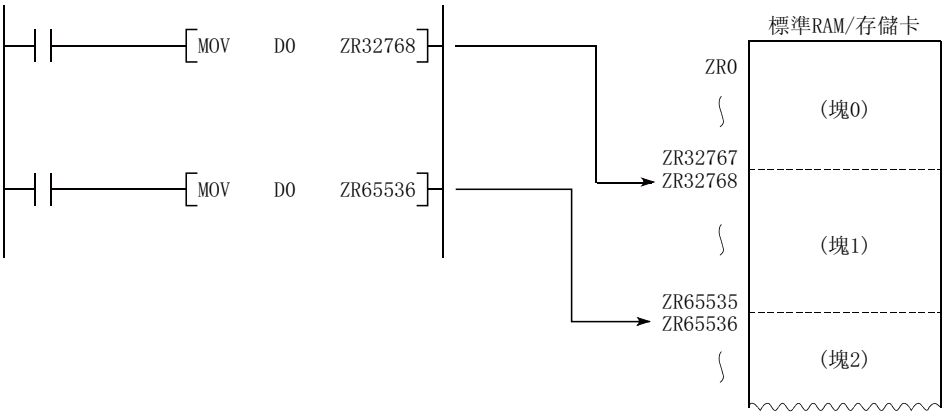
各塊的指定範圍均爲 R0 ～ R32767。



(b) 連號訪問方式

使用連號訪問方式時，通過連續的軟元件號指定超過 32k 點的文件寄存器。

多塊的文件寄存器可以作爲連續的文件寄存器使用。



(6) 將文件寄存器指定爲刷新軟元件時的設置及限制

(a) 刷新軟元件的設置

刷新軟元件的設置可在以下設置中進行：

- CC-Link IE 控制網路的刷新設置 (在 LCPU 中不能設置。)
- CC-Link IE 現場網路的刷新設置 (在基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU、序列號的前 5 位數爲 “ 12011 ” 以前的通用型 QCPU、序列號的前 5 位數爲 “ 13011 ” 以前的 LCPU 中不能設置。)
- MELSECNET/H 的刷新設置 (在 LCPU 中不能設置。)
- CC-Link 的刷新設置
- 智慧功能模組的自動刷新設置
- 多 CPU 系統的自動刷新設置 (在 LCPU 中不能設置。)

(b) 限制事項

將文件寄存器指定爲刷新軟元件時，應注意以下 1) ～ 3) 的限制事項：

1) 在 QCPU 中的可程式控制器參數中，如果進行了使用與程式同名的文件寄存器的指定，將不能正確地進行刷新。
如果使用與程式同名的文件寄存器，將被刷新到與程式設置中設置爲最後編號的程式同名的文件寄存器中。希望讀取或寫入刷新資料時，應使用 QDRSET 指令切換爲相應文件寄存器後進行指定。

2) 如果通過 QDRSET 指令更改了文件寄存器的檔案名或驅動號，將不能正確地執行刷新。(在 LCPU 中不能使用 QDRSET 指令。)
如果通過 QDRSET 指令更改了文件寄存器的檔案名或驅動號，將被鏈結刷新到 END 指令執行時的設置文件中。希望讀取或寫入刷新資料時，應指定爲在 END 指令執行時的設置文件。
但是，在除通用型 QCPU 以外的 QCPU 中，軟元件被指定爲 “ ZR ” 時，如果通過 QDRSET 指令更改了驅動號，將會發生錯誤 (LINK PARA ERROR (3101))，應加以注意。(將軟元件指定爲 “ R ” 時不會變爲出錯狀態。)

3) 當通過 RSET 指令切換了塊號時，將被刷新到切換的塊號的文件寄存器 (R) 中。
通過 RSET 指令切換了塊號時，將被刷新到 END 指令執行時的塊號的文件寄存器 (R) 中。希望讀取或寫入刷新資料時，應指定爲 END 指令執行的塊號。

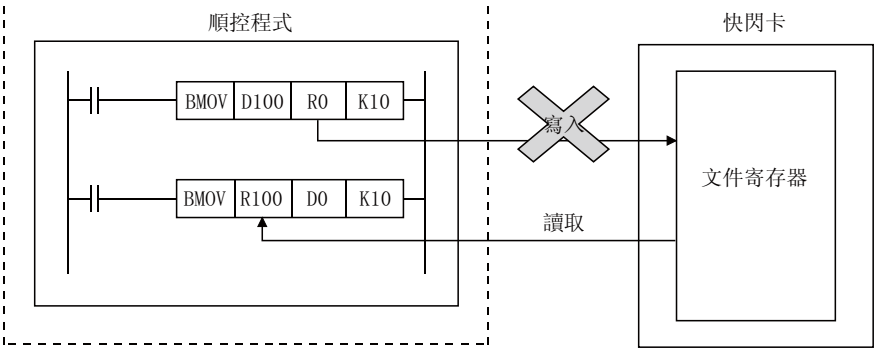
(7) 使用快閃卡中的文件寄存器時的注意事項

以下介紹可使用文件寄存器的快閃卡的注意事項。

(a) 下列快閃卡可以和 QCPU 一起使用。

· Flash 卡

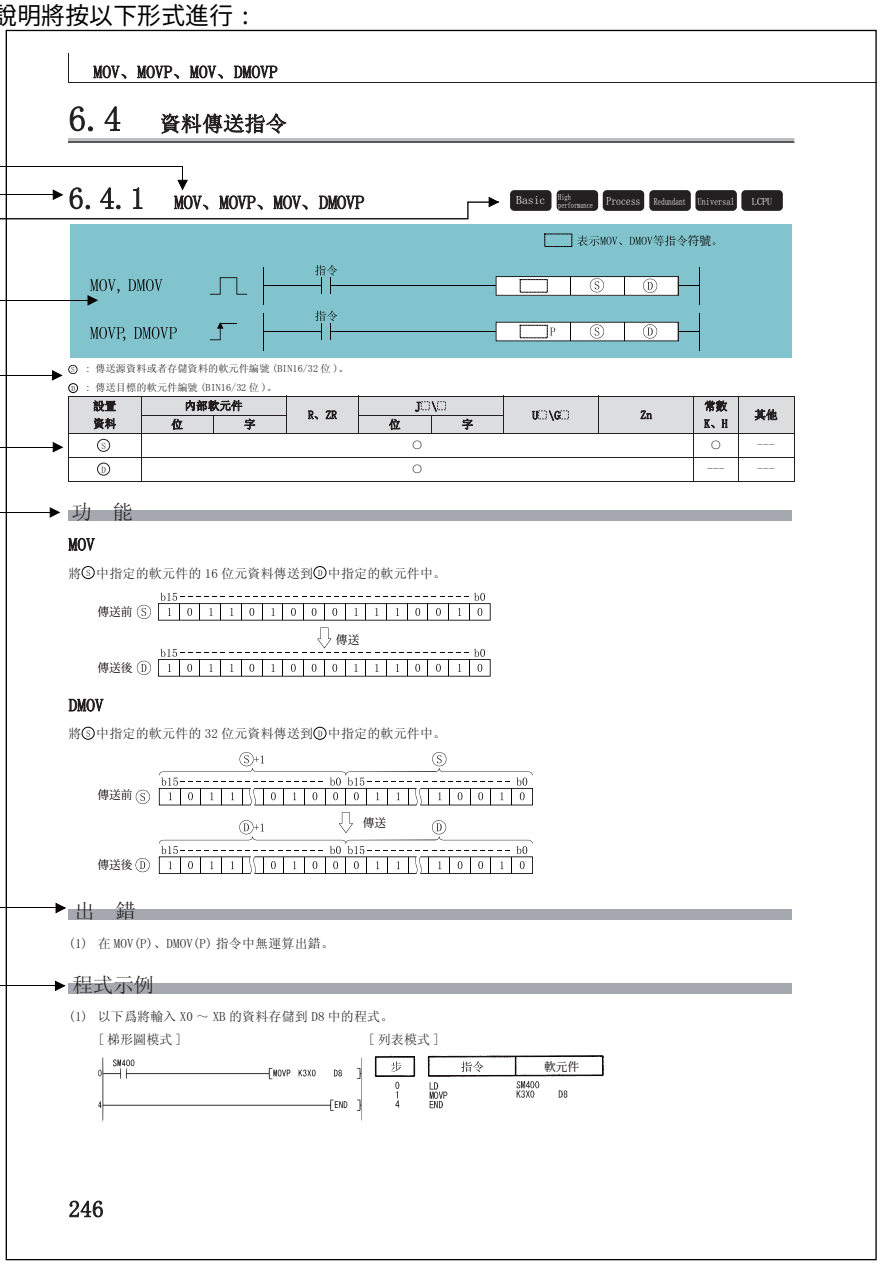
(b) 只有通過順控程式才可以讀取快閃卡中的文件寄存器。
(不能通過順控程式對快閃卡進行寫入。)



將快閃卡作爲文件寄存器使用時，應預先寫入資料。
使用編程工具，將資料寫入快閃卡。

第 4 章　如何閱讀指令

本章以後各章節中指令的說明將按以下形式進行：



1) 表示指令符號。

2) 表示章節號。

3) 表示指令能否在各 CPU 模組中使用。

圖示						內容
基本型 QCPU	高性能型 QCPU	過程 CPU	冗餘 CPU	通用型 QCPU	LCPU	
Basic	High performance	Process	Redundant	Universal	LCPU	普通圖示，表示可以使用相應指令。
Ver. Basic	Ver. High performance	Ver. Process	Ver. Redundant	Ver. Universal	Ver. LCPU	帶版本符號的圖示，表示可以使用但有某些限制。(例如： 功能版本、軟體版本)
Basic	High performance	Process	Redundant	Universal	LCPU	帶×符號的圖示，表示不可以使用的相應指令。

4) 表示梯形圖模式的表示及指令的執行條件。

執行條件	常時執行	在 ON 時執行	在 ON 時執行 1 次	在 OFF 時執行	在 OFF 時執行 1 次
說明頁面的記載符號	未記入				

5) 表示各指令設置資料的說明及資料類型。

資料類型	內容
位元	表示位元資料或位元資料的起始編號。
BIN16 位元	表示處理的 BIN16 位元資料或字軟元件的起始編號。
BIN32 位元	表示處理的 BIN32 位元資料或雙字軟元件的起始編號。
BCD4 位元	表示處理的 4 位元 BCD 資料。
BCD8 位元	表示處理的 8 位元 BCD 資料。
實數	表示處理的浮點數據。
字串	表示處理的字串資料。
軟元件名	表示處理的軟元件名。

6) 指令中可以使用的軟元件用○表示。
可以使用的軟元件型號表示如下：

設置資料	內部軟元件（系統、用戶）		文件寄存器	直接鏈結軟元件 *4 J□\□		智慧功能模組 U□\G□	變址寄存器 Zn	常數 *5	其他 *5
	位	字	R、ZR	位	字				
可用軟元件 *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、BLS、V

*1: 關於各軟元件的說明，請參閱所使用的 CPU 模組的用戶手冊（功能解說 / 程式基礎篇）。
*2: FX 和 FY 只能用於位元資料，FD 只能用於字資料。
*3: 當 T、ST 和 C 用於除以下指令以外時，只能用於字資料。（不能用於位元資料。）
[位元資料中可使用的指令]
LD、LDI、AND、ANI、OR、ORI、LDP、LDF、ANDP、ANDF、ORP、LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI、ORF、OUT、RST、BKRST
*4: 在 CC-Link IE 控制網路、CC-Link IE 現場網路、MELSECNET/H 和 MELSECNET/10 中可以使用。
*5: 可以進行設置的軟元件記錄在“常數”和“其他”欄中。

7) 表示指令的功能。

8) 表示引起出錯的條件及出錯號代碼。
關於未記載的錯誤，請參閱 100 頁 3.6 節。

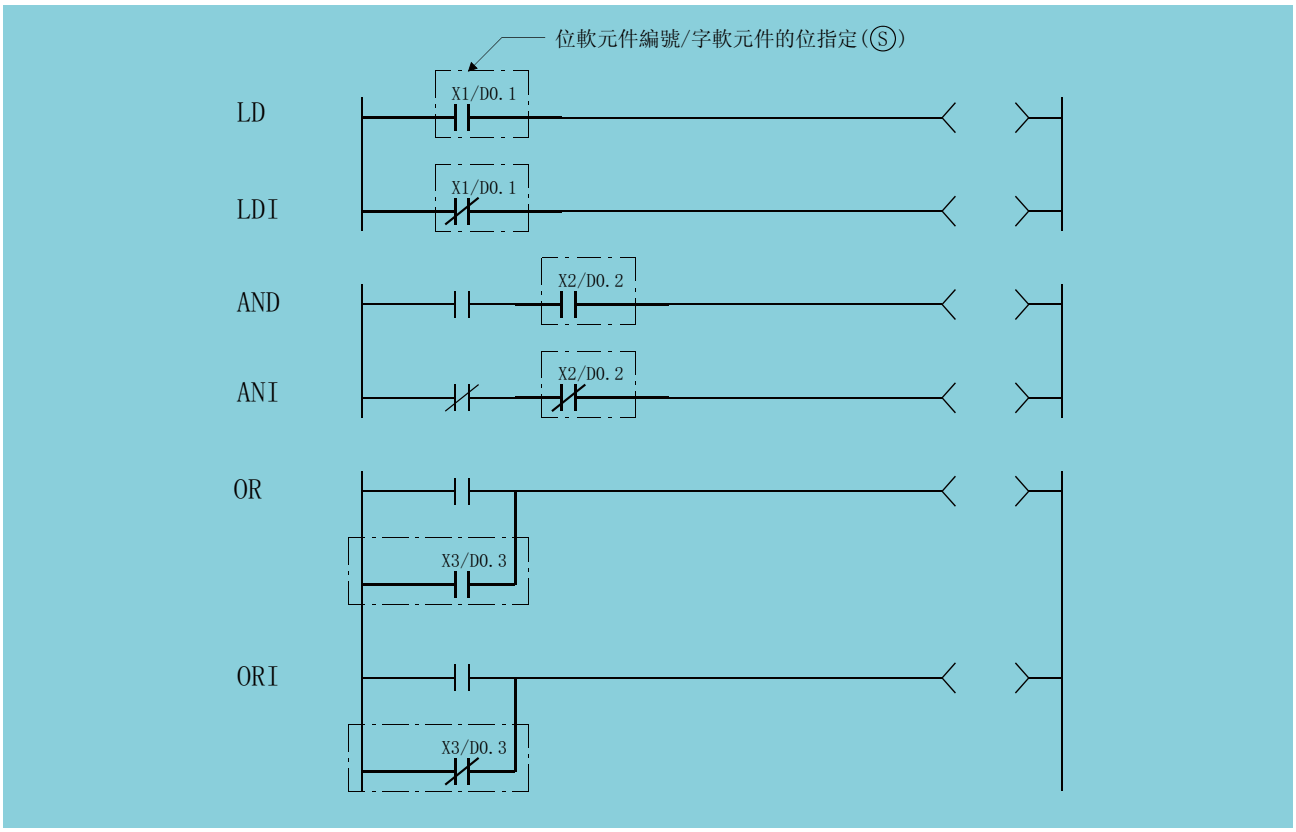
9) 以梯形圖和列表這兩種模式表示簡單的程式示例。
此外，表示該程式被執行時的各軟元件的內容。

第 5 章 順控程式指令

5.1 觸點指令

5.1.1 LD、LDI、AND、ANI、OR、ORI

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ：作爲觸點使用的軟元件（位）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				DX、BL
Ⓢ	○						--		○

功 能

LD、LDI

(1) LD 是 a 觸點運算開始指令，LDI 是 b 觸點運算開始指令。其功能是從指定的軟元件中讀取 ON/OFF 資訊 *1，並將其作爲運算結果。

*1: 進行了字軟元件的位元指定時，根據指定位元的 1/0 而 ON/OFF。

AND、ANI

(1) AND 是 a 觸點串列連接指令，ANI 是 b 觸點串列連接指令，其功能是讀取指定位元軟元件的 ON/OFF 資訊 *2，將其與至目前爲止的運算結果執行 AND 運算，並將該值作爲運算結果。

*2: 進行了字軟元件的位元指定時，根據指定位元的 1/0 而 ON/OFF。

(2) 對 AND 或 ANI 的使用沒有限制，但是在編程工具的梯形圖模式中的情況如下所示：

(a) 寫入 ..... 當 AND 和 ANI 爲串列連接時，最多可以創建 24 級的梯形圖。

(b) 讀取 ..... 當 AND 和 ANI 爲串列連接時，最多可以顯示 24 級的梯形圖。
如果超過 24 級，則最多顯示 24 級。

OR、ORI

(1) OR 是 1 個 a 觸點的並行連接指令，ORI 是 1 個 b 觸點的並行連接指令。其功能是從指定的軟元件讀取 ON/OFF 資訊 *3，將其與至目前爲止的運算結果執行 OR 運算，並將該值作爲運算結果。

*3: 字軟元件的位元指定時，根據指定位元的 1/0 而 ON/OFF。

(2) 對 OR 或 ORI 的使用沒有限制，但是在編程工具的梯形圖模式中的情況如下所示：

(a) 寫入 .....最多可以創建通過 23 個 OR 和 ORI 連接的梯形圖。

(b) 讀取 ..... 最多可以顯示通過 23 個 OR 和 ORI 連接的梯形圖。
第 24 個及以後的梯形圖不能被正確顯示。

備 註

字軟元件位的指定以 16 進制形式進行設置。
D0 的位 b11 變爲 D0.0B。
關於字軟元件的位指定，請參閱 80 頁 3.2.1 項。

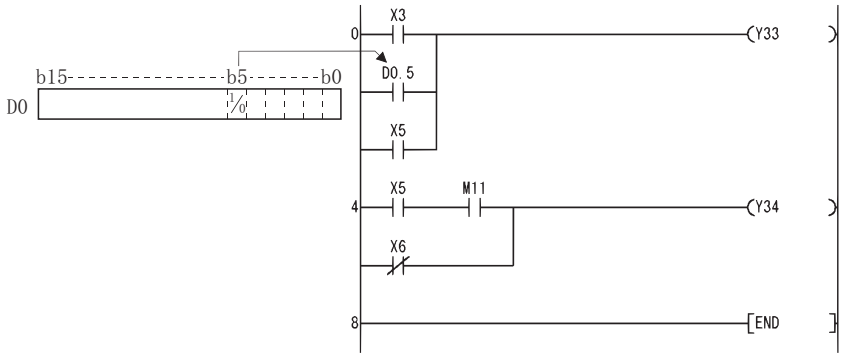
出　錯

(1) 在 LD、LDI、AND、ANI、OR、ORI 指令中無運算錯誤。

程式示例

(1) 使用了 LD、AND、OR、ORI 指令的程式。

[梯形圖模式]

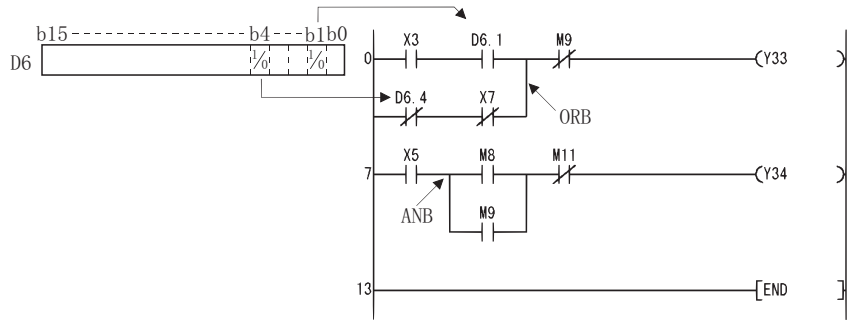


[列表模式]

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

(2) 通過使用 ANB 和 ORB 指令進行觸點連接的程式。

[梯形圖模式]

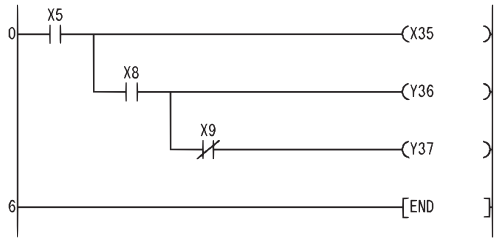


[列表模式]

步	指令	軟元件
0	LD	X3
1	AND	D6.1 } ··· 字軟元件
2	LDI	D6.4
3	ANI	X7 的位指定
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) OUT 指令的並行程式。

[梯形圖模式]

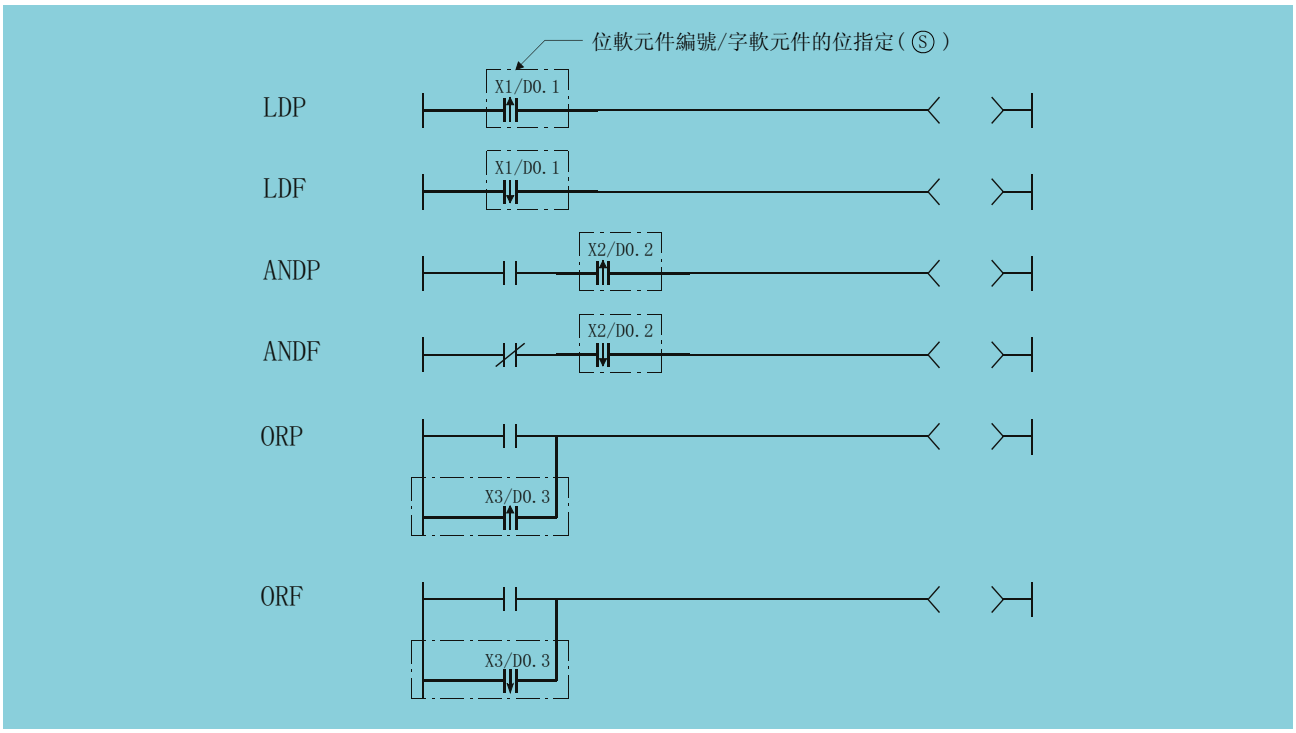


[列表模式]

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

5.1.2 LDP、LDF、ANDP、ANDF、ORP、ORF

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ : 作爲觸點使用的軟元件(位)。

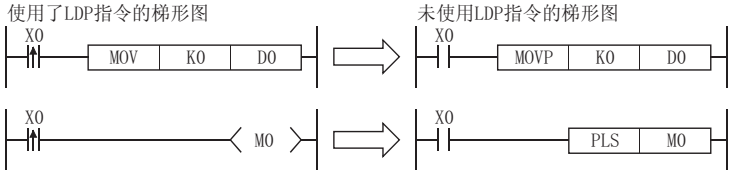
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他 DX
	位	字		位	字				
Ⓢ	○						--		○

功 能

LDP、LDF

(1) LDP 是上升沿脈衝運算開始指令，僅在指定位元軟元件的上升沿 (OFF → ON) 時導通。字軟元件的位指定時，僅在指定位從 0 變爲 1 時導通。

如果只有 1 個 LDP 指令，則與 ON 中執行指令的脈衝化指令 (□P) 相同。



(2) LDF 是下降沿脈衝運算開始指令，並且僅在指定位軟元件的下降沿 (ON → OFF) 時導通。

字軟元件的位指定時，在指定位變爲 1 → 0 時導通。

ANDP、ANDF

(1) ANDP 是上升沿脈衝串列連接指令，ANDF 是下降沿脈衝串列連接指令。其功能爲對至目前爲止的運算結果執行 AND 運算，並將結果值作爲運算結果。

ANDP、ANDF 中使用的 ON/OFF 資訊如下表所示：

ANDP、ANDF 中指定的軟元件		ANDP 的狀態	ANDF 的狀態
位軟元件	字軟元件的位指定		
OFF → ON	0 → 1	ON	OFF
OFF	0	OFF	
ON	1		
ON → OFF	1 → 0		ON

ORP、ORF

(1) ORP 是上升沿脈衝並行連接指令，ORF 是下降沿脈衝串列連接指令，其功能爲對至目前爲止的運算結果進行 OR 運算後將值作爲運算結果。

ORP、ORF 中使用的 ON/OFF 資訊如下表所示：

ORP、ORF 中指定的軟元件		ORP 的狀態	ORF 的狀態
位軟元件	字軟元件位指定		
OFF → ON	0 → 1	ON	OFF
OFF	0	OFF	
ON	1		
ON → OFF	1 → 0		ON

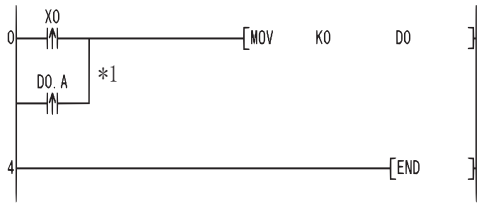
出　錯

(1) 在 LDP、LDF、ANDP、ANDF、ORP、ORF 指令中無運算出錯。

程式示例

(1) 下述程式在輸入 X0 或者在資料寄存器 D0 的 b10(位 11) 的上升沿，執行 MOV 指令：

[梯形圖模式]



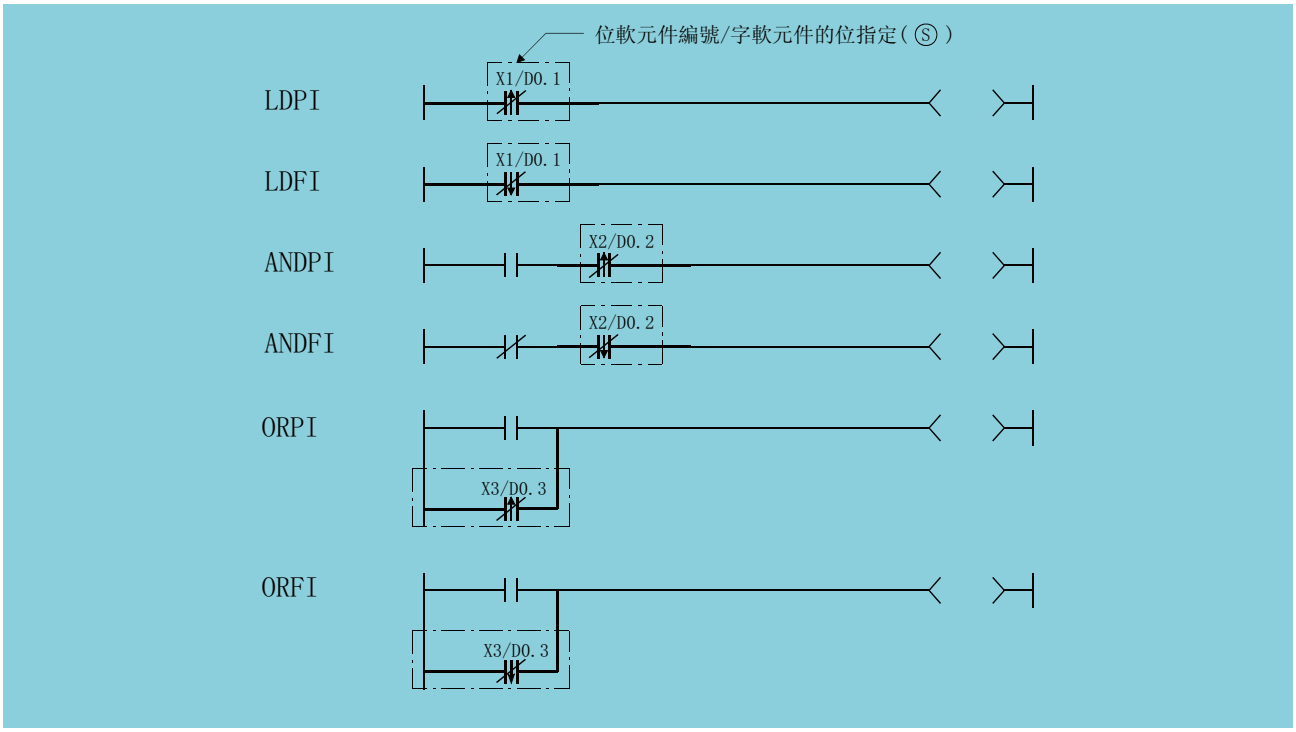
[列表模式]

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

*1:　字軟元件的位指定是以 16 進制數的形式進行的。
D0 的 b10 變爲 D0.0A。

5.1.3 LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他 DX
	位	字		位	字				
Ⓢ	○	--	○	○	--	○	--		○

功　能

LDPI、LDFI

(1) LDPI 是上升沿脈衝否運算開始指令，在指定位元軟元件爲 OFF 時、ON 時、下降沿 (ON → OFF) 時導通。字軟元件的位指定時，在指定位爲 0 時、1 時、0 → 1 變化時導通。

(2) LDFI 是下降沿脈衝否運算開始指令，在指定位元軟元件的上升沿 (OFF → ON) 時、OFF 時、ON 時導通。字軟元件的位指定時，在指定位爲 0 時、1 時、0 → 1 變化時導通。

LDPI、LDFI 中指定的軟元件		LDPI 的狀態	LDFI 的狀態
位軟元件	字軟元件位指定		
OFF → ON	0 → 1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON → OFF	1 → 0	ON	OFF

ANDPI、ANDFI

(1) ANDPI 是上升沿脈衝否串列連接指令，ANDFI 是下降沿脈衝否串列連接指令，其功能是對至目前爲止的運算結果進行 AND 運算，將值作爲運算結果。

ANDPI、ANDFI 中使用的 ON/OFF 資訊如下表所示：

ANDPI、ANDFI 中指定的軟元件		ANDPI 的狀態	ANDFI 的狀態
位軟元件	字軟元件位指定		
OFF → ON	0 → 1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON → OFF	1 → 0	ON	OFF

ORPI、ORFI

(1) ORPI 是上升沿脈衝否並行連接指令，ORFI 是下降沿脈衝否並行連接指令，其功能是對至目前爲止的運算結果進行 OR 運算，將值作爲運算結果。

ORPI、ORFI 中使用的 ON/OFF 資訊如下表所示：

ORPI、ORFI 中指定的軟元件		ORPI 的狀態	ORFI 的狀態
位軟元件	字軟元件位指定		
OFF → ON	0 → 1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON → OFF	1 → 0	ON	OFF

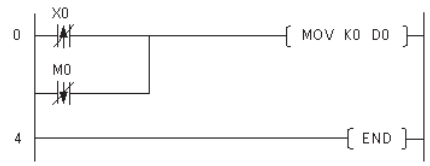
出　錯

(1) 在 LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI 指令中無運算出錯。

程式示例

(1) 下述程式在 X0 爲 ON/OFF/ON → OFF 時，或者 M0 爲 ON/OFF/OFF → ON 時，將 0 存儲到 D0 中。

[梯形圖模式]

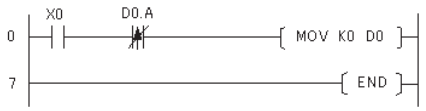


[列表模式]

步	指令	軟元件
0	LDPI	X0
3	ORFI	M0
7	MOV	K0 D0
9	END	

(2) 下述程式在 X0 爲 ON，且 D0 的 b10(位 11) 爲 ON/OFF/ON → OFF 時，將 0 存儲到 D0 中。

[梯形圖模式]



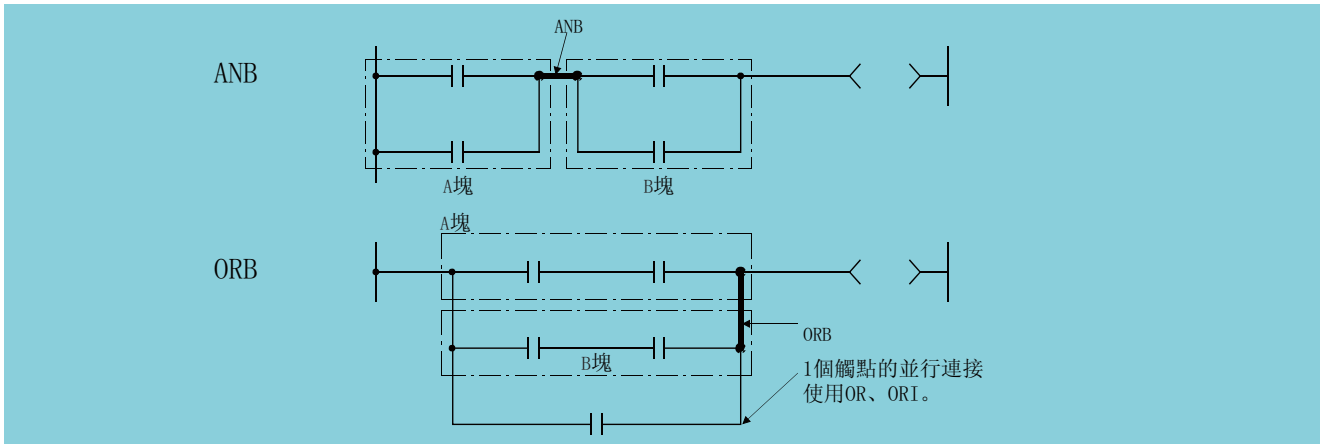
[列表模式]

步	指令	軟元件
0	LD	X0
1	ANDPI	D0.A
5	MOV	K0 D0
7	END	

5.2 連接指令

5.2.1 ANB、ORB

Basic | High performance | Process | Redundant | Universal | LCPU



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
--	--								

功　能

ANB

(1) 進行 A 塊與 B 塊的 AND 運算，並且將結果值作爲運算結果。

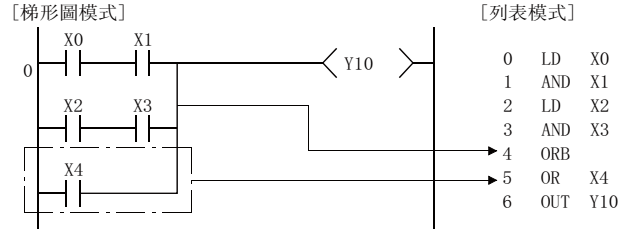
(2) ANB 的符號不是觸點符號，而是連接符號。

(3) 當在列表模式下編程時，最多可以連續寫入 15 條 ANB 指令 (16 塊)。

ORB

(1) 進行 A 塊與 B 塊的 OR 運算，並且將結果值作爲運算結果。

(2) ORB 是用於對有 2 個以上觸點的梯形圖塊執行並行連接的。
對於只有 1 個觸點的梯形圖塊，使用 OR 或 ORI，無需使用 ORB。



(3) ORB 符號不是觸點符號，而是連接符號。

(4) 在列表模式中編程時，最多可以連續使用 15 個 ORB 指令 (16 塊)。

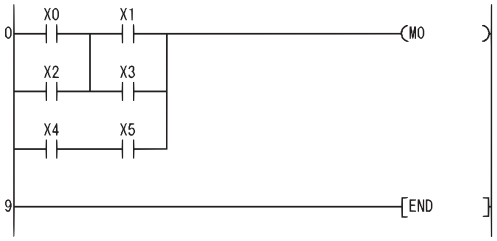
出　錯

(1) 在 ANB、ORB 指令中無運算出錯。

程式示例

(1) 在 ANB、ORB 指令中無運算出錯。

[梯形圖模式]

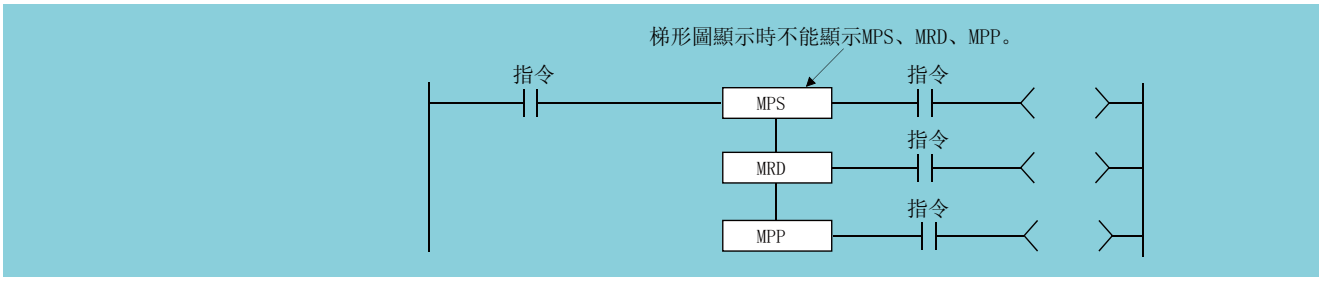


[列表模式]

步	指令	軟元件
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 MPS、MRD、MPP

Basic | High performance | Process | Redundant | Universal | LCPU



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
--	--								

功　能

MPS

(1) 記憶 MPS 指令之前的運算結果 (ON/OFF)。

(2) 最多可以連續使用 16 次 MPS 指令。
如果在中途使用了 MPP 指令，那麼 MPS 指令的使用數將會減少 1 次。

MRD

(1) 讀取通過 MPS 指令記憶的運算結果，並且使用該結果執行下一步運算。

MPP

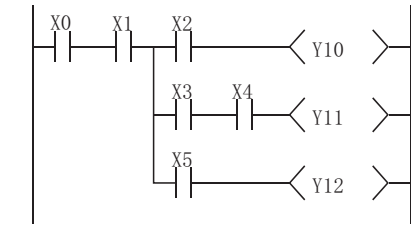
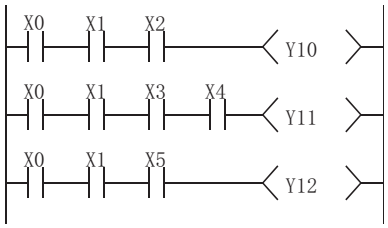
(1) 讀取通過 MPS 指令記憶的運算結果，並且使用該結果執行下一步運算。

(2) 清除通過 MPS 指令記憶的運算結果。

(3) 從 MPS 指令使用次數中減去 1。

要 點

1. 使用及未使用 MPS、MRD、MPP 時的梯形圖如下所示：

使用了 MPS、MRD、MPP 時的梯形圖	未使用 MPS、MRD、MPP 時的梯形圖
	

2. MPS 和 MPP 指令必須使用同樣的次數。
   如果 MPS 和 MPP 指令的使用數不相同，在編程工具的梯形圖模式中將不能正確顯示梯形圖。

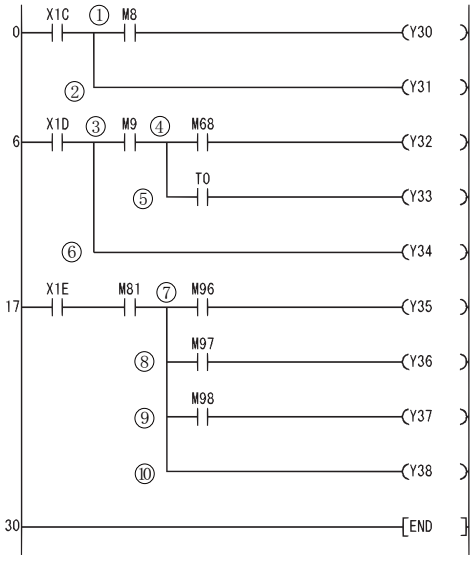
出 錯

(1) 在 MPS、MRD、MPP 指令中無運算出錯。

程式示例

(1) 使用了 MPS、MRD、MPP 的程式。

[梯形圖模式]

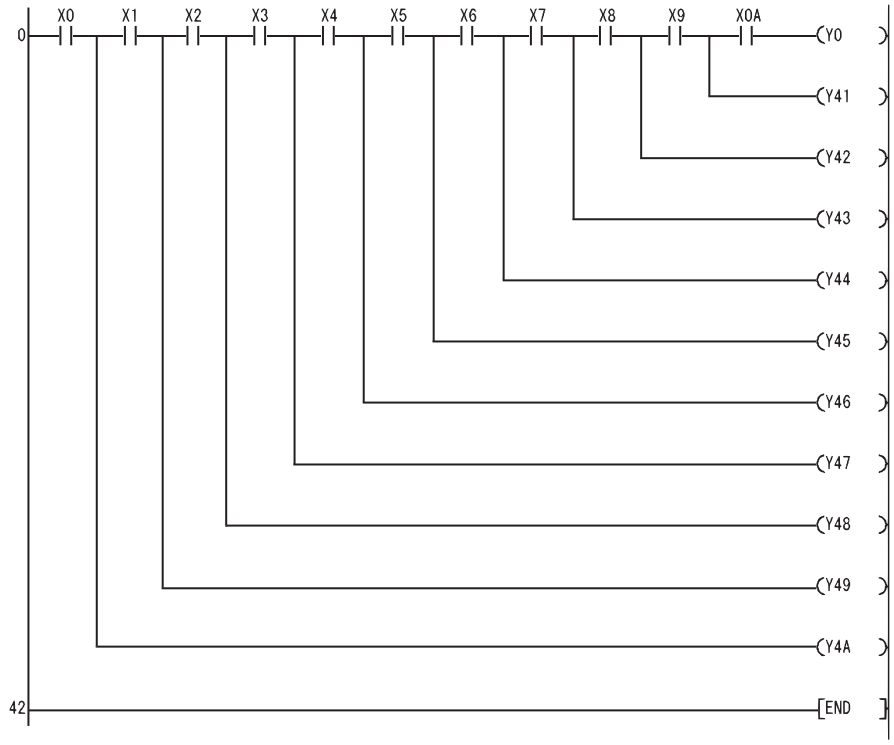


[列表模式]

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

(2) 連續使用 MPS、MPP 的程式。

[梯形圖模式]

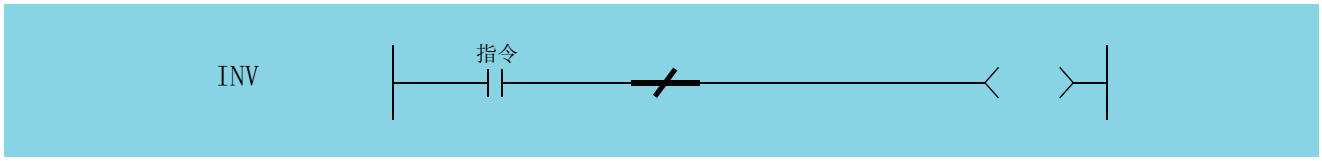


[列表模式]

步	指令	軟元件
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 INV

Basic | High performance | Process | Redundant | Universal | LCPU



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
--	--								

功　能

將 INV 指令之前的運算結果取反。

INV 指令之前的運算結果	INV 指令執行之後的運算結果
OFF	ON
ON	OFF

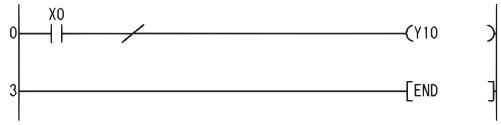
出　錯

(1) 在 INV 指令中無運算出錯。

程式示例

(1) 以下爲將 X0 的 ON/OFF 資料取反後，通過 Y10 輸出的程式。

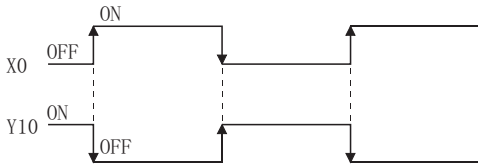
[梯形圖模式]



[列表模式]

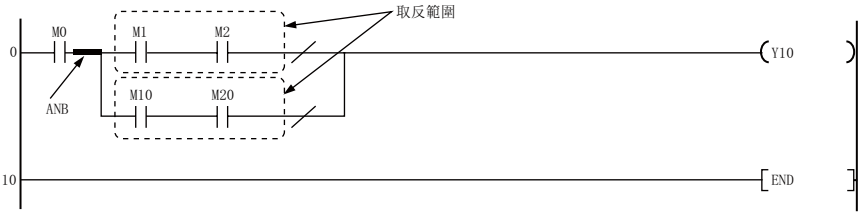
步	指令	軟元件
0	LD	X0
1	INV	
2	OUT	Y10
3	END	

[時序圖]



要 點

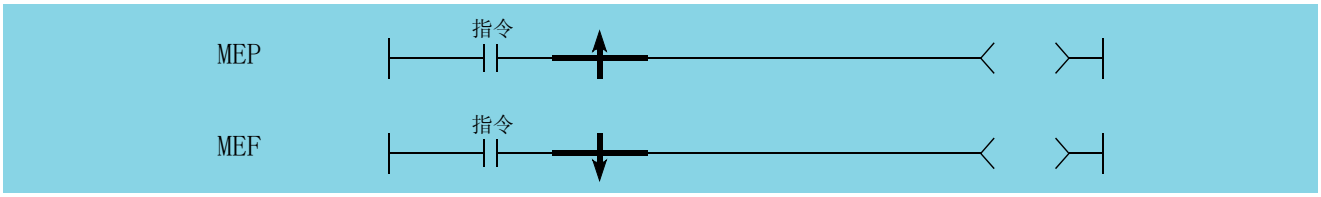
1. 由於 INV 指令是通過 INV 指令之前的運算結果執行動作，因此應在與 AND 指令相同的位置上使用。
   INV 指令不能在 LD、OR 的位置上使用。
2. 使用了梯形圖塊時，按梯形圖塊的範圍對運算結果進行取反。在使 INV 指令及 ANB 指令並用的梯形圖動作時，應注意取反範圍。



關於 ANB 指令的詳細內容，請參閱 126 頁 5.2.1 項。

5.2.4 MEP、MEF

Basic | High performance | Process | Redundant | Universal | LCPU



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
--	--								

功 能

MEP

(1) MEP 指令前的運算結果爲上升沿 (OFF → ON) 時，該指令將變爲 ON(導通狀態)。
如果 MEP 指令前的運算結果爲上升沿以外，則該指令將變爲 OFF(非導通狀態)。

(2) 當多觸點串列連接時，MEP 指令的使用簡化了脈衝化處理。

MEF

(1) MEF 指令前的運算結果爲下降沿 (ON → OFF) 時，該指令將變爲 ON(導通狀態)。
如果 MEF 指令前的運算結果爲下降沿以外，則該指令將變爲 OFF(非導通狀態)。

(2) 當多觸點串列連接時，MEF 指令的使用簡化了脈衝化處理。

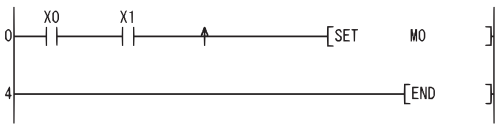
出 錯

(1) 在 MEP、MEF 指令中無運算出錯。

程式示例

(1) 以下爲對 X0 與 X1 的運算結果進行脈衝化的程式：

[梯形圖模式]



[列表模式]

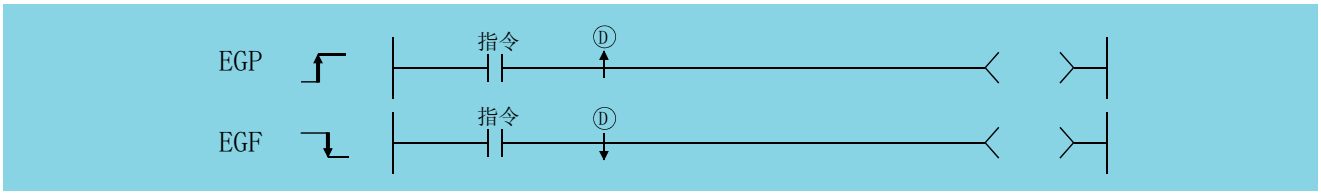
步	指令	軟元件
0	LD	X0
1	AND	X1
2	MEP	
3	SET	M0
4	END	

要 點

1. 在副程式或 FOR ～ NEXT 指令等中，如果使用 MEP 和 MEF 指令對進行了變址修飾的觸點進行脈衝化，有可能會動作不正常。
在副程式或 FOR ～ NEXT 指令等中對進行了變址修飾的觸點執行脈衝化時，應使用 EGP/EGF 指令。
2. 由於 MEP 及 MEF 指令是通過 MEP/MEF 指令之前的 LD 指令起至 MEP/MEF 指令之前爲止的運算結果執行動作的，因此應在與 AND 指令相同的位置上使用。
MEP 及 MEF 指令不能在 LD 或 OR 的位置上使用。

5.2.5 EGP、EGF

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓓ ：存儲運算結果的變址繼電器編號 (位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				V
Ⓓ	--								○

功　能

EGP

(1) 將 EGP 指令之前的運算結果記憶到變址繼電器 (V) 中。

(2) EGP 指令之前的運算結果爲上升沿 (OFF → ON) 時，該指令變爲 ON(導通狀態)。
如果 EGP 指令之前的運算結果爲除上升沿以外 (ON → ON、ON → OFF、OFF → OFF)，該指令將變爲 OFF(非導通狀態)。

(3) EGP 指令用於在副程式或 FOR ～ NEXT 之間對進行了變址修飾的程式執行脈衝運算。

(4) EGP 指令可以同 AND 指令一起使用。

EGF

(1) 將 EGF 指令之前的運算結果通過變址寄存器 (V) 進行記憶。

(2) EGF 指令之前的運算結果爲下降沿 (ON → OFF) 時，該指令將變爲 ON(導通狀態)。
EGF 指令之前的運算結果爲下降沿以外 (OFF → ON、ON → ON、OFF → OFF) 時，則變爲 OFF(非導通狀態)。

(3) EGF 指令用於在副程式及 FOR → NEXT 之間對進行了變址修飾的程式執行脈衝運算。

(4) EGP 指令可以同 AND 指令一起使用。

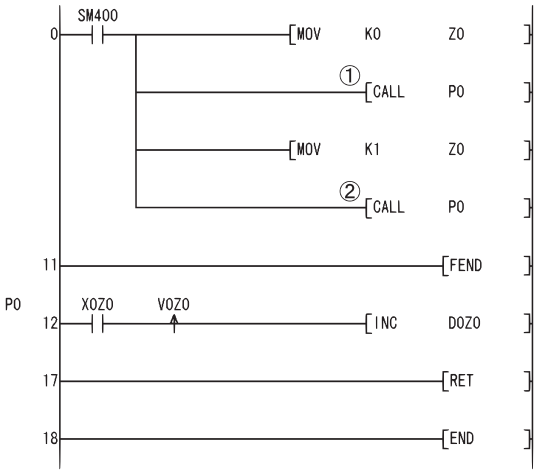
出　錯

(1) 在 EGP、EGF 指令中無運算出錯。

程式示例

(1) 以下爲在副程式中使用 EGP 指令的程式。

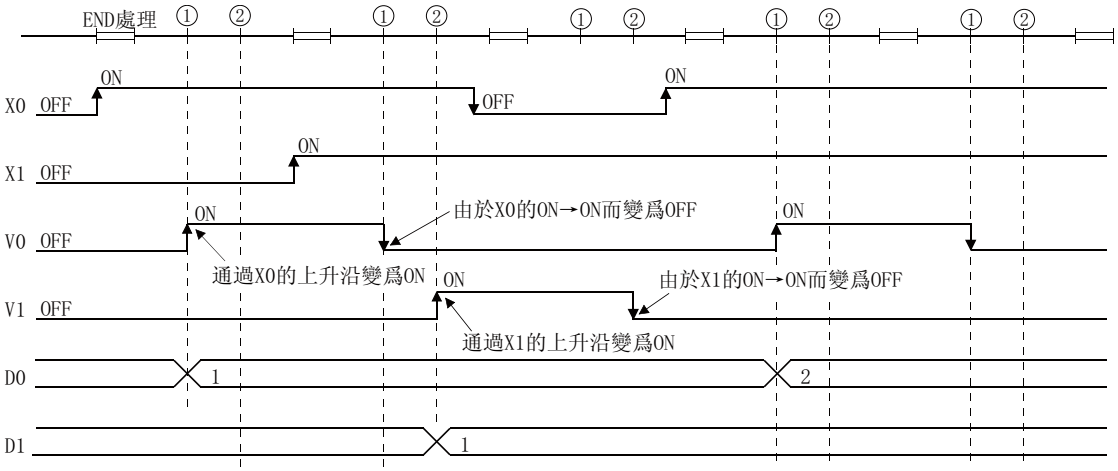
[梯形圖模式]



[列表模式]

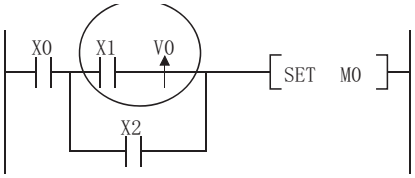
步	指令	軟元件	
0	LD	SM400	
1	MOV	K0	Z0
4	CALL	P0	
6	MOV	K1	Z0
9	CALL	P0	
11	FEND		
12	P0		
13	LD	XOZ0	
14	EGP	VOZ0	
15	INC	DOZ0	
17	RET		
18	END		

[動作]



要 點

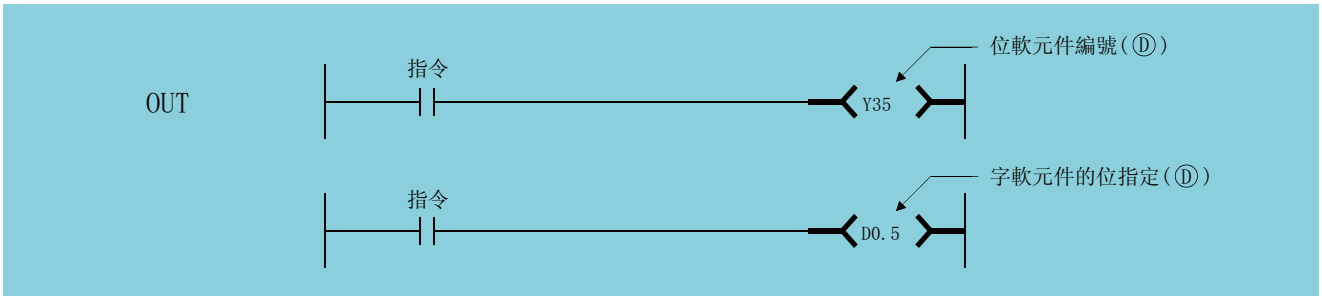
1. 由於 EGP 和 EGF 指令是通過 EGP/EGF 指令之前的 LD 指令起至 EGP/EGF 指令之前爲止的運算結果執行動作的，因此應在與 AND 指令 (參閱 120 頁 5.1.1 項) 相同的位置上使用。
   EGP 和 EGF 指令不能在 LD 或 OR 指令的位置上使用。
2. EGP 和 EGF 指令不能在下面所示的梯形圖塊位置使用。



5.3 輸出指令

5.3.1 OUT

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓓ ::ON/OFF 的軟元件編號(位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他 DY
	位	字		位	字				
Ⓓ	○(除T、C、F以外)		○				--		○

功 能

(1) 將 OUT 指令前的運算結果輸出到指定的軟元件中。

(a) 使用位軟元件時

運算結果	線圈
OFF	OFF
ON	ON

(b) 字軟元件的位指定時

運算結果	指定位
OFF	0
ON	1

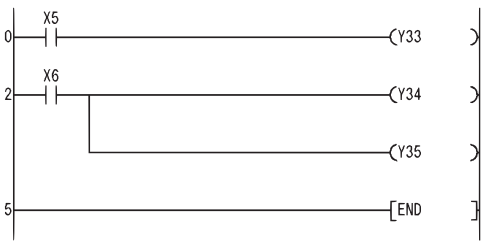
出 錯

(1) 在 OUT 指令中無運算出錯。

程式示例

(1) 使用位軟元件時

[梯形圖模式]

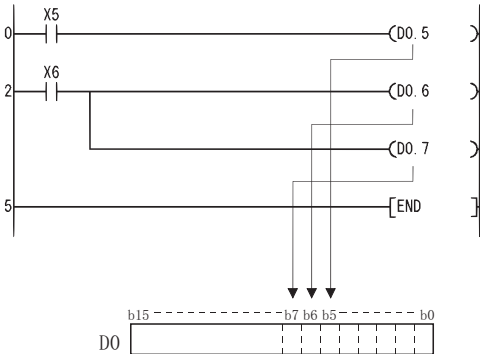


[列表模式]

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

(2) 字軟元件的位指定時

[梯形圖模式]



[列表模式]

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

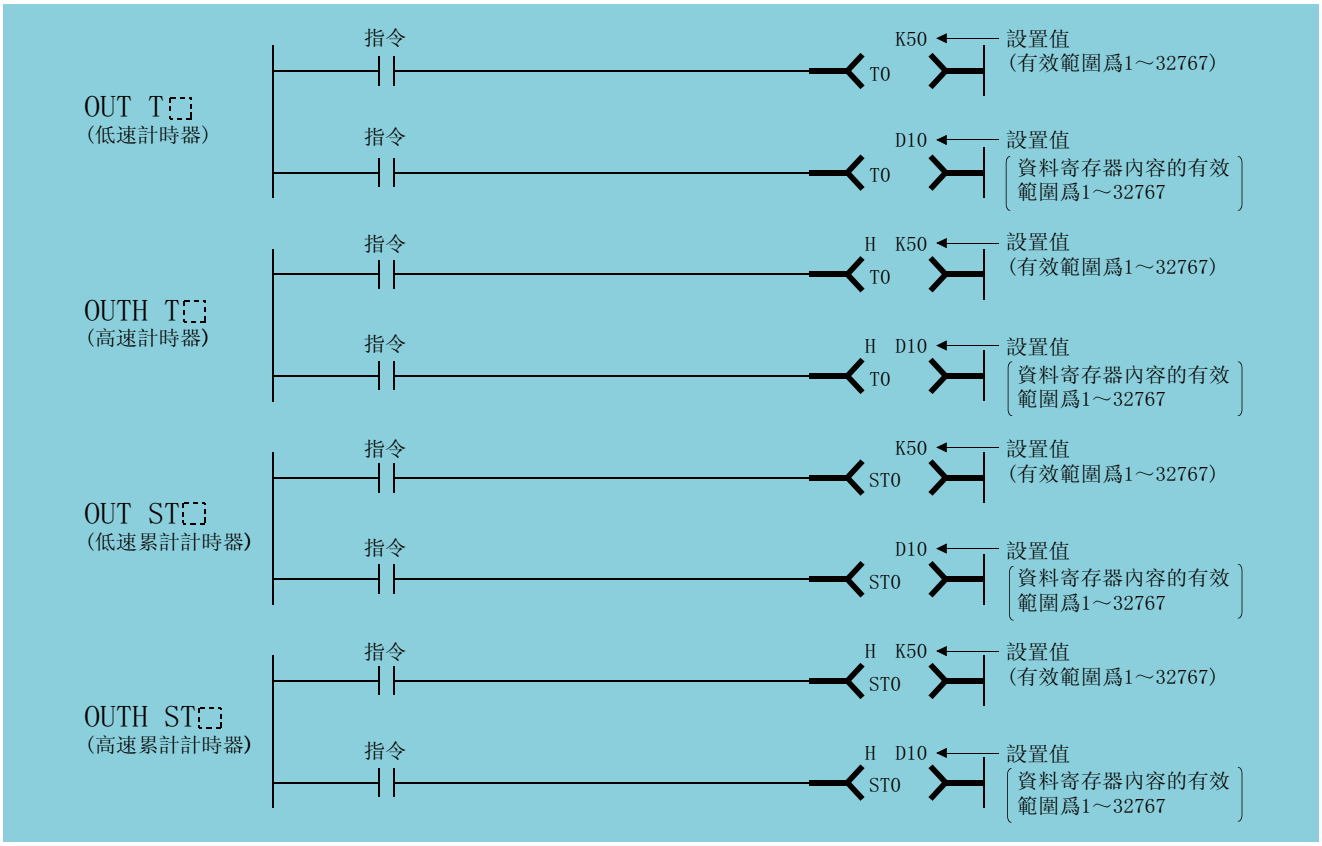
備 註

OUT 指令的基本步數如下：

・使用內部軟元件或文件寄存器 (R) 時 : 1
・使用直接訪問輸出 (DY) 時 : 2
・使用連號訪問方式文件寄存器時
(對於通用型 QCPU、LCPU) : 2
(對於基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU) : 3
・使用除上述以外的其他軟元件時 : 3

5.3.2 OUT T、OUTH T、OUT ST、OUTH ST

Basic | High performance | Process | Redundant | Universal | LCPU

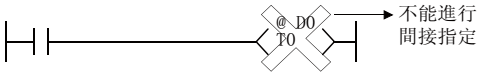


Ⓓ ：計時器編號 (位)

設置值: 計時器的設置值 (BIN16 位 *1)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K	其他
	位	字		位	字				
Ⓓ	○ (僅 T)	--	--	--	--		--	--	--
設置值	--	○ T、C 以外	○	--	○		--	○ *2	--

*1: 不能對計時器值的設置值進行間接指定。



關於間接指定，請參閱 96 頁 3.4 節。

*2: 計時器的設置值只能使用 10 進制常數 (K)。不能使用 16 進制常數 (H) 和實數。

功 能

(1) OUT 指令之前的運算結果爲 ON 時，則計時器的線圈將 ON 且計時器計測到設置值爲止，當到達 “ 時間到 ” (計數值≥設置值) 狀態時，觸點變爲如下狀態：

a 觸點	導通
b 觸點	非導通

(2) OUT 指令前的運算結果爲 ON → OFF 時，觸點變爲如下狀態：

計時器類型	計時器線圈	計時器當前值	在時間到之前		在時間到之後	
			a 觸點	b 觸點	a 觸點	b 觸點
低速計時器	OFF	0	非導通	導通	非導通	導通
高速計時器						
低速累計計時器	OFF	保持當前值	非導通	導通	非導通	非導通
高速累計計時器						

(3) 在時間到之後，通過 RST 指令進行累計計時器當前值的清除及觸點的 OFF。

(4) 不能將設置值設置爲負數 (-32768 ～ -1)。*3
如果設置值爲 0，當執行 OUT 指令時，計時器將會變爲 “ 時間到 ” 狀態。

*3: 通過字軟元件 ((D、W、R、ZR、J□\□、U□\G□) 指定計時器設置時，不進行設置值的範圍檢查。爲了防止設置值中被輸入了負數，應通過用戶程式進行設置值的範圍檢查。

(5) 執行 OUT 指令時，將進行下列處理：
· OUT T□的線圈的 ON/OFF
· OUT T□的觸點的 ON/OFF
· OUT T□的當前值的變更
如果在 OUT T□指令爲 ON 的狀態下，通過 JMP 指令等跳過了 OUT T□指令時，將不進行當前值的更新以及觸點的 ON/OFF 動作。
此外，如果在同一個掃描內同一個 OUT T□指令被執行了 2 次以上，則按執行次數對當前值進行更新。

(6) 計時器線圈 / 觸點的變址修飾只能通過 Z0 或 Z1 進行。
對於計時器的設置值，無變址修飾方面的限制。

備 註

1. 關於計時器的時限
計時器的時限是在可程式控制器參數的可程式控制器系統設置中進行設置。

計時器類型	基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU		通用型 QCPU、LCPU	
	設置範圍	設置單位	設置範圍	設置單位
低速計時器 低速累計計時器	1ms ～ 1000ms (預設值： 100ms)	1ms	1ms ～ 1000ms (預設值： 100ms)	1ms
高速計時器 高速累計計時器	0.1ms ～ 100ms (預設值： 10.0ms)	0.1ms	0.01ms ～ 100ms (預設值： 10.0ms)	0.01ms

2. 關於計時器的計數方法，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。
3. OUT T□指令的基本步數爲 4 步。

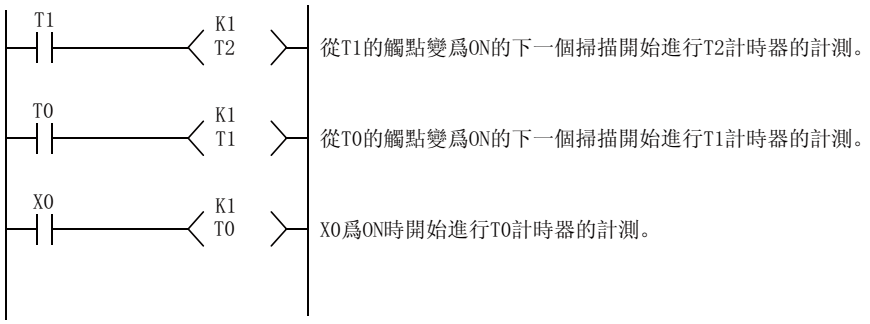
出 錯

(1) 在 OUT 指令中無運算出錯。

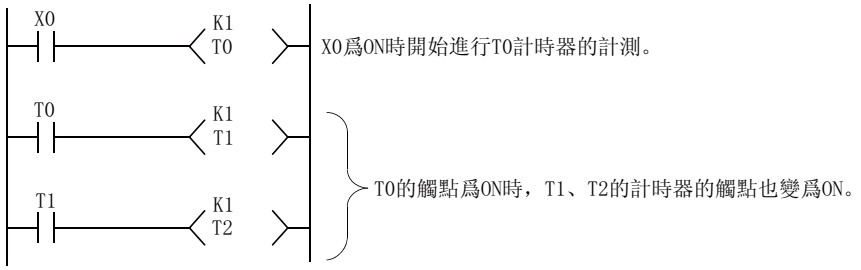
注意事項

(1) 創建通過計時器的觸點開始進行其他計時器的計測的程式時，應按照從後計測的計時器開始的順序進行編程。
在下列情況下，如果按照計測順序進行編程，則全部的計時器將在同一個掃描中變爲 ON。
· 設置值小於掃描時間時。
· 設置值爲 “ 1 ” 時。

例
· 對 T0 ～ T2 的計時器按照從後計測的計時器開始的順序進行了編程時。



· 對 T0 ～ T2 的計時器按照計測順序進行了編程時。



程式示例

(1) 以下爲在 X0 變爲 ON 的 10 秒後使 Y10、Y14 變爲 ON 的程式。

[梯形圖模式]

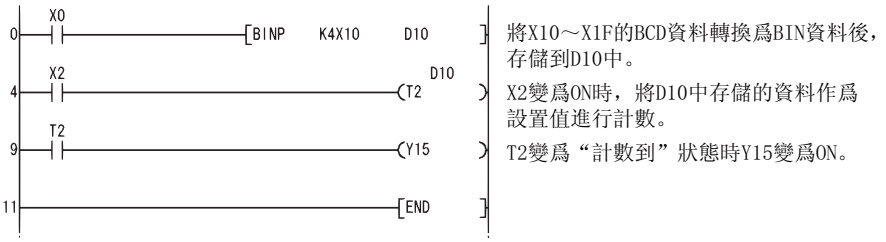
[列表模式]

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

*3: 表示低速計時器的設置值爲預設值時限 (100ms) 的情況下。

(2) 以下爲將 X10 ～ X1F 的 BCD 資料作爲計時器的設置值的程式。

[梯形圖模式]

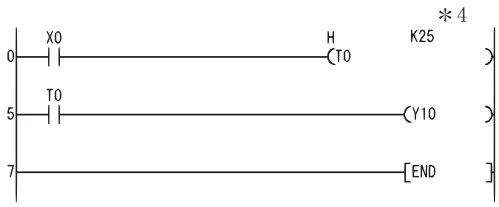


[列表模式]

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

(3) 以下爲在 X0 變爲 ON 的 250ms 後使 Y10 變爲 ON 的程式。

[梯形圖模式]



[列表模式]

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

*4: 表示高速計時器的設置值爲預設值時限 (10ms) 的情況下。

5.3.3 OUT C

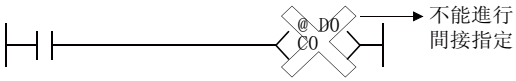
Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ : 計數器編號 (位)。

設置值: 計數器設置值 (BIN16 位 *1)

設置資料	內部軟元件		R ,ZR	J□\□		U□\G□	Zn	常數 K	其他
	位	字		位	字				
Ⓓ	○ (僅 C)	--	--	--	--		--	--	--
設置值	--	○ (除 T、C 以外)	○	--	○		--	○ *2	--

*1: 不能對計數器的設置值進行間接指定。



關於間接指定，請參閱 96 頁 3.4 節。

*2: 計時器的設置值只能使用 10 進制常數 (K)。不能使用 16 進制常數 (H) 和實數。

功 能

(1) OUT 指令之前的運算結果爲由 OFF 變爲 ON 時，在當前值 (計數值) 上加上 1，變爲 “ 計數到 ” (當前值≥設置值) 狀態時，觸點將變爲如下狀態：

a 觸點	導通
b 觸點	非導通

(2) 當運算結果爲 ON 不變時，不進行任何計數。(不需要進行計數輸入的脈衝化。)

(3) 變爲 “ 計數到 ” 狀態後，在執行 RST 指令之前，計數值及觸點的狀態不發生變化。

(4) 設定值不能爲負數 (-32768 ～ -1)。
此外，設置值爲 0 時，將進行與 1 相同的處理。

(5) 計數器線圈 / 觸點的變址修飾只能通過 Z0 或 Z1 進行。
對於計數器的設置值，無變址修飾方面的限制。

備 註

1. 關於計數器的計數方法，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。
2. OUT C□指令的基本步數爲 4 步。

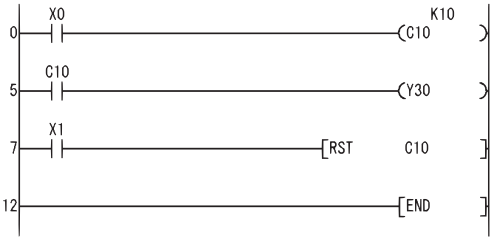
出 錯

(1) 在 OUT 指令中無運算出錯。

程式示例

(1) 以下爲 X0 10 次 ON 之後 Y30 變爲 ON，X1 變爲 ON 時對計數器進行重定的程式。

[梯形圖模式]

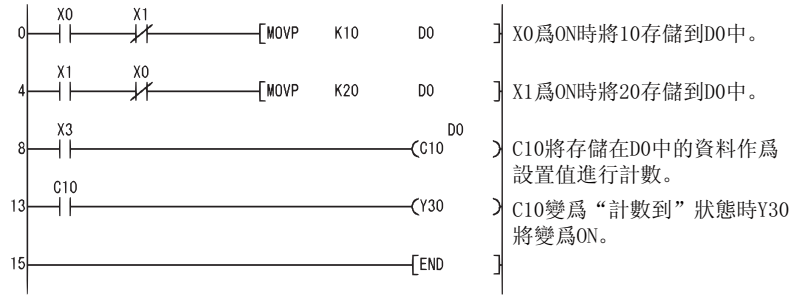


[列表模式]

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

(2) 以下爲 X0 爲 ON 時將 C10 的設置值設置爲 10，X1 爲 ON 將 C10 的設置值設置爲 20 的程式。

[梯形圖模式]



[列表模式]

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

5.3.4 OUT F

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：變爲 ON 的報警器編號 (位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	○ (僅 F)		--						

功　能

(1) 將 OUT 指令之前的運算結果輸出到指定的報警器。

(2) 當報警器 (F) 爲 ON 時的情況如下所示：

- “ USER ” / “ ERR. ” LED 亮燈。
- 將變爲 ON 的報警器號 (F 編號) 儲存到特殊寄存器 (SD64 ～ SD79) 中。
- 將 SD63 的值增加 1。

(3) SD63 的值變爲 16(有 16 個報警器已變爲 ON) 時，即使又有報警器變爲 ON，變爲 ON 的報警器編號也不會儲存到 SD64 ～ SD79 中。

(4) 通過 OUT 指令將報警器變爲 OFF 時的情況如下所示：

線圈變爲 OFF，但是“ USER ”/“ ERR. ”LED 的狀態及儲存在 SD63 ～ SD79 中的內容不發生變化。

希望使“ USER ”/“ ERR ”LED 熄燈，從 SD63 ～ SD79 中刪除通過 OUT F□指令變爲 OFF 的報警器時，可以使用 RST F 指令。

出　錯

(1) 在 OUT 指令中無運算出錯。

備 註

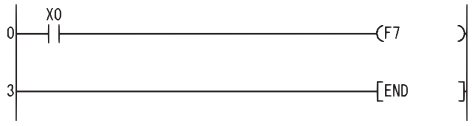
1. 關於報警器的詳細內容，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。
2. OUT F□指令的基本步數爲 2 步。
3. CPU 模組前面的 LED 顯示器或“ USER ”LED 的 CPU 模組類型如下表所示：

LED 類型	CPU 模組型號
“ USER ”LED	高性能型 QCPU、過程 CPU、冗餘 CPU、通用型 QCPU、LCPU
“ ERR. ”LED	基本型 QCPU

程式示例

(1) 以下爲 X0 爲 ON 時使 F7 變爲 ON，且將 7 存儲到 SD64 ～ SD79 中的程式。

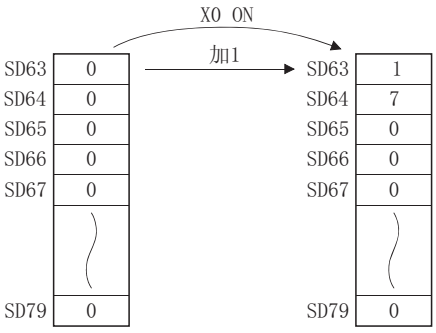
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0
1	OUT	F7
3	END	

[動作]



5.3.5 SET

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：設置 (ON) 的位軟元件編號 / 字軟元件的位指定 (位)。

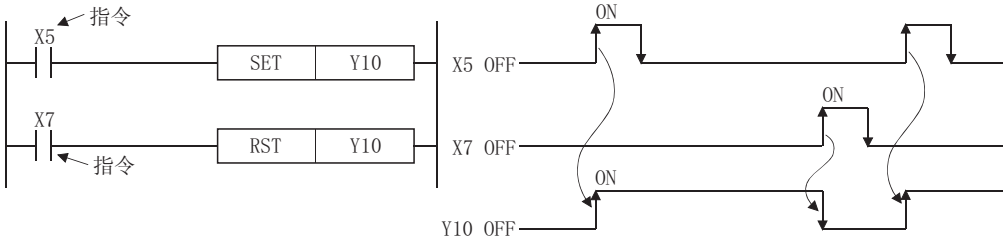
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他 BL，DY
	位	字		位	字				
Ⓓ	○	○ (除 T、C 以外)		○			--		○

功　能

(1) 當執行指令爲 ON 時，指定軟元件的狀態如下：

軟元件	軟元件狀態
位軟元件	使線圈、觸點變爲 ON。
字軟元件的位指定時	將指定位變爲 1。

(2) 對於變爲 ON 的軟元件，即使執行指令變爲 OFF，也仍將保持爲 ON 狀態不變。
通過 SET 指令變爲 ON 軟元件可以通過 RST 指令使其變爲 OFF。



(3) 當執行命令爲 OFF 時，軟元件的狀態不發生變化。

出　錯

(1) SET 指令中無運算出錯。

程式示例

(1) 以下爲在 X8 爲 ON 時對 Y8B 進行設置 (ON)，在 X9 爲 ON 時對 Y8B 進行重定 (OFF) 的程式。

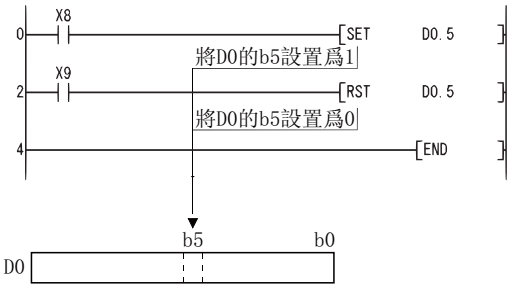
[梯形圖模式]

[列表模式]

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

(2) 以下爲在 X8 爲 ON 時將 D0 的位 5(b5) 設置爲 1，在 X9 爲 ON 時將 D0 的位 5(b5) 設置爲 0 的程式。

[梯形圖模式]



[列表模式]

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

備 註

1. SET 指令的基本步數如下所示：
   · 使用內部軟元件或文件寄存器 (R0 ～ R32767) 時 :1
   · 使用直接訪問輸出 (DY) 或 SFC 程式軟元件 (BL) 時 :2
   · 使用連號訪問方式文件寄存器時
   (對於通用型 QCPU、LCPU) :2
   (對於基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU) :3
   · 使用除上述以外的其他軟元件時 : 3
2. 當 X 被當作軟元件使用時，應使用實際輸入中未使用的軟元件號。如果使用了與實際輸入軟元件相同的號，則實際輸入的資料將被通過 SET 指令指定的輸入 X 所覆蓋。

5.3.6 RST

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：重定的位軟元件編號 / 字軟元件的位指定 (位)。
重定的位軟元件編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				DY
Ⓓ	○							--	○

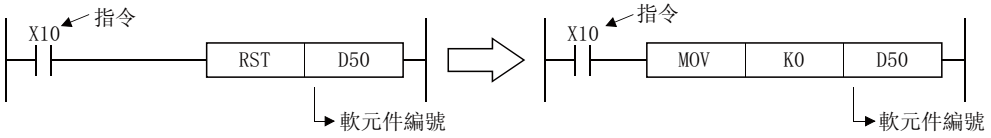
功 能

(1) 當執行命令爲 ON 時，指定軟元件的狀態如下所示：

軟元件	軟元件狀態
位軟元件	將線圈及觸點變爲 OFF。
計時器、計數器	將當前值設置爲 0，並將線圈及觸點變爲 OFF。
字軟元件的位指定時	將指定位設置爲 0。
除計時器、計數器以外的字軟元件	將內容設置爲 0。

(2) 當執行指令爲 OFF 時，軟元件的狀態不發生變化。

(3) 通過 RST 指令指定的字軟元件的功能等同於下列梯形圖：



出 錯

(1) RST 指令中無運算出錯。

備 註

RST 指令的基本步數情況如下所示：

a)對於位處理

- 內部軟元件（位軟元件或字軟元件的位指定） :1
- 直接訪問輸出 :2
- 計時器、計數器 :4
- 使用連號訪問方式文件寄存器時
- （對於通用型 QCPU、LCPU) :2
  （對於基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU) :3
- 除上述之外 :3

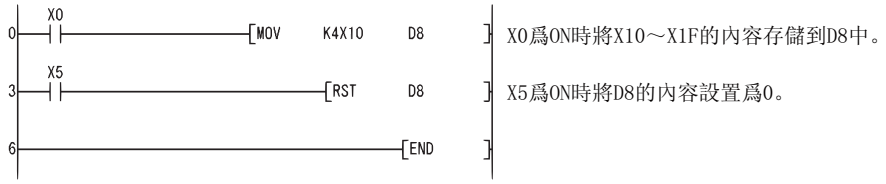
b)對於字處理

- 內部軟元件 :2
- 變址寄存器 :2
- 使用連號訪問方式文件寄存器時
- （對於通用型 QCPU、LCPU) :2
  （對於基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU) :3
- 除上述之外 :3

程式示例

(1) 以下爲將資料寄存器的值設置爲 0 的程式。

[梯形圖模式]

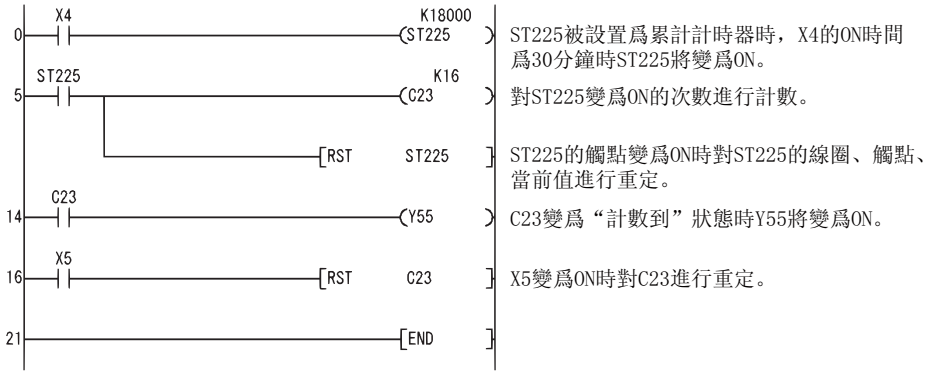


[列表模式]

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

(2) 以下爲對 100ms 累計計時器和計數器進行重定的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X4	
1	OUT	ST225	K18000
5	LD	ST225	
6	OUT	C23	K16
10	RST	ST225	
14	LD	C23	
15	OUT	Y55	
16	LD	X5	
17	RST	C23	
21	END		

5.3.7 SET F、RST F

Basic | High performance | Process | Redundant | Universal | LCPU

SET Ⓓ ：設置的報警器編號 (F 編號) (位)。

RST Ⓓ ：重定的報警器編號 (F 編號) (位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	○ (僅 F)		--						

功　能

SET

(1) 當執行指令變爲 ON 時，將Ⓓ中指定的報警器變爲 ON。

(2) 報警器 (F) 變爲 ON 的情況如下所示：

· “ USER ” LED 亮燈。*1

· 將變爲 ON 的報警器編號 (F 編號) 存儲到特殊寄存器 (SD64 ～ SD79) 中。

· SD63 的值加 1。

*1: 使用基本型 QCPU 時，“ ERR. ” LED 亮燈。

(3) SD63 的值爲 16(有 16 個報警器變爲 ON) 時，即使又有報警器變爲 ON，變爲 ON 的報警器編號也不會被存儲到 SD64 ～ SD79 中。

RST

(1) 執行指令變爲 ON 時，將Ⓓ中指定的報警器變爲 OFF。

(2) 對於已變爲 OFF 的報警器編號 (F 編號)，將其從特殊寄存器 (SD64 ～ SD79) 中刪除，且 SD63 的值被減 1。

備 註

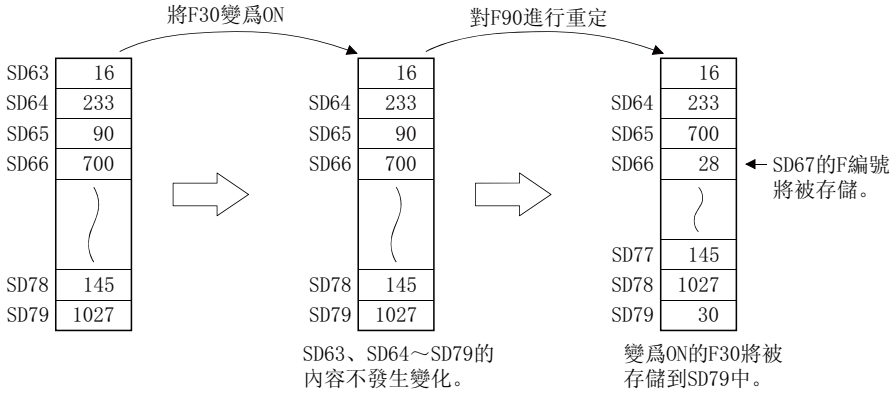
1. 關於報警器的詳細內容，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。
2. SET F□、RST F□指令的基本步數爲 2 步。

(3) 當 SD63 的值爲 16 時，通過 RST 指令將報警器號從 SD64 ～ SD79 中刪除。此外，如果未在 SD64 ～ SD79 中進行編號登錄的報警器變爲 ON 時，這些報警器號將被新建登錄。

如果 SD64 ～ SD79 的報警器號均變爲 OFF，在 CPU 模組前面的 LED 顯示或 “ USER ” LED 將會熄燈。*1

*1: 當使用基本型 QCPU 時，“ ERR. ” LED 將熄燈。

[當 SD63 爲 16 時的動作]



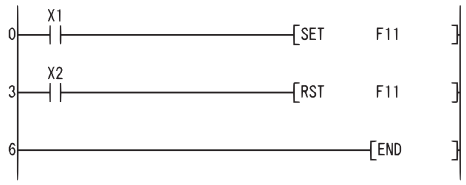
出　錯

(1) 在 SET F、RST F 指令中無運算出錯。

程式示例

(1) 以下爲 X1 變爲 ON 時，將報警器 F11 變爲 ON，並將 11 存儲到特殊寄存器 (SD64 ～ SD79) 中。此外，當 X2 變爲 ON 之後對報警器 F11 進行重定，並從特殊寄存器 (SD64 ～ SD79) 中刪除值 11。

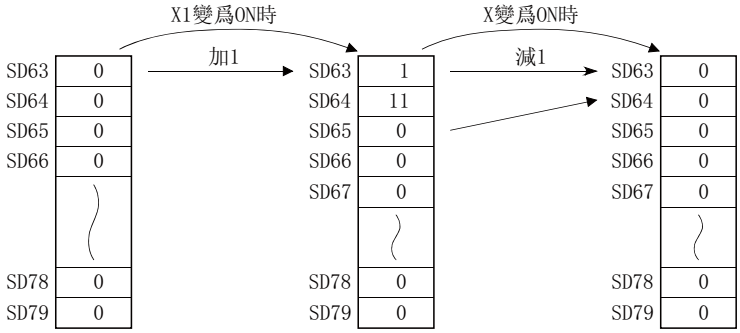
[梯形圖模式]



[列表模式]

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

[動作]



5.3.8 PLS、PLF

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：脈衝化的軟元件 (位)。

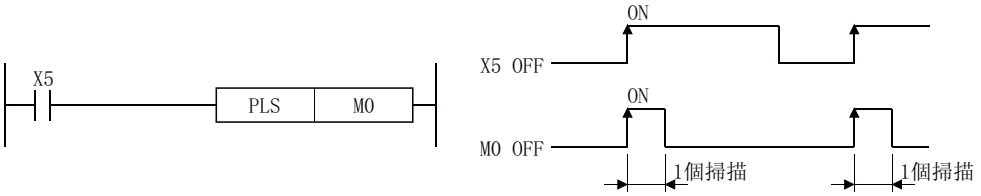
設置資料	內部軟元件		R,ZR	J□\□		U□\G□	Zn	常數	其他 DY
	位	字		位	字				
Ⓓ	○						--		○

功 能

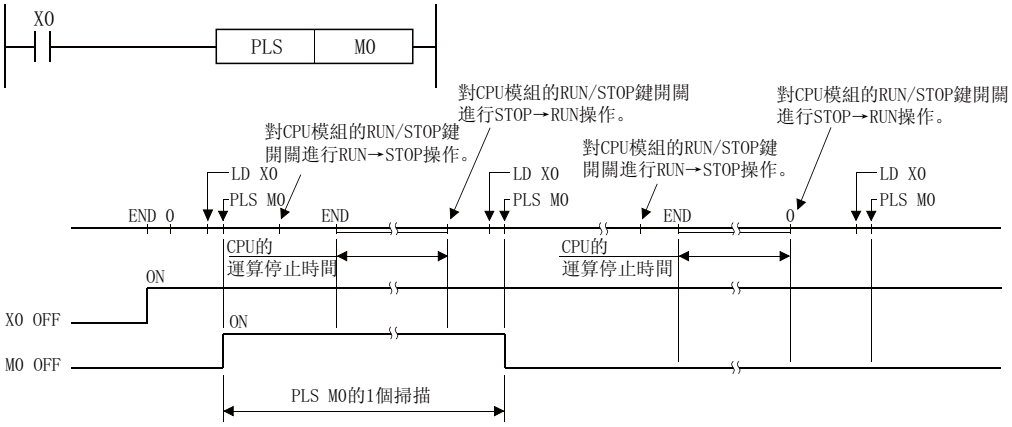
PLS

(1) 執行指令爲 OFF → ON 時將指定軟元件變爲 ON，在執行指令爲除 OFF → ON 以外 (ON → ON、ON → OFF、OFF → OFF) 時將指定軟元件變爲 OFF。

1 個掃描中的Ⓓ中指定的軟元件的 PLS 指令爲 1 個時，指定軟元件將 ON 1 個掃描。
關於在 1 個掃描中多次執行了同一軟元件的 PLS 指令時的動作，請參閱 111 頁 3.9 節。



(2) PLS 指令執行後如果進行了 RUN → STOP 操作，則即使再次置於 RUN，也不能執行 PLS 指令。

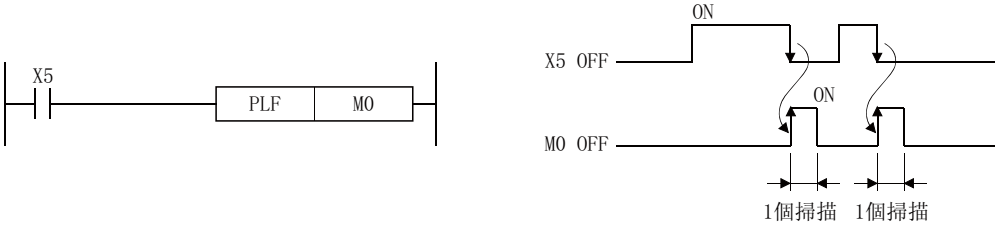


(3) 將鎖存繼電器 (L) 指定爲執行指令後，如果在鎖存繼電器爲 ON 的狀態下對電源執行了 OFF → ON 的操作，則執行指令將在第 1 個掃描中變爲 OFF → ON，PLS 指令將被執行，指定的軟元件將變爲 ON。
在電源 ON 後的第 1 個掃描中變爲 ON 的軟元件，通過下一個 PLS 指令變爲 OFF。

PLF

(1) 當執行指令爲 ON → OFF 時，PLF 指令將指定的軟元件變爲 ON，在除 ON → OFF 以外 (OFF → OFF、OFF → ON、ON → ON) 時，該指令將指定的軟元件變爲 OFF。

在 1 個掃描內只有 1 個Ⓓ中指定的軟元件的 PLF 指令時，指定的軟元件將 ON 1 個掃描周期。
關於在 1 個掃描中多次執行了同一個軟元件的 PLF 指令時的動作，請參閱 111 頁 3.9 節。



(2) PLF 指令執行後如果進行了 RUN → STOP 操作，則即使再次置於 RUN，也不能執行 PLF 指令。

要 點

如果對 PLS、PLF 指令通過 CJ 指令進行了跳轉，或未通過 CALL 指令對已執行的副程式進行調用，則Ⓓ中指定的軟元件有可能會 ON 1 個掃描以上，應加以注意。

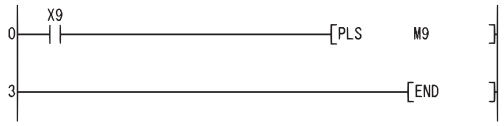
出 錯

(1) PLS、PLF 指令中無運算出錯。

程式示例

(1) 以下爲 X9 爲 ON 時執行 PLS 指令的程式。

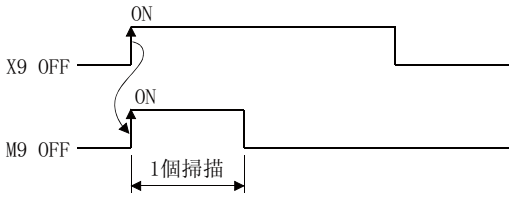
[梯形圖模式]



[列表模式]

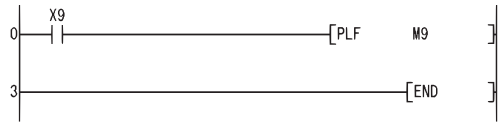
步	指令	軟元件
0	LD	X9
1	PLS	M9
3	END	

[時序圖]



(2) 以下爲 X9 爲 ON 時，執行 PLF 指令的程式。

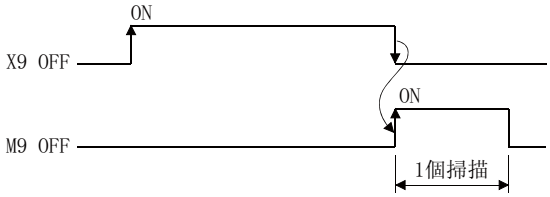
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X9
1	PLF	M9
3	END	

[時序圖]



5.3.9 FF

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：取反的軟元件編號 (位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他 DY
	位	字		位	字				
Ⓓ	○						--		○

功　能

(1) 執行指令爲 OFF → ON 時，對Ⓓ中指定的軟元件狀態進行取反。

軟元件	軟元件狀態	
	在 FF 執行之前	在 FF 執行之後
位軟元件	OFF	ON
	ON	OFF
字軟元件的位指定	0	1
	1	0

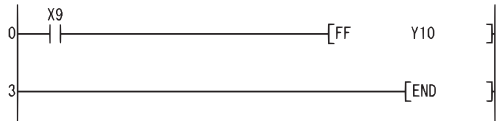
出　錯

(1) FF 指令中無運算出錯。

程式示例

(1) 以下爲 X9 變爲 ON 時，對 Y10 的輸出進行取反的程式。

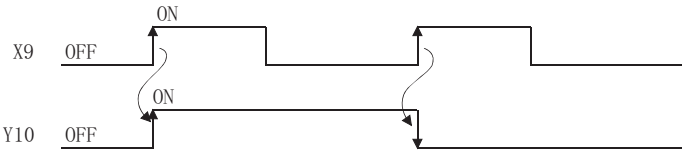
[梯形圖模式]



[列表模式]

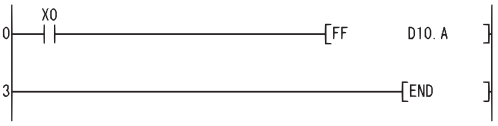
步	指令	軟元件
0	LD	X9
1	FF	Y10
3	END	

[時序圖]



(2) 以下爲 X0 變爲 ON 時，對 D10 的 b10(位 10) 進行取反的程式。

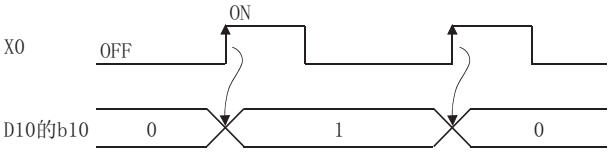
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0
1	FF	D10.A
3	END	

[時序圖]



5.3.10 DELTA、DELTAP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：脈衝化的位 (位)

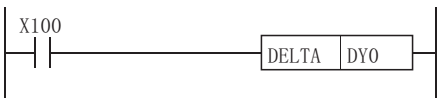
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				DY
Ⓓ	--								○

功　能

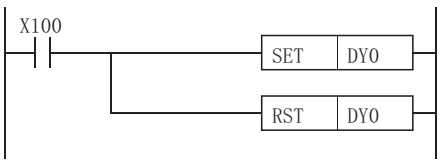
(1) 對Ⓓ中指定的直接訪問輸出 (DY) 進行脈衝輸出。

通過 DELTA　DY0 進行了指定時，其動作與下圖所示的使用了 SET/RST 指令的梯形圖的動作相同。

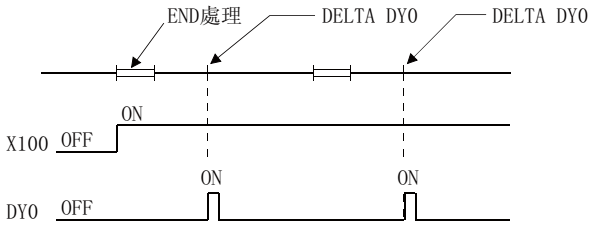
[使用了 DELTA 指令的梯形圖]



[使用了 SET/RST 指令的梯形圖]



[動作]



(2) DELTA(P) 指令被用作至智慧功能模組的啓動執行指令。

出　錯

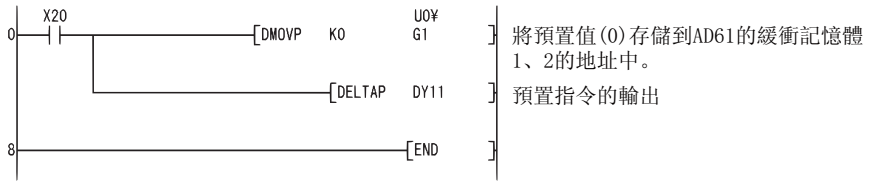
(1) 在以下情況下將變爲運算出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	指定的直接訪問輸出編號超出了 CPU 模組的輸出範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，對安裝在主基板的插槽 0 中的 AD61 的 CH1 進行預置的程式。

[梯形圖模式]



[列表模式]

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

5.4 移位元指令

5.4.1 SFT、SFTP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：移位的軟元件的起始編號（位）

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				DY
Ⓓ	○（除 T、C 以外）						---		○

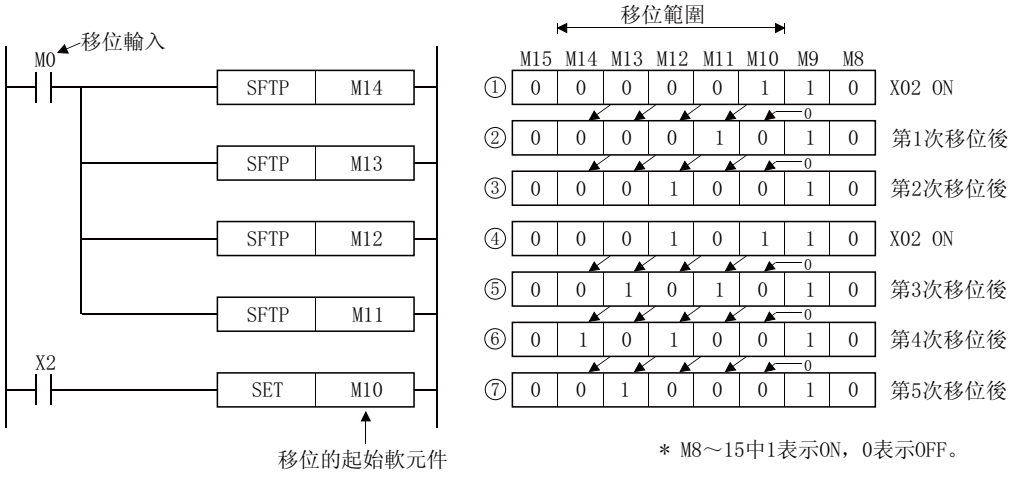
功 能

(1) 使用位軟元件時

(a) 對於在Ⓓ中指定的軟元件，將其前一個號的軟元件的 ON/OFF 狀態移位元到在Ⓓ中指定的軟元件中，並且將前一個號的軟元件置於 OFF。
例如，通過 SFT 指令指定了 M11 時，在 SFT 指令被執行時將 M10 的 ON/OFF 狀態移位元至 M11 中，並將 M10 置於 OFF。

(b) 應通過 SET 指令將要移位元的起始軟元件置於 ON。

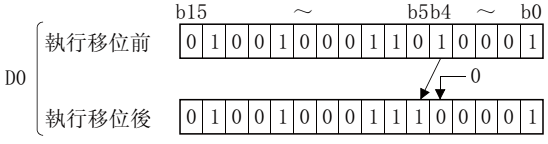
(c) 當 SFT 和 SFTP 被連續使用時，應將程式編制爲從較大號的軟元件處開始執行程式。



＊ M8～15中1表示ON，0表示OFF。

(2) 使用字軟元件的位指定時

(a) 對於在Ⓓ中指定的軟元件，將其前一個號的位元的 1/0 狀態移位元到在Ⓓ中指定的位中，並且將前一個號的位置於 0。
例如，通過 SFT 指令指定了 D0.5[D0 的位 5(b5)] 時，SFT 指令執行時將 D0 的 b4 的 1/0 移位到 b5 中，並將 b4 置於 0。



出　錯

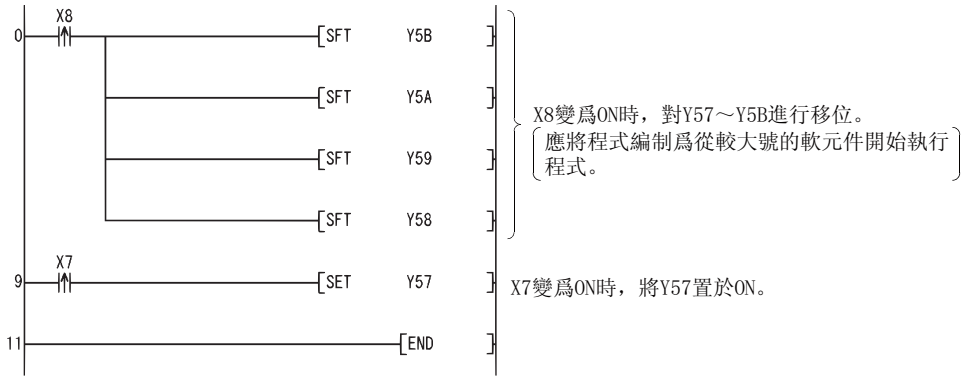
(1) 在以下情況下將變爲運算出錯狀態，出錯標誌 (SM0) 將變爲 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	指定的軟元件超出了相應軟元件範圍時	○	○	○	○	○	○

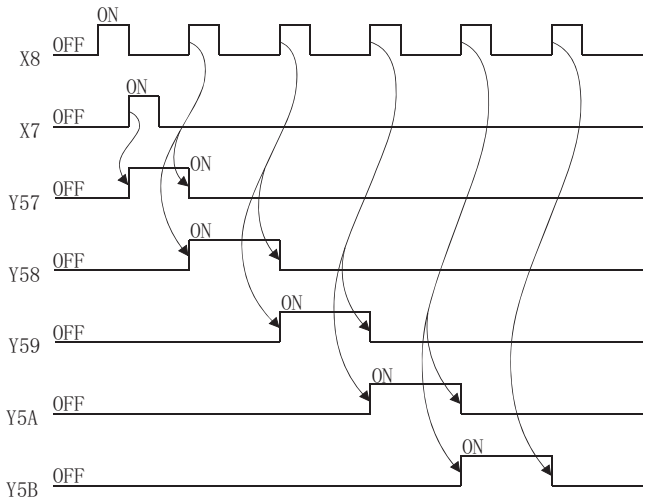
程式示例

(1) 以下爲 X8 變爲 ON 時，對 Y57 ～ Y5B 進行移位元的程式。

[梯形圖模式]



[時序圖]



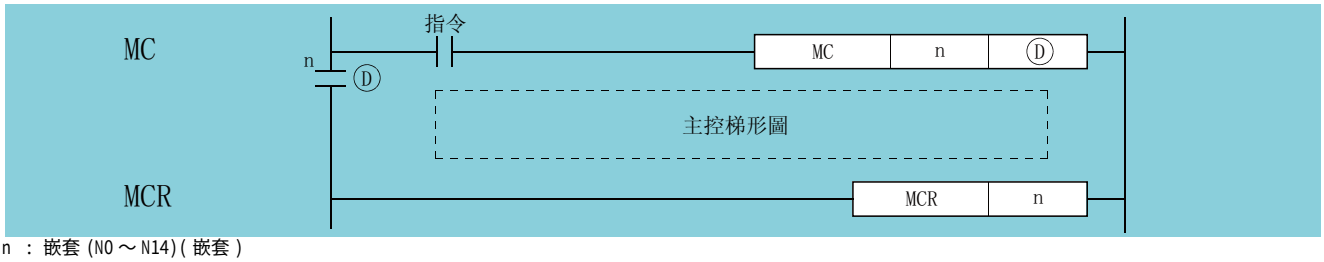
[列表模式]

步	指令	軟元件
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 MC、MCR

Basic | High performance | Process | Redundant | Universal | LCPU



n ： 嵌套 (N0 ～ N14)(嵌套)

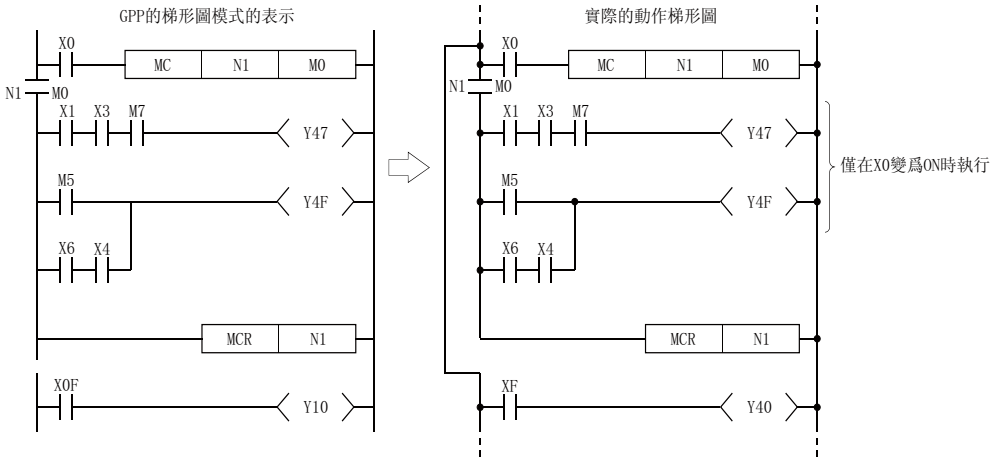
Ⓓ： 使其變爲 ON 的軟元件編號 (位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他	
	位	字		字位	字				N	DY
n	---						---		○	---
Ⓓ	○						---		---	○

功 能

主控指令是用於創建通過梯形圖公共母線的開閉以進行高效的梯形圖切換的順控程式的指令。

使用了主控的梯形圖如下所示：



備 註

在編程工具的寫入模式下進行編程時，無需在縱母線上輸入觸點。
在創建了梯形圖之後，執行 “ 轉換 ” 操作，置於讀取模式時將會自動顯示。

MC

(1) 主控開始時，如果 MC 指令的執行指令爲 ON，則在 MC 指令與 MCR 指令之間的運算結果如指令 (上述梯形圖) 所示。
如果 MC 的執行指令爲 OFF，則 MC 指令與 MCR 指令之間的運算結果如下所示：

軟元件	軟元件狀態
高速計時器 低速計時器	計數器值變爲 0，將線圈和觸點均置於 OFF。
高速累計計時器 低速累計計時器 計數器	線圈變爲 OFF 狀態，但計數值及觸點均保持爲當前狀態。
在 OUT 指令中的軟元件	全部置於 OFF。
SET、RST、 SFT 指令中 基本、應用 } 指令中的軟元件	保持爲當前狀態。

(2) 當 MC 指令爲 OFF 時，MC 指令與 MCR 指令之間的指令仍將被執行，因此掃描時間將不會縮短。

要 點

如果在使用了主控的梯形圖中存在有不需要觸點指令的指令 (FOR ～ NEXT、EI、DI 指令等)，則與 MC 指令的執行指令無關，CPU 模組將執行這些指令。

(3) 通過改變ⓓ中指定的軟元件，MC 指令能夠任意次使用同一個嵌套 (N) 號。

(4) MC 指令爲 ON 時，將ⓓ中指定的軟元件線圈置於 ON。
此外，在 OUT 指令等中使用同一軟元件時，將會變爲雙線圈，因此不要在其他指令中使用ⓓ中指定的軟元件。

MCR

(1) 該指令爲主控的解除指令，表示主控範圍的結束。

(2) 在 MCR 指令之前不附加觸點指令。

(3) 將具有相同嵌套號的 MC 指令與 MCR 指令配套使用。
但是，在 MCR 指令集中在 1 個位置處的嵌套結構的情況下，通過最小號的 1 個嵌套 (N) 號即可使所有的主控均結束。
(參閱程式示例中的 “ 嵌套結構時的注意事項 ”)

出　錯

(1) 在 MC 及 MCR 指令中無運算出錯。

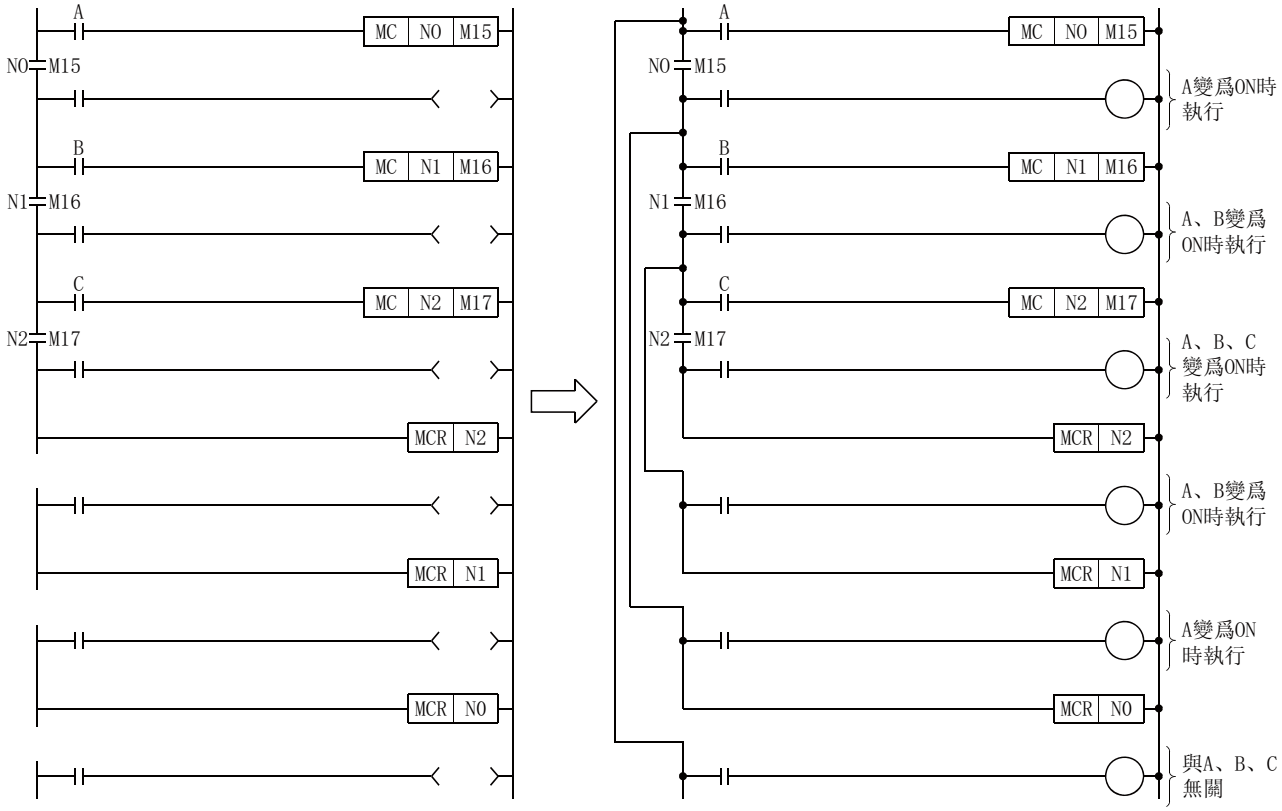
程式示例

主控指令可在嵌套結構中使用。通過嵌套 (N) 把各個主控區分開來。嵌套的最大使用範圍爲 N0 ～ N14。

通過使用嵌套結構，可以創建對程式的執行條件進行逐項限制的梯形圖。

使用了嵌套結構的梯形圖如下所示：

[GPP 梯形圖模式下顯示的梯形圖]　　[實際的動作梯形圖]



嵌套結構時的注意事項

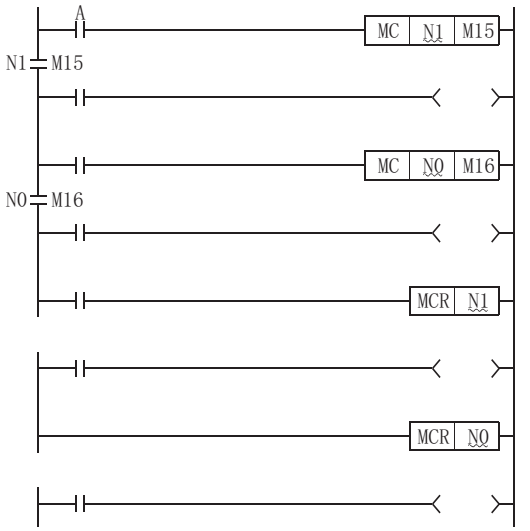
(1) 最多可使用 15 個嵌套 (N0 ～ N14)。

使用嵌套時，在 MC 指令中是按嵌套 (N) 號的從小到大的順序使用的，而在 MCR 指令中是按從大到小的順序使用的。

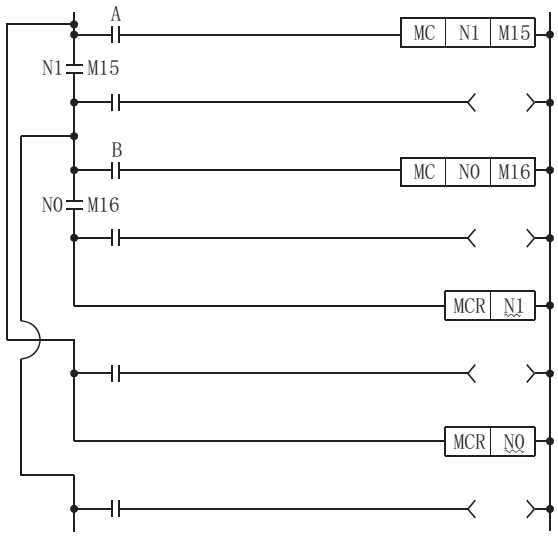
如果該順序逆向，將不能成爲嵌套結構，因此 CPU 模組也不能進行正常運算。

例如，如果在 MC 指令中將嵌套按 N1 → N0 的順序進行了指定，且在 MCR 指令中也按 N1 → N0 的順序進行了指定，則縱母線將會交叉。因此不能構成正常的主控梯形圖。

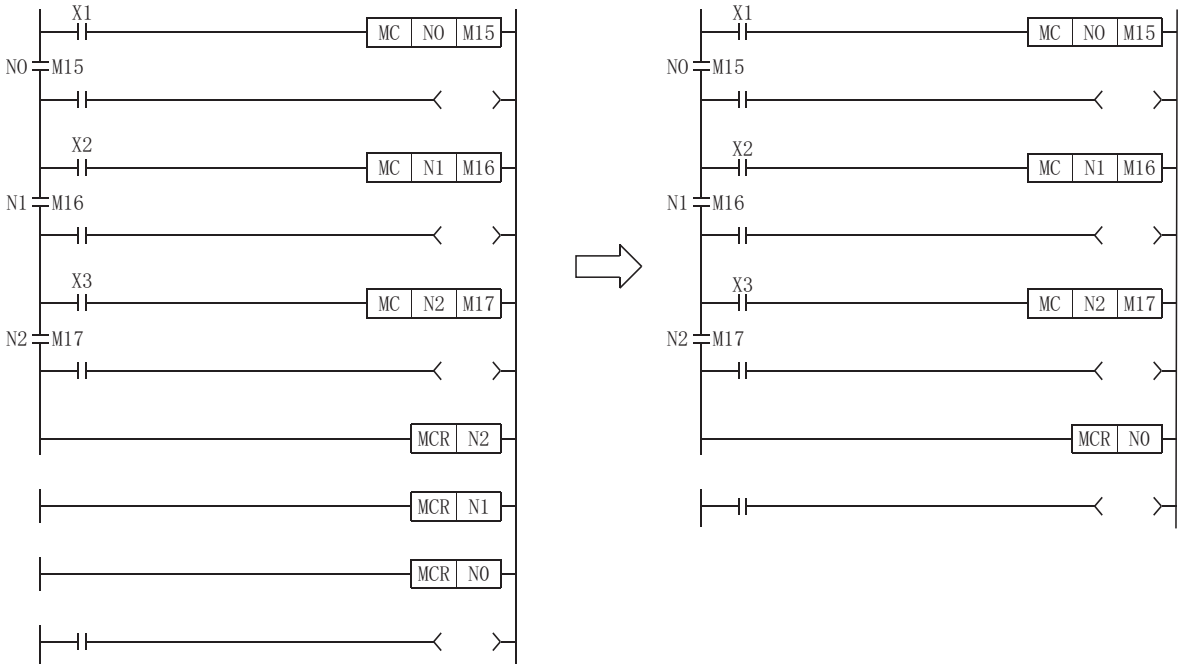
[GPP 梯形圖模式下顯示的梯形圖]



[實際的動作梯形圖]



(2) 在 MCR 指令集中在一個位置的嵌套結構的情況下，通過最小號的 1 個嵌套 (N) 號即可使所有的主控均結束。



5.6 結束指令

5.6.1 FEND

Basic | High performance | Process | Redundant | Universal | LCPU

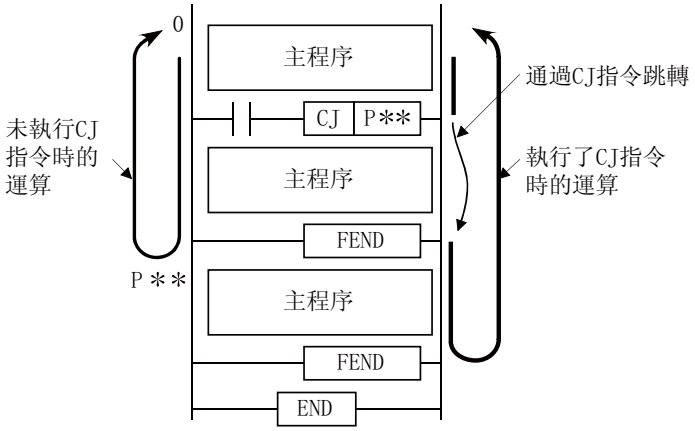
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功 能

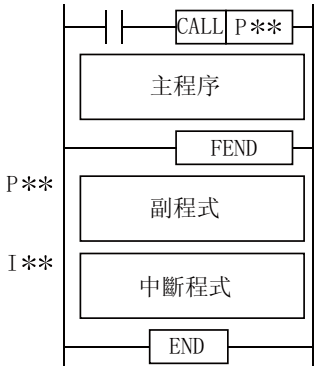
(1) 當通過 CJ 指令等被對順控程式運算進行分支時，可使用 FEND 指令將主程序從副程式或中斷程式中分離出來。

(2) 執行 FEND 指令時，將結束 CPU 模組正在執行的程式。

(3) FEND 指令之後的順控程式也可通過編程工具進行梯形圖顯示。
(編程工具持續顯示梯形圖直至 END 指令爲止。)



(a) 使用CJ指令時



(b) 存在有副程式、中斷程式時

出 錯

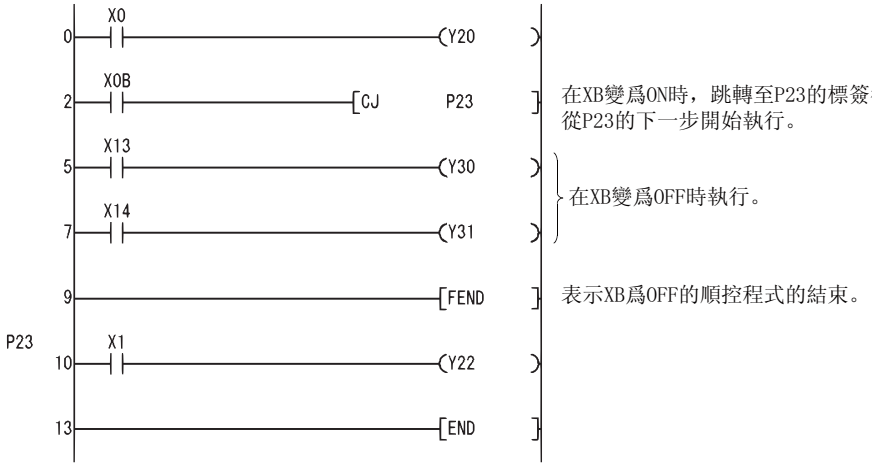
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4200	在執行 FOR 指令後，執行 NEXT 指令之前執行了 FEND 指令時。	○	○	○	○	○	○
4211	在執行 CALL/FCALL/ECALL/EFCALL 指令後，執行 RET 指令之前執行了 FEND 指令時。	○	○	○	○	○	○
4221	在中斷程式的執行過程中，在執行 IRET 指令之前執行了 FEND 指令時。	○	○	○	○	○	○
4230	在 CHKCIR ～ CHKEND 指令之間執行了 FEND 指令時。	○	○	○	○	○	○
4231	在 IX ～ IXEND 指令之間執行了 FEND 指令時。	○	○	○	○	○	○

程式示例

(1) 下列爲使用了 CJ 指令的程式。

[梯形圖模式]

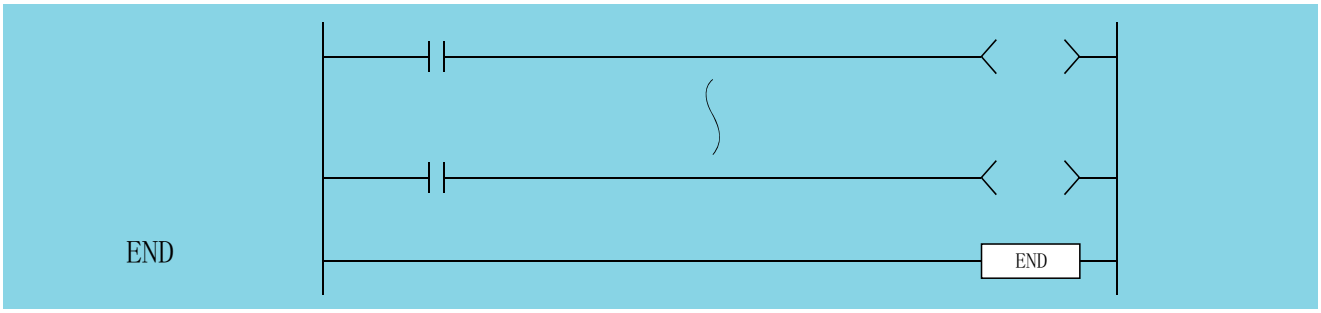


[列表模式]

步	指令	軟元件
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 END

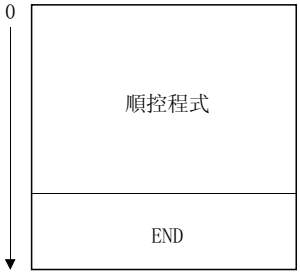
Basic | High performance | Process | Redundant | Universal | LCPU



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功　能

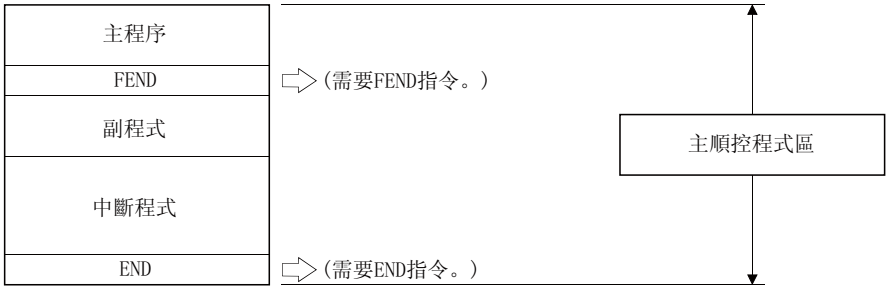
(1) 表示包括主程序、副程式和中斷程式的程式的結束。
如果執行了 END 指令，CPU 模組正在執行的程式將被結束。



(2) 在主順控程式的執行過程中不能使用 END 指令。
如果在程式執行過程中需要進行 END 處理，應使用 FEND 指令。

(3) 在編程工具的梯形圖模式下進行編程時，無需輸入 END 指令。

(4) 在存在有主程序、副程式和中斷程式的情況下， END 和 FEND 指令的使用分類如下所示：



出　錯

(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4200	在執行 FOR 指令後，執行 NEXT 指令之前執行了 END 指令時。	○	○	○	○	○	○
4211	在執行 CALL/FCALL/ECALL/EFCALL 指令後，執行 RET 指令之前執行了 END 指令時。	○	○	○	○	○	○
4221	在中斷程式的執行過程中，在執行 IRET 指令之前執行了 END 指令時。	○	○	○	○	○	○
4230	在 CHKCIR ～ CHKEND 指令之間執行了 END 指令時。	○	○	○	○	○	○
4231	在 IX ～ IXEND 指令之間執行了 END 指令時。	○	○	○	○	○	○

5.7 其他指令

5.7.1 STOP

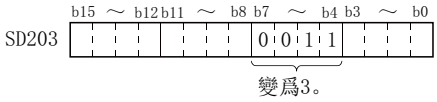
Basic | High performance | Process | Redundant | Universal | LCPU

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功 能

(1) 當執行指令爲 ON 時，輸出 Y 被重定且 CPU 模組的運算被停止。
(與開關被置於 STOP 側時相同。)

(2) 執行 STOP 指令時，特殊寄存器 SD203 的 b4 ～ b7 的值將變爲 3。



(3) 在執行 STOP 指令之後若要重新開始 CPU 模組的運算，則應對開關進行 RUN → STOP 操作後，再次置於 RUN 位置。

出 錯

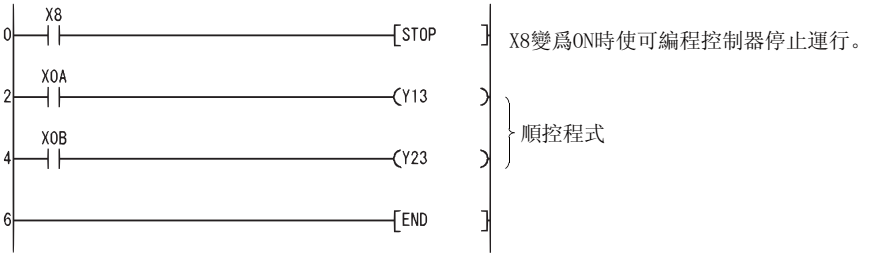
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4200	在執行 FOR 指令後，執行 NEXT 指令之前執行了 STOP 指令時。	○	○	○	○	○	○
4211	在執行 CALL/FCALL/EFCALL/XCALL 指令後，執行 RET 指令之前執行了 STOP 指令時。	---	---	---	---	○	○
4221	在中斷程式的執行過程中，在執行 IRET 指令之前執行了 STOP 指令時。	○	○	○	○	○	○
4223	在恒定周期執行型程式的執行過程中執行了 STOP 指令時。	○	○	○	○	○	○
4230	在 CHKCIR ～ CHKEND 指令之間執行了 STOP 指令時。	○	○	○	○	○	○
4231	在 IX ～ IXEND 指令之間執行了 STOP 指令時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X8 變爲 ON 時使 CPU 模組停止運行的程式。

[梯形圖模式

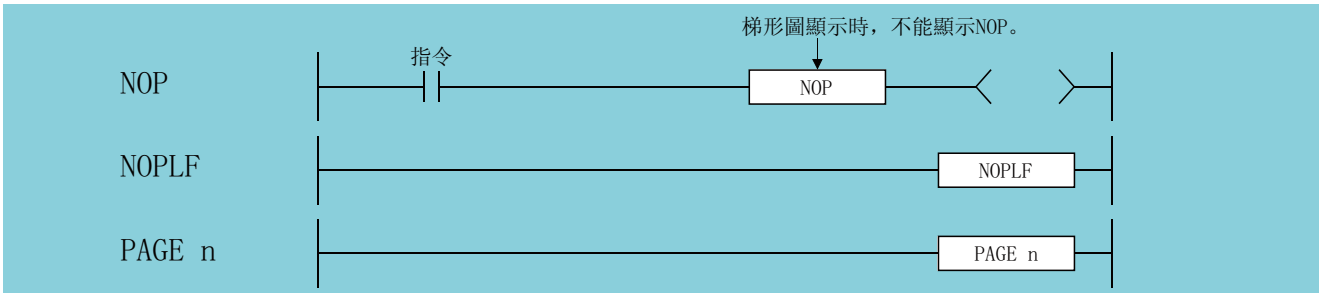


[列表模式]

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

5.7.2 NOP、NOPLF、PAGE n

Basic | High performance | Process | Redundant | Universal | LCPU



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功　能

NOP

(1) 該指令爲無處理指令，對至目前爲止的運算不會施加任何影響。

(2) NOP 指令適用於下列情況：

(a) 爲順控程式調試插入空間。

(b) 在不改變步數的狀況下刪除指令。(用 NOP 替換指令)

(c) 臨時性刪除指令。

NOPLF

(1) 該指令爲無處理指令，對至目前爲止的運算不會施加任何影響。

(2) 在通過編程工具進行列印時，可以使用 NOPLF 指令在任意位置進行換頁。

(a) 列印梯形圖時

- 在梯形圖塊之間如果有 NOPLF 指令，則將進行換頁列印。
- 在梯形圖塊中如果存在有 NOPLF 指令，則梯形圖將無法正常顯示。
  不要在梯形圖快中插入 NOPLF 指令。

(b) 列印指令列表時

- 在列印 NOPLF 指令之後將換頁列印。

(3) 關於通過編程工具進行列印輸出的詳細內容，請參閱所使用的編程工具的操作手冊。

PAGE n

(1) 該指令爲無處理指令，對至目前爲止的運算不會施加任何影響。

(2) 是在編程工具中也執行無處理的指令。

出　錯

(1) 在 NOP、NOPLF、PAGE 指令中無運算出錯。

程式示例

NOP

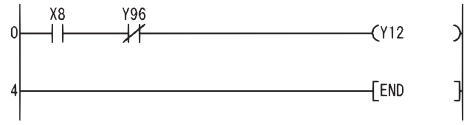
(1) 觸點短接 . . . 刪除 AND 或 AN1 指令。

[梯形圖模式]　　[列表模式]

變更前

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

變更後

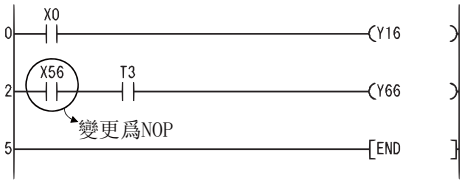


步	指令	軟元件
0	LD	X8
1	NOP	
2	ANI	Y96
3	OUT	Y12
4	END	

(2) 觸點短接 . . . 將 LD、LDI 變更爲 NOP(請注意將 LD 和 LDI 變更爲爲 NOP 時，梯形圖將被完全改變。)

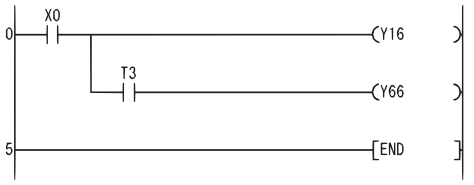
[梯形圖模式]　　[列表模式]

變更前



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

變更後



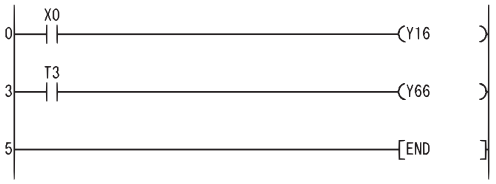
步	指令	軟元件
0	LD	X0
1	OUT	Y16
2	NOP	
3	AND	T3
4	OUT	Y66
5	END	

[梯形圖模式]　　[列表模式]

變更前

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

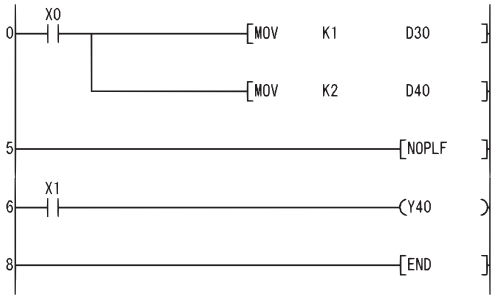
變更後



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

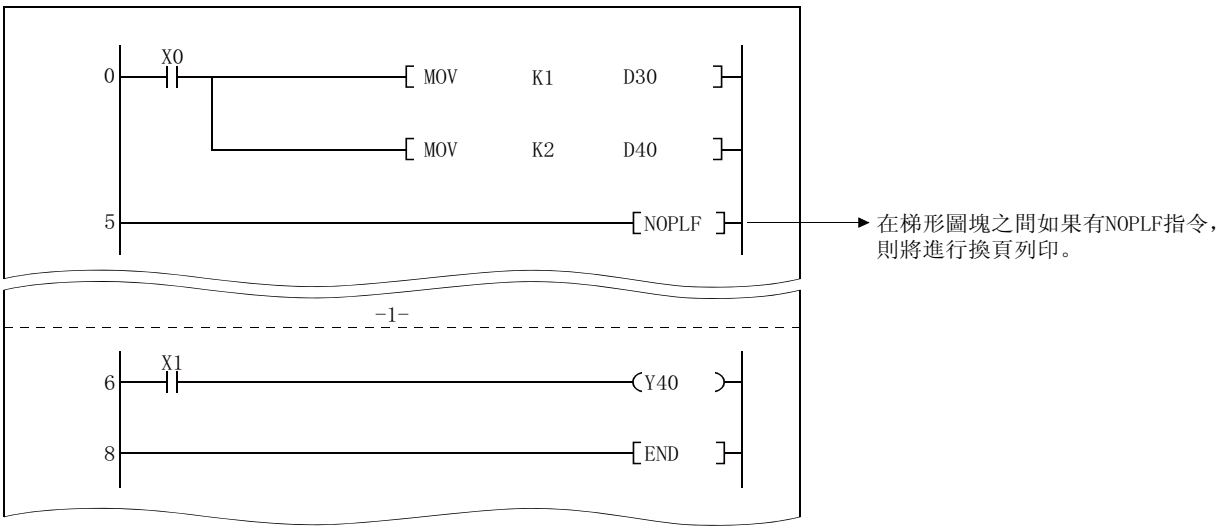
NOPLF

[梯形圖模式]　　[列表模式]



步	指令	軟元件	
0	LD	X0	
1	MOV	K1	D30
3	MOV	K2	D40
5	NOPLF		
6	LD	X1	
7	OUT	Y40	
8	END		

· 執行梯形圖列印時的情況如下所示。

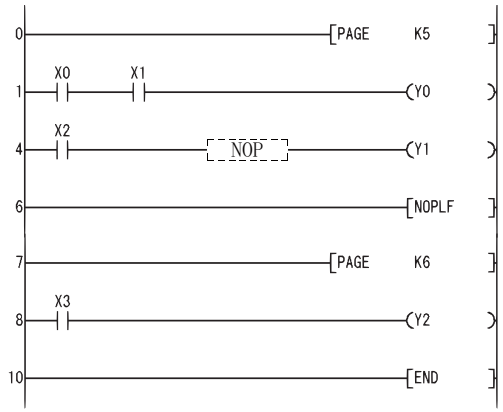


・執行列表列印時的情況如下所示。

```
0     LD      X0
1     MOV     K1      D30
3     MOV     K2      D40
5     NOPLF                  ──→ 在列印NOPLF指令之後將換頁列印。
                -1-
6     LD      X1
7     OUT     Y40
8     END
```

PAGE n

[梯形圖模式]



[列表模式]

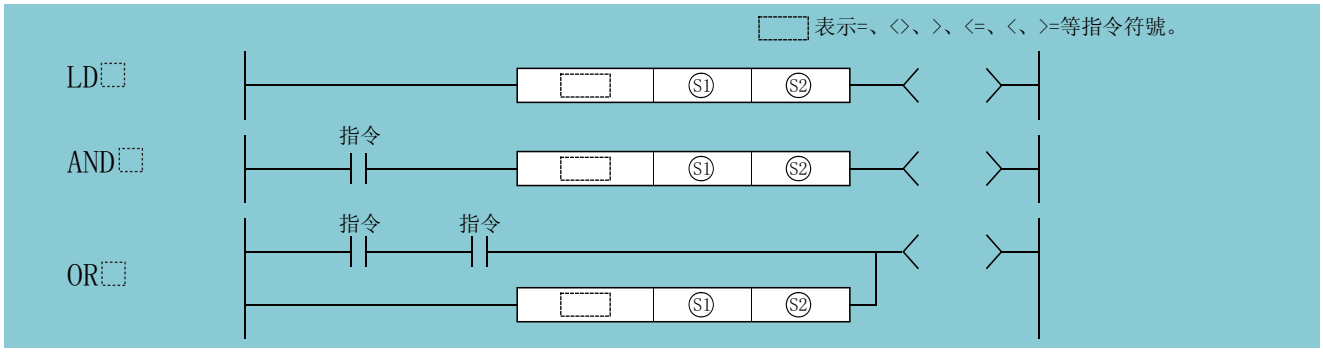
步	指令	軟元件
0	PAGE	K5
1	LD	X0
2	AND	X1
3	OUT	Y0
4	LD	X2
5	NOP	
6	OUT	Y1
7	NOPLF	
8	PAGE	K6
9	LD	X3
10	OUT	Y2
11	END	

第 6 章 基本指令

6.1 比較運算指令

6.1.1 =、<>、>、<=、<、>=

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ1、Ⓢ2 ： 比較資料或者存儲比較資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	○								---
Ⓢ2	○								---

功 能

(1) 將Ⓢ1中指定的軟元件的 BIN16 位元資料與Ⓢ2中指定的軟元件的 BIN16 位元資料使用 a 觸點進行比較運算。

(2) 各指令的比較運算結果如下所示：

□中的指令符號	條件	比較運算結果	□中的指令符號	條件	比較運算結果
=	Ⓢ1=Ⓢ2	導通狀態	=	Ⓢ1≠Ⓢ2	非導通狀態
< >	Ⓢ1≠Ⓢ2		< >	Ⓢ1=Ⓢ2	
>	Ⓢ1>Ⓢ2		>	Ⓢ1≤Ⓢ2	
<=	Ⓢ1≤Ⓢ2		<=	Ⓢ1>Ⓢ2	
<	Ⓢ1<Ⓢ2		<	Ⓢ1≥Ⓢ2	
>=	Ⓢ1≥Ⓢ2		>=	Ⓢ1<Ⓢ2	

(3) 在Ⓢ1、Ⓢ2中指定了 16 進制常數的情況下，如果指定了最高位 (b15) 爲 1 的數值 (8 ～ F)，則該值將被視爲負的 BIN 值進行比較運算。

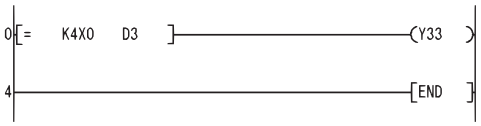
出 錯

(1) 在 =、< >、>、<=、<、>= 指令中無運算出錯。

程式示例

(1) 以下爲將 X0 ～ XF 的資料與 D3 的資料進行比較，X0 ～ XF 的資料與 D3 的資料一致時，將 Y33 變爲 ON 的程式。

[梯形圖模式]

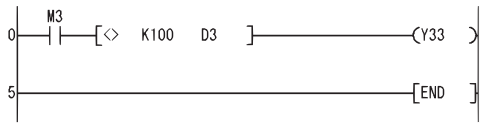


[列表模式]

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

(2) 以下爲將 BIN 值的 K100 與 D3 的資料進行比較，D3 的資料爲除 100 以外時置於導通狀態的程式。

[梯形圖模式]

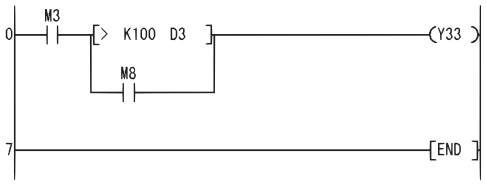


[列表模式]

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

(3) 以下爲將 BIN 值的 100 與 D3 的資料進行比較，D3 的資料小於 100 時置於導通狀態的程式。

[梯形圖模式]

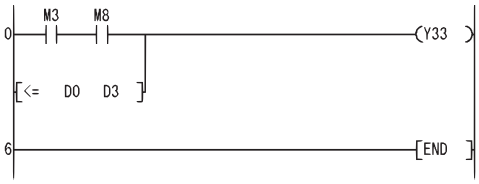


[列表模式]

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

(4) 以下爲將 D0 與 D3 的資料進行比較，(D0 的資料) ≤ (D3 的資料) 時置於導通狀態的程式。

[梯形圖模式]

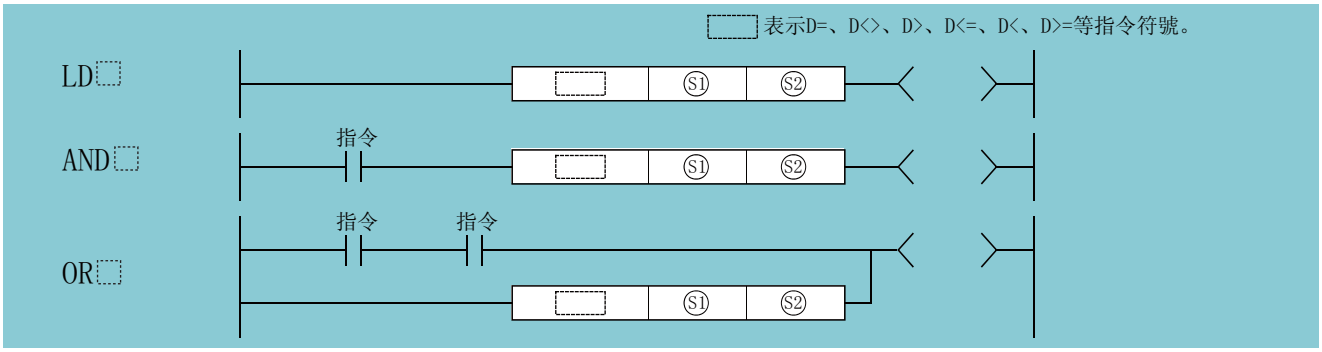


[列表模式]

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

6.1.2 D=、D<>、D>、D<=、D<、D>=

Basic | High performance | Process | Redundant | Universal | LCPU



S1、S2 ： 比較資料或者存儲比較資料的軟元件的起始編號 (BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
S1				○					---
S2				○					---

功　能

(1) 將Ⓢ1中指定的軟元件的 BIN32 位元資料與Ⓢ2中指定的軟元件的 BIN32 位元資料使用 a 觸點進行比較運算。

(2) 各指令的比較運算結果如下所示：

□中的指令符號	條件	比較運算結果	□中的指令符號	條件	比較運算結果
D=	Ⓢ1=Ⓢ2	導通狀態	D=	Ⓢ1≠Ⓢ2	非導通狀態
D< >	Ⓢ1≠Ⓢ2		D< >	Ⓢ1=Ⓢ2	
D>	Ⓢ1>Ⓢ2		D>	Ⓢ1≤Ⓢ2	
D<=	Ⓢ1≤Ⓢ2		D<=	Ⓢ1>Ⓢ2	
D<	Ⓢ1<Ⓢ2		D<	Ⓢ1≥Ⓢ2	
D>=	Ⓢ1≥Ⓢ2		D>=	Ⓢ1<Ⓢ2	

(3) 在Ⓢ1、Ⓢ2中指定了 16 進制常數的情況下，如果指定了最高位 (b31) 爲 1 的數值 (8 ～ F)，則該值將被視爲負的 BIN 值進行比較運算。

(4) 用於比較的資料應通過 32 位元指令 (DMOV 指令等) 進行指定。
如果通過 16 位元指令 (MOV 指令等) 進行了指定，則大小值的比較將不能正常執行。

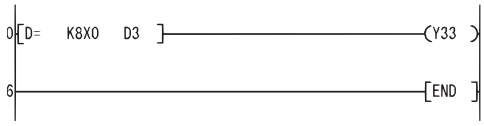
出　錯

(1) 在 D=、D< >、D>、D<=、D<、D>= 指令中無運算出錯。

程式示例

(1) 以下爲將 X0 ～ X1F 的資料與 D3、D4 的資料進行比較，X0 ～ X1F 的資料與 D3、D4 的資料一致時，將 Y33 變爲 ON 的程式。

[梯形圖模式]

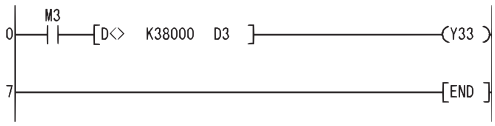


[列表模式]

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

(2) 以下爲將 BIN 值的 K38000 與 D3、D4 的資料進行比較，D3、D4 的資料爲除 38000 以外時置於導通狀態的程式。

[梯形圖模式]

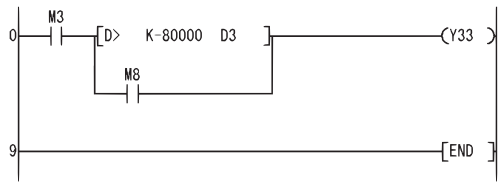


[列表模式]

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

(3) 以下爲將 BIN 值的 K-80000 與 D3、D4 的資料進行比較，D3、D4 的資料小於 -80000 時置於導通狀態的程式。

[梯形圖模式]

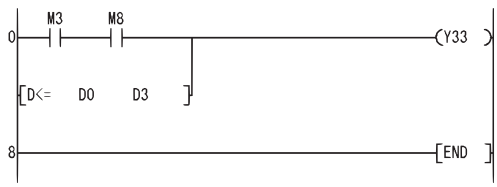


[列表模式]

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

(4) 以下爲將 D0、D1 與 D3、D4 的資料進行比較，(D0、D1 的資料) ≤ (D3、D4 的資料) 時置於導通狀態的程式。

[梯形圖模式]



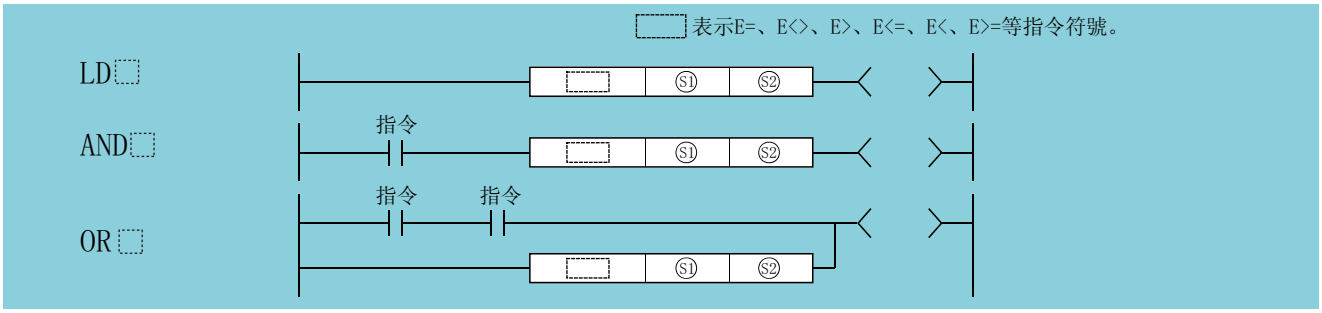
[列表模式]

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

6.1.3 E=、E<>、E>、E<=、E<、E>=

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後基本型 QCPU 中可以使用。



Ⓢ1、Ⓢ2 ： 比較資料或者存儲比較資料的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ1	---	○		---	○		○ *1	○	---
Ⓢ2	---	○		---	○		○ *1	○	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

(1) 將Ⓢ1中指定的軟元件的 32 位浮點數據與Ⓢ2中指定的軟元件的 32 位浮點數據使用 a 觸點進行比較運算。

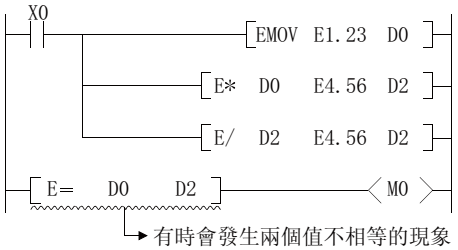
(2) 各指令的比較運算結果如下所示：

□中的指令符號	條件	比較運算結果	□中的指令符號	條件	比較運算結果
E=	Ⓢ1=Ⓢ2	導通狀態	E=	Ⓢ1≠Ⓢ2	非導通狀態
E< >	Ⓢ1≠Ⓢ2		E< >	Ⓢ1=Ⓢ2	
E>	Ⓢ1>Ⓢ2		E>	Ⓢ1≤Ⓢ2	
E<=	Ⓢ1≤Ⓢ2		E<=	Ⓢ1>Ⓢ2	
E<	Ⓢ1<Ⓢ2		E<	Ⓢ1≥Ⓢ2	
E>=	Ⓢ1≥Ⓢ2		E>=	Ⓢ1<Ⓢ2	

要 點

注意，使用了 E= 指令時，有時會發生由於誤差而導致兩個值不相等的現象。

例



出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

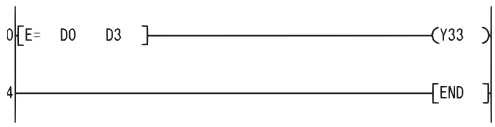
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定軟元件的內容爲 -0 時。*2	○	○	○	○	---	---
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定了 -0 也不會變爲運算出錯狀態。有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

(1) 以下爲將 D0、D1 的 32 位元浮點實數資料與 D3、D4 的 32 位元浮點實數資料進行比較的程式。

[梯形圖模式]

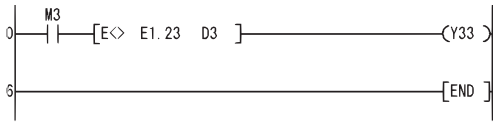


[列表模式]

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

(2) 以下爲將浮點實數 1.23 與 D3、D4 的 32 位元浮點實數資料進行比較的程式。

[梯形圖模式]

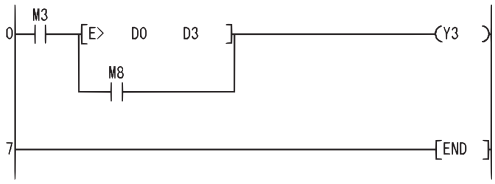


[列表模式]

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

(3) 以下爲將 D0、D1 的 32 位元浮點實數資料與 D3、D4 的 32 位元浮點實數資料進行比較的程式。

[梯形圖模式]

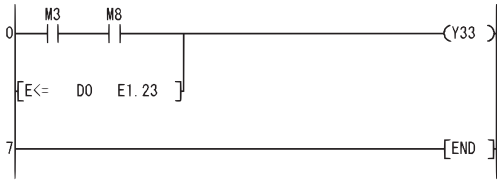


[列表模式]

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

(4) 以下爲將 D0、D1 的 32 位元浮點實數資料與浮點實數 1.23 進行比較的程式。

[梯形圖模式]

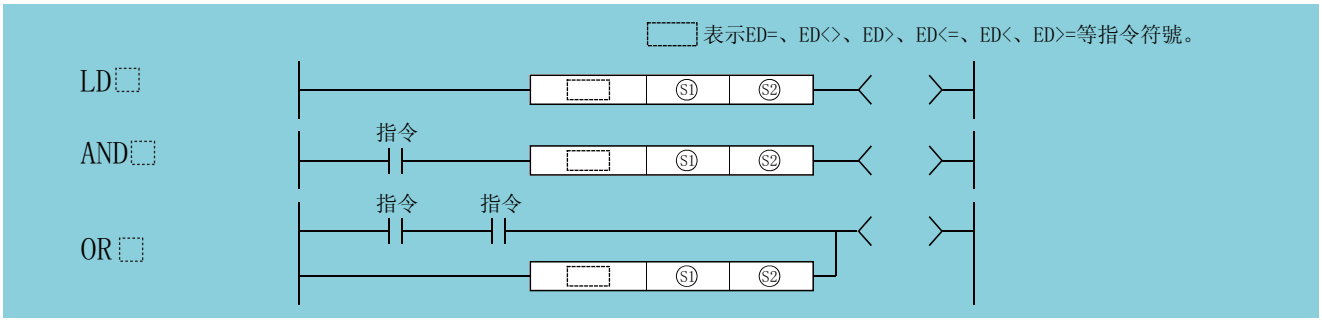


[列表模式]

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

6.1.4 ED=、ED<>、ED>、ED<=、ED<、ED>=

Basic High performance Process Redundant Universal LCPU



S1、S2 ：比較資料或者存儲比較資料的軟元件的起始編號（實數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
S1	---	○		---			---	○	---
S2	---	○		---			---	○	---

功　能

(1) 將S1中指定的軟元件的 64 位浮點數據與S2中指定的軟元件的 64 位浮點數據使用 a 觸點進行比較運算。

(2) 各指令的比較運算結果如下所示：

□中的指令符號	條件	比較運算結果	□中的指令符號	條件	比較運算結果
ED=	S1=S2	導通狀態	ED=	S1≠S2	非導通狀態
ED< >	S1≠S2		ED< >	S1=S2	
ED>	S1>S2		ED>	S1≤S2	
ED<=	S1≤S2		ED<=	S1>S2	
ED<	S1<S2		ED<	S1≥S2	
ED>=	S1≥S2		ED>=	S1<S2	

出　錯

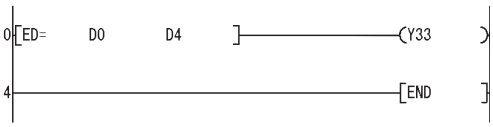
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍內時。 0、2^{-1022} ≤｜指定軟元件的內容｜ <2^{1024} 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 D0 ～ D3 的 64 位元浮點實數資料與 D4 ～ D7 的 64 位元浮點實數資料進行比較的程式。

[梯形圖模式]

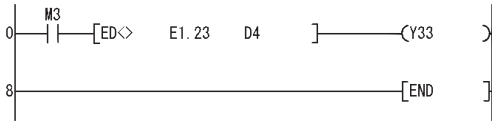


[列表模式]

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

(2) 以下爲將浮點實數 1.23 與 D4 ～ D7 的 64 位元浮點實數資料進行比較的程式。

[梯形圖模式]

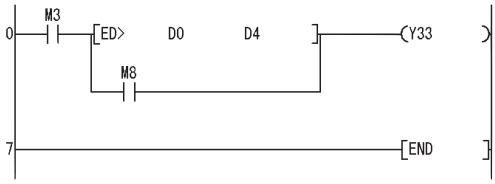


[列表模式]

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

(3) 以下爲將 D0 ～ D3 的 64 位元浮點實數資料與 D4 ～ D7 的 64 位元浮點實數資料進行比較的程式。

[梯形圖模式]

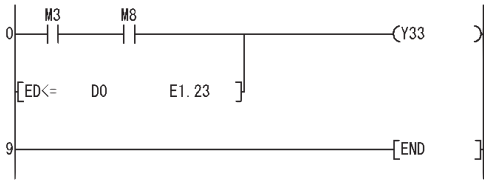


[列表模式]

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

(4) 以下爲將 D0 ～ D3 的 64 位元浮點實數資料與浮點實數 1.23 進行比較的程式。

[梯形圖模式]



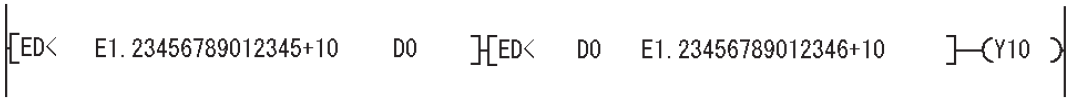
[列表模式]

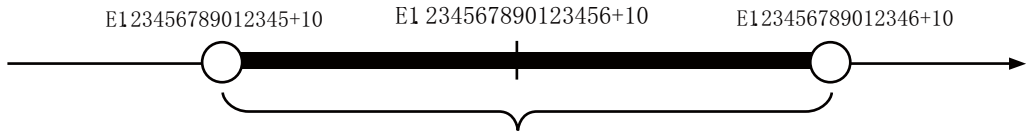
步	指令	軟元件	
0	LD	M3	
1	AND	M8	
2	ORED<=	D0	E1.23
8	OUT	Y33	
9	END		

注意事項

(1) 因爲可以通過編程工具輸入的實數位數最多爲 15 位，所以本項中所示的指令不能與有效位元數爲 16 位以上的實數進行比較。
當用本項中的指令判斷與有效位元數爲 16 位以上的實數的一致 / 不一致時，需要將其與待比較的實數的近似值進行大小比較以判斷其是否一致。

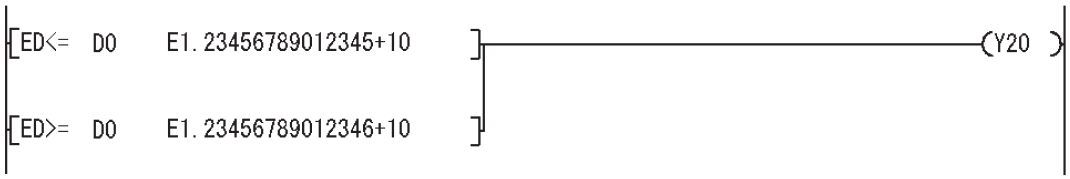
例 當判斷 E1.234567890123456+10(有效位數爲 16 位) 與雙精度浮點數據的一致性時。

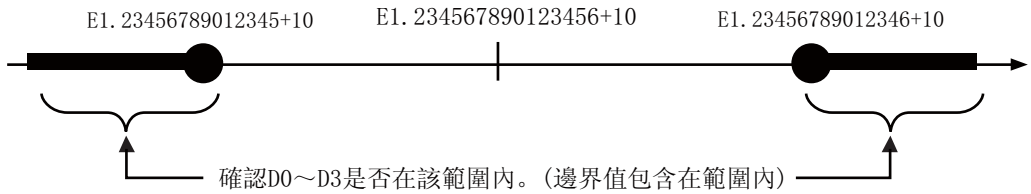




確認D0～D3是否在該範圍內。(邊界值不包含在範圍內)

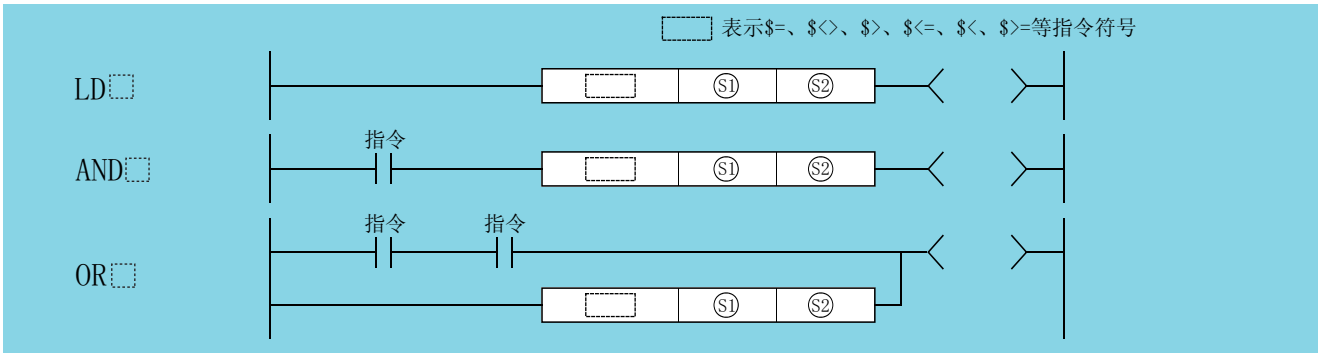
例 當判斷 E1.234567890123456+10(有效位數爲 16 位) 與雙精度浮動小數點資料的不一致性時。





6.1.5 $=、$<>、$>、$<=、$<、$>=

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ1、Ⓢ2 ：比較資料或者存儲比較資料的軟元件的起始編號（字串）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數$	其他
	位	字		位	字				
Ⓢ1	---	○		---				○	---
Ⓢ2	---	○		---				○	---

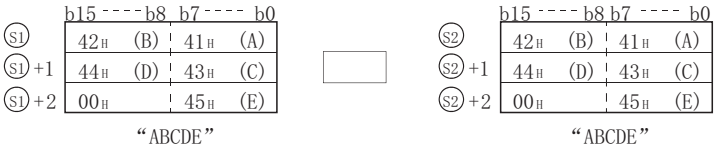
功　能

(1) 將Ⓢ1中指定的字串資料與Ⓢ2中指定的字串資料使用 a 觸點進行比較運算。

(2) 比較運算將字串的 ASCII 碼從字串的起始開始逐字進行比較。

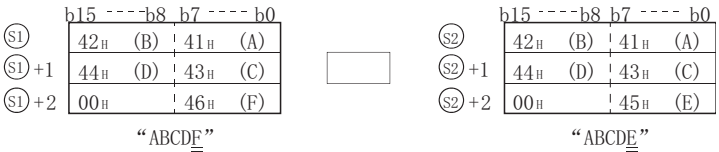
(3) Ⓢ1、Ⓢ2的字串包含從指定的軟元件號開始至存儲了“ 00_H ”的軟元件號爲止中的所有字元。

(a) 如果所有字串都一致，則比較結果一致。



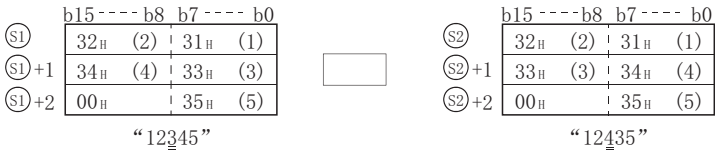
□中的指令符號	比較運算結果	□中的指令符號	比較運算結果
$=	導通狀態	$<=	導通狀態
$< >	非導通狀態	$<	非導通狀態
$>	非導通狀態	$>=	導通狀態

(b) 在字串不同的情況下，則帶較大字元代碼的字串算作大的一方。



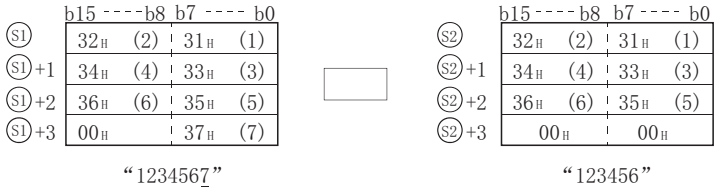
□中的指令符號	比較運算結果	□中的指令符號	比較運算結果
$=	非導通狀態	$<=	非導通狀態
$< >	導通狀態	$<	非導通狀態
$>	導通狀態	$>=	導通狀態

(c) 在字串不同的情況下，則由第一個不同字元代碼的大小決定字串的大小。



□中的指令符號	比較運算結果	□中的指令符號	比較運算結果
$=	非導通狀態	$<=	導通狀態
$< >	導通狀態	$<	導通狀態
$>	非導通狀態	$>=	非導通狀態

(4) 在(S1)與(S2)中的字串資料的長度不同的情況下，有較長字串的資料算作大的一方。



□中的指令符號	比較運算結果	□中的指令符號	比較運算結果
$=	非導通狀態	$<=	非導通狀態
$< >	導通狀態	$<	非導通狀態
$>	導通狀態	$>=	導通狀態

出　錯

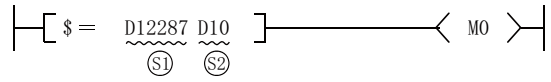
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	在(S1)、(S2)中指定的軟元件號的後面，相應軟元件的範圍中不存在 "00_H" 時。 (S1)、(S2)的字串超過了 16383 個字元時。	---	○	○	○	○	○

要 點

在字串資料比較運算指令中，在進行字串比較的同時也進行軟元件範圍檢查。
因此，即使在相應軟元件的範圍中不存在 "00H" 時，只要在軟元件範圍內檢測出字串不一致，則不變爲運算出錯狀態，對比較運算結果進行輸出。

例



(S1)的資料

D12287	"B"	"A"
W0	00H	"C"

(S2)的資料

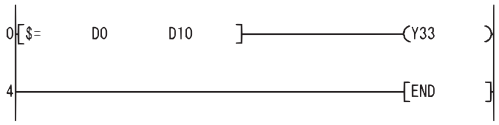
D10	"Z"	"A"
D11	00H	"C"

在上述(S1)和(S2)的資料的情況下，由於(S1)的第 2 個字元與(S2)的第 2 個字元不相同，因此(S1) ≠ (S2)，運算結果爲 " 非導通狀態 "。
此時雖然 (S1) 的軟元件範圍中不存在 "00_H" 代碼，但是由於檢測出不一致的軟元件爲 D12287(在軟元件範圍內)，因此不變爲運算出錯狀態。

程式示例

(1) 以下爲將存儲在 D0 後面的字串與存儲在 D10 後面的字元進行比較的程式。

[梯形圖模式]

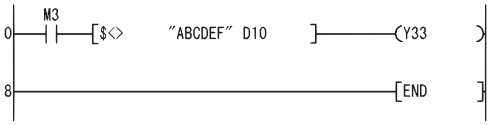


[列表模式]

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

(2) 以下爲將字串 “ ABCDEF ” 與存儲在 D10 後面的字串進行比較的程式。

[梯形圖模式]

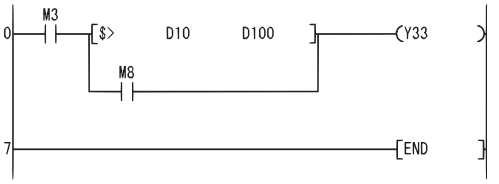


[列表模式]

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

(3) 以下爲將存儲在 D10 後面的字串與存儲在 D100 後面的字串進行比較的程式。

[梯形圖模式]

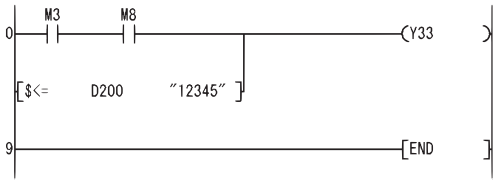


[列表模式]

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

(4) 以下爲將存儲在 D200 後面的字串與字串 “ 12345 ” 進行比較的程式。

[梯形圖模式]

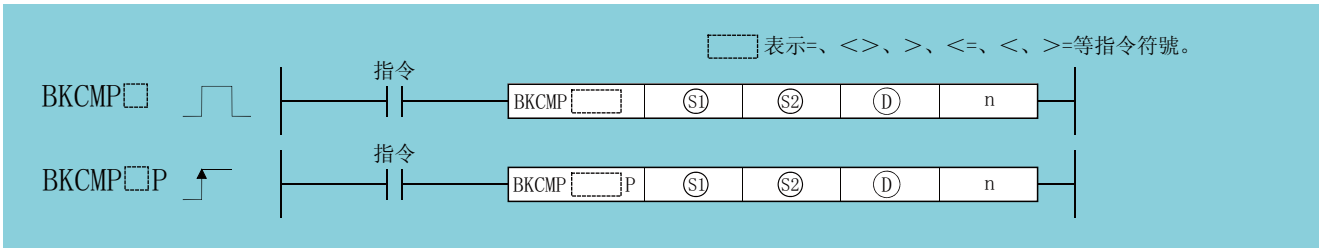


[列表模式]

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

6.1.6 BKCMP □、BKCMP □ P

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ1 ： 比較資料或者存儲比較資料的軟元件的起始編號 (BIN16 位)。

Ⓢ2 ： 存儲比較資料的軟元件的起始編號 (BIN16 位)。

Ⓓ ： 存儲運算結果的軟元件的起始編號 (位)。

n ： 比較的資料數 (BIN16 位元)。

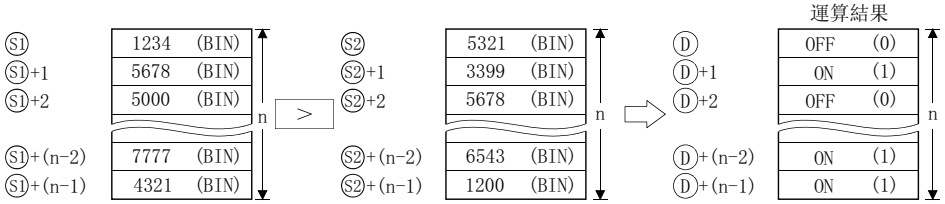
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	---	○		---				○	---
Ⓢ2	---	○		---				---	---
Ⓓ	○	○		---				---	---
n	○	○		○				○	---

功　能

(1) 將從Ⓢ1中指定的軟元件號開始的 n 點的 BIN16 位元資料與Ⓢ2中指定的軟元件號開始的 n 點的 BIN16 位元資料進行比較，並將運算結果存儲到Ⓓ中指定的軟元件的後面。

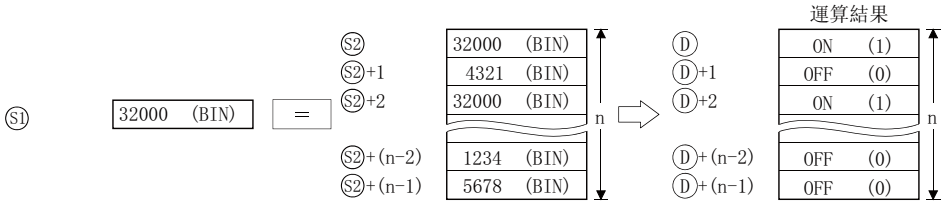
(a) 如果比較條件成立，則Ⓓ的相應軟元件將變爲 ON。

(b) 如果比較條件不成立，則Ⓓ的相應軟元件將變爲 OFF。



(2) 比較運算以 16 位元爲單位進行。

(3) Ⓢ1中可指定的常數範圍爲 -32768 ～ 32767(BIN16 位)。



(4) 各指令的比較運算結果如下所示：

指令符號	條件	比較運算結果	指令符號	條件	比較運算結果
BKCMP=	Ⓢ1=Ⓢ2	ON(1)	BKCMP=	Ⓢ1≠Ⓢ2	OFF(0)
BKCMP< >	Ⓢ1≠Ⓢ2		BKCMP< >	Ⓢ1=Ⓢ2	
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Ⓓ開始的 n 點中存儲的比較運算結果全部爲 ON(1)，則 SM704(塊比較信號) 將變爲 ON。

出　錯

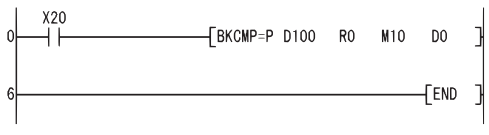
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓢ1、Ⓢ2、Ⓓ的軟元件開始的 n 點的範圍超出了相應軟元件範圍時。 從Ⓢ1開始第 n 點爲止的軟元件範圍與從Ⓓ開始的至第 n 點爲止的軟元件範圍相重復時。 從Ⓢ2開始至第 n 點爲止的軟元件範圍與從Ⓓ開始的至第 n 點爲止的軟元件範圍相重復時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D100 ～ D103 中存儲的資料值與 R0 ～ R3 中存儲的資料值進行比較運算，並將其結果存儲到 M10 後面的程式。

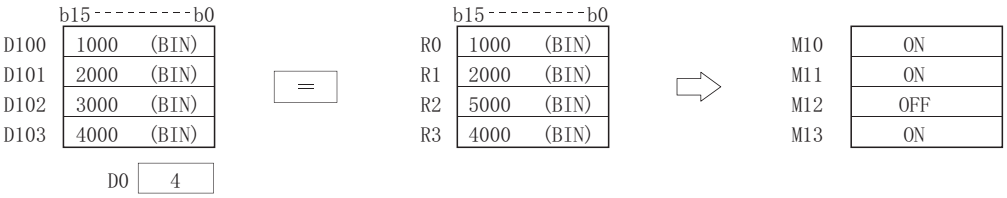
[梯形圖模式]



[列表模式]

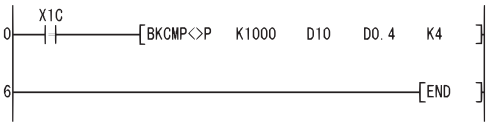
步	指令	軟元件
0	LD	X20
1	BKCMP=P	D100 R0 M10 D0
6	END	

[動作]



(2) 以下爲 X1C 變爲 ON 時，將常數 K1000 與 D10 ～ D13 中存儲的資料值進行比較運算，並將其結果存儲到 D0 的 b4 ～ b7 中的程式。

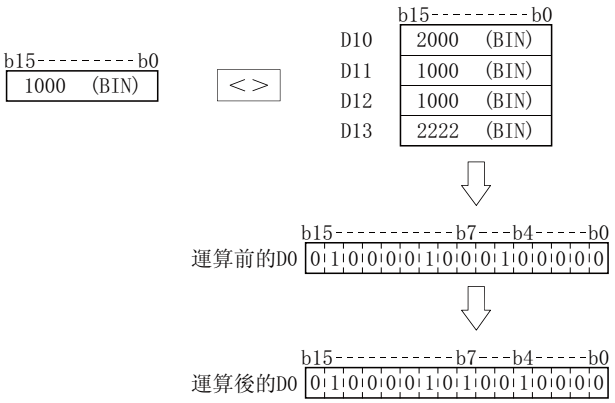
[梯形圖模式]



[列表模式]

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

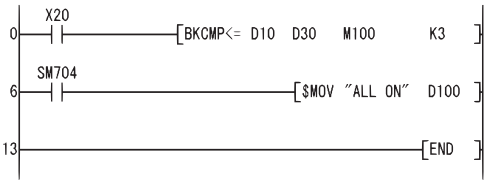
[動作]



(3) 以下爲 X20 變爲 ON 時，將 D10 ～ D12 中存儲的資料與 D30 ～ D32 中存儲的資料進行比較，並將其結果存儲到 M100 後面的程式。

M100 後面的軟元件全部變爲 1(ON) 時，將字串 “ ALL ON ” 傳送到 D100 後面。

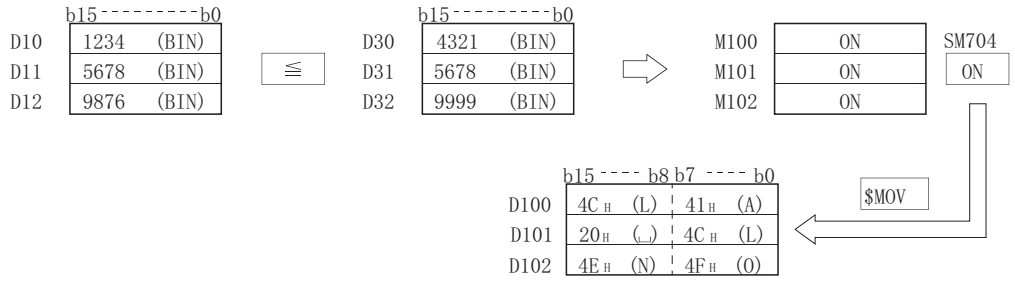
[梯形圖模式]



[列表模式]

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

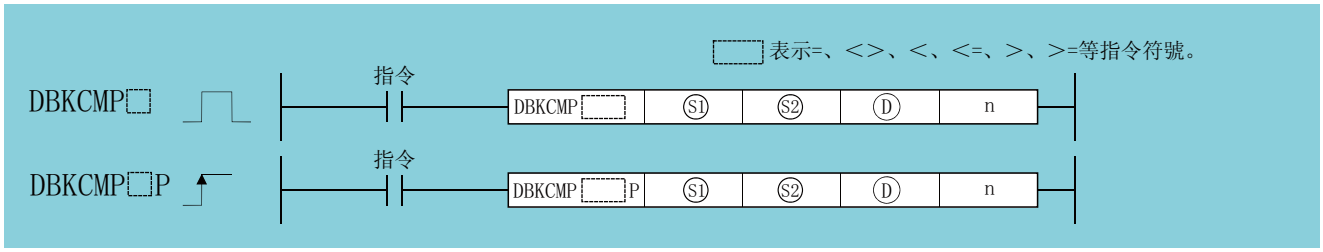
[動作]



6.1.7 DBKCMP □、DBKCMP □ P

Basic ✗ | High performance ✗ | Process ✗ | Redundant ✗ | Universal Ver. | LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。



Ⓢ1 ： 比較資料或者存儲比較資料的軟元件的起始編號 (BIN32 位)。

Ⓢ2 ： 存儲比較資料的軟元件的起始編號 (BIN32 位)。

Ⓓ ： 存儲運算結果的軟元件的起始編號 (位)。

n ： 比較的資料數 (BIN16 位元)。

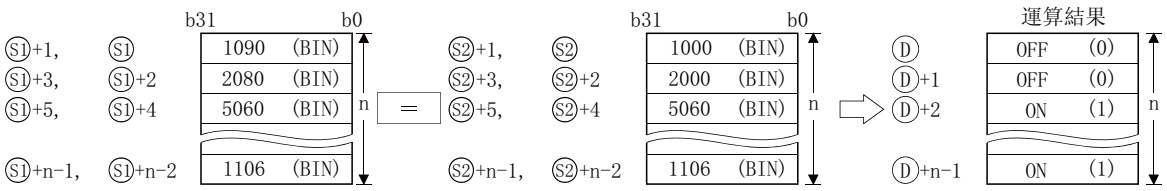
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	---	○	○	---				○	---
Ⓢ2	---	○	○	---				---	---
Ⓓ	○	---	○	---				---	---
n	---	○	○	○				○	---

功　能

(1) 將從Ⓢ1中指定的軟元件號開始的 n 點的 BIN32 位元資料或常數與Ⓢ2中指定的軟元件號開始的 n 點的 BIN32 位元資料進行比較，並將運算結果存儲到Ⓓ中指定的軟元件的後面。

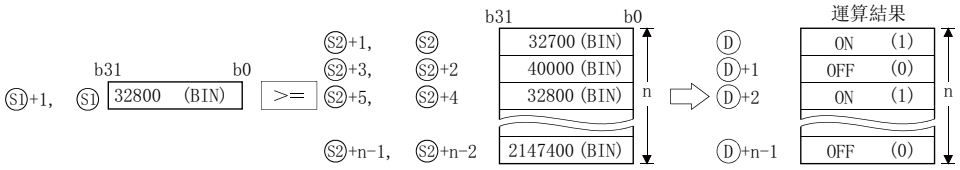
(a) 如果比較條件成立，則Ⓓ的相應軟元件將變爲 ON。

(b) 如果比較條件不成立，則Ⓓ的相應軟元件將變爲 OFF。



(2) 比較運算以 32 位元爲單位進行。

(3) Ⓢ1中可指定的常數範圍爲 -2147483648 ～ 2147483647(BIN32 位)。



(4) Ⓓ的指定應在從Ⓢ1開始的 n 點的軟元件範圍及從Ⓢ2開始的 n 點的軟元件範圍以外。

(5) 各指令的比較運算結果如下所示：

指令符號	條件	比較運算結果	指令符號	條件	比較運算結果
DBKCMP=	Ⓢ1=Ⓢ2	ON(1)	DBKCMP=	Ⓢ1≠Ⓢ2	OFF(0)
DBKCMP< >	Ⓢ1≠Ⓢ2		DBKCMP< >	Ⓢ1=Ⓢ2	
DBKCMP>	Ⓢ1>Ⓢ2		DBKCMP>	Ⓢ1≤Ⓢ2	
DBKCMP<=	Ⓢ1≤Ⓢ2		DBKCMP<=	Ⓢ1>Ⓢ2	
DBKCMP<	Ⓢ1<Ⓢ2		DBKCMP<	Ⓢ1≥Ⓢ2	
DBKCMP>=	Ⓢ1≥Ⓢ2		DBKCMP>=	Ⓢ1<Ⓢ2	

(6) 根據從ⓓ開始的 n 點中存儲的比較運算結果全部爲 ON(1) 時，或有一個爲 OFF(0) 時的條件，特殊繼電器的 ON/OFF 情況如下所示。

序號	編號	比較運算結果全部爲 ON(1) 時			比較運算結果中包含有 OFF(0) 時		
		初始化執行 / 掃描	中斷 (I45 除外) / 恒定周期執行	中斷 (I45)	初始化執行 / 掃描	中斷 (I45 除外) / 恒定周期執行	中斷 (I45)
1	SM704	ON	ON	ON	OFF	OFF	OFF
2	SM716	ON	---	---	OFF	---	---
3	SM717	---	ON	---	---	OFF	---
4	SM718	---	---	ON	---	---	OFF

在待機程式的情況下，爲基於調用根源程式的特殊繼電器的 ON/OFF 情況。

(7) n 中指定的值爲 0 時將執行無處理。

出　錯

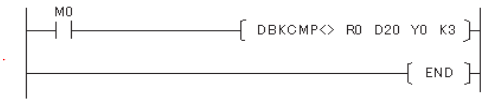
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/ Q00/ Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 中指定了負值時。	---	---	---	---	○	○
4101	從ⓢ1、ⓢ2、ⓓ 中指定的軟元件開始至第 n 點爲止的範圍超出了相應軟元件的範圍時。 從ⓢ1開始至第 n 點爲止的軟元件範圍與從ⓓ 開始的至第 n 點爲止的軟元件範圍相重復時。 從ⓢ2開始至第 n 點爲止的軟元件範圍與從ⓓ 開始的至第 n 點爲止的軟元件範圍相重復時。	---	---	---	---	○	○

程式示例

(1) 以下爲 M0 變爲 ON 時，將 R0 ～ R5 中存儲的值與 D20 ～ D25 中存儲的值進行比較，並將運算結果存儲到 Y0 ～ Y2 中的程式。

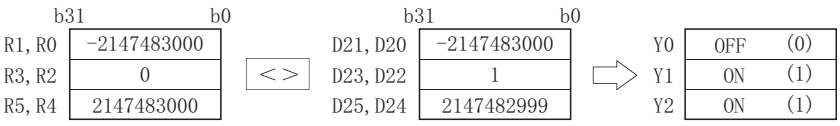
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	M0
1	DBKCMP<>	R0 D20 Y0 K3
6	END	

[動作]



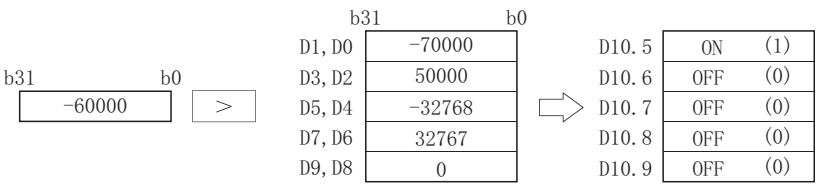
(2) 以下爲 M0 變爲 ON 時，將常數與 D0 ～ D9 中存儲的值進行比較，並將運算結果存儲到 D10.5 ～ D10.9 中的程式。

[梯形圖模式]

[列表模式]

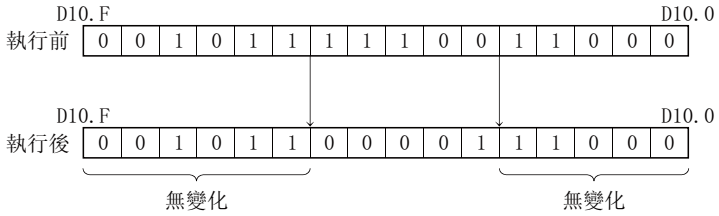
步	指令	軟元件
0	LD	M0
1	DBKCMP>	K-60000 D0 D10.5 K5
6	END	

[動作]



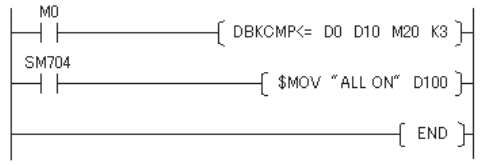
要 點

進行了字軟元件的位元指定時，除存儲運算結果的位元指定軟元件以外的其他軟元件不發生變化。



(3) 以下爲 M0 變爲 ON 時，將 D0 ～ D5 中存儲的值與 D10 ～ D15 中存儲的值進行比較，並將運算結果存儲到 M20 ～ M22 中的程式，當 M20 ～ M22 的軟元件全部變爲 ON 時，將字串 “ ALL ON ” 傳送到 D100 的後面。

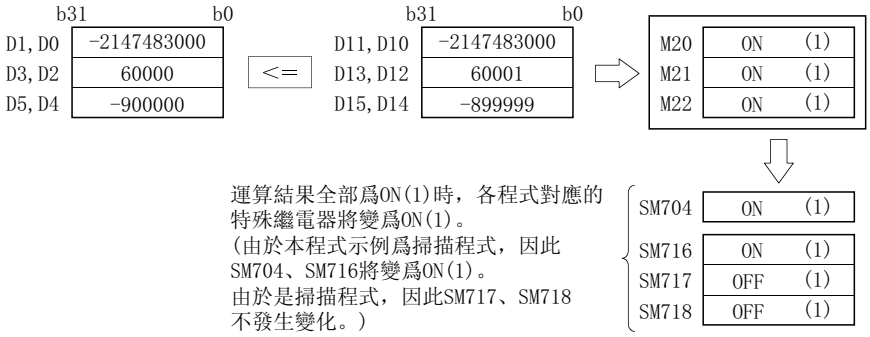
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	M0
1	DBKCMP<=	D0 D10 M20 K3
6	LD	SM704
7	$MOV	“ALLON” D100
13	END	

[動作]

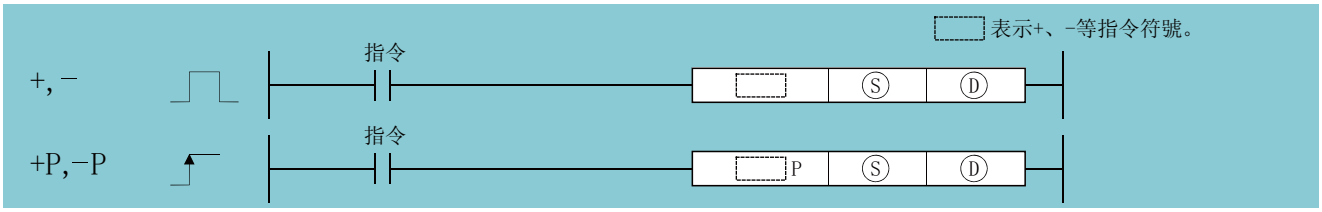


6.2 算術運算指令

6.2.1 +、+P、-、-P

Basic | High performance | Process | Redundant | Universal | LCPU

1 設置資料爲 2 個時 (Ⓓ+Ⓢ→Ⓓ、Ⓓ-Ⓢ→Ⓓ)



Ⓢ : 加法減法資料或者存儲加法減法資料的軟元件的起始編號 (BIN16 位)。

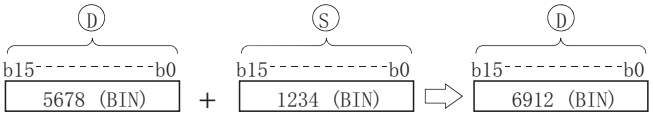
Ⓓ : 存儲加法減法資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ				○				○	---
Ⓓ				○				---	---

功 能

+

(1) 將Ⓓ中指定的 BIN16 位元資料與Ⓢ中指定的 BIN16 位元資料進行加法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ、Ⓓ中可指定的範圍爲 -32768 ～ 32767(BIN16 位)。

(3) 資料的正負判定是在最高位 (b15) 中進行。

· 0 .....正

· 1 .....負

(4) 運算結果中發生了下溢或者上溢時的情況如下所示。

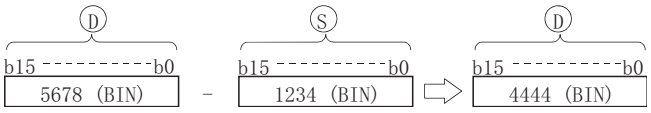
在這種情況下，進位元標誌不變爲 ON。

• K32767 (7FFFH) +K2 (0002H) → K-32767 (8001H) ··· 由於b15爲1，因此爲負值。

• K-32768 (8000H) +K-2 (FFFEH) → K32766 (7FFEH) ····· 由於b15爲0，因此爲正值。

-

(1) 將Ⓓ中指定的 BIN16 位元資料與Ⓢ中指定的 BIN16 位元資料進行減法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ、Ⓓ中可指定的範圍爲 -32768 ～ 32767(BIN16 位)。

(3) 資料的正負判定是在最高位 (b15) 中進行。

· 0 ... 正

· 1 ... 負

(4) 運算結果中發生了下溢或者上溢時的情況如下所示。
在這種情況下，進位元標誌不變爲 ON。

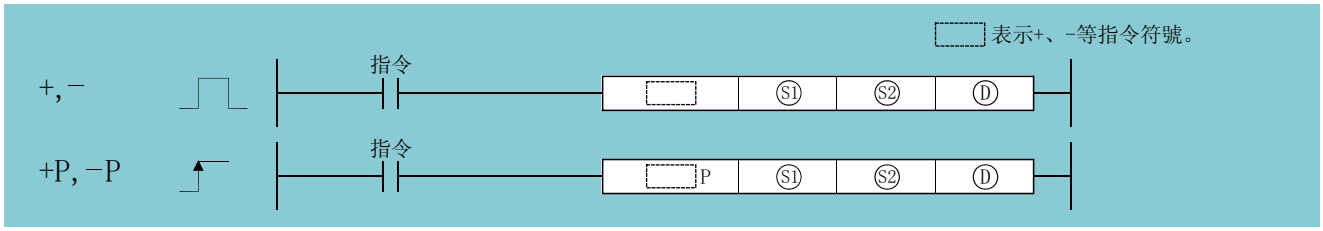
・K-32768- K2 ⟶ K32766·····由於b15爲0，因此爲正值。
(8000H) (0002H) (7FFEH)

・K32767 -K-2 ⟶ K-32767 ··· 由於b15爲1，因此爲負值。
(7FFFH) (FFFEH) (8001H)

出　錯

(1) 在 +(P)、-(P) 指令中無運算出錯。

2 設置資料爲 3 個時 (Ⓢ1+Ⓢ2→Ⓓ、Ⓢ1-Ⓢ2→Ⓓ)



Ⓢ1 ： 被加數被減數資料或者存儲被加數被減數資料的軟元件的起始編號 (BIN16 位)。

Ⓢ2 ： 加數減數資料或者存儲加數減數資料的軟元件的起始編號 (BIN16 位)。

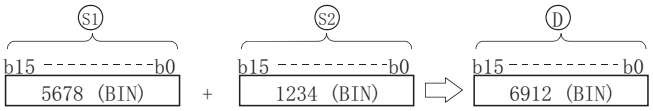
Ⓓ ： 存儲運算結果的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	○							○	---
Ⓢ2	○							○	---
Ⓓ	○							---	---

功　能

+

(1) 將Ⓢ1中指定的 BIN16 位元資料與Ⓢ2中指定的 BIN16 位元資料進行加法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的範圍爲 -32768 ～ 32767(BIN16 位)。

(3) 資料的正負判定是在最高位 (b15) 中進行。
・0　正
・1　負

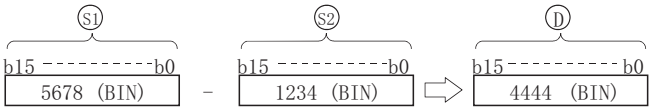
(4) 運算結果中發生了下溢或者上溢時的情況如下所示。
在這種情況下，進位元標誌不變爲 ON。

・K32767 +K2 ⟶ K-32767 ··· 由於b15爲1，因此爲負值。
(7FFFH) (0002H) (8001H)

・K-32768 +K-2 ⟶ K32766·····由於b15爲0，因此爲正值。
(8000H) (FFFEH) (7FFEH)

-

(1) 將ⓢ1中指定的 BIN16 位元資料與ⓢ2中指定的 BIN16 位元資料進行減法運算，並將運算結果存儲到ⓓ中指定的軟元件中。



(2) ⓢ1、ⓢ2、ⓓ中可可指定的範圍爲 -32768 ～ 32767(BIN16 位)。

(3) 資料的正負判定是在最高位 (b15) 中進行。

· 0 ....正

· 1 ....負

(4) 運算結果中發生了下溢或者上溢時的情況如下所示。

在這種情況下，進位元標誌不變爲 ON。

• K－32768－K2 (8000H) (0002H) ⟶ K32766·····由於b15爲0，因此爲正值。 (7FFEH)

• K32767 －K-2 (7FFFH) (FFFEH) ⟶ K－32767 ···由於b15爲1，因此爲負值。 (8001H)

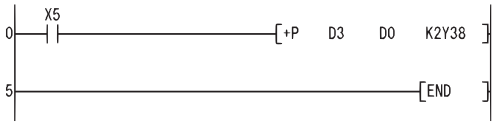
出　錯

(1) 在 +(P)、-(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X5 變爲 ON 時，將 D3 與 D0 的內容進行加法運算，並將運算結果存儲到 Y38 ～ Y3F 中的程式。

[梯形圖模式]

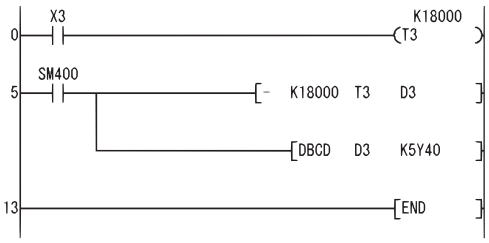


[列表模式]

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

(2) 以下爲將計時器 T3 的設置值與當前值的差以 BCD 格式輸出到 Y40 ～ Y53 中的程式。

[梯形圖模式]



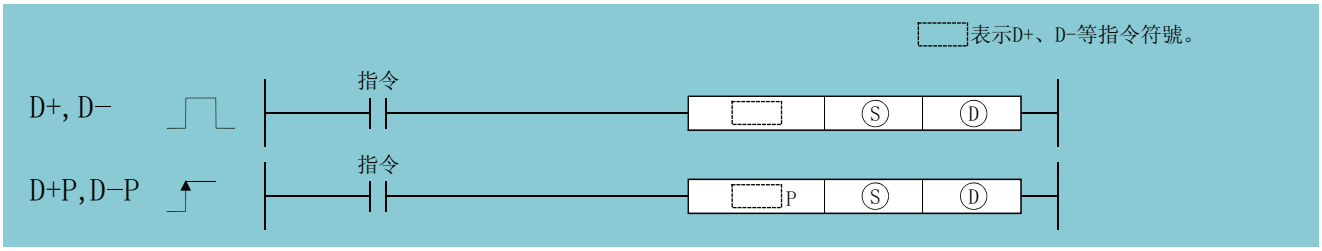
[列表模式]

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

6.2.2 D+、D+P、D-、D-P

Basic | High performance | Process | Redundant | Universal | LCPU

1 設置資料爲 2 個時 ((Ⓓ+1、Ⓓ)+(Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ)、(Ⓓ+1、Ⓓ) - (Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ))



Ⓢ ：加法減法資料或者存儲加法減法資料的軟元件的起始編號 (BIN32 位)。

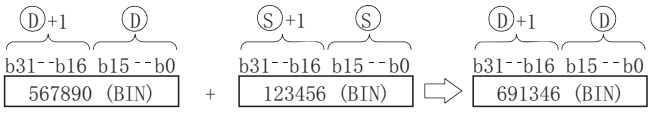
Ⓓ ：存儲加法減法資料的軟元件的起始編號 (BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功 能

D+

(1) 將Ⓓ中指定的 BIN32 位元資料與Ⓢ中指定的 BIN32 位元資料進行加法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ、Ⓓ中可指定的範圍爲 -2147483648 ～ 2147483647(BIN32 位)。

(3) 資料的正負判定是在最高位 (b31) 中進行。

· 0 ... 正

· 1 ... 負

(4) 運算結果中發生了下溢或者上溢時的情況如下所示。

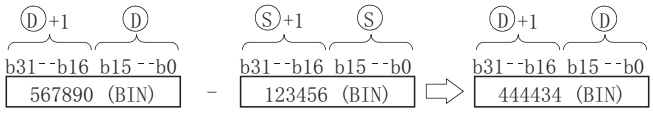
在這種情況下，進位元標誌不變爲 ON。

・K2147483647 (7FFFFFFFH) +K2 (00000002H) → K-2147483647 (80000001H) ··· 由於b31爲1，因此爲負值。

・K-2147483648 (80000000H) +K-2 (FFFFFFFEH) → K2147483646 (7FFFFFFEH) ····· 由於b31爲0，因此爲正值。

D-

(1) 將Ⓓ中指定的 BIN32 位元資料與Ⓢ中指定的 BIN32 位元資料進行減法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ、Ⓓ中可指定的範圍爲 -2147483648 ～ 2147483647(BIN32 位)。

(3) 資料的正負判定是在最高位 (b31) 中進行。

· 0 .... 正

· 1 .... 負

(4) 運算結果中發生了下溢或者上溢時的情況如下所示。
在這種情況下，進位元標誌不變爲 ON。

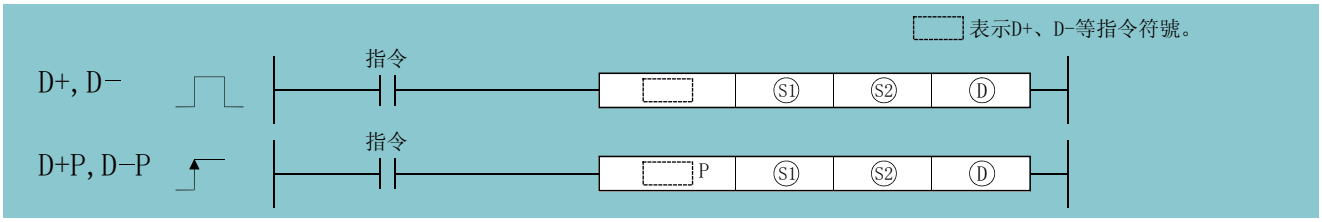
・K-2147483648 (80000000H) − K2 (00000002H) ⟶ K2147483646 (7FFFFFFEH) ····· 由於b31爲0，因此爲正值。

・K2147483647 (7FFFFFFFH) − K-2 (FFFFFFFEH) ⟶ K-2147483647 (80000001H) ··· 由於b31爲1，因此爲負值。

出　錯

(1) 在 D+(P)、D-(P) 指令中無運算出錯。

2 設置資料爲 3 個時 ((Ⓢ1+1、Ⓢ1)+(Ⓢ2+1、Ⓢ2) → (Ⓓ+1、Ⓓ)、 (Ⓢ1+1、Ⓢ1) - (Ⓢ2+1、Ⓢ2) → (Ⓓ+1、Ⓓ))



Ⓢ1 ： 被加數被減數資料或者存儲被加數被減數資料的軟元件的起始編號 (BIN32 位)。

Ⓢ2 ： 加數減數資料或者存儲加數減數資料的軟元件的起始編號 (BIN32 位)。

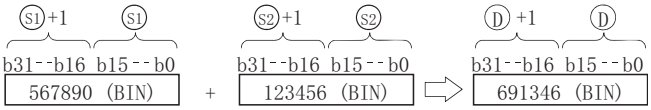
Ⓓ ： 存儲運算結果的軟元件的起始編號 (BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	○							○	---
Ⓢ2	○							○	---
Ⓓ	○							---	---

功　能

D+

(1) 將Ⓢ1中指定的 BIN32 位元資料與Ⓢ2中指定的 BIN32 位元資料進行加法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的範圍爲 -2147483648 ～ 2147483647(BIN32 位)。

(3) 資料的正負判定是在最高位 (b31) 中進行。
・0 ....正
・1 ....負

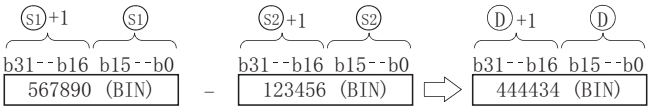
(4) 運算結果中發生了下溢或者上溢時的情況如下所示。
在這種情況下，進位元標誌不變爲 ON。

・K2147483647 (7FFFFFFFH) +K2 (00000002H) ⟶ K- 2147483647 (80000001H) ··· 由於b31爲1，因此爲負值。

・K-2147483648 (80000000H) +K-2 (FFFFFFFEH) ⟶ K2147483646 (7FFFFFFEH) ····· 由於b31爲0，因此爲正值。

D-

(1) 將Ⓢ1中指定的 BIN32 位元資料與Ⓢ2中指定的 BIN32 位元資料進行減法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的範圍爲 -2147483648 ～ 2147483647(BIN32 位)。

(3) 資料的正負判定是在最高位 (b31) 中進行。

· 0 .... 正

· 1 .... 負

(4) 運算結果中發生了下溢或者上溢時的情況如下所示。

在這種情況下，進位元標誌不變爲 ON。

• K-2147483648 (80000000H) -K2 (00000002H) ⟶ K2147483646·····由於b31爲0，因此爲正值。 (7FFFFFFEH)

• K2147483647 (7FFFFFFFH) -K-2 (FFFFFFFEH) ⟶ K-2147483647 (80000001H) ···由於b31爲1，因此爲負值。

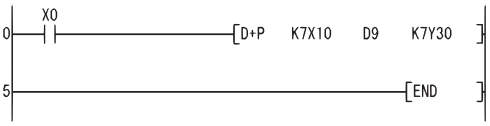
出　錯

(1) 在 D+(P)、D-(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X0 變爲 ON 時，將 X10 ～ X2B 的 28 位元資料與 D9、D10 的資料進行加法運算，並將運算結果輸出到 Y30 ～ Y4B 中的程式。

[梯形圖模式]

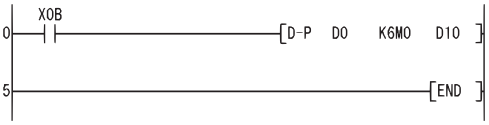


[列表模式]

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

(2) 以下爲 XB 變爲 ON 時，將 D0、D1 的資料與 M0 ～ M23 的資料進行減法運算，並將運算結果存儲到 D10、D11 中的程式。

[梯形圖模式]

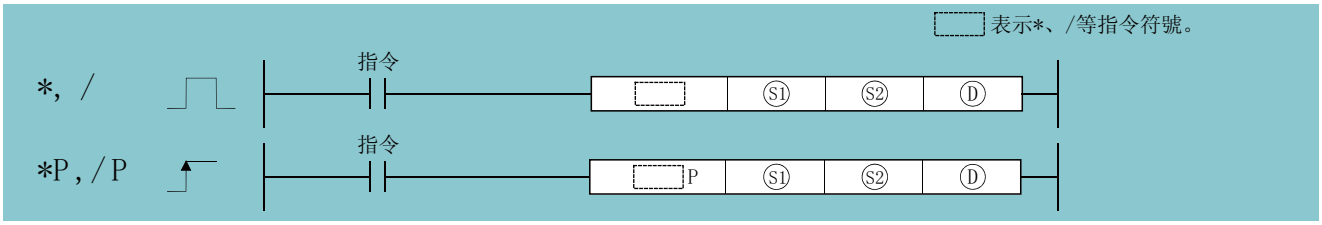


[列表模式]

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

6.2.3 *、*P、/、/P

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ1 ：被乘數被除數資料或者存儲被乘數被除數資料的軟元件的起始編號 (BIN16 位)。

Ⓢ2 ：乘數除數資料或者存儲乘數除數資料的軟元件的起始編號 (BIN16 位)。

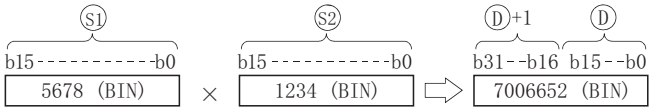
Ⓓ ：存儲運算結果的軟元件的起始編號 (BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	○							○	---
Ⓢ2	○							○	---
Ⓓ	○							---	---

功　能

*

(1) 將Ⓢ1中指定的 BIN16 位元資料與Ⓢ2中指定的 BIN16 位元資料進行乘法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓓ爲位軟元件時則從低位開始指定。

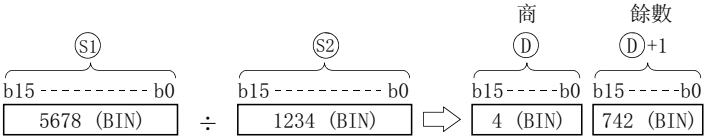
例
K1 .....低 4 位 (b0 ～ b3)
K4 .....低 16 位 (b0 ～ b15)
K8 .....32 位 (b0 ～ b31)

(3) Ⓢ1、Ⓢ2中可指定的範圍爲 -32768 ～ 32767(BIN16 位)。

(4) Ⓢ1、Ⓢ2、Ⓓ的資料的正負判定是在最高位 (Ⓢ1、Ⓢ2爲 b15、Ⓓ爲 b31) 中進行。
· 0 ... 正
· 1 ... 負

/

(1) 將Ⓢ1中指定的 BIN16 位元資料與Ⓢ2中指定的 BIN16 位元資料進行除法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) 在字軟元件的情況下，則將除法運算結果的商和餘數以 32 位元進行存儲；在位軟元件的情況下，則以 16 位元格式僅存儲商。
商 ..... 存儲在低 16 位元中。
餘數 ... 存儲在高 16 位元中。(只有在字軟元件的情況下才可以存儲。)

(3) Ⓢ1、Ⓢ2中可指定的範圍爲 -32768 ～ 32767(BIN16 位)。

(4) ⓢ1、ⓢ2、ⓓ、ⓓ+1 的資料的正負判定是在最高位 (b15) 中進行。
(商及餘數均附加符號。)
· 0 .....正
· 1 .....負

出　錯

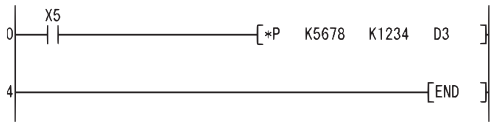
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	除數爲 0 時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X5 變爲 ON 時，將 BIN 的 “ 5678 ” 與 “ 1234 ” 的乘法運算結果存儲到 D3、D4 中的程式。

[梯形圖模式]

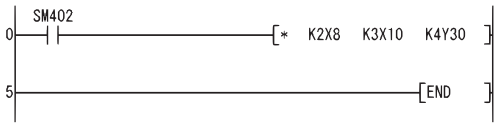


[列表模式]

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

(2) 以下爲將 X8 ～ XF 的 BIN 資料與 X10 ～ X1B 的 BIN 資料的乘法結果輸出到 Y30 ～ Y3F 中的程式。

[梯形圖模式]

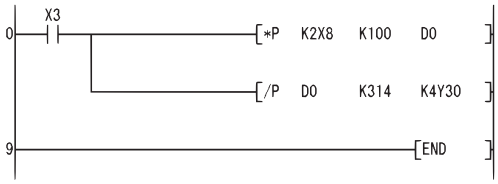


[列表模式]

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

(3) 以下爲 X3 變爲 ON 時，將 X8 ～ XF 的資料用 3.14 相除，並將運算結果輸出到 Y30 ～ Y3F 中的程式。

[梯形圖模式]

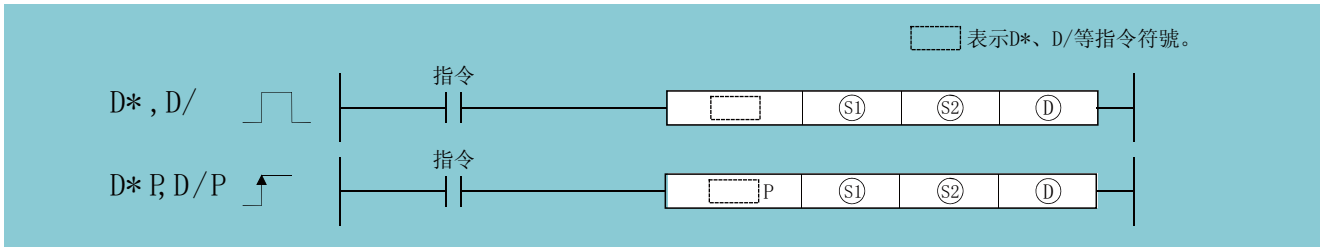


[列表模式]

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

6.2.4 D*、D*P、D/、D/P

Basic　High performance　Process　Redundant　Universal　LCPU



ⓢ1 ： 被乘數被除數資料或者存儲被乘數被除數資料的軟元件的起始編號 (BIN32 位)。

ⓢ2 ： 乘數除數資料或者存儲乘數除數資料的軟元件的起始編號 (BIN32 位)。

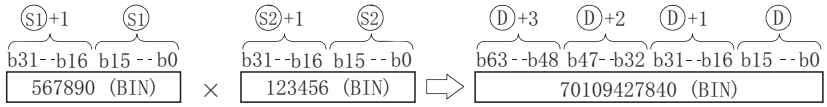
ⓓ ： 存儲運算結果的軟元件的起始編號 (BIN64 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
ⓢ1	○			○					---
ⓢ2	○			○					---
ⓓ	○			---					---

功　能

D*

(1) 將Ⓢ1中指定的 BIN32 位元資料與Ⓢ2中指定的 BIN32 位元資料進行乘法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓓ爲位軟元件時，則乘法結果的低 32 位元之前成爲物件，不能對高 32 位進行指定。

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

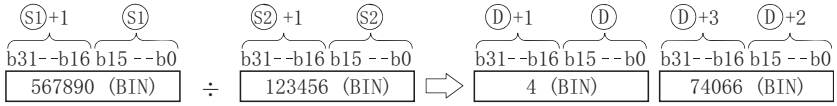
在位軟元件中需要使用乘法運算結果的高 32 位元資料時，則應預先將資料暫存在字軟元件中，然後將字軟元件的 (Ⓓ+2)、(Ⓓ+3) 的資料傳送到指定的位元軟元件中。

(3) Ⓢ1、Ⓢ2中可指定的範圍爲 -2147483648 ～ 2147483647(BIN32 位)。

(4) Ⓢ1、Ⓢ2、Ⓓ的資料的正負判定是在最高位 (Ⓢ1、Ⓢ2爲 b31、Ⓓ爲 b63) 中進行。
· 0 ....正
· 1 ....負

D/

(1) 將Ⓢ1中指定的 BIN32 位元資料與Ⓢ2中指定的 BIN32 位元資料進行除法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) 在字軟元件的情況下，則將除法運算結果的商和餘數以 64 位元進行存儲；在位軟元件的情況下，則以 32 位元格式僅存儲商。

商 ..... 存儲在低 32 位元中。
餘數 ... 存儲在高 32 位元中。(只有在字軟元件的情況下才可以存儲。)

(3) Ⓢ1、Ⓢ2可指定的範圍爲 -2147483648 ～ 2147483647(BIN32 位)。

(4) Ⓢ1、Ⓢ2、Ⓓ、Ⓓ+2 的資料的正負判定是在最高位 (b31) 中進行。
(商及餘數均附加符號。)
· 0 .....正
· 1 .....負

出　錯

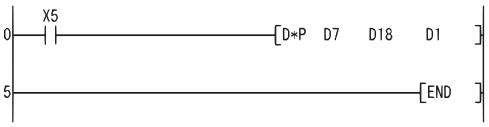
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	除數爲 0 時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X5 變爲 ON 時，將 D7、D8 的 BIN 資料與 D18、D19 的 BIN 資料的乘法運算結果存儲到 D1 ～ D4 中的程式。

[梯形圖模式]

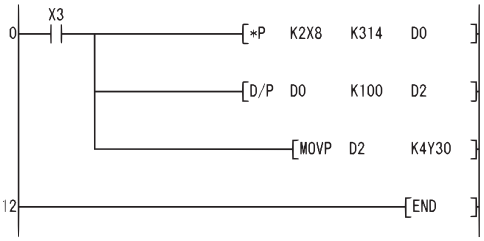


[列表模式]

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

(2) 以下爲 X3 變爲 ON 時，將 X8 ～ XF 的資料用 3.14 相除，並將運算結果輸出到 Y30 ～ Y3F 中的程式。

[梯形圖模式]



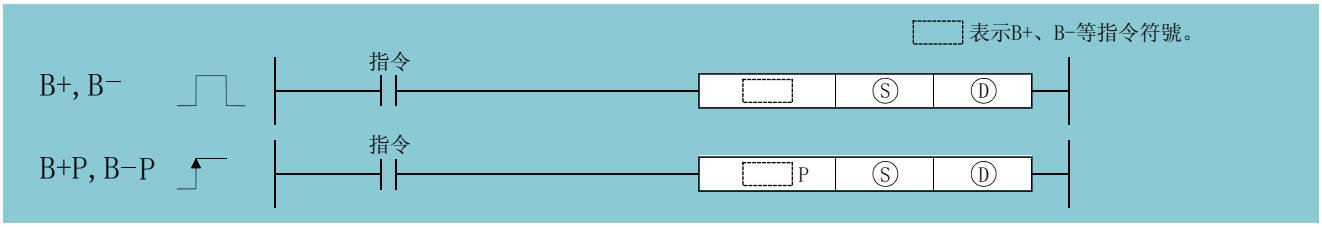
[列表模式]

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

6.2.5 B+、B+P、B-、B-P

Basic | High performance | Process | Redundant | Universal | LCPU

1 BCD4 位加法和減法運算 (Ⓓ+Ⓢ→Ⓓ、Ⓓ-Ⓢ→Ⓓ)



Ⓢ ： 加法減法資料或者存儲加法減法資料的軟元件的起始編號 (BCD4 位)。

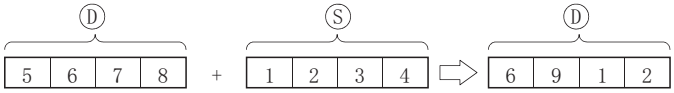
Ⓓ ： 存儲加法減法資料的軟元件的起始編號 (BCD4 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功　能

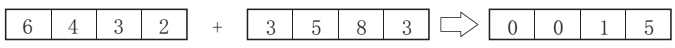
B+

(1) 將Ⓓ中指定的 BCD4 位元資料與Ⓢ中指定的 BCD4 位元資料進行加法運算，並將加法運算結果存儲到Ⓓ中指定的軟元件中。



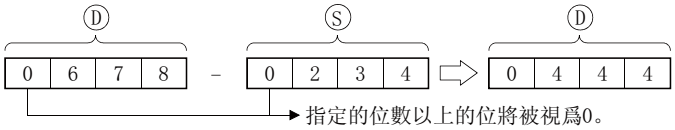
(2) Ⓢ、Ⓓ中可指定的範圍爲 0 ～ 9999(BCD4 位)。

(3) 加法運算結果超過了 9999 時，進位將被視爲無效。
在這種情況下，進位元標誌不變爲 ON。



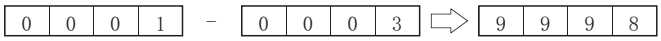
B-

(1) 將Ⓢ中指定的 BCD4 位元資料與Ⓓ中指定的 BCD4 位元資料進行減法運算，並將減法運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ、Ⓓ中可指定的範圍爲 0 ～ 9999(BCD4 位)。

(3) 減法運算結果中發生了下溢時的情況如下所示。
在這種情況下，進位元標誌不變爲 ON。



出　錯

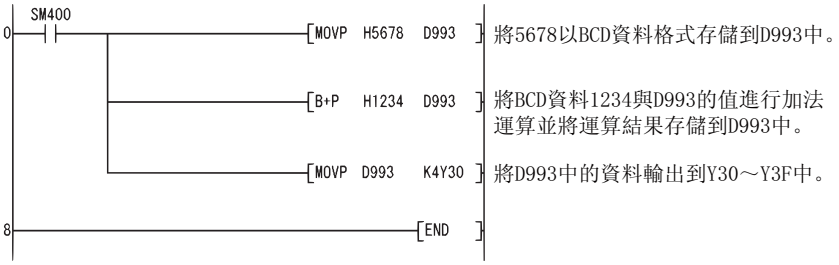
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ、Ⓓ的 BCD 資料超出了 0 ～ 9999 的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲將 5678 及 1234 的 BCD 資料進行加法運算，在將運算結果存儲到 D993 中的同時並輸出到 Y30 ～ Y3F 中的程式。

[梯形圖模式]

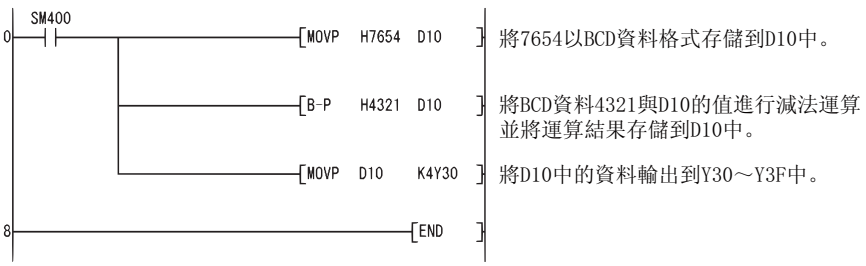


[列表模式]

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

(2) 以下爲將 7654 及 4321 的 BCD 資料進行減法運算，在將運算結果存儲到 D10 中的同時並輸出到 Y30 ～ Y3F 中的程式。

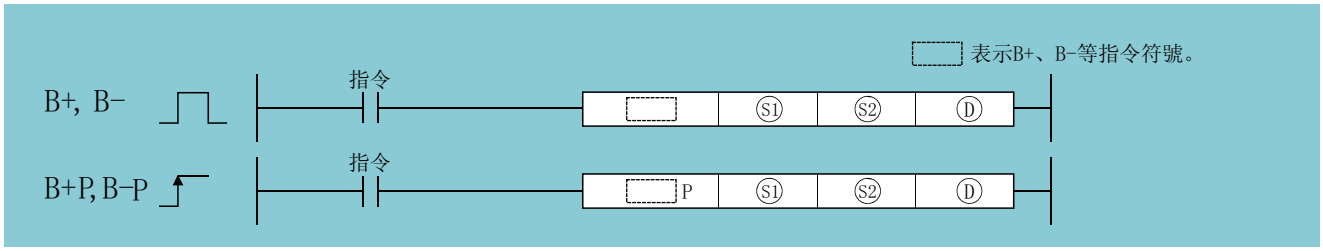
[梯形圖模式]



[列表模式]

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

2 設置資料爲 3 個時 (Ⓢ1+Ⓢ2→Ⓓ、Ⓢ1-Ⓢ2→Ⓓ)



Ⓢ1 ： 被加數被減數資料或者存儲被加數被減數資料的軟元件的起始編號 (BCD4 位)。

Ⓢ2 ： 加數減數資料或者存儲加數減數資料的軟元件的起始編號 (BCD4 位)。

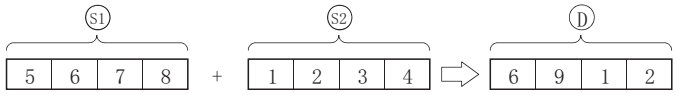
Ⓓ ： 存儲運算結果的軟元件的起始編號 (BCD4 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	○							○	---
Ⓢ2	○							○	---
Ⓓ	○							---	---

功　能

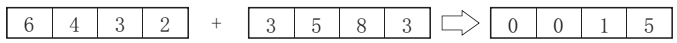
B+

(1) 將Ⓢ1中指定的 BCD4 位元資料與Ⓢ2中指定的 BCD4 位元資料進行加法運算，並將加法運算結果存儲到Ⓓ中指定的軟元件中。



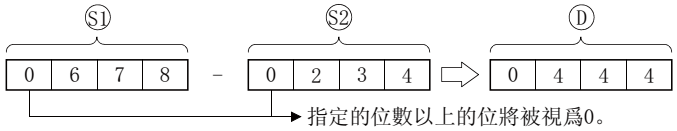
(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的範圍爲 0 ～ 9999(BCD4 位)。

(3) 加法運算結果超過了 9999 時，進位將被視爲無效。
在這種情況下，進位元標誌不變爲 ON。



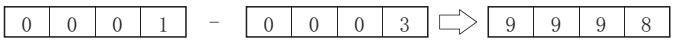
B-

(1) 將Ⓢ1中指定的 BCD4 位元資料與Ⓢ2中指定的 BCD4 位元資料進行減法運算，並將減法運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的範圍爲 0 ～ 9999(BCD4 位)。

(3) 減法運算結果中發生了下溢時的情況如下所示。
在這種情況下，進位元標誌不變爲 ON。



出　錯

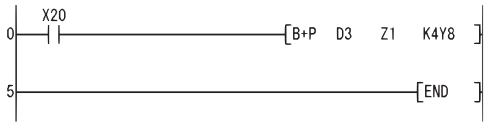
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ1、Ⓢ2的 BCD 資料超出了 0 ～ 9999 的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D3 的 BCD 資料與 Z1 的 BCD 資料進行加法運算，並將運算結果輸出到 Y8 ～ Y17 中的程式。

[梯形圖模式]

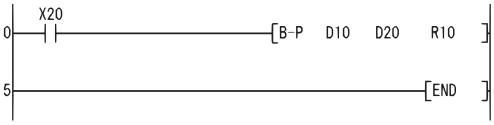


[列表模式]

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

(2) 以下爲 X20 變爲 ON 時，將 D10 的 BCD 資料與 D20 的 BCD 資料進行減法運算，並將運算結果輸出到 R10 中的程式。

[梯形圖模式]



[列表模式]

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

6.2.6 DB+、DB+P、DB-、DB-P

Basic | High performance | Process | Redundant | Universal | LCPU

1 設置資料爲 2 個時 ((Ⓓ+1、Ⓓ)+(Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ)、(Ⓓ+1、Ⓓ) - (Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ))

Ⓢ ：加法減法資料或者存儲加法減法資料的軟元件的起始編號 (BCD8 位)。

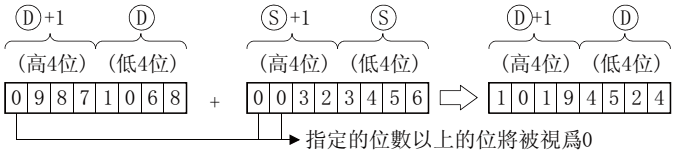
Ⓓ ：存儲加法減法資料的軟元件的起始編號 (BCD8 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ				○				○	---
Ⓓ				○				---	---

功　能

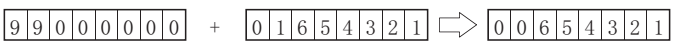
DB+

(1) 將Ⓓ中指定的 BCD8 位元資料與Ⓢ中指定的 BCD8 位元資料進行加法運算，並將加法運算結果存儲到Ⓓ中指定的軟元件中。



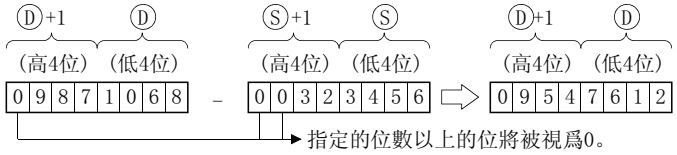
(2) Ⓢ、Ⓓ中可指定的範圍爲 0 ～ 99999999(BCD8 位)。

(3) 加法運算結果超過了 99999999 時，進位將被視爲無效。
在這種情況下，進位元標誌不變爲 ON。



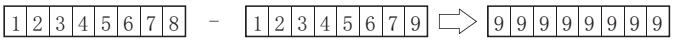
DB-

(1) 將ⓓ中指定的 BCD8 位元資料與Ⓢ中指定的 BCD8 位元資料進行減法運算，並將減法運算結果存儲到ⓓ中指定的軟元件中。



(2) Ⓢ、ⓓ中可指定的範圍爲 0 ～ 99999999(BCD8 位)。

(3) 減法運算結果中發生了下溢時的情況如下所示。在這種情況下，進位元標誌不變爲 ON。



出　錯

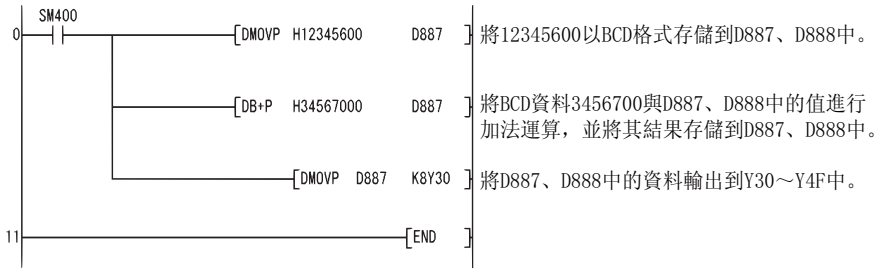
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ、ⓓ的 BCD 資料超出了 0 ～ 9999 的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲將 12345600 及 34567000 的 BCD 資料進行加法運算，在將運算結果存儲到 D887、D888 中的同時並輸出到 Y30 ～ Y4F 中的程式。

[梯形圖模式]

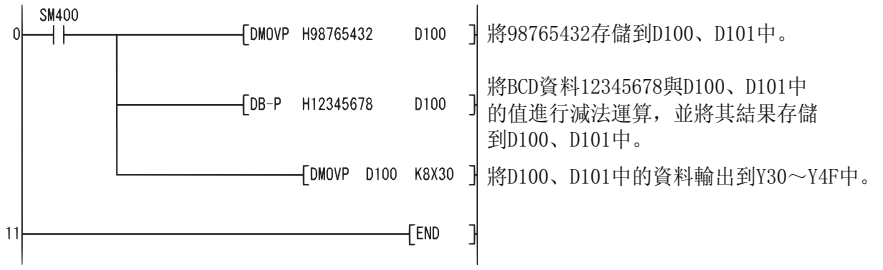


[列表模式]

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

(2) 以下爲將 98765432 及 12345678 的 BCD 資料進行減法運算，在將運算結果存儲到 D100、D101 中的同時並輸出到 Y30 ～ Y4F 中的程式。

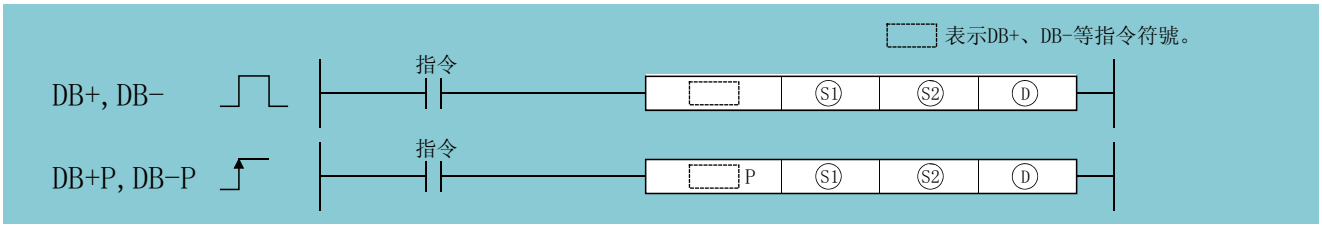
[梯形圖模式]



[列表模式]

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

2 設置資料爲 3 個時 ((Ⓢ1+1、Ⓢ1)+(Ⓢ2+1、Ⓢ2) → (Ⓓ+1、Ⓓ)、 (Ⓢ1+1、Ⓢ1) - (Ⓢ2+1、Ⓢ2) → (Ⓓ+1、Ⓓ))



Ⓢ1 ： 被加數被減數資料或者存儲被加數被減數資料的軟元件的起始編號 (BCD8 位)。

Ⓢ2 ： 加數減數資料或者存儲加數減數資料的軟元件的起始編號 (BCD8 位)。

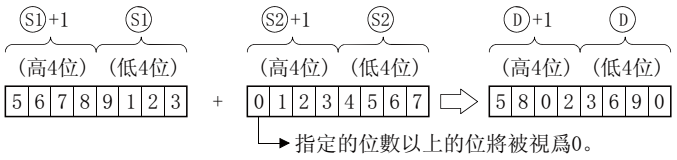
Ⓓ ： 存儲運算結果的軟元件的起始編號 (BCD8 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	○							○	---
Ⓢ2	○							○	---
Ⓓ	○							---	---

功 能

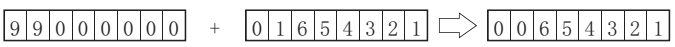
DB+

(1) 將Ⓢ1中指定的 BCD8 位元資料與Ⓢ2中指定的 BCD8 位元資料進行加法運算，並將加法運算結果存儲到Ⓓ中指定的軟元件中。



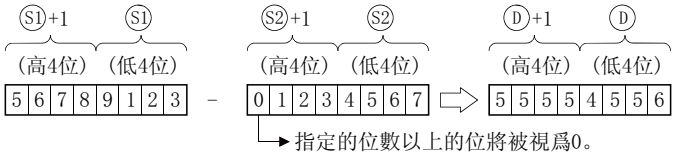
(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的範圍爲 0 ～ 99999999(BCD8 位)。

(3) 加法運算結果超過了 99999999 時，進位將被視爲無效。
在這種情況下，進位元標誌不變爲 ON。



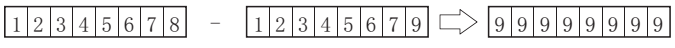
DB-

(1) 將Ⓢ1中指定的 BCD8 位元資料與Ⓢ2中指定的 BCD8 位元資料進行減法運算，並將減法運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的範圍爲 0 ～ 99999999(BCD8 位)。

(3) 減法運算結果中發生了下溢時的情況如下所示。
在這種情況下，進位元標誌不變爲 ON。



出　錯

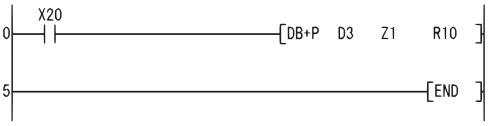
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	ⓢ1、ⓢ2、ⓓ的 BCD 資料超出了 0 ～ 9999 的範圍時。	○	○	○	○	○	○

程式示例

(1)　以下爲 X20 變爲 ON 時，將 D3、D4 的 BCD 資料與 Z1、Z2 的 BCD 資料進行加法運算，並將運算結果存儲到 R10、R11 中的程式。

[梯形圖模式]

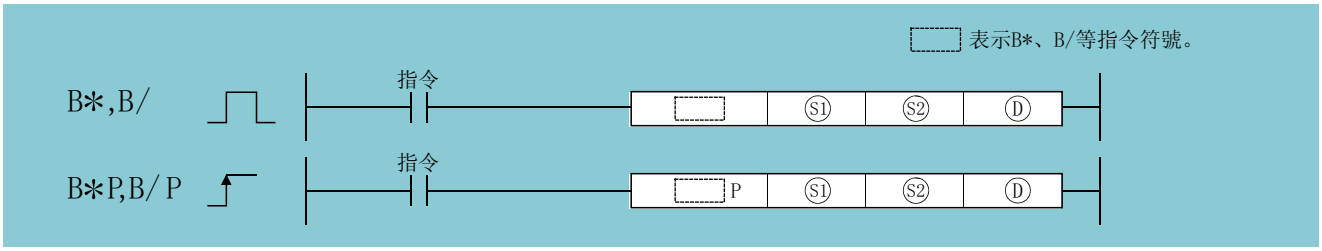


[列表模式]

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

6.2.7　B*、B*P、B/、B/P

Basic　High performance　Process　Redundant　Universal　LCPU



ⓢ1 ：被乘數被除數資料或者存儲被乘數被除數資料的軟元件的起始編號 (BCD4 位)。

ⓢ2 ：乘數除數資料或者存儲乘數除數資料的軟元件的起始編號 (BCD4 位)。

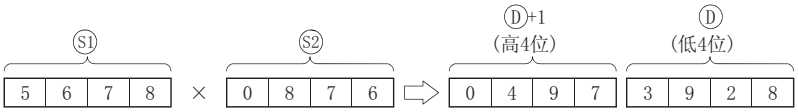
ⓓ ：存儲運算結果的軟元件的起始編號 (BCD8 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
ⓢ1	○							○	---
ⓢ2	○							○	---
ⓓ	○							---	---

功　能

B*

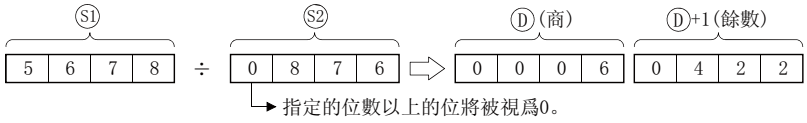
(1)　將ⓢ1中指定的軟元件的 BCD 資料與ⓢ2中指定的軟元件的 BCD 資料進行乘法運算，並將運算結果存儲到ⓓ中指定的軟元件中。



(2)　ⓢ1、ⓢ2中可指定的範圍爲 0 ～ 9999(BCD4 位)。

B/

(1)　將ⓢ1中指定的軟元件的 BCD 資料與ⓢ2中指定的軟元件的 BCD 資料進行除法運算，並將運算結果存儲到ⓓ中指定的軟元件中。



(2) 將除法運算結果的商和餘數以 32 位元進行存儲。
商 (BCD4 位元) .... 存儲在低 16 位元中。
餘數 (BCD4 位元) .. 存儲在高 16 位元中。

(3) 對ⓓ進行了位元軟元件指定時，不存儲除法運算結果的餘數。

出　錯

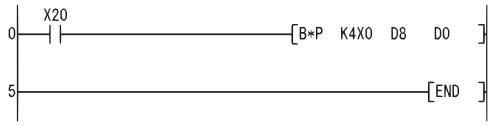
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	ⓢ1、ⓢ2的 BCD 資料超出了 0 ～ 9999 的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 X0 ～ XF 的 BCD 資料與 D8 的 BCD 資料進行乘法運算，並將運算結果存儲到 D0、D1 中的程式。

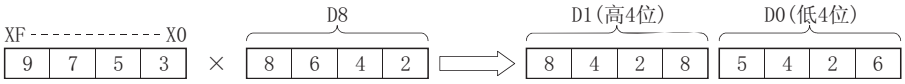
[梯形圖模式]



[列表模式]

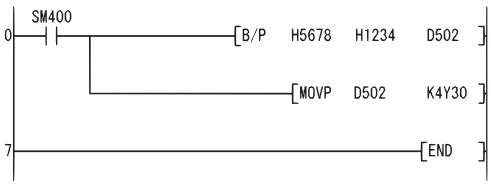
步	指令	軟元件		
0	LD	X20		
1	B*P	K4X0	D8	D0
5	END			

[動作]



(2) 以下爲將 5678 及 1234 的 BCD 資料進行除法運算，在將運算結果存儲到 D502、D503 中的同時並將商輸出到 Y30 ～ Y3F 中的程式。

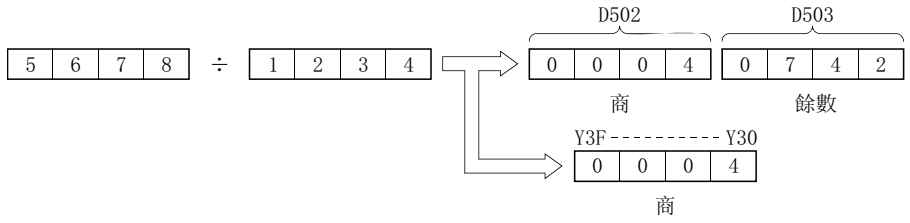
[梯形圖模式]



[列表模式]

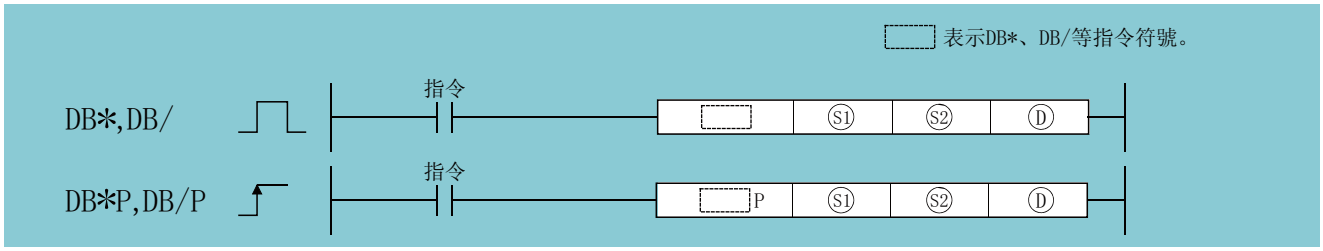
步	指令	軟元件		
0	LD	SM400		
1	B/P	H5678	H1234	D502
5	MOVP	D502	K4Y30	
7	END			

[動作]



6.2.8 DB*、DB*P、DB/、DB/P

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ1 ： 被乘數被除數資料或者存儲被乘數被除數資料的軟元件的起始編號 (BCD8 位)。

Ⓢ2 ： 乘數除數資料或者存儲乘數除數資料的軟元件的起始編號 (BCD8 位)。

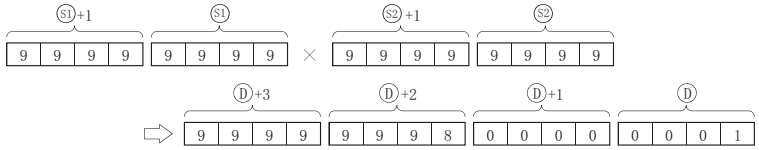
Ⓓ ： 存儲運算結果的軟元件的起始編號 (BCD16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1		○				○			---
Ⓢ2		○				○			---
Ⓓ		○				---			---

功　能

DB*

(1) 將Ⓢ1中指定的軟元件的 BCD8 位元資料與Ⓢ2中指定的軟元件的 BCD8 位元資料進行乘法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



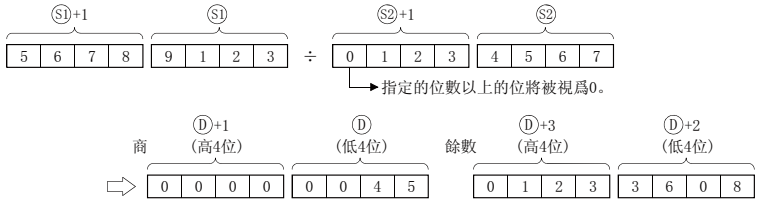
(2) Ⓓ中指定的軟元件爲位軟元件時，乘法運算結果以至低位 8 位數 (低 32 位元) 爲物件，不能對高位 8 位元數 (高 32 位) 進行指定。

K1……低位 1 位數 (b0 ～ 3)，K4……低位 4 位數 (b0 ～ 15)，K8……低位 8 位數 (b0 ～ 31)

(3) Ⓢ1、Ⓢ2中可指定的範圍爲 0 ～ 99999999(BCD8 位)。

DB/

(1) 將Ⓢ1中指定的軟元件的 BCD8 位元資料與Ⓢ2中指定的軟元件的 BCD8 位元資料進行除法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) 將除法運算結果的商和餘數以 64 位元進行存儲。

商 (BCD8 位元) .... 存儲在低 32 位元中。

餘數 (BCD8 位元) .. 存儲在高 32 位元中。

(3) 對Ⓓ進行了位元軟元件指定時，不存儲除法運算結果的餘數。

出　錯

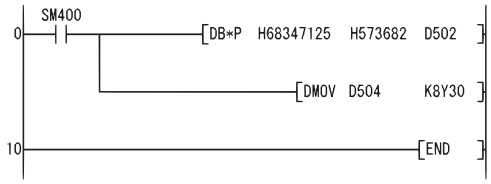
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	⑤1、⑤2的 BCD 資料超出了 0 ～ 9999 的範圍時。 除數⑤2爲 0 時。	○	○	○	○	○	○

程式示例

(1) 以下爲將 67347125 及 573682 的 BCD 資料進行乘法運算，在將運算結果存儲到 D502 ～ D505 中的同時並將高 8 位輸出到 Y30 ～ Y4F 中的程式。

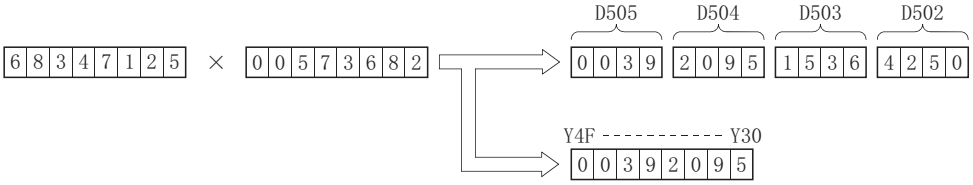
[梯形圖模式]



[列表模式]

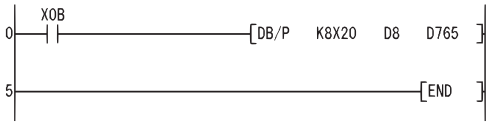
步	指令	軟元件
0	LD	SM400
1	DB*P	H68347125 H573682 D502
7	DMOV	D504 K8Y30
10	END	

[動作]



(2) 以下爲 X0B 變爲 ON 時，將 X20 ～ X3F 的 BCD 資料與 D8、D9 的 BCD 資料進行除法運算，並將運算結果存儲到 D765 ～ D768 中的程式。

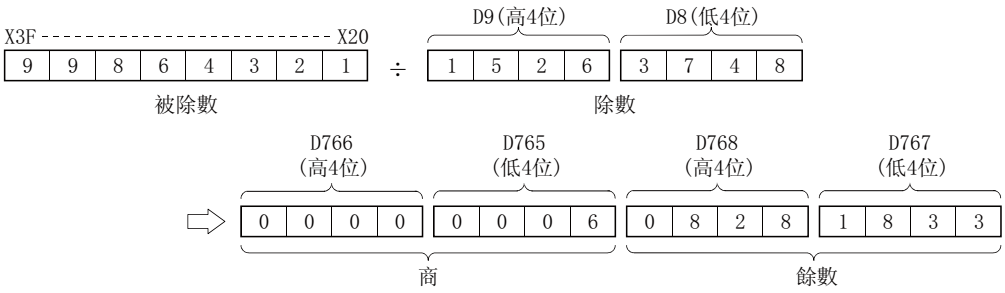
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0B
1	DB/P	K8X20 D8 D765
5	END	

[動作]

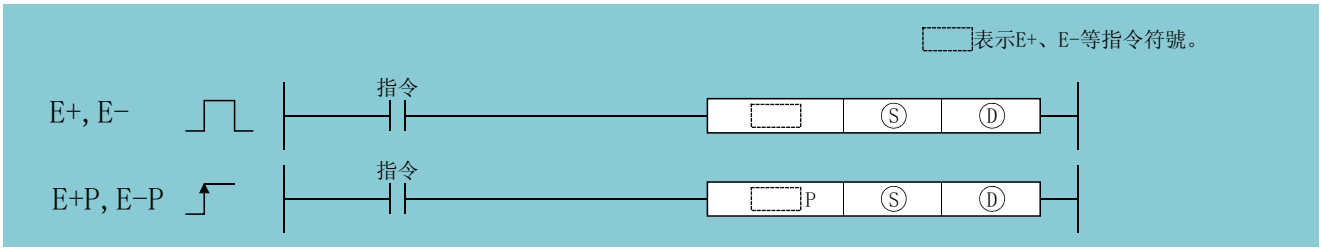


Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

6.2.9 E+、E+P、E-、E-P

1 設置資料爲 2 個時 ((Ⓓ+1、Ⓓ)+(Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ)、(Ⓓ+1、Ⓓ) - (Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ))



Ⓢ ：加法減法資料或者存儲加法減法資料的軟元件的起始編號 (實數)。

Ⓓ ：存儲加法減法資料的軟元件的起始編號 (實數)。

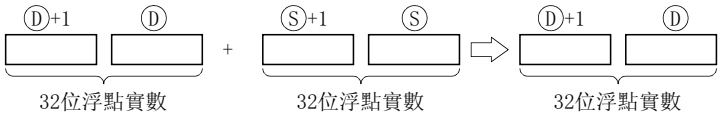
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1：在通用型 QCPU、LCPU 中可以使用。

功　能

E+

(1) 將Ⓓ中指定的 32 位浮點實數與Ⓢ中指定的 32 位浮點實數進行加法運算，並將加法運算結果存儲到Ⓓ中指定的軟元件中。

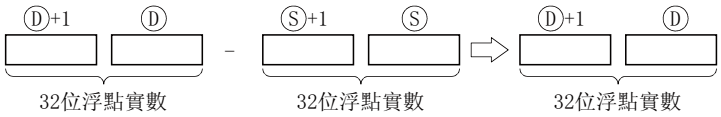


(2) Ⓢ、Ⓓ中可指定的值以及可存儲的值如下所示：

0、$2^{-126} \leq$ | 指定值 (存儲值) | $< 2^{128}$

E-

(1) 將Ⓓ中指定的 32 位浮點實數與Ⓢ中指定的 32 位浮點實數進行減法運算，並將減法運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ、Ⓓ中可指定的值以及可存儲的值如下所示：

0、$2^{-126} \leq$ | 指定值 (存儲值) | $< 2^{128}$

出　錯

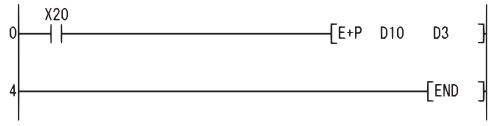
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定的軟元件的內容不在以下範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定的軟元件的內容爲 -0 時。	○	○	○	○	---	---
4141	運算結果超出以下範圍時。 (發生了溢出時。) $2^{128} \leq$ | 運算結果 |	○	○	○	○	○	○
4140	指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D3、D4 的 32 位浮點實數與 D10、D11 的 32 位浮點實數進行加法運算，並將加法運算結果存儲到 D3、D4 中的程式。

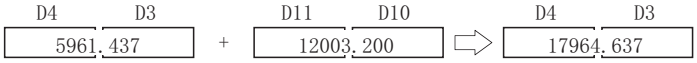
[梯形圖模式]



[列表模式]

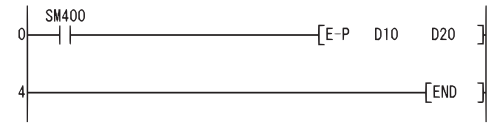
步	指令	軟元件	
0	LD	X20	
1	E+P	D10	D3
4	END		

[動作]



(2) 以下爲將 D20、D21 的 32 位浮點實數與 D10、D11 的 32 位浮點實數進行減法運算，並將減法運算結果存儲到 D20、D21 中的程式。

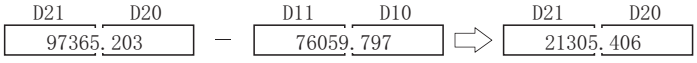
[梯形圖模式]



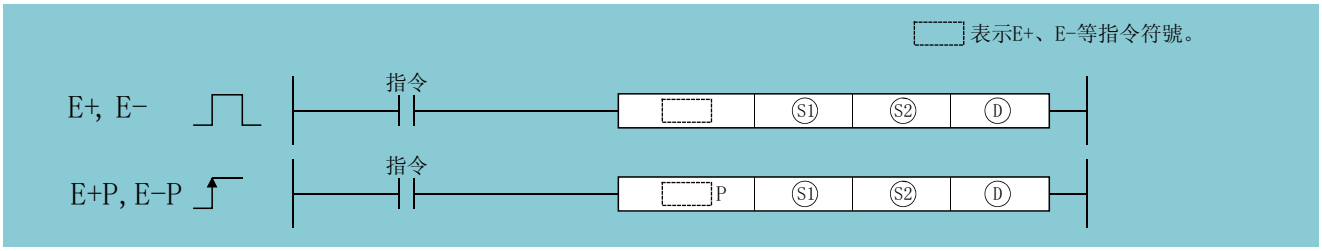
[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	E-P	D10	D20
4	END		

[動作]



2 設置資料爲 3 個時 ((Ⓢ1+1、Ⓢ1)+(Ⓢ2+1、Ⓢ2) → (Ⓓ+1、Ⓓ)、(Ⓢ1+1、Ⓢ1) - (Ⓢ2+1、Ⓢ2) → (Ⓓ+1、Ⓓ))



Ⓢ1 ： 被加數被減數資料或者存儲被加數被減數資料的軟元件的起始編號（實數）。

Ⓢ2 ： 加數減數資料或者存儲加數減數資料的軟元件的起始編號（實數）。

Ⓓ ： 存儲運算結果的軟元件的起始編號（實數）。

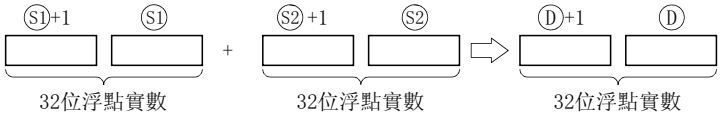
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ1	---	○		---	○		○ *1	○	---
Ⓢ2	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCP 中可以使用。

功 能

E+

(1) 將Ⓢ1中指定的 32 位浮點實數與Ⓢ2中指定的 32 位浮點實數進行加法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。

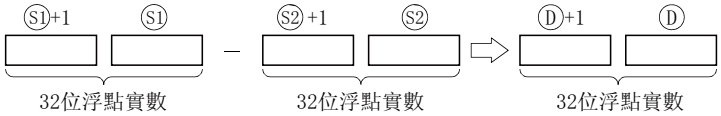


(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的值以及可存儲的值如下所示。

0、$2^{-126} \leq$ | 指定值（存儲值） | $< 2^{128}$

E-

(1) 將Ⓢ1中指定的 32 位浮點實數與Ⓢ2中指定的 32 位浮點實數進行減法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的值以及可存儲的值如下所示。

0、$2^{-126} \leq$ | 指定值（存儲值） | $< 2^{128}$

出　錯

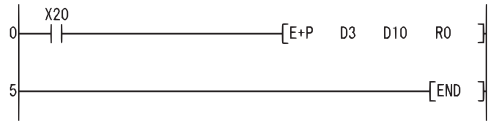
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定的軟元件的內容不在以下範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定的軟元件的內容爲 -0 時。	○	○	○	○	---	---
4141	運算結果超出以下範圍時。 (發生了溢出時。) $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○
4140	指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D3、D4 的 32 位浮點實數與 D10、D11 的 32 位浮點實數進行加法運算，並將加法運算結果存儲到 R0、R1 中的程式。

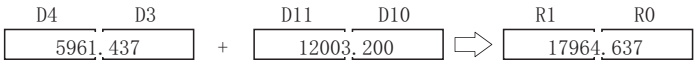
[梯形圖模式]



[列表模式]

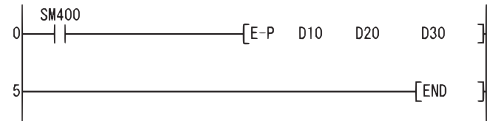
步	指令	軟元件		
0	LD	X20		
1	E+P	D3	D10	R0
5	END			

[動作]



(2) 以下爲將 D10、D11 的 32 位浮點實數與 D20、D21 的 32 位浮點實數進行減法運算，並將減法運算結果存儲到 D30、D31 中的程式。

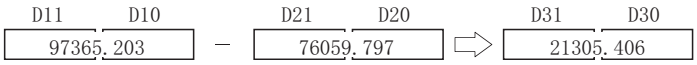
[梯形圖模式]



[列表模式]

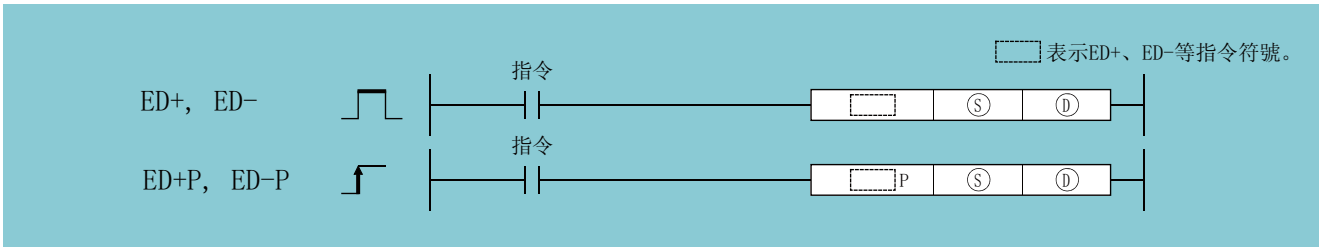
步	指令	軟元件		
0	LD	SM400		
1	E-P	D10	D20	D30
5	END			

[動作]



6.2.10 ED+、ED+P、ED-、ED-P

1 設置資料爲 2 個時 ((Ⓓ+3、Ⓓ+2、Ⓓ+1、Ⓓ)+(Ⓢ+3、Ⓢ+2、Ⓢ+1、Ⓢ) → (Ⓓ+3、Ⓓ+2、Ⓓ+1、Ⓓ)、(Ⓓ+3、Ⓓ+2、Ⓓ+1、Ⓓ) - (Ⓢ+3、Ⓢ+2、Ⓢ+1、Ⓢ) → (Ⓓ+3、Ⓓ+2、Ⓓ+1、Ⓓ))



Ⓢ ：加法減法資料或者存儲加法減法資料的軟元件的起始編號（實數）。

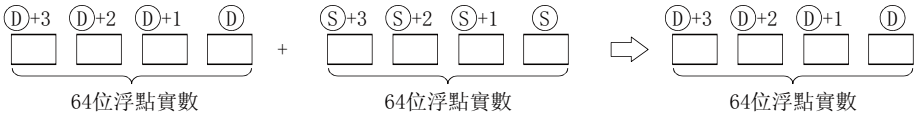
Ⓓ ：存儲加法減法資料的軟元件的起始編號（實數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

ED+

(1) 將Ⓓ中指定的 64 位浮點實數與Ⓢ中指定的 64 位浮點實數進行加法運算，並將加法運算結果存儲到Ⓓ中指定的軟元件中。

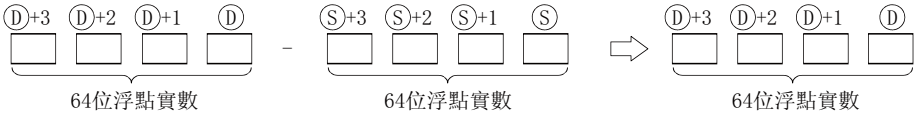


(2) Ⓢ、Ⓓ中可指定的值以及可存儲的值如下所示。

0、$2^{-1022} \leq$ | 指定值（存儲值） | $< 2^{1024}$

ED-

(1) 將Ⓓ中指定的 64 位浮點實數與Ⓢ中指定的 64 位浮點實數進行減法運算，並將減法運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ、Ⓓ中可指定的值以及可存儲的值如下所示。

0、$2^{-1022} \leq$ | 指定值（存儲值） | $< 2^{1024}$

出　錯

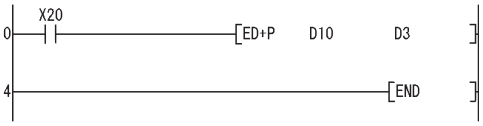
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定的軟元件的內容不在以下範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定的軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出以下範圍時。 (發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D3 ～ D6 的 64 位浮點實數與 D10 ～ D13 的 64 位浮點實數進行加法運算，並將加法運算結果存儲到 D3 ～ D6 中的程式。

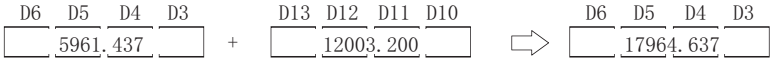
[梯形圖模式]



[列表模式]

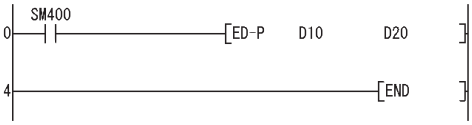
步	指令	軟元件	
0	LD	X20	
1	ED+P	D10	D3
4	END		

[動作]



(2) 以下爲將 D20 ～ D23 的 64 位浮點實數與 D10 ～ D13 的 64 位浮點實數進行減法運算，並將減法運算結果存儲到 D20 ～ D23 中的程式。

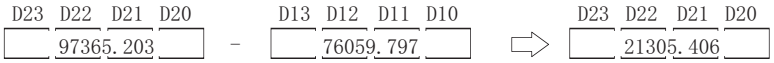
[梯形圖模式]



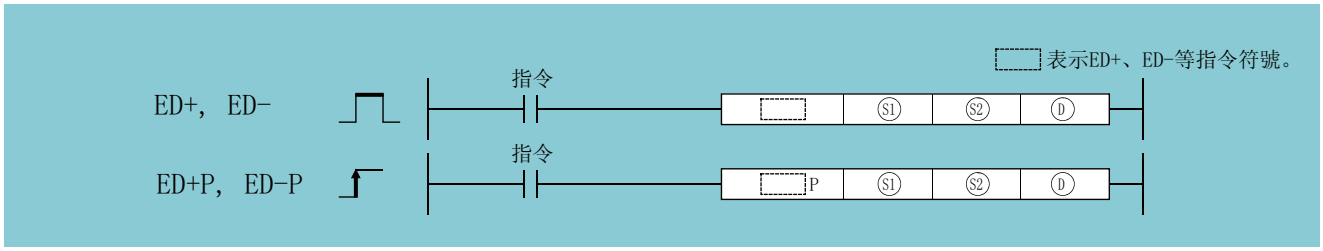
[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	ED-P	D10	D20
4	END		

[動作]



2 設置資料爲 3 個時 ((Ⓢ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 ： 被加數被減數資料或者存儲被加數被減數資料的軟元件的起始編號（實數）。

Ⓢ2 ： 加數減數資料或者存儲加數減數資料的軟元件的起始編號（實數）。

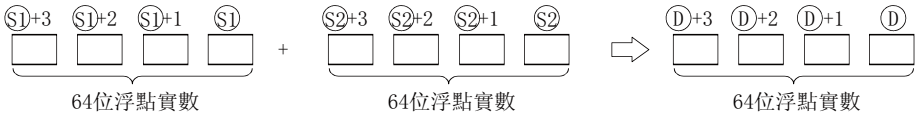
Ⓓ ： 存儲運算結果的軟元件的起始編號（實數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ1	---	○		---				○	---
Ⓢ2	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

ED+

(1) 將Ⓢ1中指定的 64 位浮點實數與Ⓢ2中指定的 64 位浮點實數進行加法運算，並將運算結果存儲到Ⓓ中指定的軟元件中。

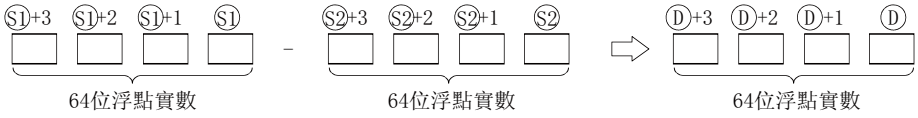


(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的值以及可存儲的值如下所示。

0、 $2^{-1022} \leq$ | 指定值（存儲值） | $< 2^{1024}$

ED-

(1) 將Ⓢ1中指定的 64 位浮點實數與Ⓢ2中指定的 64 位浮點實數進行減法運算，並將減法運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ1、Ⓢ2、Ⓓ中可指定的值以及可存儲的值如下所示。

0、 $2^{-1022} \leq$ | 指定值（存儲值） | $< 2^{1024}$

出　錯

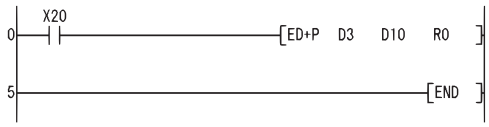
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定的軟元件的內容不在以下範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定的軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出以下範圍時。(發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D3 ～ D6 的 64 位浮點實數與 D10 ～ D13 的 64 位浮點實數進行加法運算，並將加法運算結果存儲到 R0 ～ R3 中的程式。

[梯形圖模式]



[列表模式]

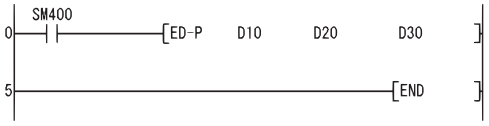
步	指令	軟元件		
0	LD	X20		
1	ED+P	D3	D10	R0
5	END			

[動作]

D6	D5	D4	D3		D13	D12	D11	D10		R3	R2	R1	R0
	5961.	437		+		12003.	200		⇨		17964.	637	

(2) 以下爲將 D10 ～ D13 的 64 位浮點實數與 D20 ～ D23 的 64 位浮點實數進行減法運算，並將減法運算結果存儲到 D30 ～ D33 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	SM400		
1	ED-P	D10	D20	D30
5	END			

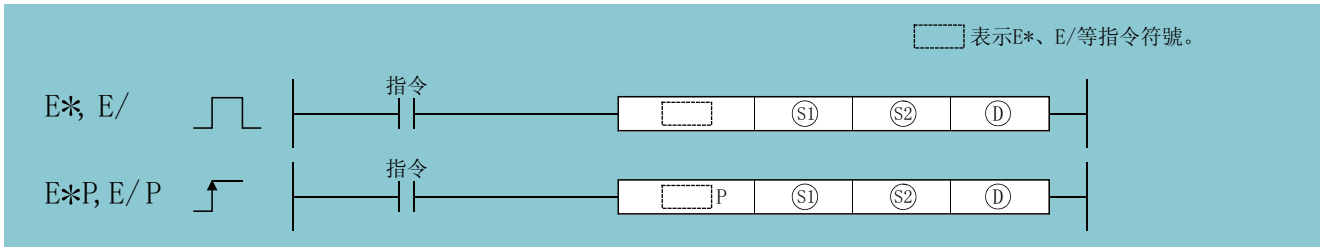
[動作]

D13	D12	D11	D10		D23	D22	D21	D20		D33	D32	D31	D30
	97365.	203		−		76059.	797		⇨		21305.	406	

6.2.11 E*、E*P、E/、E/P

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。



S1 ： 被乘數被除數資料或者存儲被乘數被除數資料的軟元件的起始編號 (實數)。

S2 ： 乘數除數資料或者存儲乘數除數資料的軟元件的起始編號 (實數)。

D ： 存儲運算結果的軟元件的起始編號 (實數)。

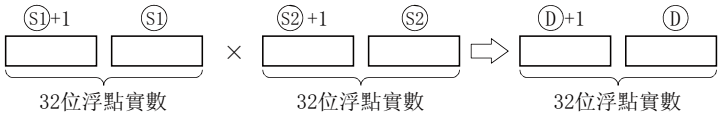
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
S1	---	○		---	○		○ *1	○	---
S2	---	○		---	○		○ *1	○	---
D	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

E*

(1) 將S1中指定的 32 位浮點實數與S2中指定的 32 位浮點實數進行乘法運算，並將乘法運算結果存儲到D中指定的軟元件中。

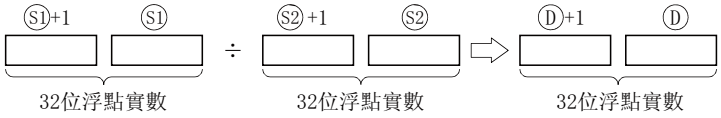


(2) S1、S2、D中可指定的值以及可存儲的值如下所示。

0、$2^{-126} \leq$ | 指定值 (存儲值) | $< 2^{128}$

E/

(1) 將S1中指定的 32 位浮點實數與S2中指定的 32 位浮點實數進行除法運算，並將除法運算結果存儲到D中指定的軟元件中。



(2) S1、S2、D中可指定的值以及可存儲的值如下所示。

0、$2^{-126} \leq$ | 指定值 (存儲值) | $< 2^{128}$

出　錯

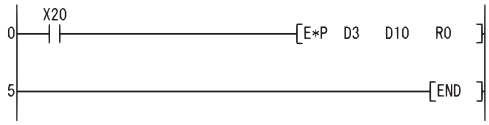
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定的軟元件的內容不在以下範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定的軟元件的內容爲 -0 時。	○	○	○	○	---	---
	除數爲 0 時。	○	○	○	○	○	---
4141	運算結果超出以下範圍時。 (發生了溢出時。) $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○
4140	指定軟元件的內容爲 0、非正規數、非數、±∞時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D3、D4 的 32 位浮點實數與 D10、D11 的 32 位浮點實數進行乘法運算，並將乘法運算結果存儲到 R0、R1 中的程式。

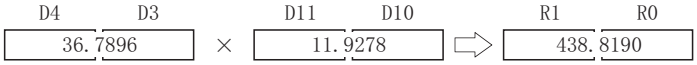
[梯形圖模式]



[列表模式]

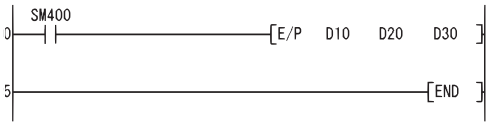
步	指令	軟元件		
0	LD	X20		
1	E*P	D3	D10	R0
5	END			

[動作]



(2) 以下爲將 D10、D11 的 32 位浮點實數與 D20、D21 的 32 位浮點實數進行除法運算，並將除法運算結果存儲到 D30、D31 中的程式。

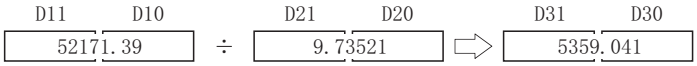
[梯形圖模式]



[列表模式]

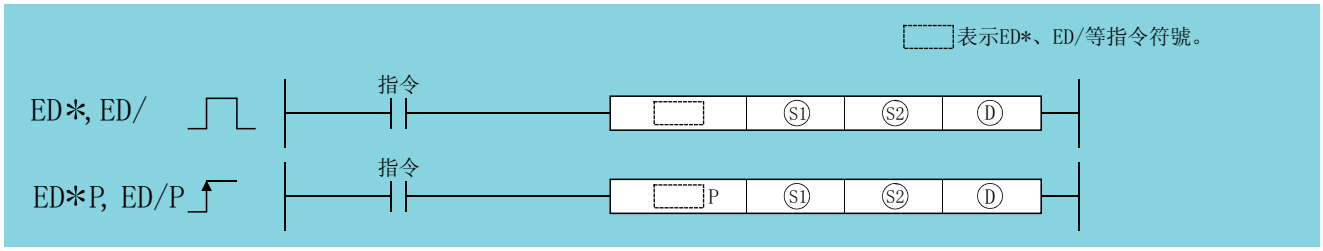
步	指令	軟元件		
0	LD	SM400		
1	E/P	D10	D20	D30
5	END			

[動作]



6.2.12 ED*、ED*P、ED/、ED/P

Basic High performance Process Redundant Universal LCPU



ⓢ1 ： 被乘數被除數資料或者存儲被乘數被除數資料的軟元件的起始編號（實數）。

ⓢ2 ： 乘數除數資料或者存儲乘數除數資料的軟元件的起始編號（實數）。

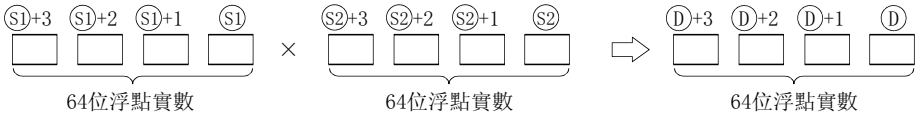
Ⓓ ： 存儲運算結果的軟元件的起始編號（實數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
ⓢ1	---	○		---				○	---
ⓢ2	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

ED*

(1) 將ⓢ1中指定的 64 位浮點實數與ⓢ2中指定的 64 位浮點實數進行乘法運算，並將乘法運算結果存儲到Ⓓ中指定的軟元件中。



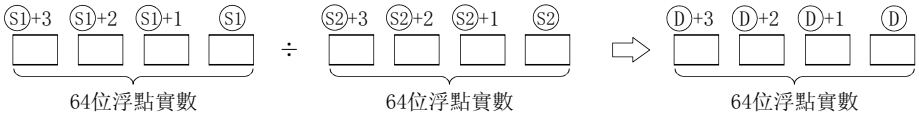
(2) ⓢ1、ⓢ2、Ⓓ中可指定的值以及可存儲的值如下所示。

0、$2^{-1022} \leq$ | 指定值（存儲值） | $< 2^{1024}$

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 處理。

ED/

(1) 將ⓢ1中指定的 64 位浮點實數與ⓢ2中指定的 64 位浮點實數進行除法運算，並將除法運算結果存儲到Ⓓ中指定的軟元件中。



(2) ⓢ1、ⓢ2、Ⓓ中可指定的值以及可存儲的值如下所示。

0、$2^{-1022} \leq$ | 指定值（存儲值） | $< 2^{1024}$

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 處理。

出　錯

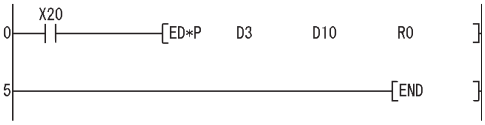
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定的軟元件的內容不在以下範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定的軟元件的內容爲 -0 時。	---	---	---	---	○	○
4100	除數爲 0 時。	---	---	---	---	○	○
4141	運算結果超出以下範圍時。(發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D3 ～ D6 的 64 位浮點實數與 D10 ～ D13 的 64 位浮點實數進行乘法運算，並將乘法運算結果存儲到 R0 ～ R3 中的程式。

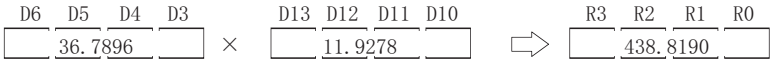
[梯形圖模式]



[列表模式]

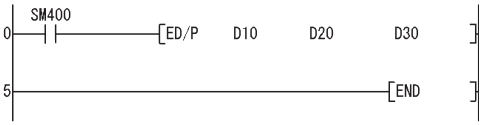
步	指令	軟元件		
0	LD	X20		
1	ED*P	D3	D10	R0
5	END			

[動作]



(2) 以下爲將 D10 ～ D13 的 64 位浮點實數與 D20 ～ D23 的 64 位浮點實數進行除法運算，並將除法運算結果存儲到 D30 ～ D33 中的程式。

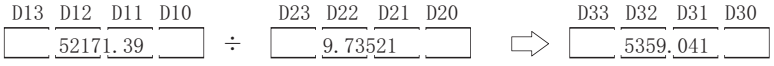
[梯形圖模式]



[列表模式]

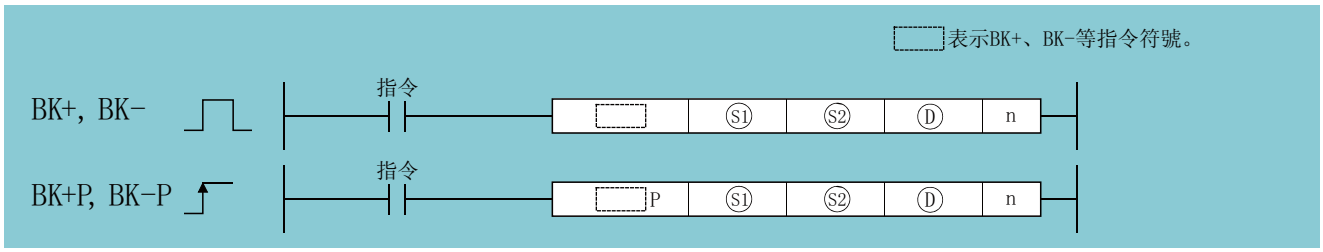
步	指令	軟元件		
0	LD	SM400		
1	ED/P	D10	D20	D30
5	END			

[動作]



6.2.13 BK+、BK+P、BK-、BK-P

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ1 ： 被加數被減數資料或者存儲被加數被減數資料的軟元件的起始編號 (BIN16 位)。

Ⓢ2 ： 加數減數資料或者存儲加數減數資料的軟元件的起始編號 (BIN16 位)。

Ⓓ ： 存儲運算結果的軟元件的起始編號 (BIN16 位)。

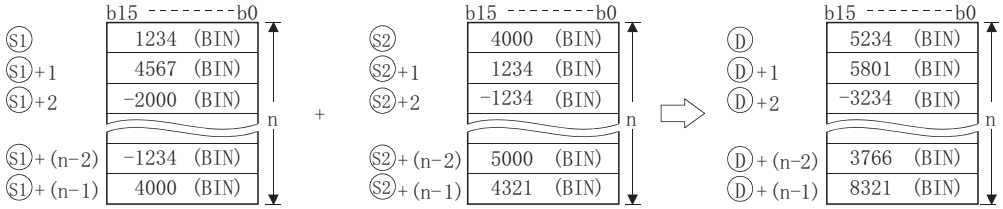
n ： 加法減法運算資料個數 (BIN16 位元)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	---	○		---				---	---
Ⓢ2	---	○		---				○	---
Ⓓ	---	○		---				---	---
n	○	○		○				○	---

功　能

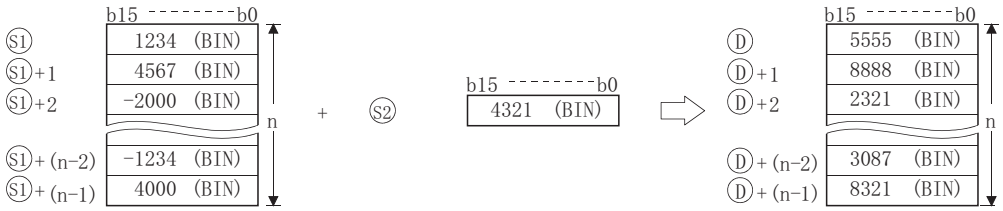
BK+

(1) 將從Ⓢ1中指定的軟元件開始的 n 點的 BIN 資料與Ⓢ2中指定的軟元件開始的 n 點的 BIN 資料進行加法運算，並將運算結果存儲到Ⓓ中指定的軟元件後面。



(2) 塊加法運算是以 16 位元爲單位進行。

(3) Ⓢ2中可指定的常數範圍爲 -32768 ～ 32767(BIN16 位)。

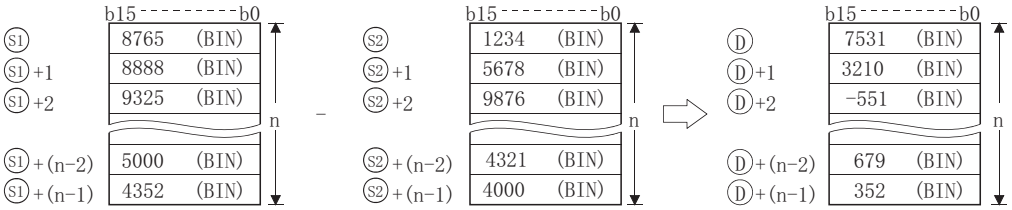


(4) 運算結果中發生了下溢或者上溢時的情況如下所示。
在這種情況下，進位元標誌不變爲 ON。

- K32767 (7FFFH) +K2 (0002H) ⟶ K-32767 (8001H)
- K-32767 (8001H) +K-2 (FFFEH) ⟶ K32767 (7FFFH)

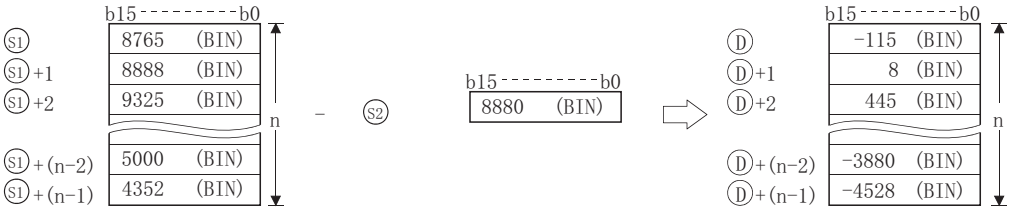
BK-

(1) 將從Ⓢ1中指定的軟元件開始的 n 點的 BIN 資料與Ⓢ2中指定的軟元件開始的 n 點的 BIN 資料進行減法運算，並將運算結果存儲到Ⓓ中指定的軟元件後面。



(2) 塊減法運算是以 16 位元爲單位進行。

(3) Ⓢ2中可指定的常數範圍爲 -32768 ～ 32767(BIN16 位)。



(4) 運算結果中發生了下溢或者上溢時的情況如下所示。
在這種情況下，進位元標誌不變爲 ON。

- K-32768 (8000H) −K2 (0002H) ⟶ K32766 (7FFEH)
- K32767 (7FFFH) −K-2 (FFFEH) ⟶ -32767 (8001H)

出　錯

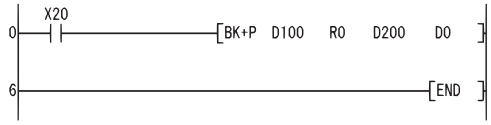
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓢ1、Ⓢ2、Ⓓ 的軟元件開始的 n 點的範圍超出了相應軟元件範圍時。 從Ⓢ1開始的 n 點的軟元件範圍與從Ⓓ 開始的 n 點的軟元件範圍相重復時。(Ⓢ1與Ⓓ 中指定了同一個軟元件時除外。) 從Ⓢ2開始的 n 點的軟元件範圍與從Ⓓ 開始的 n 點的軟元件範圍相重復時。(Ⓢ2與Ⓓ 中指定了同一個軟元件時除外。)	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D100 ～ D103 中存儲的資料值與 R0 ～ R3 中存儲的資料值進行加法運算，並將其結果存儲到 D200 後面的程式。

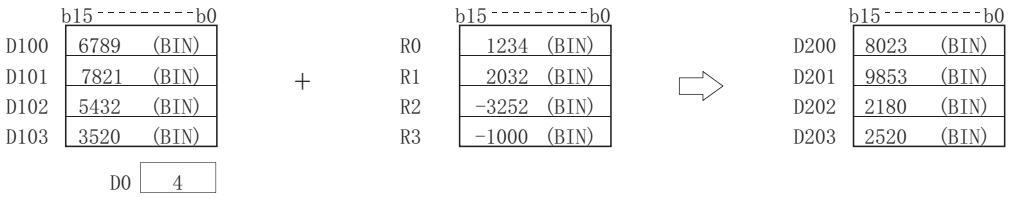
[梯形圖模式]



[列表模式]

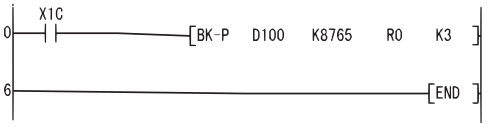
步	指令	軟元件
0	LD	X20
1	BK+P	D100 R0 D200 D0
6	END	

[動作]



(2) 以下爲 X1C 變爲 ON 時，將常數 D100 ～ D102 的資料與常數 8765 進行減法運算，並將其結果存儲到 R0 後面的程式。

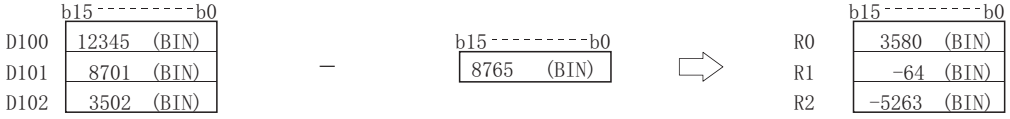
[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X1C			
1	BK-P	D100	K8765	R0	K3
6	END				

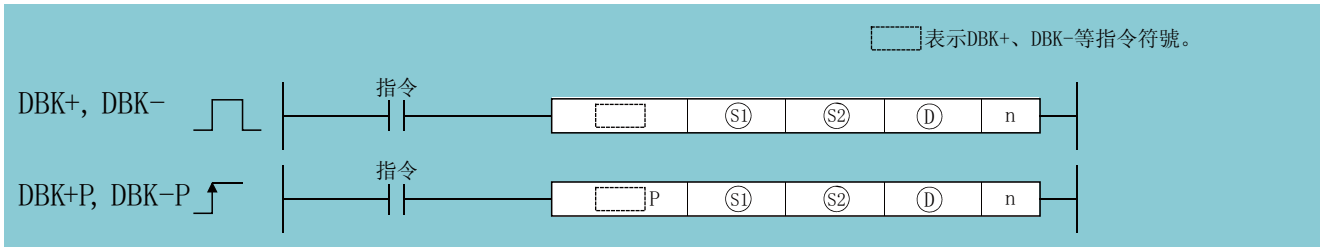
[動作]



Basic　High performance　Process　Redundant　Universal　LCPU

6.2.14 DBK+、DBK+P、DBK-、DBK-P

·在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。



Ⓢ1 ： 被加數被減數資料或者存儲被加數被減數資料的軟元件的起始編號 (BIN32 位)。

Ⓢ2 ： 加數減數資料或者存儲加數減數資料的軟元件的起始編號 (BIN32 位)。

Ⓓ ： 存儲運算結果的軟元件的起始編號 (BIN32 位)。

n ： 加法減法運算資料個數 (BIN16 位元)

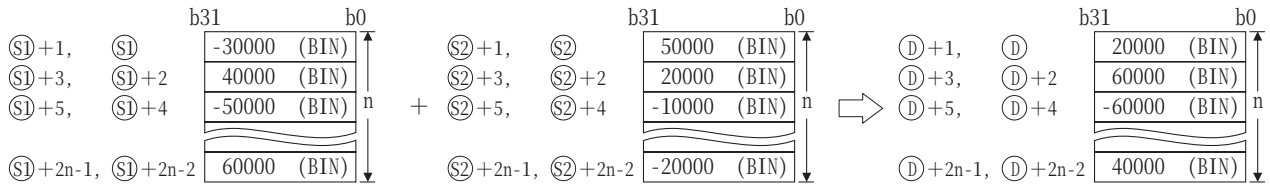
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	---	○		---				---	---
Ⓢ2	---	○		---				○	---
Ⓓ	---	○		---				---	---
n	---	○		○				○	---

功　能

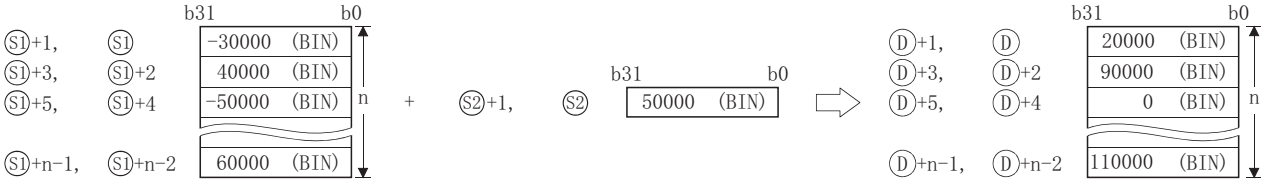
DBK+

(1) 將從Ⓢ1中指定的軟元件開始的 n 點的 BIN32 位元資料與Ⓢ2中指定的軟元件開始的 n 點的 BIN32 位元資料或者常數進行加法運算，並將運算結果存儲到Ⓓ中指定的軟元件後面。

Ⓢ2中指定了軟元件時



Ⓢ2中指定了常數時



(2) 塊加法運算是以 32 位元爲單位進行。

(3) Ⓢ2中可指定的常數範圍爲 -2147483648 ～ 2147483647(BIN32 位)。

(4) n 中指定的值爲 0 時將變爲無處理。

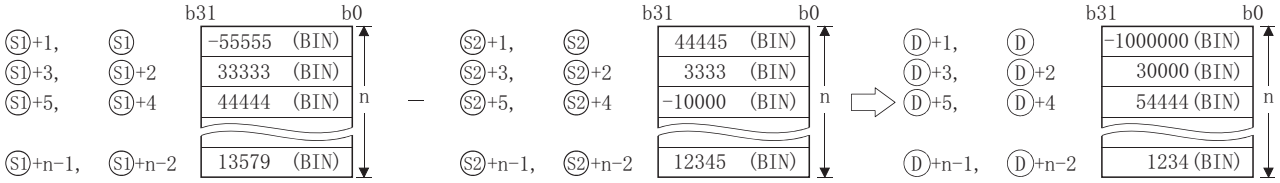
(5) 運算結果中發生了溢出時的情況如下所示。
此時，進位元標誌不變爲 ON。

- K2147483647 (7FFFFFFFH) +K2 (00000002H) → K-2147483647 (80000001H)
- K-2147483647 (80000001H) +K-2 (FFFFFFFEH) → K2147483647 (7FFFFFFFH)

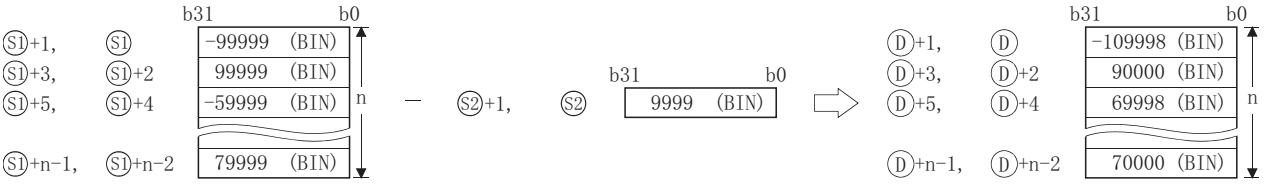
DBK-

(1) 將從Ⓢ1中指定的軟元件開始的 n 點的 BIN32 位元資料與Ⓢ2中指定的軟元件開始的 n 點的 BIN32 位元資料或者常數進行減法運算，並將運算結果存儲到Ⓓ中指定的軟元件後面。

Ⓢ2中指定了軟元件時



Ⓢ2中指定了常數時



(2) 塊減法運算是以 32 位元爲單位進行。

(3) Ⓢ2中可指定的常數範圍爲 -2147483648 ～ 2147483647(BIN32 位)。

(4) n 中指定的值爲 0 時將變爲無處理。

(5) Ⓓ的指定應在從Ⓢ1開始的 n 點的軟元件範圍及從Ⓢ2開始的 n 點的軟元件範圍以外。
但是，與Ⓢ1或者Ⓢ2指定了同一個軟元件時除外。

(6) 運算結果中發生了溢出時的情況如下所示。
此時，進位元標誌不變爲 ON。

- K2147483647 (7FFFFFFFH) -K - 2 (00000002H) → K-2147483647 (8000001H)
- K-2147483647 (80000001H) -K 2 (FFFFFFFEH) → K2147483647 (7FFFFFFFH)

出　錯

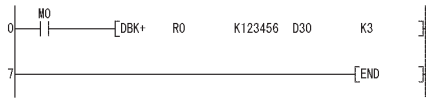
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 中指定了負值時。	○	○	○	○	○	○
4101	從(S1)、(S2)、(D) 中指定的軟元件開始的 n 點的範圍超出了相應軟元件的範圍時。 從(S1)開始的 n 點的軟元件範圍與從(D) 開始的 n 點的軟元件範圍重復時。 ((S1)與(D) 中指定了同一個軟元件時除外。) 從(S2)開始的 n 點的軟元件範圍與從(D) 開始的 n 點的軟元件範圍重復時。 ((S2)與(D) 中指定了同一個軟元件時除外。)	○	○	○	○	○	○

程式示例

(1)　以下爲 M0 變爲 ON 時，將 R0 ～ R5 中存儲的值與常數進行加法運算，並將運算結果存儲到 D30 ～ D35 中的程式。

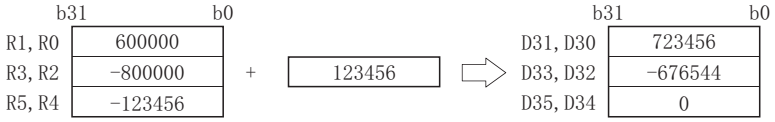
[梯形圖模式]



[列表模式]

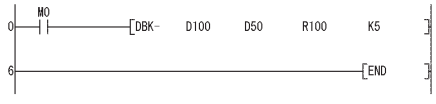
步	指令	軟元件			
0	LD	M0			
1	DBK+	R0	K123456	D30	K3
7	END				

[動作]



(2)　以下爲 M0 變爲 ON 時，將 D100 ～ D109 中存儲的值與 D50 ～ D59 中存儲的值進行減法運算，並將運算結果存儲到 R100 ～ R109 中的程式。

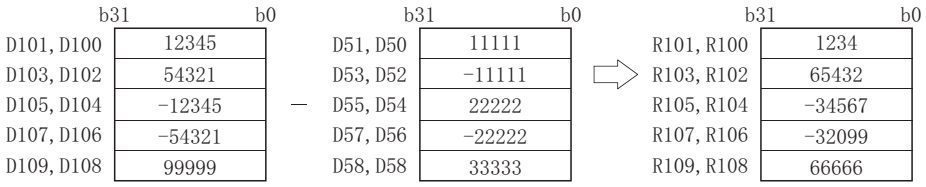
[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	M0			
1	DBK−	D100	D50	R100	K5
6	END				

[動作]



6.2.15 $+、$+P

Basic (×) | High performance | Process | Redundant | Universal | LCPU

1 設置資料爲 2 個時 (Ⓓ+Ⓢ→Ⓓ)

Ⓢ ：合併的資料或者存儲合併資料的軟元件的起始編號 (字串)。

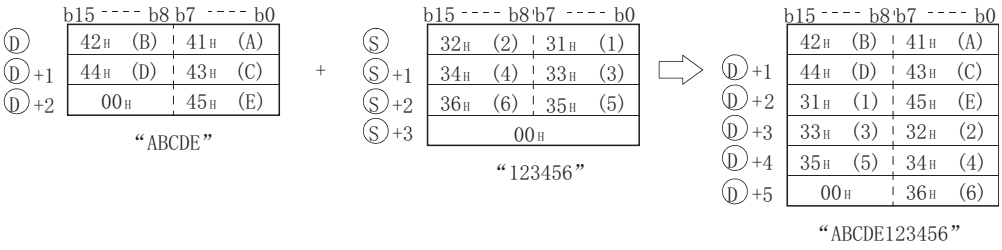
Ⓓ ：存儲合併資料的軟元件的起始編號 (字串)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 將Ⓓ中指定的字串資料連接到Ⓢ中指定的字串資料的後面，並將結果存儲到Ⓓ中指定編號的軟元件的後面。

字串資料是以從Ⓓ、Ⓢ中指定的軟元件編號開始，至存儲了 “ 00_H ” 的軟元件編號爲止中存儲的字串資料爲物件。



(2) 字串合併時，表示Ⓓ中指定的字串的結束的 “ 00_H ” 將被視爲無效，Ⓓ中指定的字串將被連接到Ⓢ中字串的最後字元處。

出　錯

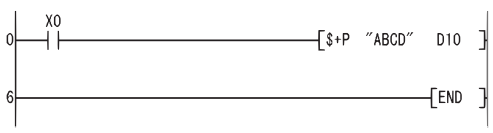
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓓ中指定的軟元件編號後面起，至相應軟元件的最終軟元件號爲止的點數不能容納合併後的字串時。 Ⓢ與Ⓓ中指定的存儲字串的軟元件編號重復時。 Ⓢ與Ⓓ中的字串超過了 16383 個字元時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D10 ～ D12 中存儲的字串與字串 “ ABCD ” 合併的程式。

[梯形圖模式]

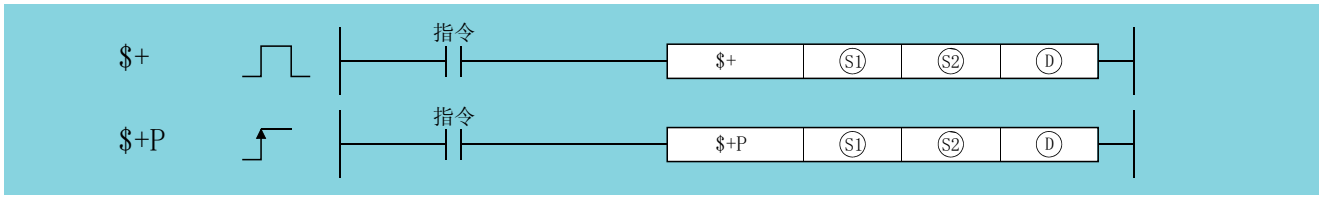


[列表模式]

步	指令	軟元件
0	LD	X0
1	$+P	"ABCD" D10
6	END	

[動作]

2 設置資料爲 3 個時 (Ⓢ1+Ⓢ2→Ⓓ)



Ⓢ1 ： 合併的資料或者存儲合併資料的軟元件的起始編號 (字串)。

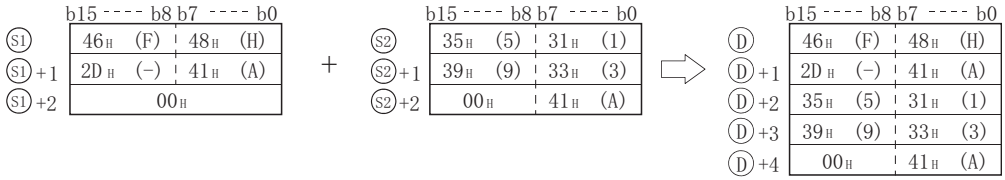
Ⓢ2 ： 被合併的資料或者存儲被合併資料的軟元件的起始編號 (字串)。

Ⓓ ： 儲合併結果的軟元件的起始編號 (字串)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ1	---	○		---				○	---
Ⓢ2	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 將Ⓢ1中指定的字串資料連接到Ⓢ2中指定的字串資料的後面，並將結果存儲到Ⓓ中指定編號的軟元件的後面。



(2) 字串合併時，表示Ⓢ1中指定的字串的結束的 “ 00_H ” 將被視爲無效，Ⓢ1中指定的字串將被連接到Ⓢ2中字串的最後字元處。

出　錯

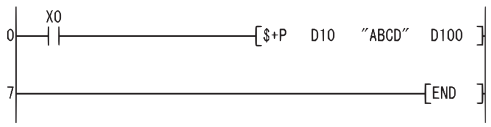
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓓ中指定的軟元件編號後面起，至相應軟元件的最終軟元件號爲止的點數不能容納合併後的字串時。 Ⓢ1與Ⓢ2中指定的軟元件編號重復時。 Ⓢ2與Ⓓ中指定的軟元件編號重復時。 Ⓢ1、Ⓢ2、Ⓓ中的字串超過了 16383 個字元時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D10 ～ D12 中存儲的字串與字串 “ ABCD ” 合併，並將結果存儲到 D100 後面的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X0	
1	$+P	D10	"ABCD" D100
7	END		

[動作]

6.2.16 INC、INCP、DEC、DECP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ : 執行 INC(+1)、DEC(-1) 的軟元件的起始編號 (BIN16 位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	○							---	

功　能

INC

(1) 對Ⓓ中指定的軟元件 (16 位元資料) 進行 +1。

(2) 在Ⓓ中指定的軟元件的值爲 32767 時如果執行了 INC、INCP，將在Ⓓ中指定的軟元件中存儲 -32768。

DEC

(1) 對Ⓓ中指定的軟元件 (16 位元資料) 進行 -1。

(2) 在Ⓓ中指定的軟元件的值爲 -32768 時如果執行了 DEC、DECP，將在Ⓓ中指定的軟元件中存儲 32767。

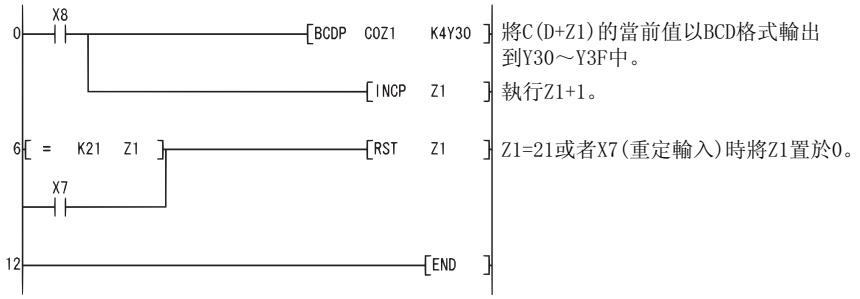
出　錯

(1) 在 INC(P)/DEC(P) 指令中無運算出錯。

程式示例

(1) 以下爲每當 X8 變爲 ON 時，將計數器 C0 ～ C20 的當前值以 BCD 格式輸出到 Y30 ～ Y3F 中的程式。
(當前值＜ 9999 時)

[梯形圖模式]

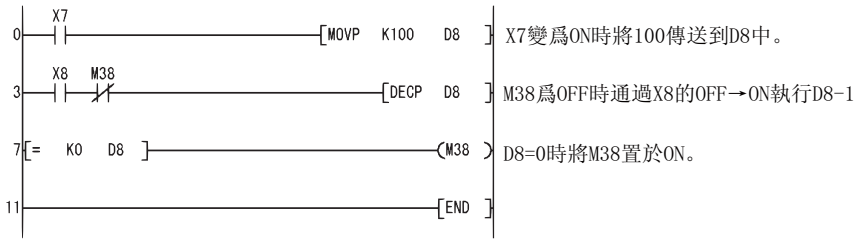


[列表模式]

步	指令	軟元件	
0	LD	X8	
1	BCDP	C0Z1	K4Y30
4	INCP	Z1	
6	LD=	K21	Z1
9	OR	X7	
10	RST	Z1	
12	END		

(2) 減法計數器的程式

[梯形圖模式]



[列表模式]

步	指令	軟元件	
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		

6.2.17 DINC、DINCP、DDEC、DDECP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：執行 DINC(+1)、DDEC(-1) 的軟元件的起始編號 (BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	○							---	

功　能

DINC

(1) 對Ⓓ中指定的軟元件 (32 位元資料) 進行 +1。

(2) 在Ⓓ中指定的軟元件的值爲 2147483647 時如果執行了 DINC、DINCP，將在Ⓓ中指定的軟元件中存儲 -2147483648。

DDEC

(1) 對Ⓓ中指定的軟元件 (32 位元資料) 進行 -1。

(2) 在Ⓓ中指定的軟元件的值爲 0 時如果執行了 DDEC、DDECP，將在Ⓓ中指定的軟元件中存儲 -1。

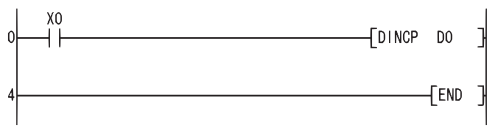
出　錯

(1) 在 DINC(P)/DDEC(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X0 變爲 ON 時，對 D0、D1 的資料進行 +1 的程式。

[梯形圖模式]

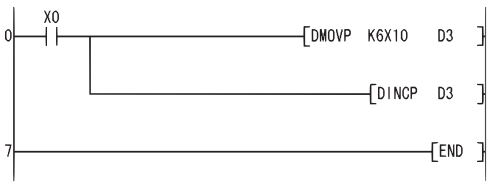


[列表模式]

步	指令	軟元件
0	LD	X0
1	DINCP	D0
4	END	

(2) 以下爲 X0 變爲 ON 時，對設置到 X10 ～ X27 中的資料進行 +1，並將結果存儲到 D3、D4 中的程式。

[梯形圖模式]

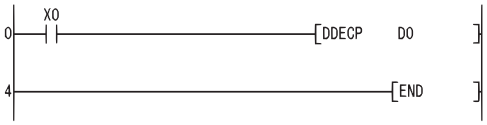


[列表模式]

步	指令	軟元件
0	LD	X0
1	DMOVP	K6X10 D3
4	DINCP	D3
7	END	

(3) 以下爲 X0 變爲 ON 時，對 D0、D1 的資料進行 -1 的程式。

[梯形圖模式]

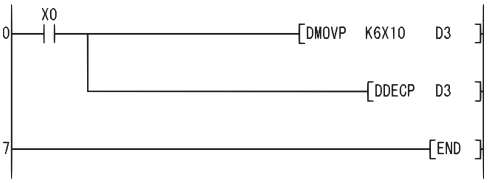


[列表模式]

步	指令	軟元件
0	LD	X0
1	DDECP	D0
4	END	

(4) 以下爲 X0 變爲 ON 時，對設置到 X10 ～ X27 中的資料進行 -1，並將結果存儲到 D3、D4 中的程式。

[梯形圖模式]



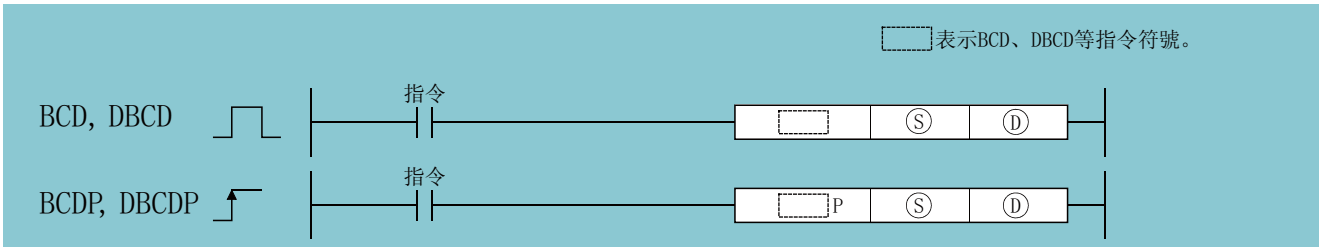
[列表模式]

步	指令	軟元件
0	LD	X0
1	DMOVP	K6X10 D3
4	DDECP	D3
7	END	

6.3 資料轉換指令

6.3.1 BCD、BCDP、DBCD、DBCDP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：BIN 資料或者存儲 BIN 資料的軟元件的起始編號 (BIN16/32 位)。

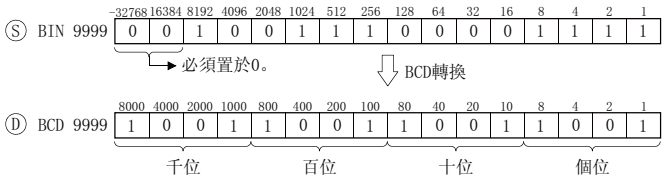
Ⓓ ：存儲 BCD 資料的軟元件的起始編號 (BCD4/8 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ				○				○	---
Ⓓ				○				---	---

功　能

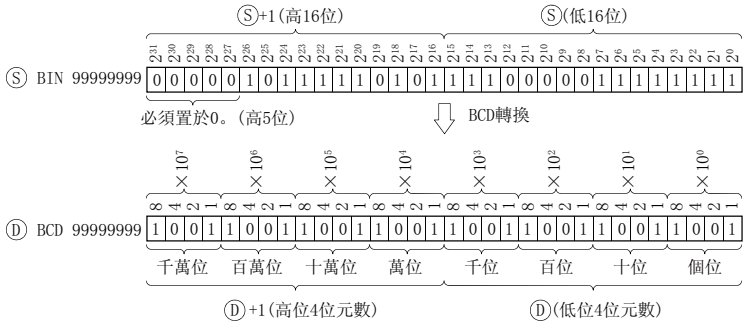
BCD

對Ⓢ中指定的軟元件的 BIN 資料 (0 ～ 9999) 進行 BCD 轉換後，存儲到Ⓓ中指定的軟元件中。



DBCD

對Ⓢ中指定的軟元件的 BIN 資料 (0 ～ 99999999) 進行 BCD 轉換後，存儲到Ⓓ中指定的軟元件中。



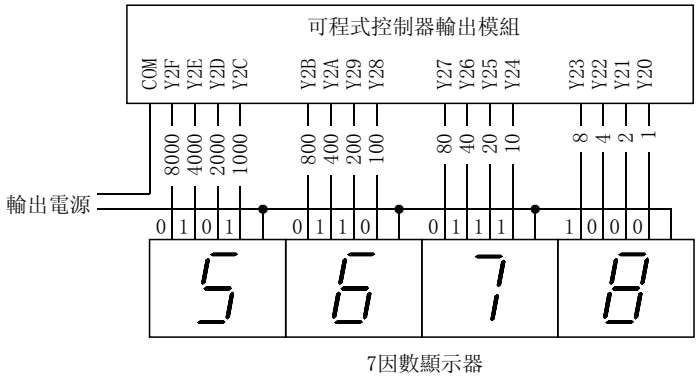
出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	BCD 指令時Ⓢ的資料超出了 0 ～ 9999 的範圍時。	○	○	○	○	○	○
4100	DBCD 指令時Ⓢ+1、Ⓢ的資料超出了 0 ～ 99999999 的範圍時。	○	○	○	○	○	○

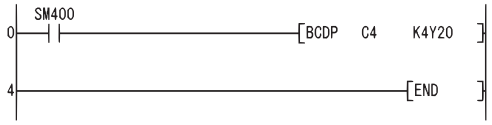
程式示例

(1) 以下爲將 C4 的當前值從 Y20 ～ Y2F 輸出到 BCD 顯示器中的程式。



7因數顯示器

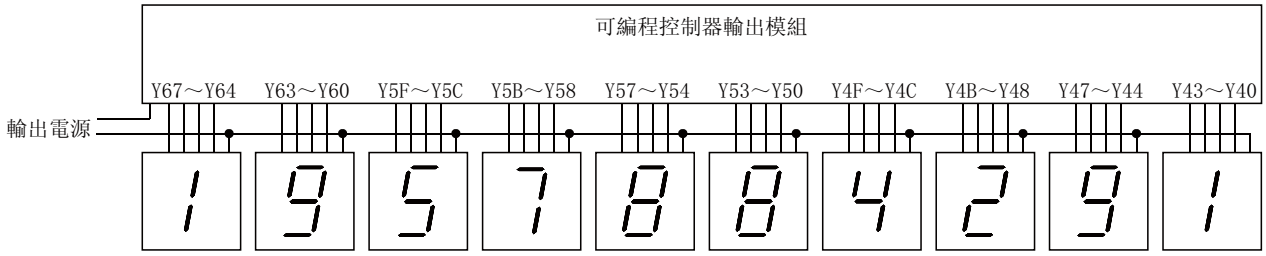
[梯形圖模式]



[列表模式]

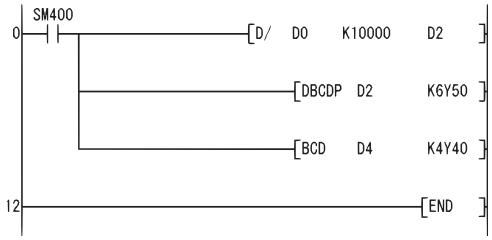
步	指令	軟元件	
0	LD	SM400	
1	BCDP	C4	K4Y20
4	END		

(2) 以下爲將 D0 ～ D1 的 32 位元資料輸出到 Y40 ～ Y67 中的程式。



7因數顯示器

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	SM400		
1	D/	D0	K10000	D2
6	DBCDP	D2	K6Y50	
9	BCD	D4	K4Y40	
12	END			

6.3.2 BIN、BINP、DBIN、DBINP

Basic | High performance | Process | Redundant | Universal | LCPU

□表示BIN、DBIN等指令符號。

Ⓢ ： BCD 資料或者存儲 BCD 資料的軟元件的起始編號 (BCD4/8 位)。

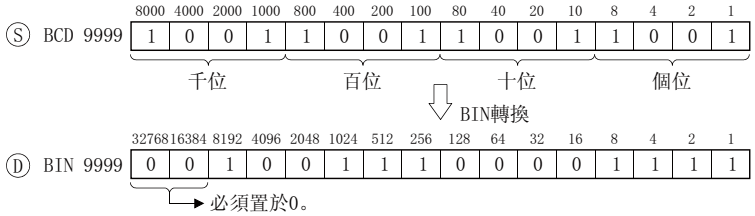
Ⓓ ： 存儲 BIN 資料的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓢ	○							○	---
Ⓓ	○							---	---

功　能

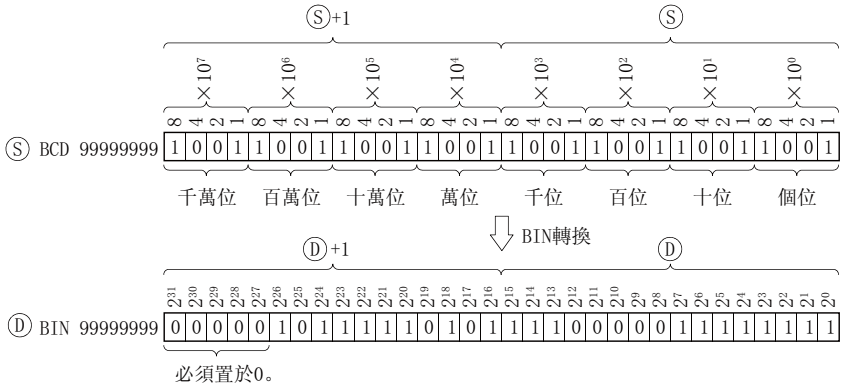
BIN

將Ⓢ中指定的軟元件的 BCD 資料 (0 ～ 9999) 轉換爲 BIN 後，存儲到Ⓓ中指定的軟元件中。



DBIN

對Ⓢ中指定的軟元件的 BCD 資料 (0 ～ 99999999) 進行 BIN 轉換後，存儲到Ⓓ中指定的軟元件中。



出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ的各位數中存在有超出 0 ～ 9 範圍的值時。	○	○	○	○	○	○

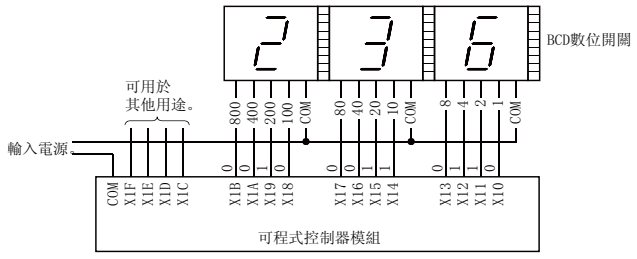
通過預先將 SMA722 置爲 ON，可以避免發生上述出錯。

此時，設置了超出範圍的數值時，與 SM722 的 ON/OFF 狀態無關，不執行指令。

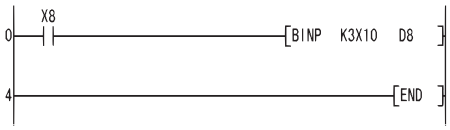
此外，BINP/DBINP 指令的情況下，與有無出錯無關，在對指令 (執行條件) 進行 OFF → ON 操作之前，不執行運算。

程式示例

(1) 以下爲 X8 變爲 ON 時，將 X10 ～ X1B 的 BCD 資料進行 BIN 轉換後，存儲到 D8 中的程式。



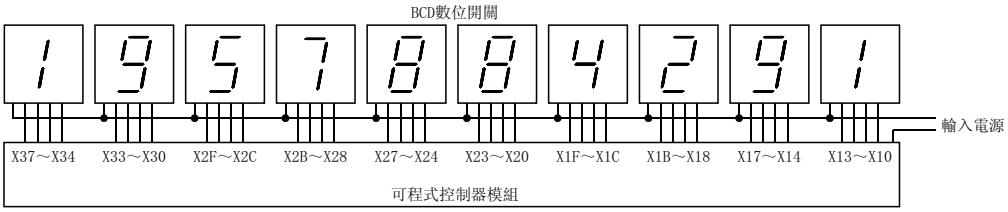
[梯形圖模式]



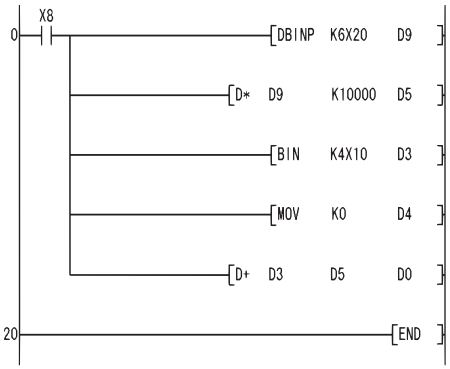
[列表模式]

步	指令	軟元件
0	LD	X8
1	BINP	K3X10 D8
4	END	

(2) 以下爲 X8 變爲 ON 時，將 X10 ～ X37 的 BCD 資料進行 BIN 轉換後，存儲到 D0、D1 中的程式。
(將 X20 ～ X37 的 BCD 資料進行了 BIN 轉換後的值與將 X10 ～ X1F 的 BCD 資料進行 BIN 轉換後的值進行加法運算。)



[梯形圖模式]



[列表模式]

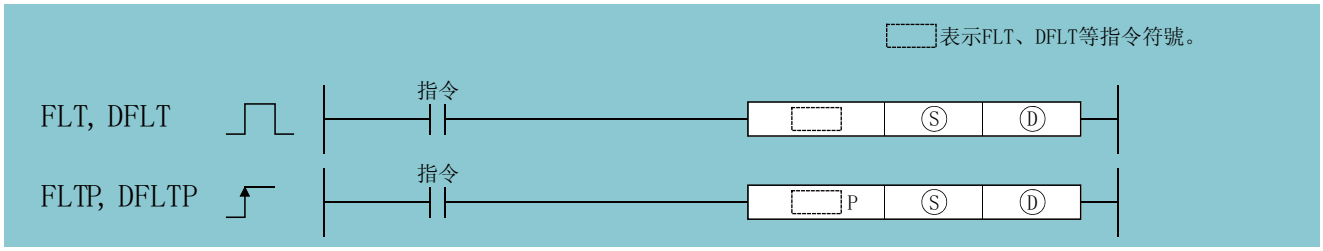
步	指令	軟元件		
0	LD	X8		
1	DBINP	K6X20	D9	
4	D*	D9	K10000	D5
9	BIN	K4X10	D3	
12	MOV	K0	D4	
14	D+	D3	D5	D0
20	END			

在 X10 ～ X37 中設置了超過 2147483647 的 BCD 值的情況下，由於超出了 32 位軟元件的數值處理範圍，因此 D0、D1 中的值將變爲負數。

6.3.3 FLT、FLTP、DFLT、DFLTP

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 基本型 QCPU：序列號的前 5 位數爲 “ 04122 ” 以後



Ⓢ ：轉換爲 32 位元浮點數據的整數或者存儲整數資料的起始軟元件號 (BIN16/32 位)。

Ⓓ ：存儲轉換爲 32 位元浮點數據的起始軟元件號 (實數)。

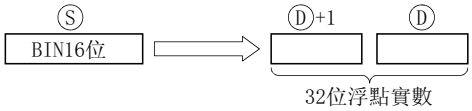
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○		○	○		○	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

FLT

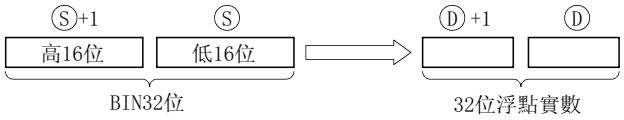
(1) 將Ⓢ中指定的 BIN16 位元資料轉換爲 32 位元浮點實數後，存儲到Ⓓ中指定編號的軟元件中。



(2) Ⓢ中指定的值爲 BIN 值且在 -32768 ～ 32767 的範圍內。

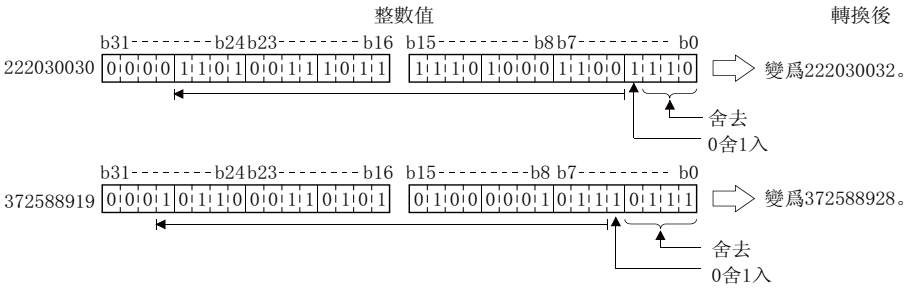
DFLT

(1) 將Ⓢ中指定的 BIN32 位元資料轉換爲 32 位元浮點實數後，存儲到Ⓓ中指定編號的軟元件中。



(2) Ⓢ+1、Ⓢ中指定的值爲 BIN 值且在 -2147483648 ～ 2147483647 的範圍內。

(3) 由於 32 位浮點實數是以 32 位單精確度進行處理，2 進制數表示時有效位數爲 24 位，10 進制數表示時有效位數約爲 7 位數。
因此，整數值超出了 -16777216 ～ 16777215(24 位 BIN 值) 的範圍時，轉換的值中將產生誤差。
將轉換結果從整數值的高位起第 25 位元進行 0 舍 1 入，第 26 位以後將被舍去。



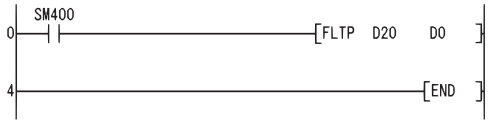
出　錯

(1) FLT(P)、DFLT(P) 指令中無運算出錯。

程式示例

(1) 以下爲將 D20 的 BIN16 位元資料轉換爲 32 位元浮點實數後，存儲到 D0、D1 中的程式。

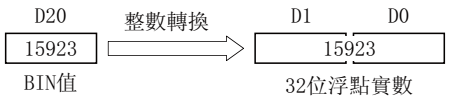
[梯形圖模式]



[列表模式]

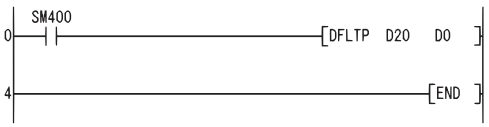
步	指令	軟元件	
0	LD	SM400	
1	FLTP	D20	D0
4	END		

[動作]



(2) 以下爲將 D20、D21 的 BIN32 位元資料轉換爲 32 位元浮點實數後，存儲到 D0、D1 中的程式。

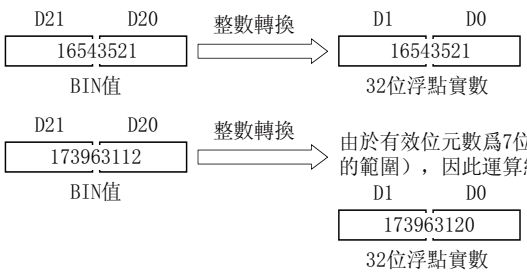
[梯形圖模式]



[列表模式]

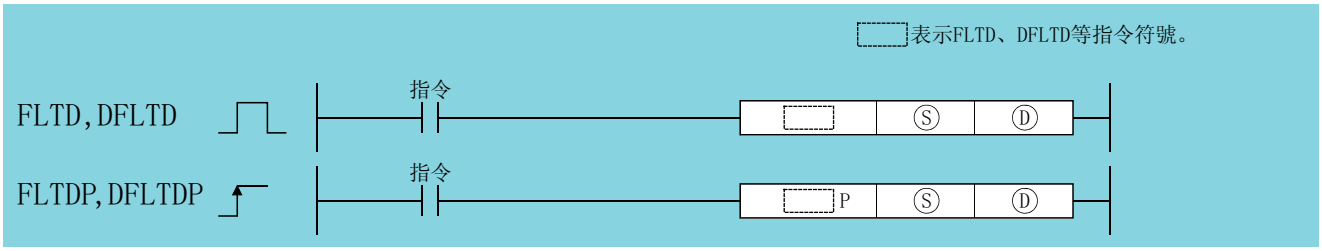
步	指令	軟元件	
0	LD	SM400	
1	DFLTP	D20	D0
4	END		

[動作]



6.3.4 FLTD、FLTDP、DFLTD、DFLTDP

Basic High performance Process Redundant Universal LCPU



Ⓢ ：轉換爲 64 位元浮點數據的整數或者存儲整數資料的起始軟元件號 (BIN16/32 位)。

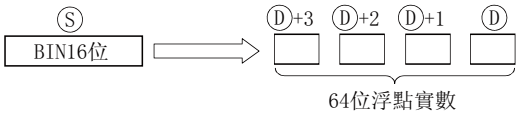
Ⓓ ：存儲轉換爲 64 位元浮點數據的起始軟元件號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---			○		---
Ⓓ	---	○		---			---		---

功　能

FLTD

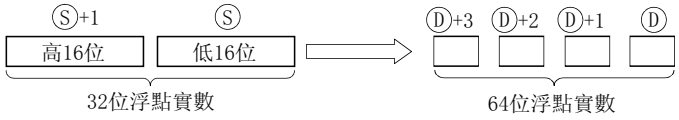
(1) 將Ⓢ中指定的 BIN16 位元資料轉換爲 64 位元浮點實數後，存儲到Ⓓ中指定編號的軟元件中。



(2) Ⓢ中指定的值爲 BIN 值且在 -32768 ～ 32767 的範圍內。

DFLTD

(1) 將Ⓢ中指定的 BIN32 位元資料轉換爲 64 位元浮點實數後，存儲到中指定編號的軟元件中。



(2) Ⓢ+1、Ⓢ中指定的值爲 BIN 值且在 -2147483648 ～ 2147483647 的範圍內。

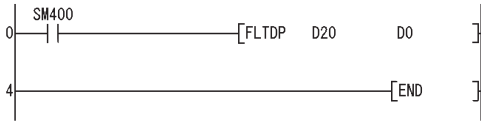
出　錯

(1) 在 FLTD(P)，DFLTD(P) 指令中沒有運算出錯。

程式示例

(1) 以下爲將 D20 的 BIN16 位元資料轉換爲 64 位元浮點實數後，存儲到 D0 ～ D3 中的程式。

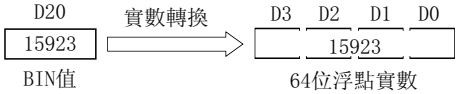
[梯形圖模式]



[列表模式]

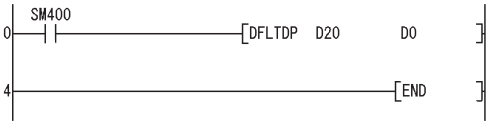
步	指令	軟元件	
0	LD	SM400	
1	FLTDP	D20	D0
4	END		

[動作]



(2) 以下爲將 D20、D21 的 BIN32 位元資料轉換爲 64 位元浮點實數後，存儲到 D0 ～ D3 中的程式。

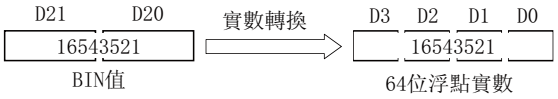
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	DFLTDP	D20	D0
4	END		

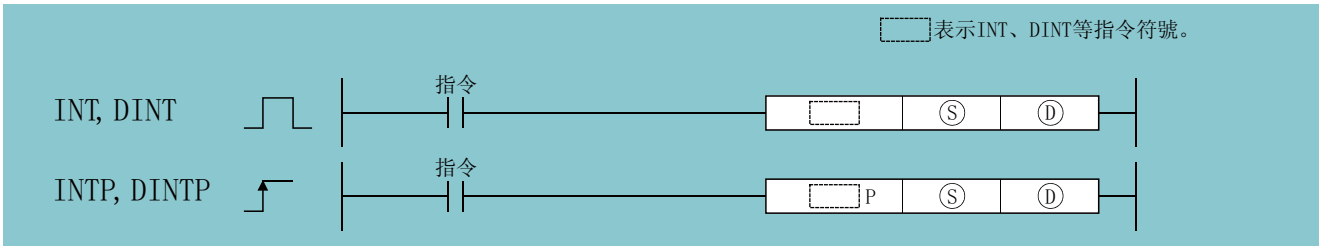
[動作]



Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

6.3.5 INT、INTP、DINT、DINTP



Ⓢ ： 要轉換爲 BIN 值的 32 位元浮點數據或者存儲浮點數據的起始軟元件號 (實數)。

Ⓓ ： 存儲轉換後的 BIN 值的起始軟元件號 (BIN16/32 位)。

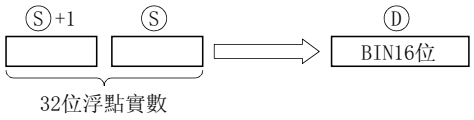
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	○	○		○	○		○	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

INT

(1) 將Ⓢ中指定的 32 位浮點實數轉換爲 BIN16 位元資料後，存儲到Ⓓ中指定編號的軟元件中。



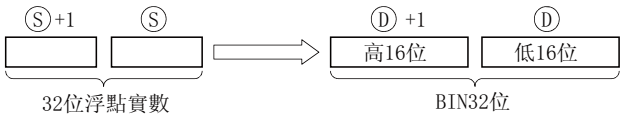
(2) Ⓢ+1、Ⓢ中指定的 32 位浮點實數的可指定範圍爲 -32768 ～ 32767。

(3) Ⓓ中存儲的整數值是以 BIN16 位元格式存儲的。

(4) 轉換後的實數資料的小數點以下第 1 位被四捨五入。

DINT

(1) 將Ⓢ中指定的 32 位浮點實數轉換爲 BIN32 位元資料後，存儲到Ⓓ中指定編號的軟元件中。



(2) Ⓢ+1、Ⓢ中指定的 32 位浮點實數的可指定範圍爲 -2147483648 ～ 2147483647。

(3) Ⓓ+1、Ⓓ中存儲的整數值是以 BIN32 位元格式存儲的。

(4) 轉換後的實數資料的小數點以下第 1 位被四捨五入。

出　錯

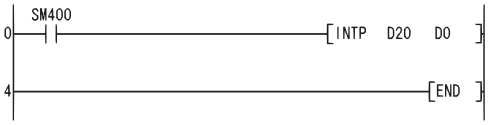
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容超出了下述範圍時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 0、非正規數、非數、±∞時。	---	---	---	---	○	○
4100	使用 INT 指令時，Ⓢ 中設置的 32 位元浮點型資料超出了 -32768 ～ 32767 的範圍時。	○	○	○	○	○	○
4100	使用 DINT 指令時，Ⓢ 中設置的 32 位元浮點型資料超出了 -2147483648 ～ 2147483647 的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲將 D20、D21 的 32 位浮點實數轉換爲 BIN16 位元資料後，存儲到 D0 中的程式。

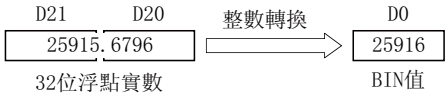
[梯形圖模式]

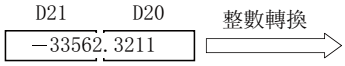


[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	INTP	D20	D0
4	END		

[動作]

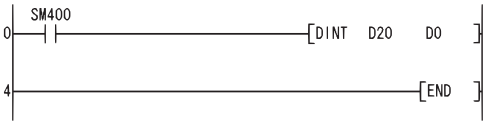




由於設置資料＜-32768，因此變爲運算出錯狀態。

(2) 以下爲將 D20、D21 的 32 位浮點實數轉換爲 BIN32 位元資料後，存儲到 D0、D1 中的程式。

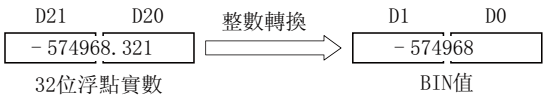
[梯形圖模式]

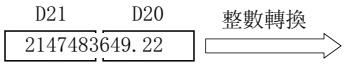


[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	DINT	D20	D0
4	END		

[動作]

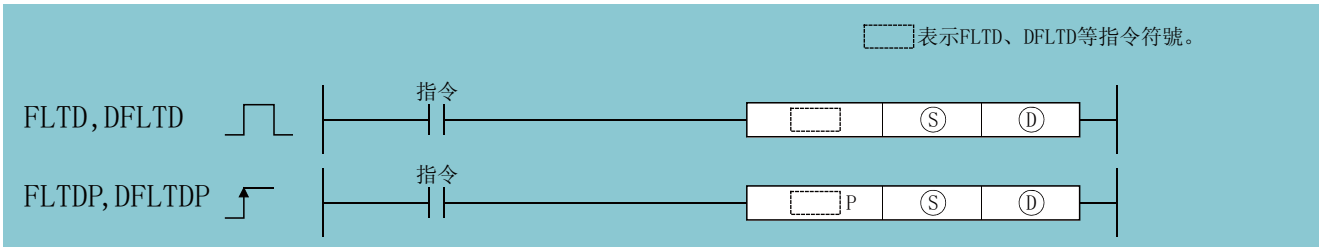




由於設置資料＞2147483647，因此變爲運算出錯狀態。

6.3.6 INTD、INTDP、DINTD、DINTDP

Basic High performance Process Redundant Universal LCPU



Ⓢ ：要轉換爲 BIN 值的 64 位元浮點實數資料或者存儲浮點實數資料的起始軟元件號 (實數)。

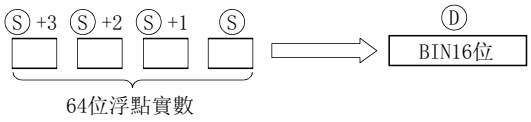
Ⓓ ：存儲轉換後的 BIN 值的起始軟元件號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---			---	○	---
Ⓓ	---	○		---			○	---	---

功　能

INTD

(1) 將Ⓢ中指定的 64 位浮點實數轉換爲 BIN16 位元資料後，存儲到Ⓓ中指定編號的軟元件中。



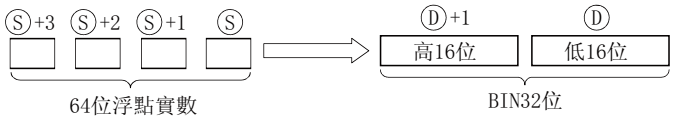
(2) Ⓢ+3、Ⓢ+2、Ⓢ+1、Ⓢ中指定的 64 位浮點實數的可指定範圍爲 -32768 ～ 32767。

(3) Ⓓ中存儲的整數值是以 BIN16 位元格式存儲的。

(4) 轉換後的 64 位元浮點實數資料的小數點以下第 1 位被四捨五入。

DINTD

(1) 將Ⓢ中指定的 64 位浮點實數轉換爲 BIN32 位元資料後，存儲到Ⓓ中指定編號的軟元件中。



(2) Ⓢ+3、Ⓢ+2、Ⓢ+1、Ⓢ中指定的 64 位浮點實數的可指定範圍爲 -2147483648 ～ 2147483647。

(3) Ⓓ+1、Ⓓ中存儲的整數值是以 BIN32 位元格式存儲的。

(4) 轉換後的 64 位元浮點實數資料的小數點以下第 1 位被四捨五入。

出　錯

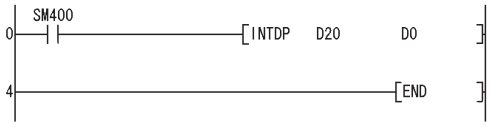
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容超出了下述範圍時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 0、非正規數、非數、±∞時。	---	---	---	---	○	○
4100	使用 INTD 指令時，中設置的 64 位元浮點型資料超出了 -32768 ～ 32767 的範圍時。	---	---	---	---	○	○
4100	使用 DINTD 指令時，Ⓢ 中設置的 64 位元浮點型資料超出了 -2147483648 ～ 2147483647 的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 D20 ～ D23 的 64 位浮點實數轉換爲 BIN16 位元資料後，存儲到 D0 中的程式。

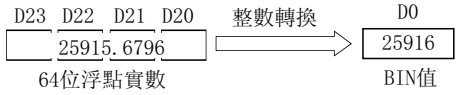
[梯形圖模式]

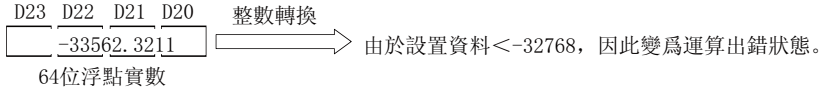


[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	INTDP	D20	D0
4	END		

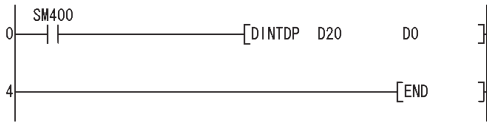
[動作]





(2) 以下爲將 D20 ～ D23 的 64 位浮點實數轉換爲 BIN32 位元資料後，存儲到 D0、D1 中的程式。

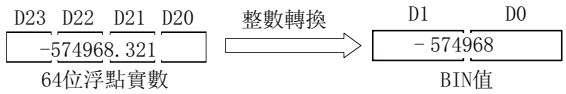
[梯形圖模式]

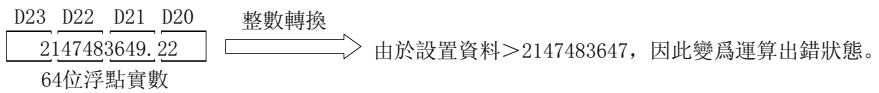


[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	DINTDP	D20	D0
4	END		

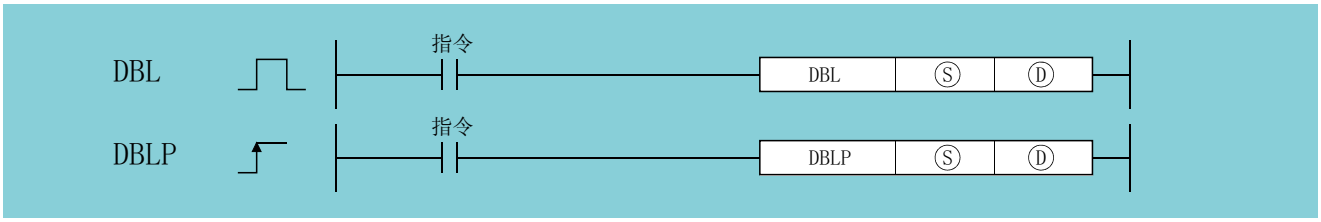
[動作]





6.3.7 DBL、DBLP

Basic | High performance | Process | Redundant | Universal | LCPU



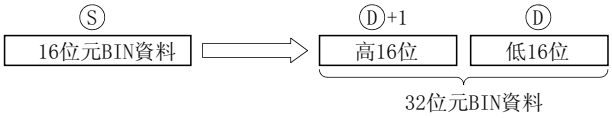
Ⓢ ：BIN16 位元資料或者存儲 BIN16 位元資料的起始軟元件號 (BIN16 位)。

Ⓓ ：存儲轉換後的 BIN32 位元資料的起始軟元件號 (BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ				○				○	---
Ⓓ				○				---	---

功　能

將Ⓢ中指定的 BIN16 位元資料轉換爲帶符號 BIN32 位元資料後，存儲到Ⓓ中指定編號的軟元件中。



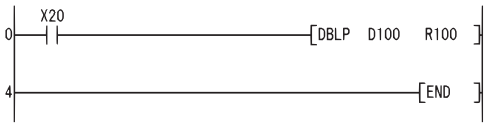
出　錯

(1) 在 DBL(P) 指令中無出錯。

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D100 的 BIN16 位元資料轉換爲 BIN32 位元資料後，存儲到 R100、R101 中的程式。

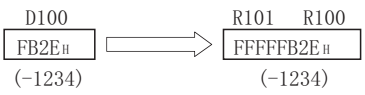
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X20	
1	DBLP	D100	R100
4	END		

[動作]



6.3.8 WORD、WORDP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：BIN32 位元資料或者存儲 BIN32 位元資料的起始軟元件號 (BIN32 位)。

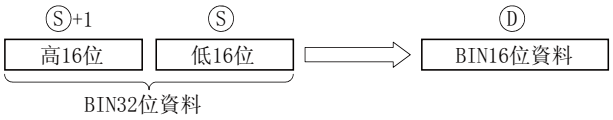
Ⓓ ：存儲轉換後的 BIN16 位元資料的起始軟元件號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ				○				○	---
Ⓓ				○				---	---

功　能

將Ⓢ中指定的 BIN32 位元資料轉換爲帶符號 BIN16 位元資料後，存儲到Ⓓ中指定編號的軟元件中。

可指定的範圍爲 -32768 ～ 32767。



出　錯

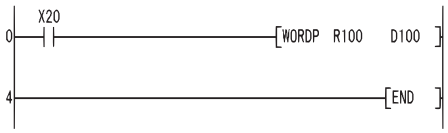
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ+1、Ⓢ中指定的軟元件的內容超出了 -32768 ～ 32767 的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 R100、R101 的 BIN32 位元資料轉換爲 BIN16 位元資料後，存儲到 D100 中的程式。

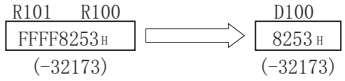
[梯形圖模式]



[列表模式]

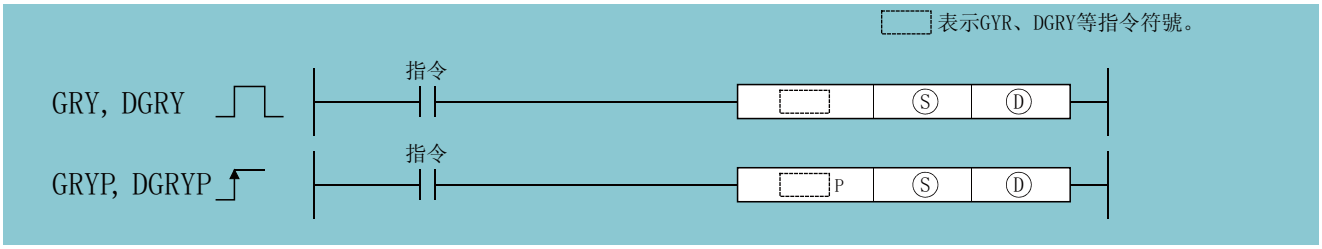
步	指令	軟元件	
0	LD	X20	
1	WORDP	R100	D100
4	END		

[動作]



6.3.9 GRY、GRYP、DGRY、DGRYP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：BIN 資料或者存儲 BIN 資料的起始軟元件號 (BIN16/32 位)。

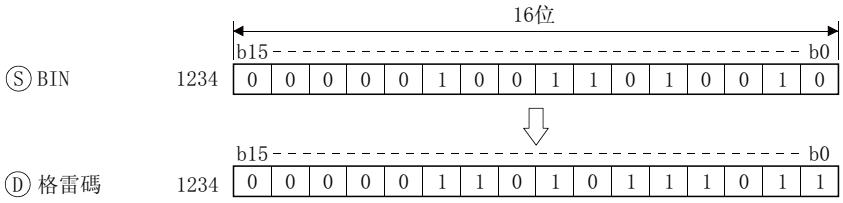
Ⓓ ：存儲轉換後的格雷碼的起始軟元件號 (BIN16/32 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓢ	○							○	---
Ⓓ	○							---	---

功　能

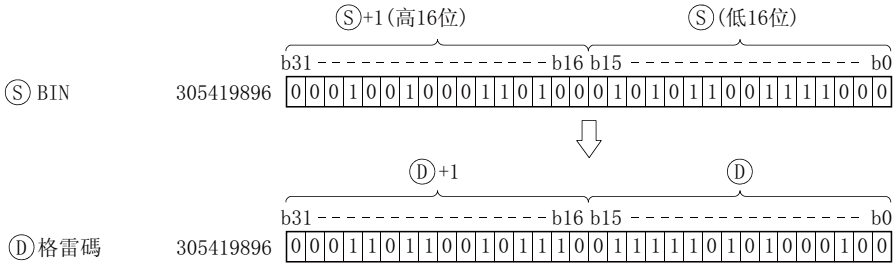
GRY

將Ⓢ中指定的軟元件的 BIN16 位元資料轉換爲格雷碼後，存儲到中指定編號的軟元件中。



DGRY

將Ⓢ中指定的軟元件的 BIN32 位元資料轉換爲格雷碼後，存儲到Ⓓ中指定編號的軟元件中。



出　錯

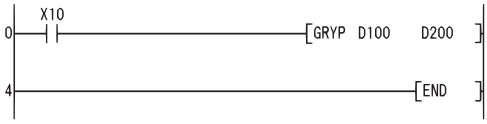
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ的資料爲負數時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X10 變爲 ON 時，將 D100 的 BIN 資料轉換爲格雷碼後，存儲到 D200 中的程式。

[梯形圖模式]

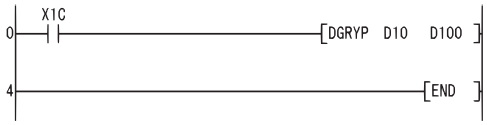


[列表模式]

步	指令	軟元件	
0	LD	X10	
1	GRYP	D100	D200
4	END		

(2) 以下爲 X1C 變爲 ON 時，將 D10、D11 的 BIN 資料轉換爲格雷碼後，存儲到 D100、D101 中的程式。

[梯形圖模式]

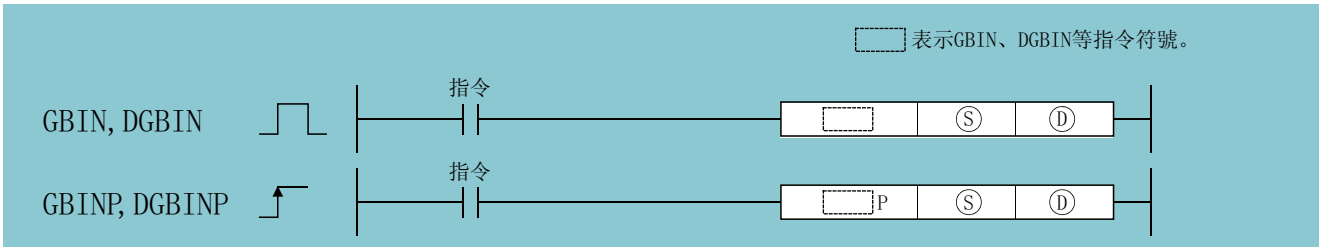


[列表模式]

步	指令	軟元件	
0	LD	X1C	
1	DGRYP	D10	D100
4	END		

6.3.10 GBIN、GBINP、DGBIN、DGBINP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ： 格雷碼資料或者存儲格雷碼資料的起始軟元件號 (BIN16/32 位)。

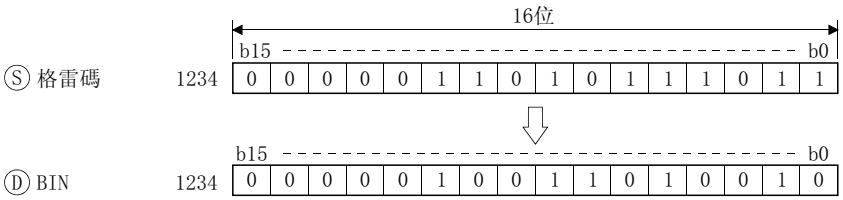
Ⓓ ： 存儲轉換後的 BIN 值的起始軟元件號 (BIN16/32 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓢ	○							○	---
Ⓓ	○							---	---

功　能

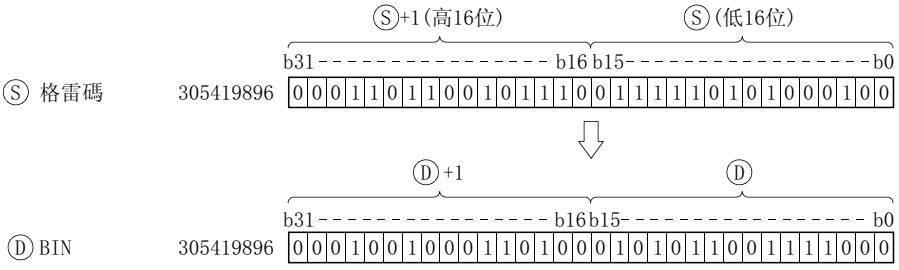
GBIN

將Ⓢ中指定的軟元件中存儲的格雷碼資料轉換爲 BIN16 位元資料後，存儲到Ⓓ中指定的軟元件中。



DGBIN

將Ⓢ中指定的軟元件中存儲的格雷碼資料轉換爲 BIN32 位元資料後，存儲到Ⓓ中指定的軟元件中。



出　錯

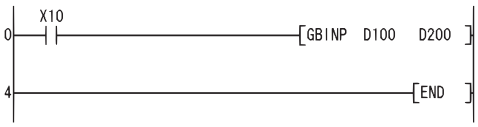
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	使用 GBIN 指令時，Ⓢ的資料超出了 0 ～ 32767 的範圍時。	○	○	○	○	○	○
4100	使用 DGBIN 指令時，Ⓢ的資料超出了 0 ～ 2147483647 的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X10 變爲 ON 時，將 D100 的格雷碼資料轉換爲 BIN 資料後，存儲到 D200 中的程式。

[梯形圖模式]

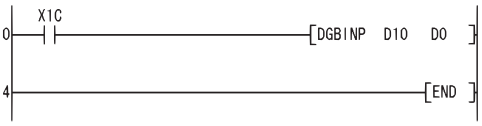


[列表模式]

步	指令	軟元件	
0	LD	X10	
1	GBINP	D100	D200
4	END		

(2) 以下爲 X1C 變爲 ON 時，將 D10、D11 的格雷碼資料轉換爲 BIN 資料後，存儲到 D0、D1 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X1C	
1	DGBINP	D10	D0
4	END		

6.3.11 NEG、NEGP、DNEG、DNEGP

Basic　High performance　Process　Redundant　Universal　LCPU

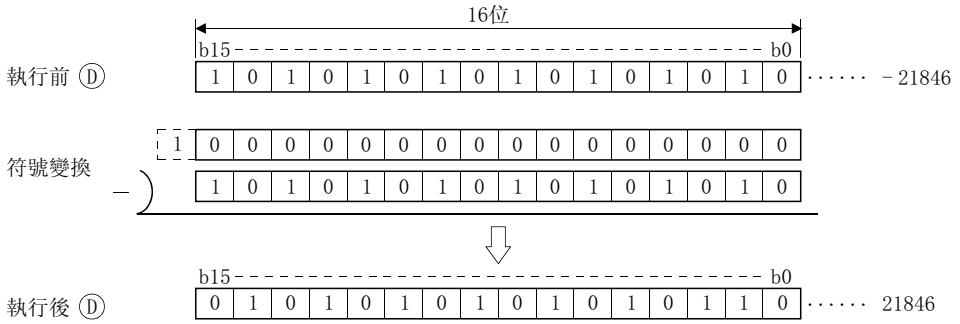
Ⓓ ：存儲進行 2 進制補數的資料的起始軟元件號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	○							---	

功　能

NEG

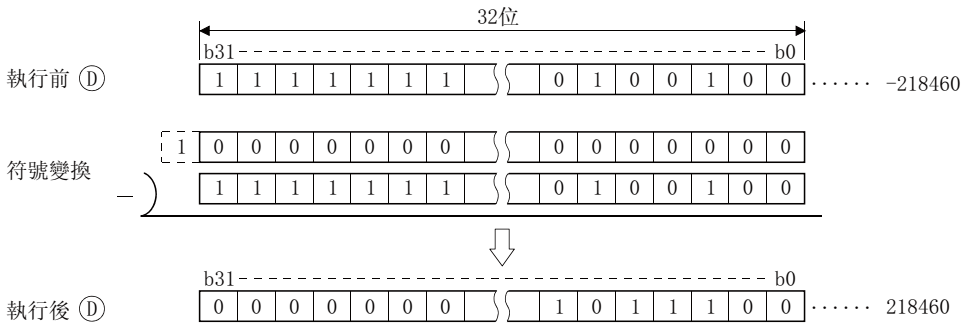
(1) 將ⓓ中指定的 16 位元軟元件的符號取反後，存儲到ⓓ中指定的軟元件中。



(2) 用於對正負符號進行取反。

DNEG

(1) 將ⓓ中指定的 32 位元軟元件的符號取反後，存儲到ⓓ中指定的軟元件中。



(2) 用於對正負符號進行取反。

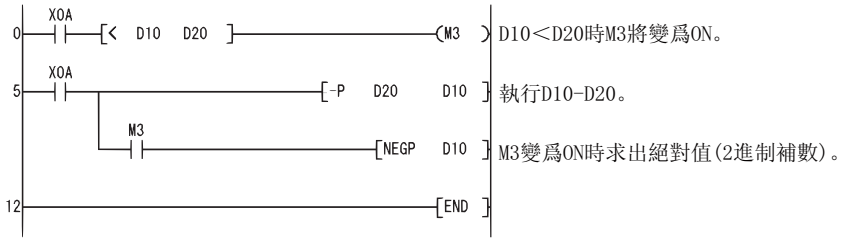
出　錯

(1) 在 NEG(P)、DNEG(P) 指令中無出錯。

程式示例

(1) 以下爲 XA 變爲 ON 時，進行 D10-D20 的計算，其結果爲負時求出其絕對值的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件	
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		

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

6.3.12 ENEG、ENEGP

Ⓓ ： 存儲進行符號取反的 32 位元浮點數據的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	---	○		---	○		○ *1	---	

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

(1) 將Ⓓ中指定的軟元件的 32 位元浮點實數資料的符號取反後，存儲到Ⓓ中指定的軟元件中。

(2) 用於對正負符號進行取反。

出 錯

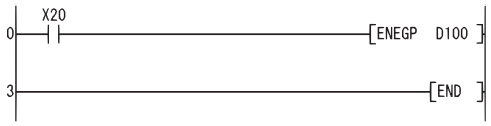
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 0、非正規數、非數、±∞時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D100、D101 的 32 位元浮點實數資料的符號取反後，存儲到 D100、D101 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X20
1	ENEGP	D100
3	END	

[動作]

6.3.13 EDNEG、EDNEGP

Basic High performance Process Redundant Universal LCPU

Ⓓ ：存儲進行符號取反的 64 位元浮點數據的軟元件的起始編號（實數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	---	○		---					

功　能

(1) 將Ⓓ中指定的軟元件的 64 位元浮點實數資料的符號取反後，存儲到Ⓓ中指定的軟元件中。

(2) 用於對正負符號進行取反。

出　錯

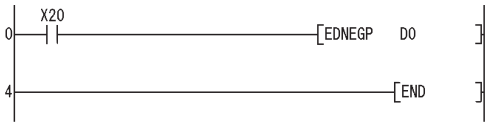
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 0 時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D0 ～ D3 的 64 位元浮點實數資料的符號取反後，存儲到 D0 ～ D3 中的程式。

[梯形圖模式]



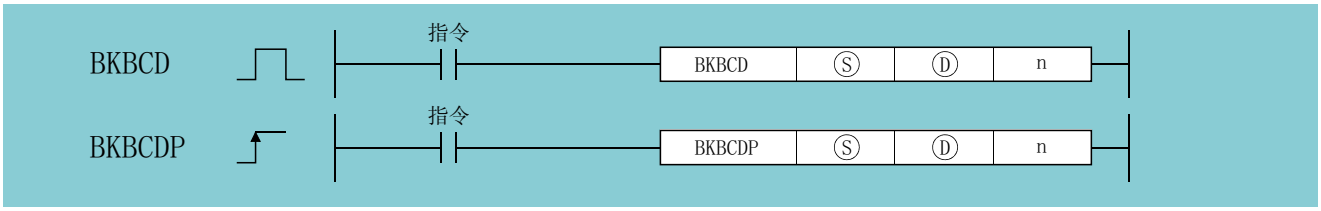
[列表模式]

步	指令	軟元件
0	LD	X20
1	EDNEGP	D0
3	END	

[動作]

6.3.14 BKBCD、BKBCDP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：存儲 BIN 資料的軟元件的起始編號 (BIN16 位)

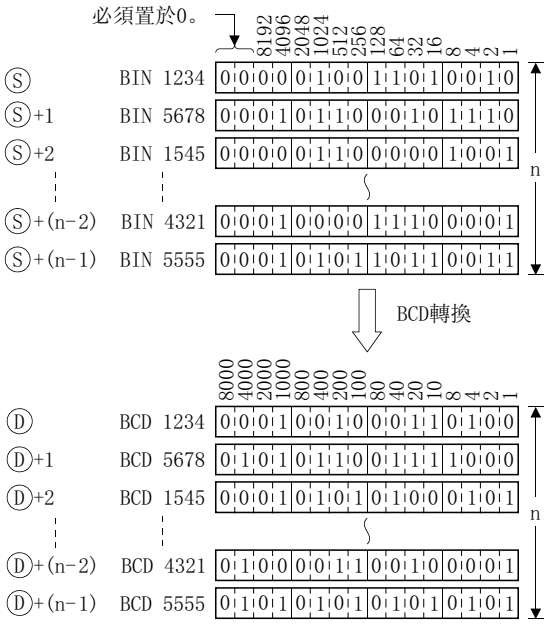
Ⓓ ：存儲轉換後的 BCD 資料的軟元件的起始編號 (BCD4 位)

n ：變數資料數 (BIN16 位元)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---					---
Ⓓ	---	○		---					---
n	○	○		○					---

功　能

(1) 將從Ⓢ中指定的軟元件開始的 n 點的 BIN 資料 (0 ～ 9999) 進行 BCD 轉換後，存儲到Ⓓ中指定的軟元件的後面。



出　錯

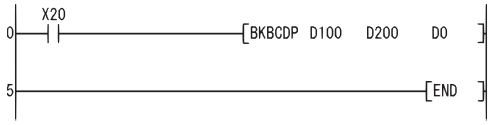
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	從Ⓢ的軟元件開始的 n 點的資料超出 0 ～ 9999 的範圍時。	○	○	○	○	○	○
4101	從Ⓢ、Ⓓ的軟元件開始的 n 點的範圍超出了相應軟元件時。 Ⓢ、Ⓓ的軟元件重復時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D100 ～ D102 中存儲的 BIN 資料值進行 BCD 轉換後，將其結果存儲到 D200 後面的程式。

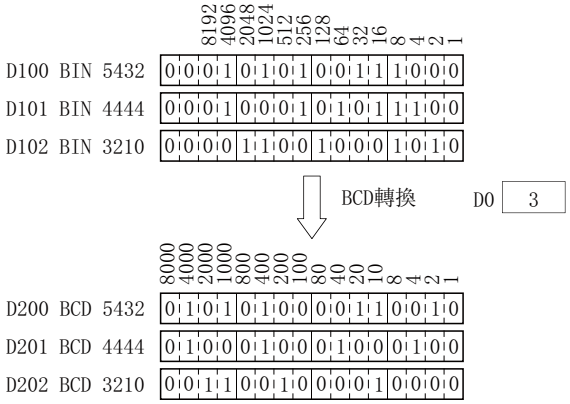
[梯形圖模式]



[列表模式]

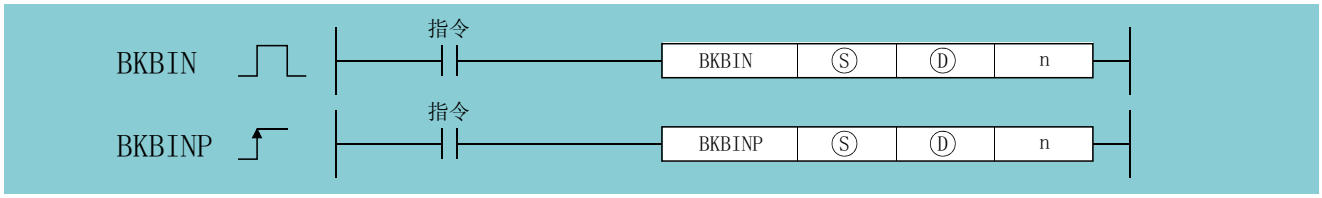
步	指令	軟元件		
0	LD	X20		
1	BKBCDP	D100	D200	D0
5	END			

[動作]



6.3.15 BKBIN、BKBINP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ： 存儲 BCD 資料的軟元件的起始編號 (BCD4 位)。

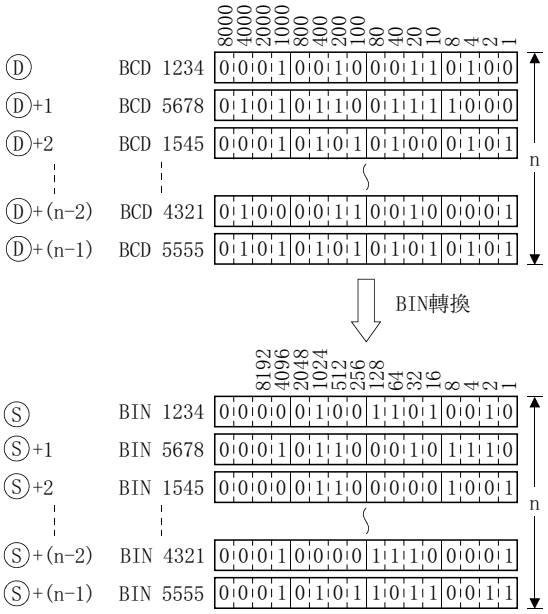
Ⓓ ： 存儲轉換後的 BIN 資料的軟元件的起始編號 (BIN16 位)。

n ： 變數資料數 (BIN16 位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---					---
Ⓓ	---	○		---					---
n	○	○		○					---

功　能

(1) 將從Ⓢ中指定的軟元件開始的 n 點的 BCD 資料 (0 ～ 9999) 進行 BIN 轉換後，存儲到Ⓓ中指定的軟元件的後面。



出　錯

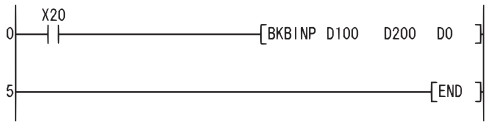
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	從Ⓢ的軟元件開始的 n 點的資料超出 0 ～ 9999 的範圍時。	○	○	○	○	○	○
4101	從Ⓢ、Ⓓ的軟元件開始的 n 點的範圍超出了相應軟元件時。 Ⓢ、Ⓓ的軟元件重復時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D100 ～ D102 中存儲的 BCD 資料值進行 BIN 轉換後，將其結果存儲到 D200 後面的程式。

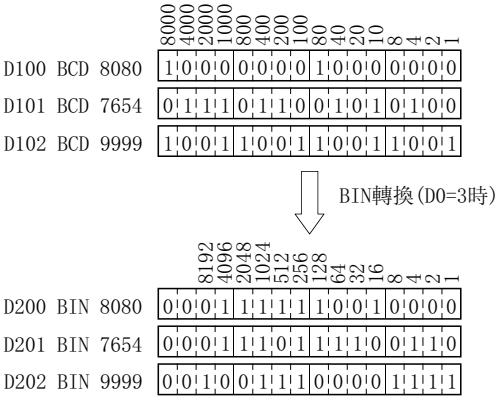
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X20		
1	BKBINP	D100	D200	D0
5	END			

[動作]



6.3.16 ECON、ECONP

Basic High performance Process Redundant Universal LCPU

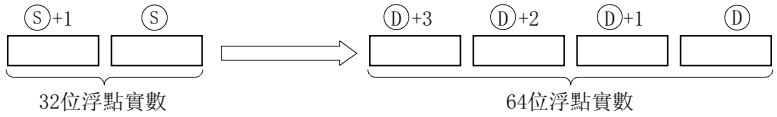
Ⓢ ：轉換源資料或者存儲轉換源資料的軟元件的起始編號（實數（單精確度））。

Ⓓ ：存儲轉換後的資料的軟元件的起始編號（實數（雙精度））。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---			○		---
Ⓓ	---	○		---			---		---

功　能

將Ⓢ中指定的 32 位浮點實數轉換爲 64 位浮點實數後，將轉換結果存儲到Ⓓ中指定的軟元件中。



出　錯

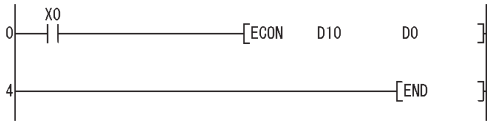
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 0、非正規數、非數、±∞時。	---	---	---	---	○	○

程式示例

(1)　以下爲 X0 變爲 ON 時，將 D10 ～ D11 中的 32 位浮點實數轉換爲 64 位浮點實數後，輸出到 D0 ～ D3 中的程式。

[梯形圖模式]

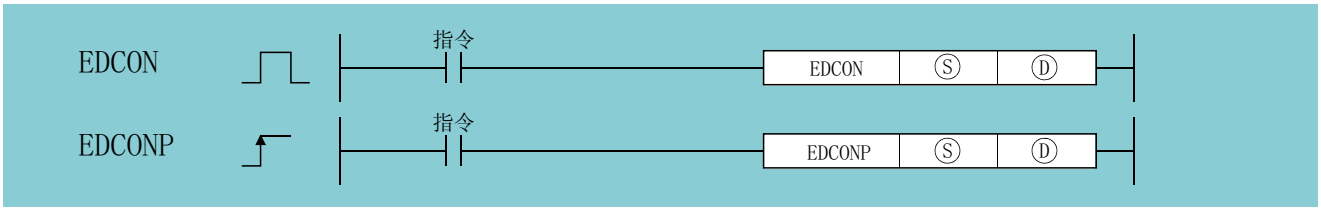


[列表模式]

步	指令	軟元件	
0	LD	X0	
1	ECON	D10	D0
4	END		

6.3.17 EDCON、EDCONP

Basic High performance Process Redundant Universal LCPU



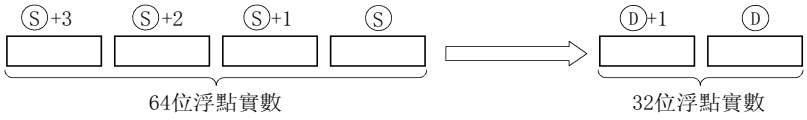
Ⓢ ：轉換源資料或者存儲轉換源資料的軟元件的起始編號（實數（雙精度））。

Ⓓ ：存儲轉換後的資料的軟元件的起始編號（實數（單精確度））。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---			---	○	---
Ⓓ	---	○		---			○	---	---

功　能

將Ⓢ中指定的 64 位浮點實數轉換爲 32 位浮點實數後，將轉換結果存儲到Ⓓ中指定的軟元件中。



出　錯

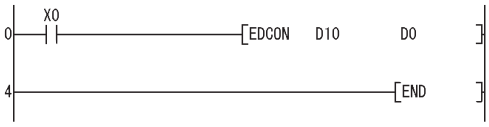
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 0 時。	---	---	---	---	○	○
4141	转换结果超出了以下范围时。（发生了溢出时。） $2^{128} \leq$ | 轉換結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D10 ～ D13 中的 64 位浮點實數轉換爲 32 位浮點實數後，輸出到 D0 ～ D1 中的程式。

[梯形圖模式]



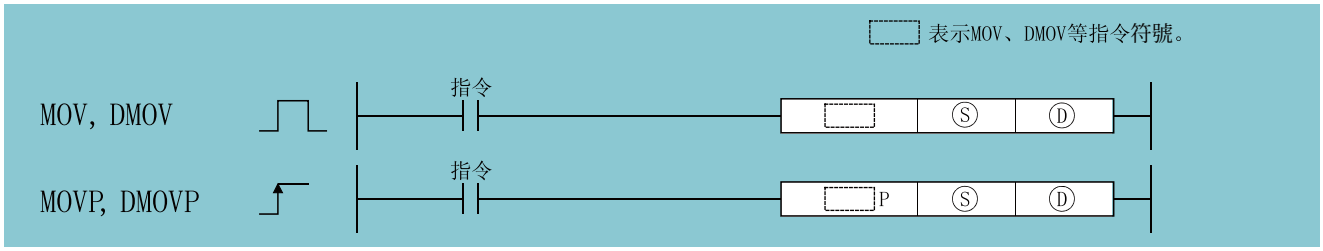
[列表模式]

步	指令	軟元件	
0	LD	X0	
1	EDCON	D10	D0
4	END		

6.4 資料傳送指令

6.4.1 MOV、MOVP、MOV、DMOVP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：傳送源資料或者存儲資料的軟元件編號 (BIN16/32 位)。

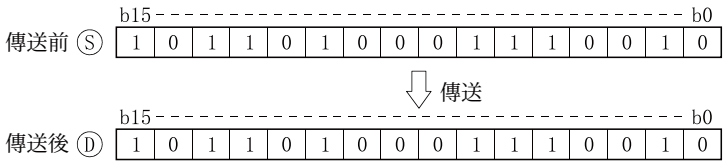
Ⓓ ：傳送目標的軟元件編號 (BIN16/32 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓢ	○							○	---
Ⓓ	○							---	---

功　能

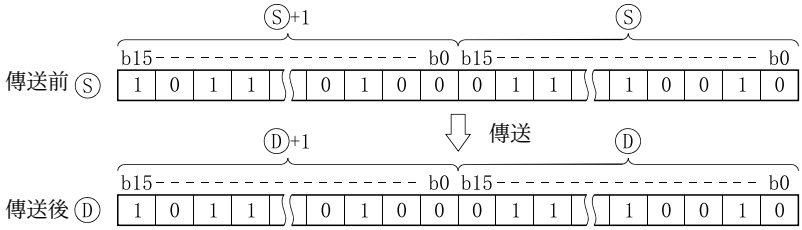
MOV

將Ⓢ中指定的軟元件的 16 位元資料傳送到Ⓓ中指定的軟元件中。



DMOV

將Ⓢ中指定的軟元件的 32 位元資料傳送到Ⓓ中指定的軟元件中。



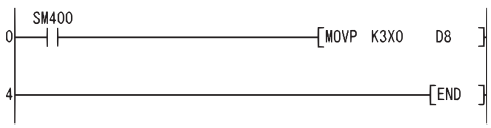
出　錯

(1) 在 MOV(P)、DMOV(P) 指令中無運算出錯。

程式示例

(1) 以下爲將輸入 X0 ～ XB 的資料存儲到 D8 中的程式。

[梯形圖模式]

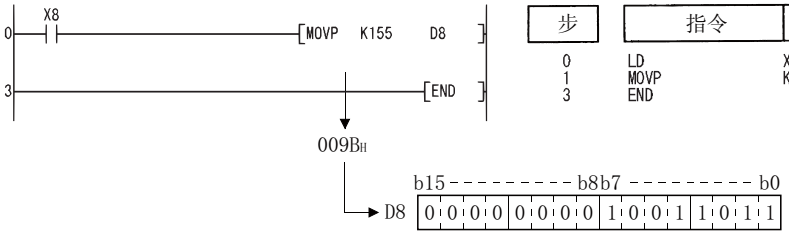


[列表模式]

步	指令	軟元件
0	LD	SM400
1	MOVP	K3X0 D8
4	END	

(2) 以下爲 X8 變爲 ON 時，將常數 K155 存儲到 D8 中的程式。

[梯形圖模式]

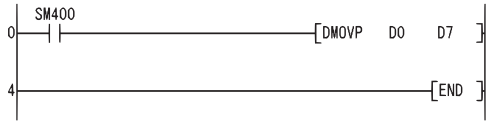


[列表模式]

步	指令	軟元件	
0	LD	X8	
1	MOVP	K155	D8
3	END		

(3) 以下爲將 D0、D1 的資料存儲到 D7、D8 中的程式。

[梯形圖模式]

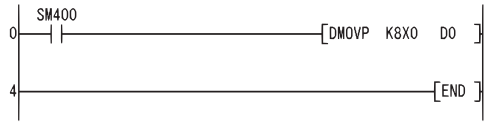


[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	DMOVP	D0	D7
4	END		

(4) 以下爲將 X0 ～ X1F 的資料存儲到 D0、D1 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	DMOVP	K8X0	D0
4	END		

6.4.2 EMOV、EMOVP

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 基本型 QCPU：序列號的前 5 位數爲 “ 04122 ” 以後

Ⓢ ：傳送資料或者存儲傳送資料的軟元件編號（實數）

Ⓓ ：存儲傳送目標資料的軟元件編號（實數）

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

將Ⓢ中指定的軟元件中存儲的 32 位元浮點實數資料傳送到Ⓓ中指定的軟元件中。

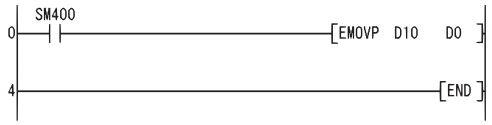
出 錯

(1) EMOV(P) 指令中無運算出錯。

程式示例

(1) 以下爲將 D10、D11 的實數存儲到 D0、D1 中的程式。

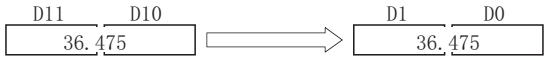
[梯形圖模式]



[列表模式]

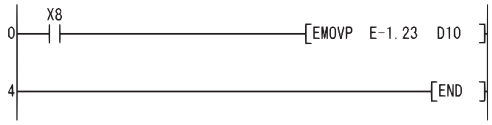
步	指令	軟元件	
0	LD	SM400	
1	EMOVP	D10	D0
4	END		

[動作]



(2) 以下爲 X8 變爲 ON 時，將實數 -1.23 存儲到 D10、D11 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X8	
1	EMOVP	E-1.23	D10
4	END		

[動作]

6.4.3 EDMOV、EDMOVP

Basic (×) High performance (×) Process (×) Redundant (×) Universal LCPU

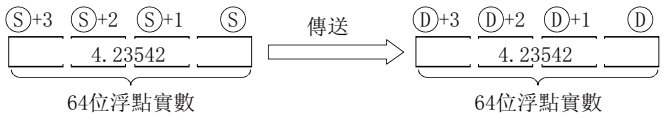
Ⓢ ：傳送資料或者存儲傳送資料的軟元件編號 (實數)。

Ⓓ ：存儲傳送目標資料的軟元件編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

將Ⓢ中指定的軟元件中存儲的 64 位元浮點實數資料傳送到Ⓓ中指定的軟元件中。



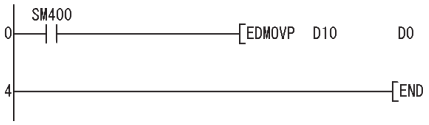
出　錯

(1) EDMOV(P) 指令中無運算出錯。

程式示例

(1) 以下爲將 D10 ～ D13 的 64 位元浮點實數存儲到 D0 ～ D3 中的程式。

[梯形圖模式]



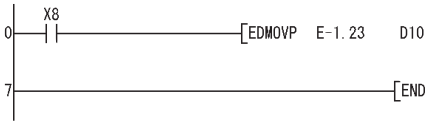
[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	EDMOVP	D10	D0
4	END		

[動作]

(2) 以下爲 X8 變爲 ON 時，將實數 -1.23 存儲到 D10 ～ D13 中的程式。

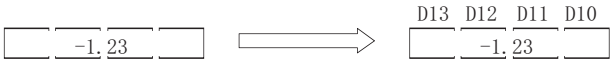
[梯形圖模式]



[列表模式]

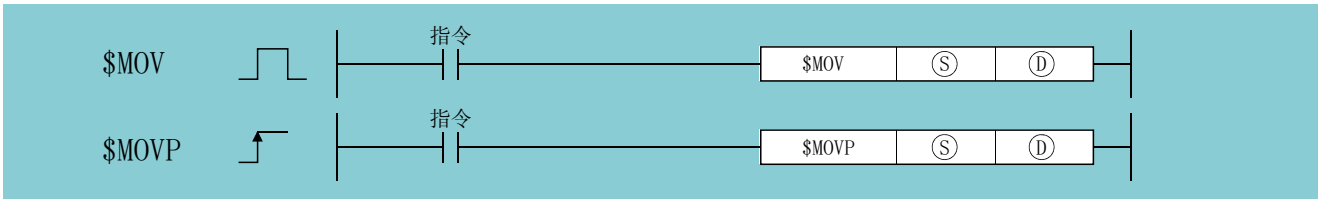
步	指令	軟元件	
0	LD	X8	
1	EDMOVP	E-1.23	D10
7	END		

[動作]



6.4.4 $MOV、$MOVP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：傳送字串（最大字串： 32 個字元）或者存儲傳送字串的軟元件的起始編號（字串）。

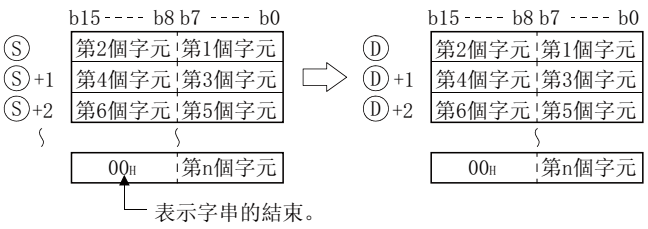
Ⓓ ：存儲傳送字串的軟元件的起始編號（字串）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

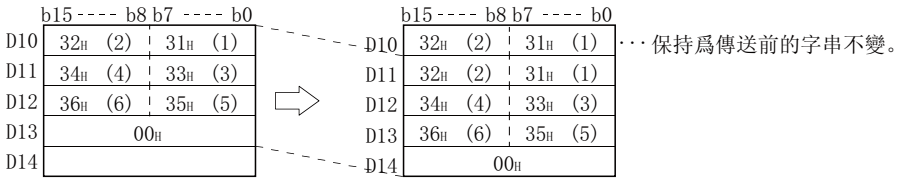
(1) 將Ⓢ中指定的字串資料傳送到Ⓓ中指定的軟元件後面。

在字串傳送中，對Ⓢ中指定的用 “ ”（雙引號）圍住的字串或者從中指定的軟元件號開始至存儲了 “ 00_H ” 的軟元件號爲止的字串進行一次傳送。

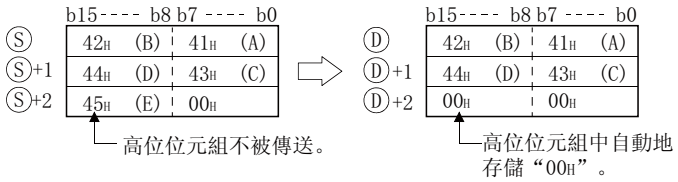


(2) 即使在存儲要傳送的字串資料的軟元件範圍 (Ⓢ～Ⓢ+n) 与與存儲已傳送的字串資料的軟元件範圍 (Ⓓ～Ⓓ+n) 相互重疊的情況下，也將正常進行處理。

將存儲在 D10 ～ D13 中的字串資料傳送到 D11 ～ D14 中時的情況如下所示：



(3) Ⓢ+n 的低位元位元組中存儲了 “ 00_H ” 時，Ⓓ+n 的高位位元組、低位元位元組中均將存儲 “ 00_H ”。



(4) Ⓢ中指定了漢字等 2 位元組資料時，將被轉換爲移位 JIS 碼。

如果執行了 $MOV 指令，Ⓓ中存儲的高位位元組與低位元位元組將被相互換位元。(參閱程式示例 (3))

出　錯

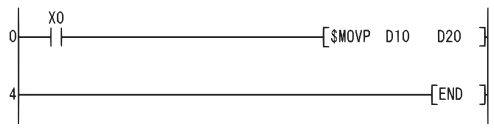
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓢ 中指定的軟元件號以後至相應軟元件號爲止之間不存在 “ 00_H ” 時。 Ⓢ 中指定的軟元件號以後至相應軟元件的最終軟元件號爲止的點數無法容納全部的字串時。 Ⓢ 的字串超過了 16383 個字元時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D10 ～ D12 中存儲的字串資料傳送到 D20 ～ D22 中的程式。

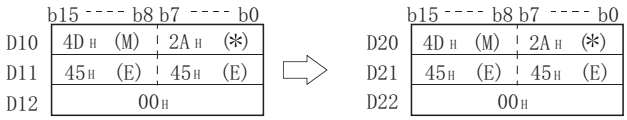
[梯形圖模式]



[列表模式]

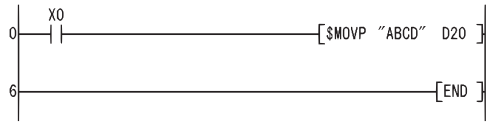
步	指令	軟元件	
0	LD	X0	
1	$MOVP	D10	D20
4	END		

[動作]



(2) 以下爲 X0 變爲 ON 時，將字串 “ ABCD ” 傳送到 D20 ～ D21 中的程式。

[梯形圖模式]

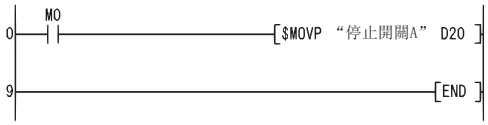


[列表模式]

步	指令	軟元件	
0	LD	X0	
1	$MOVP	"ABCD"	D20
6	END		

(3) 以下爲 X0 變爲 ON 時，將字串 “ 停止開關 A ” 傳送到 D20 ～ D24 中的程式。

[梯形圖模式]



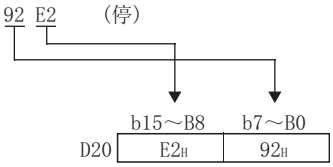
[列表模式]

步	指令	軟元件	
0	LD	M0	
1	$MOVP	“停止開關A”	D20
9	END		

[動作]

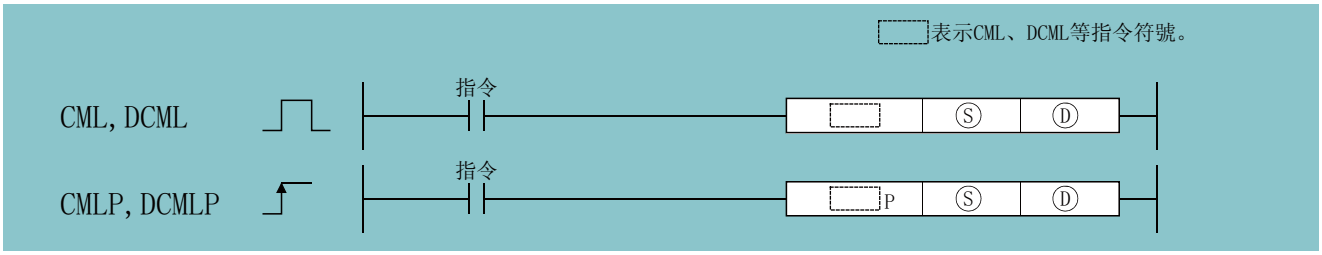
	b15～B8	b7～B0	
D20	$7E_H$ *1	92_H *1	(停)
D21	$7E_H$	$8E_H$	(止)
D22	$B2_H$	BD_H	(開)
D23	$C1_H$	AF_H	(關)
D24	00_H	41_H	(A)

*1: 2 個位元組的情況下，存儲的高位位元組與低位元位元組將被相互換位元。
停止開關 A 的 “ 停 ” 的移位碼爲 $92E2_H$，通過執行了 $MOV 指令，D20 中將存儲 $E292_H$。



6.4.5 CML、CMLP、DCML、DCMLP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ : 取反資料或者存儲取反資料的軟元件編號 (BIN16/32 位)。

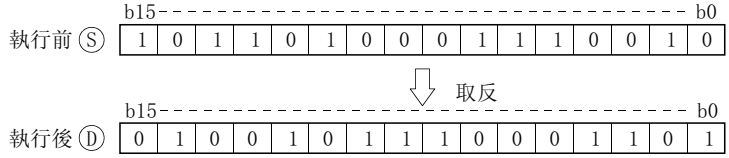
Ⓓ : 存儲取反結果的軟元件編號 (BIN16/32 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓢ	○							○	---
Ⓓ	○							---	---

功 能

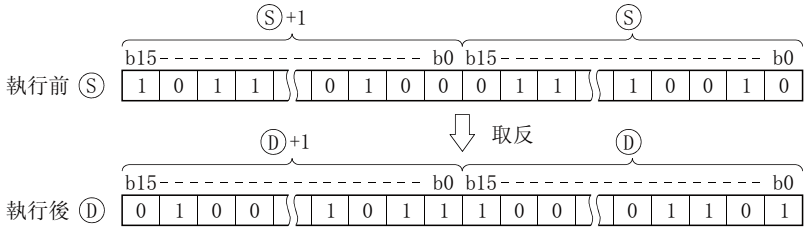
CML

對Ⓢ中指定的 16 位元資料進行逐位元取反， 並將其結果傳送到Ⓓ中指定的軟元件中。



DCML

對Ⓢ中指定的 32 位元資料進行逐位元取反， 並將其結果傳送到Ⓓ中指定的軟元件中。



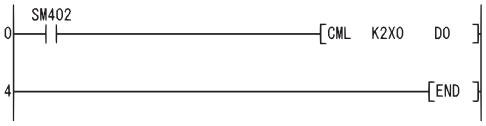
出　錯

(1) CML(P)、DCML(P) 指令中無運算出錯。

程式示例

(1) 以下爲對 X0 ～ X7 的資料進行取反後，將其傳送到 D0 中的程式。

[梯形圖模式]

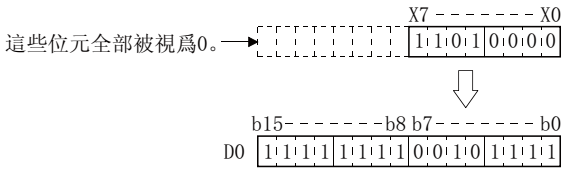


[列表模式]

步	指令	軟元件	
0	LD	SM402	
1	CML	K2X0	D0
4	END		

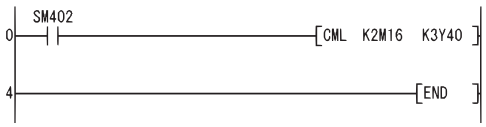
[動作]

Ⓢ 的位數＜Ⓓ的位數時



(2) 以下爲對 M16 ～ M23 的資料進行取反後，將其傳送到 Y40 ～ Y47 中的程式。

[梯形圖模式]

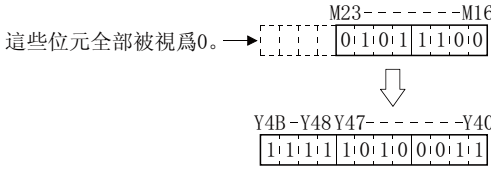


[列表模式]

步	指令	軟元件	
0	LD	SM402	
1	CML	K2M16	K3Y40
4	END		

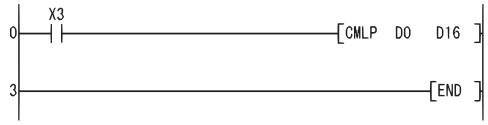
[動作]

Ⓢ 的位數＜Ⓓ的位數時



(3) 以下爲 X3 變爲 ON 時，對 D0 的資料進行取反後，將其傳送到 D16 中的程式。

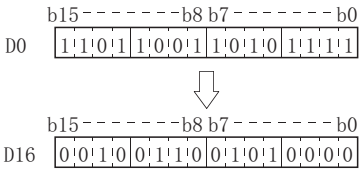
[梯形圖模式]



[列表模式]

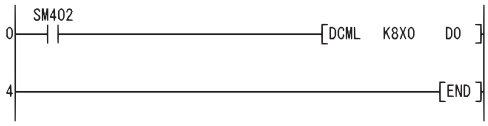
步	指令	軟元件	
0	LD	X3	
1	CMLP	D0	D16
3	END		

[動作]



(4) 以下爲對 X0 ～ X1F 的資料進行取反後，將其傳送到 D0、D1 中的程式。

[梯形圖模式]

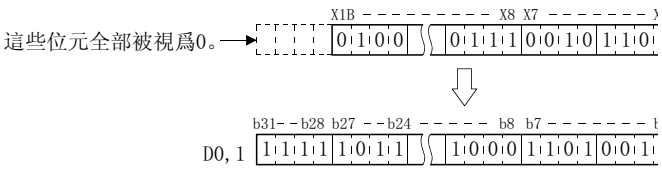


[列表模式]

步	指令	軟元件	
0	LD	SM402	
1	DCML	K8X0	D0
4	END		

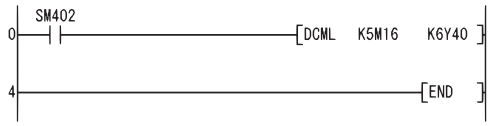
[動作]

Ⓢ 的位數＜Ⓓ的位數時



(5) 以下爲對 M16 ～ M35 的資料進行取反後，將其傳送到 Y40 ～ Y63 中的程式。

[梯形圖模式]

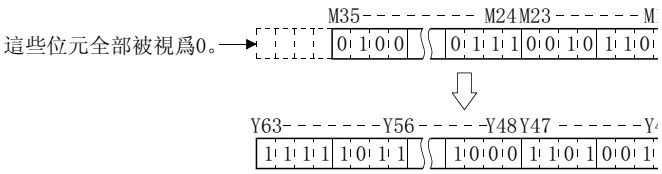


[列表模式]

步	指令	軟元件	
0	LD	SM402	
1	DCML	K5M16	K6Y40
4	END		

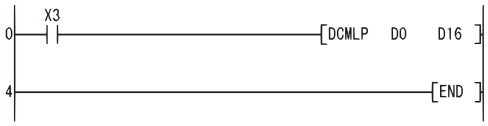
[動作]

Ⓢ 的位數＜Ⓓ的位數時



(6) 以下爲 X3 變爲 ON 時，對 D0、D1 的資料進行取反後，將其傳送到 D16、D17 中的程式。

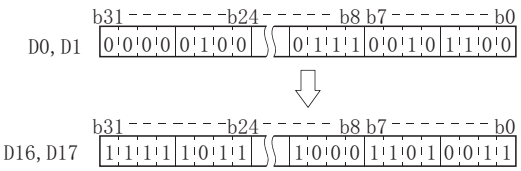
[梯形圖模式]



[列表模式]

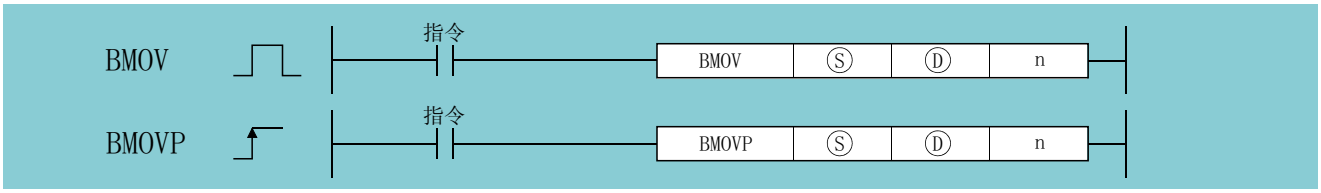
步	指令	軟元件	
0	LD	X3	
1	DCMLP	D0	D16
4	END		

[動作]



6.4.6 BMOV、BMOVP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：傳送資料或者存儲傳送資料的軟元件的起始編號 (BIN16 位)。

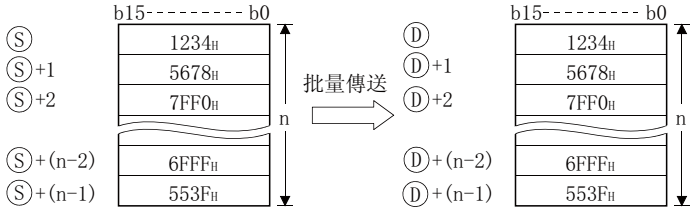
Ⓓ ：存儲傳送目標軟元件的起始編號 (BIN16 位)。

n ：傳送數 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ			○				---		---
Ⓓ			○				---		---
n			○				○		---

功　能

(1) 將Ⓢ中指定的軟元件開始的 n 點的 16 位元資料批量傳送到Ⓓ中指定的軟元件開始的 n 點中。



(2) 傳送源與傳送目標的軟元件重復時，也可進行傳送。

傳送到軟元件號小的一方的情況下，從Ⓢ開始傳送；傳送到軟元件號大的一方的情況下，從Ⓢ+(n-1) 開始傳送。

但是，從 R 傳送到 ZR 或者從 ZR 傳送到 R 時，應注意不要使下述 ZR 與 R 的各自的傳送範圍重疊。

從 R 傳送到 R 以及從 ZR 傳送到 ZR 時不會有問題。

・ZR 的傳送範圍 ((指定的 ZR 起始號) ～ (指定的 ZR 起始號 + 傳送數 -1))

・R 的傳送範圍 ((指定的 R 起始號 + 文件寄存器塊號× 32768) ～

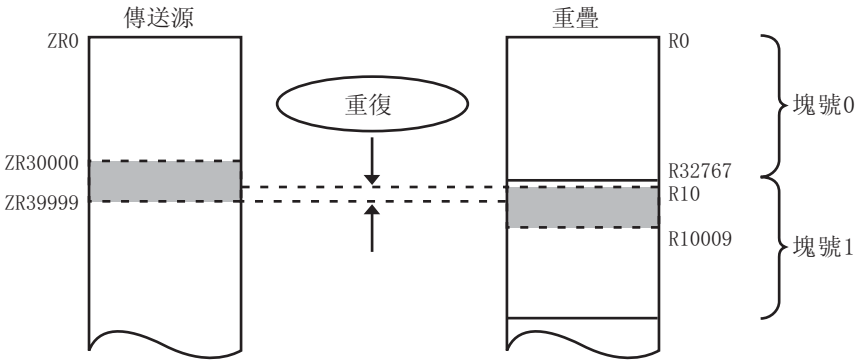
(指定的 R 起始號 + 文件寄存器塊號× 32768+ 傳送數 -1))

例 將 10000 點的資料從傳送源 ZR30000 傳送到傳送目標塊號 1 的 R10 中時，傳送範圍重疊。

· ZR 的傳送範圍→ (30000) ～ (30000+10000-1) → (30000) ～ (39999)

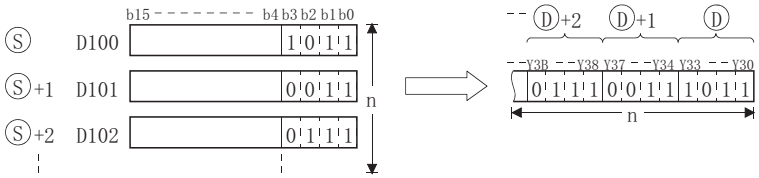
· R 的傳送範圍→ (10+(1 × 32768)) ～ (10+(1 × 32768)+10000-1)
→ (32778) ～ (42777)

由於 32778 起至 39999 爲止的範圍重疊，因此無法進行正常的值傳送。



(3) 在Ⓢ爲字軟元件而Ⓓ爲位元軟元件的情況下，字軟元件的物件爲位元軟元件的位數指定中指定的位數。

Ⓓ中指定了 K1Y30 時，Ⓢ中指定的字軟元件的低 4 位元將成爲物件。



(4) Ⓢ、Ⓓ中指定了位軟元件時，必須將Ⓢ、Ⓓ的位數設置爲相同。

(5) Ⓢ、Ⓓ中使用鏈結直接軟元件及智慧功能模組軟元件時，只能指定Ⓢ或Ⓓ中的一個。

(6) 軟元件範圍檢查的執行有無選擇

通過軟元件範圍檢查禁止標誌 (SM237)，可以選擇在執行 BMOV 指令時是否進行軟元件範圍檢查。(僅在子集條件成立時)

SM237 變爲 ON 時，不對Ⓢ～Ⓢ+(n-1)、Ⓓ～Ⓓ+(n-1) 進行是否在軟元件範圍內的檢查。

注意事項

SM237 變爲 ON 時，不要進行以下訪問：

· 變址修飾目標超出了軟元件範圍的訪問

·Ⓓ～Ⓓ+(n-1) 跨越了軟元件範圍邊界的訪問 *1

· 在未進行文件寄存器設置狀態下至文件寄存器的訪問

· 對不存在多 CPU 高速通信區軟元件的區域進行的訪問 （僅對於 QCPU）

*1: 請參閱 DFMOV 指令有關內容。

要 點

SM237 只能使用下述 CPU 模組：

· 序列號的前 5 位數爲 “ 10012 ” 以後的通用型 QCPU

· LCPU

出　錯

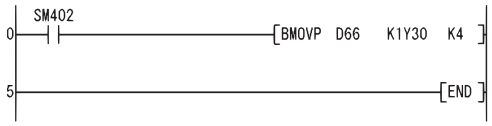
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓢ、Ⓓ開始的 n 點的軟元件範圍超出了相應軟元件範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲將 D66 ～ D69 的低 4 位元資料以 4 點爲單位輸出到 Y30 ～ Y3F 中的程式。

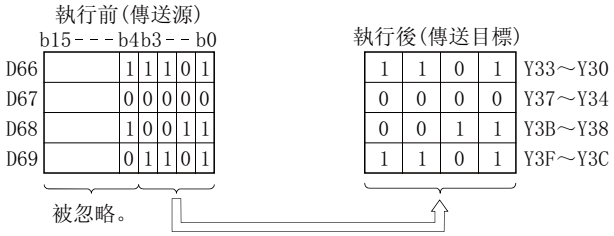
[梯形圖模式]



[列表模式]

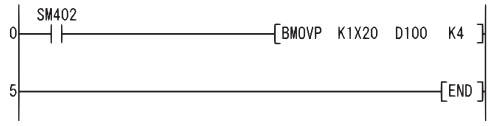
步	指令	軟元件		
0	LD	SM402		
1	BMOVP	D66	K1Y30	K4
5	END			

[動作]



(2) 以下爲將 X20 ～ X2F 的資料以 4 點爲單位輸出到 D100 ～ D103 中的程式。

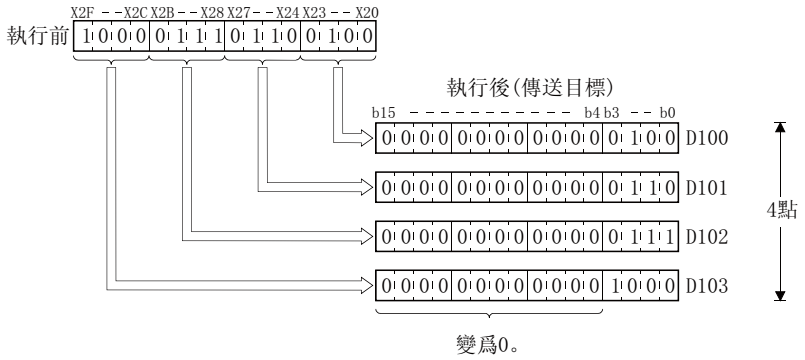
[梯形圖模式]



[列表模式]

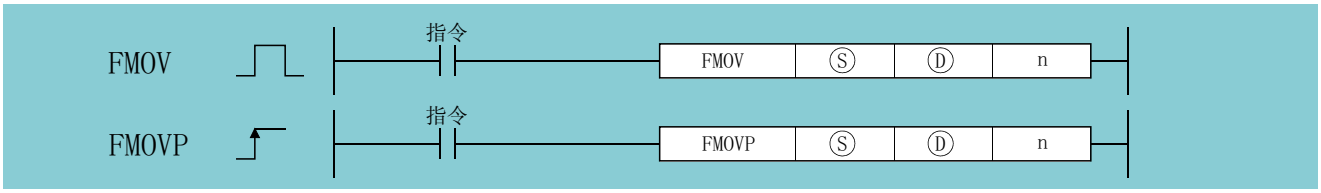
步	指令	軟元件		
0	LD	SM402		
1	BMOVP	K1X20	D100	K4
5	END			

[動作]



6.4.7 FMOV、FMOVP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：傳送資料或者存儲傳送資料的軟元件的起始編號 (BIN16 位)。

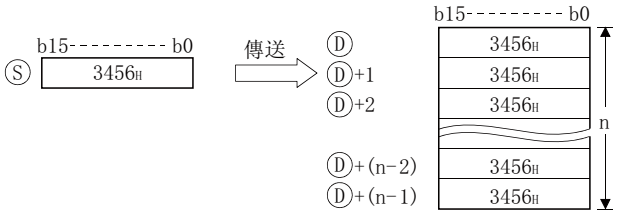
Ⓓ ：傳送目標的軟元件的起始編號 (BIN16 位)。

n ：傳送數 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ			○				○		---
Ⓓ			○				---		---
n			○				○		---

功　能

(1) 將Ⓢ中指定的軟元件的 16 位元資料傳送到從Ⓓ中指定的軟元件開始的 n 點中。



(2) 在Ⓢ爲字軟元件而Ⓓ爲位元軟元件的情況下，Ⓢ的字軟元件的物件爲位元軟元件的位數指定中指定的位數。

Ⓓ中指定了 K1Y30 時，Ⓢ中指定的字軟元件的低 4 位元將成爲物件。

(3) Ⓢ、Ⓓ中指定了位軟元件時，必須將Ⓢ、Ⓓ的位數設置爲相同。

(4) 軟元件範圍檢查的執行有無選擇

通過軟元件範圍檢查禁止標誌 (SM237)，可以選擇在執行 FMOV 指令時是否進行軟元件範圍檢查。

(僅在子集條件成立時)

SM237 變爲 ON 時，不進行Ⓓ～Ⓓ+(n-1) 是否在軟元件範圍內的檢查。

關於 SM237 的詳細內容，請參閱所使用的 CPU 模組的用戶手冊 (硬體設計 / 維護點檢篇)。

注意事項

軟元件範圍檢查的執行有無選擇

- 變址修飾目標超出了軟元件範圍的訪問
- Ⓓ～Ⓓ+(n-1) 跨越了軟元件範圍邊界的訪問 *1
- 在未進行文件寄存器設置狀態下至文件寄存器的訪問
- 對不存在多 CPU 高速通信區軟元件的區域進行的訪問 （僅對於 QCPU）

*1:　請參閱 DFMOV 指令有關內容。

要 點

SM237 只能使用下述 CPU 模組：
・序列號的前 5 位數爲 “ 10012 ” 以後的通用型 QCPU
・LCPU

出 錯

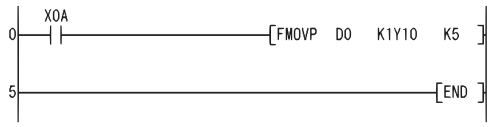
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從ⓓ開始的 n 點的軟元件範圍超出了相應軟元件範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 XA 變爲 ON 時，將 D0 的低 4 位元資料以 4 點爲單位輸出到 Y10 ～ Y23 中的程式。

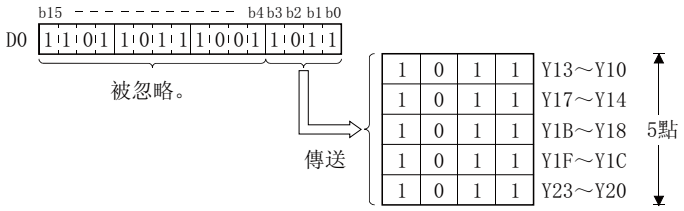
[梯形圖模式]



[列表模式]

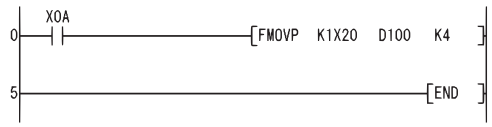
步	指令	軟元件		
0	LD	XOA		
1	FMOVP	D0	K1Y10	K5
5	END			

[動作]



(2) 以下爲 XA 變爲 ON 時，將 X20 ～ X23 的資料輸出到 D100 ～ D103 中的程式。

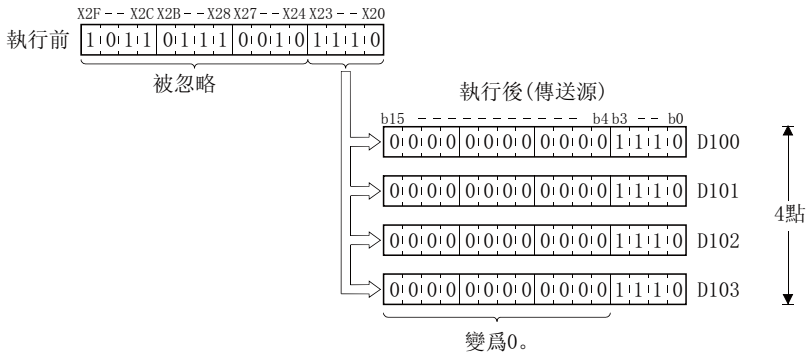
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	XOA		
1	FMOVP	K1X20	D100	K4
5	END			

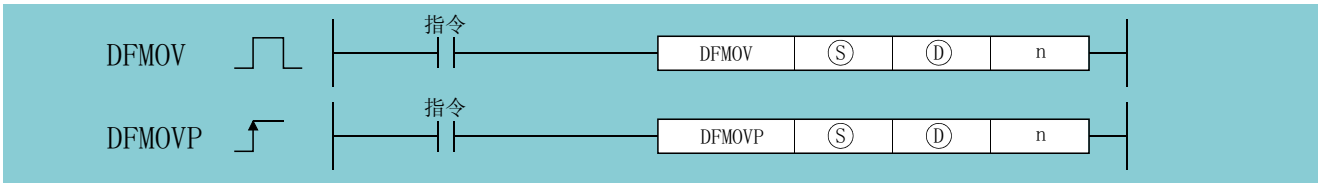
[動作]



6.4.8 DFMOV、DFMOVP

Basic / High performance / Process / Redundant / Ver. Universal / LCPU

·在序列號的前 5 位數爲“10102”以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。



Ⓢ ：傳送資料或者存儲傳送資料的軟元件的起始編號 (BIN32 位)。

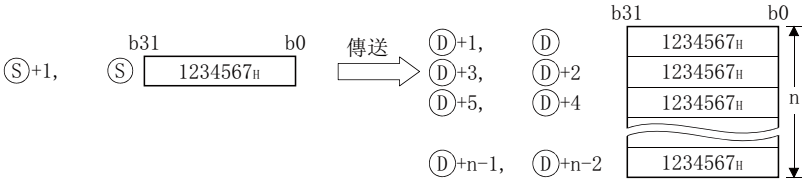
Ⓓ ：傳送目標的軟元件的起始編號 (BIN32 位)。

n ：傳送數 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ			○				○		---
Ⓓ			○				---		---
n			○				○		---

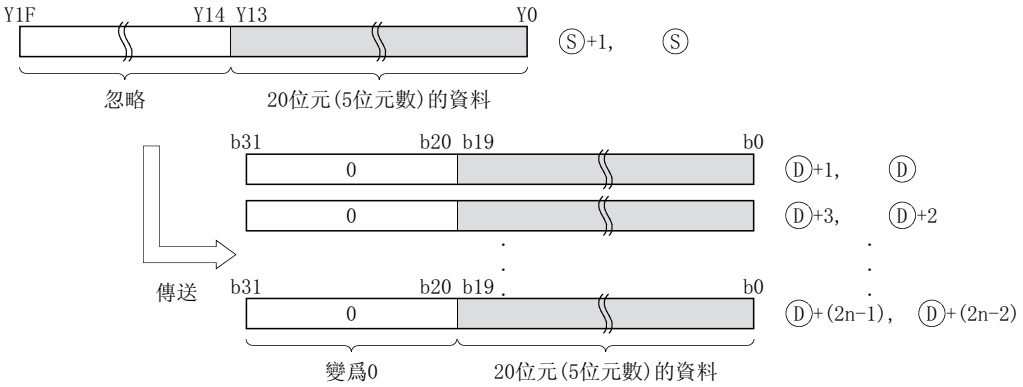
功　能

(1) 將Ⓢ中指定的軟元件的 32 位元資料傳送到從Ⓓ中指定的軟元件開始的 n 點中。



(2) 在Ⓢ中進行了位數指定時，將只按位元數指定量進行資料傳送。

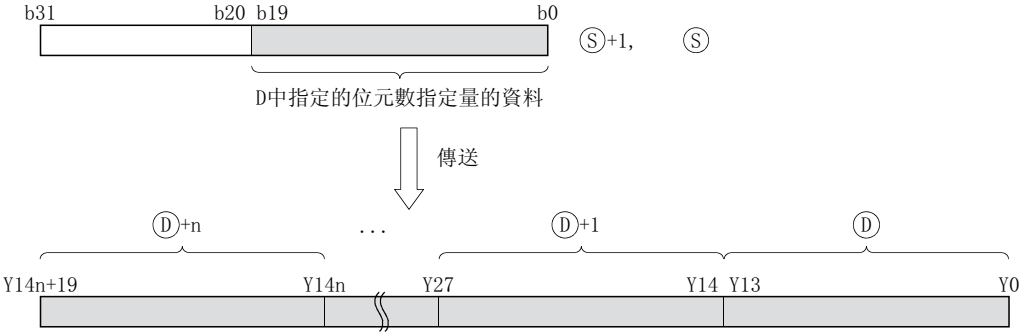
在Ⓢ中指定了 K5Y0 時，Ⓢ的字軟元件的低 20 位 (5 位元數) 將成爲物件。



(3) 在Ⓓ中進行了位數指定時，將只按Ⓓ中進行的位元數指定量進行資料傳送。

在Ⓓ中指定了 K5Y0 時，Ⓢ的字軟元件的低 20 位元將成爲物件。

Ⓢ、Ⓓ二者中均進行了位數指定時，與位數無關，將只按Ⓓ中指定的位元數指定量進行資料傳送。



(4) n 中指定的值爲 0 時將執行無處理。

(5) 通過軟元件範圍檢查禁止標誌 (SM237)，可以選擇在執行 DFMOV 指令時是否進行軟元件範圍檢查。
(僅在子集條件成立時)

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 中指定的資料爲負數時。	---	---	---	---	○	○
4101	傳送的資料點數 n 超過了Ⓓ的軟元件範圍時。	---	---	---	---	○	○

程式示例

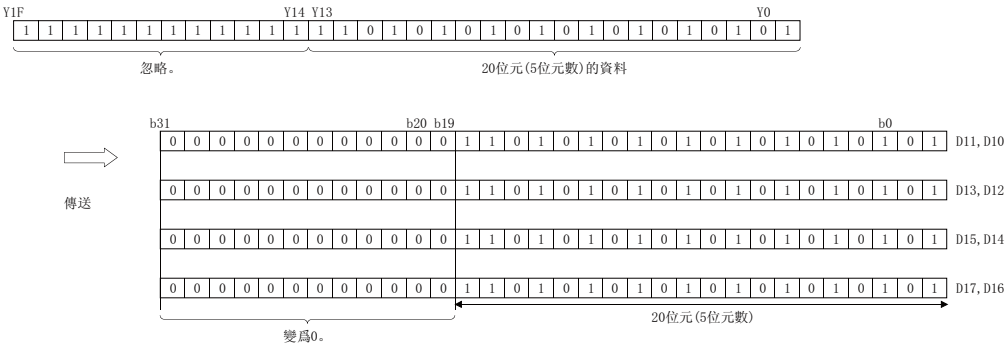
(1) 以下爲 M0 變爲 ON 時，將 Y0 ～ Y13(20 位元) 的資料存儲到 D10 ～ D17 中的程式。

[梯形圖模式]

[列表模式]

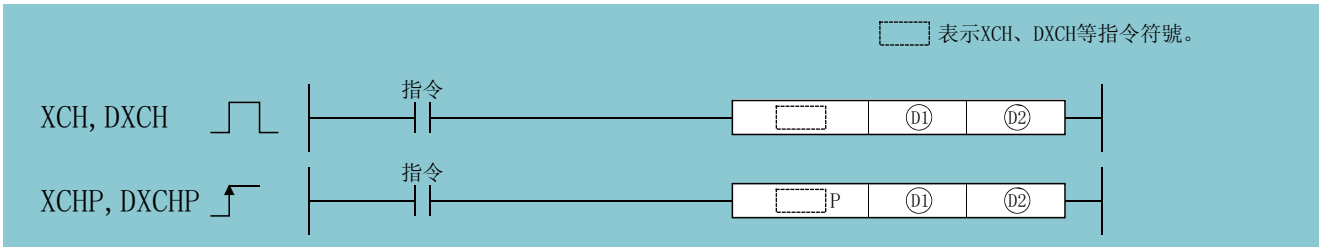
步	指令	軟元件		
0	LD	M0		
1	DFMOV	K5Y0	D10	K4
5	END			

[動作]



6.4.9 XCH、XCHP、DXCH、DXCHP

Basic | High performance | Process | Redundant | Universal | LCPU



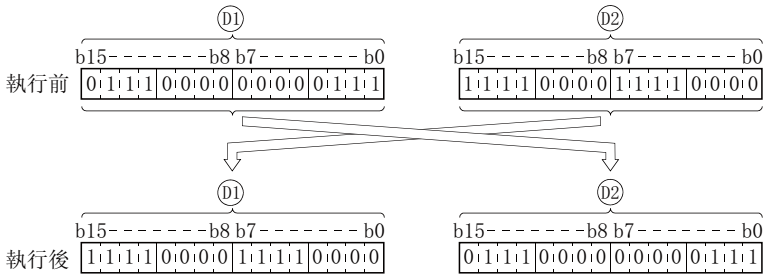
(D1)、(D2) ： 存儲交換資料的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數	其他
(D1)		○						---	
(D2)		○						---	

功　能

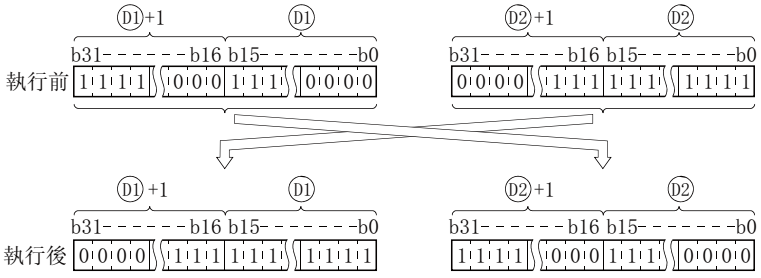
XCH

對(D1)、(D2)的 16 位元資料進行交換。



DXCH

對(D1)+1、(D1)與(D2)+1、(D2)的 32 位元資料進行交換。



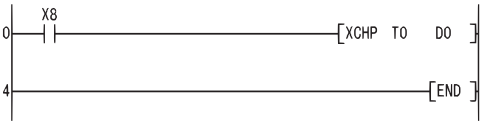
出　錯

(1) 在 XCH(P)、DXCH(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X8 變爲 ON 時，將 T0 的當前值與 D0 的內容進行交換的程式。

[梯形圖模式]

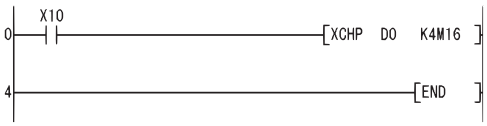


[列表模式]

步	指令	軟元件	
0	LD	X8	
1	XCHP	T0	D0
4	END		

(2) 以下爲 X10 變爲 ON 時，將 D0 的內容與 M16 ～ M31 的資料進行交換的程式。

[梯形圖模式]

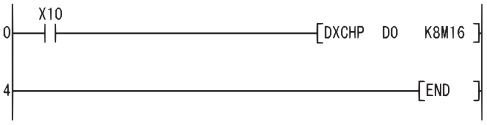


[列表模式]

步	指令	軟元件	
0	LD	X10	
1	XCHP	D0	K4M16
4	END		

(3) 以下爲 X10 變爲 ON 時，將 D0、D1 的內容與 M16 ～ M47 的資料進行交換的程式。

[梯形圖模式]

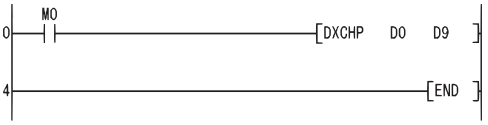


[列表模式]

步	指令	軟元件	
0	LD	X10	
1	DXCHP	D0	K8M16
4	END		

(4) 以下爲 M0 變爲 ON 時，將 D0、D1 的內容與 D9、D10 的資料進行交換的程式。

[梯形圖模式]

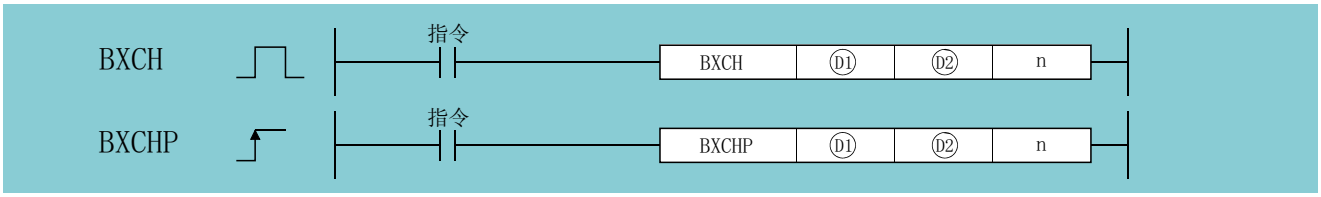


[列表模式]

步	指令	軟元件	
0	LD	M0	
1	DXCHP	D0	D9
4	END		

6.4.10 BXCH、BXCHP

Basic | High performance | Process | Redundant | Universal | LCPU



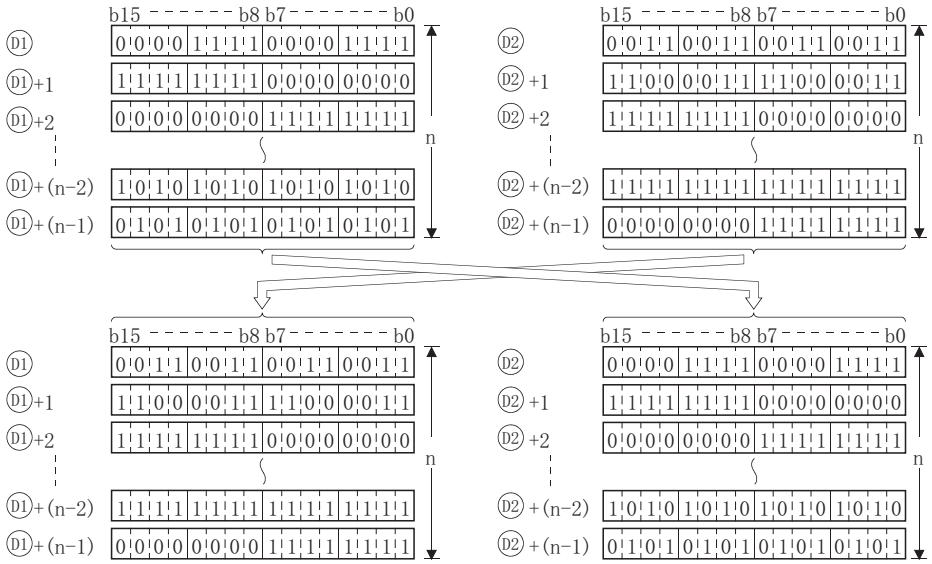
(D1)、(D2)： 存儲交換資料的軟元件的起始編號 (BIN16 位)。

n ：交換數 (BIN16 位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
(D1)	---	○		---					---
(D2)	---	○		---					---
n	○	○		○					---

功　能

將從(D1)中指定的軟元件開始的 n 點的 16 位元資料，與從(D2)中指定的軟元件開始的 n 點的 16 位元資料進行交換。



出　錯

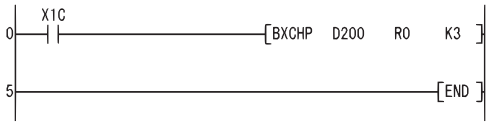
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從(D1)、(D2)的軟元件開始的 n 點的範圍超出了相應軟元件範圍時。 (D1)、(D2)的軟元件重復時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X1C 變爲 ON 時，將從 D200 開始的 3 點的資料與從 R0 開始的 3 點的資料進行交換的程式。

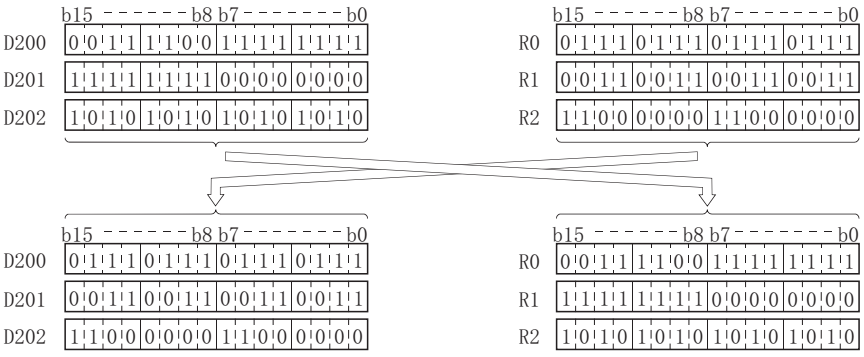
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X1C		
1	BXCHP	D200	R0	K3
5	END			

[動作]



6.4.11 SWAP、SWAPP

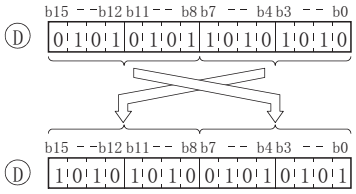
Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：存儲交換資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	○							---	

功　能

對Ⓓ中指定的軟元件的高低各 8 位的值進行交換。



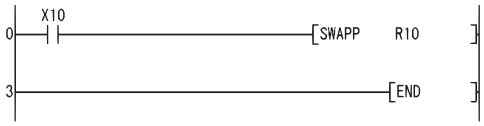
出　錯

(1)　在 SWAP(P) 指令中無運算出錯。

程式示例

(1)　以下爲 X10 變爲 ON 時，將 R10 的高 8 位與低 8 位元進行交換的程式。

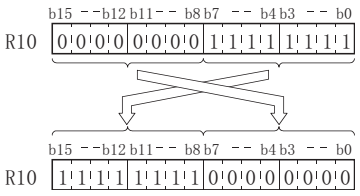
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X10
1	SWAPP	R10
3	END	

[動作]



6.5 程式分支指令

6.5.1 CJ、SCJ、JMP

Basic | High performance | Process | Redundant | Universal | LCPU

P＊＊ ：跳轉目標的指標編號（軟元件名）

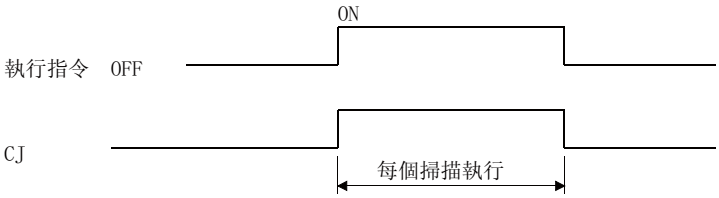
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				P
P	---								○

功 能

CJ

(1) 執行指令變爲 ON 時，執行同一程式文件內的指定的指標號的程式。

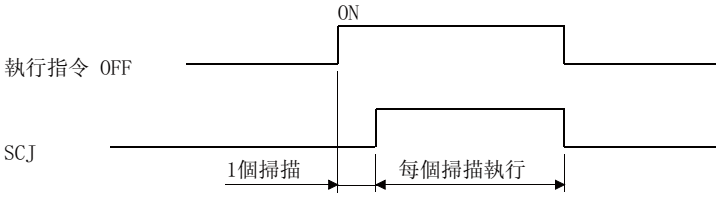
(2) 執行指令爲 OFF 時，執行下一步的程式。



SCJ

(1) 執行指令 OFF → ON 時從下一個掃描開始，執行同一程式文件內的指定的指標號的程式。

(2) 執行指令爲 OFF 或者 OFF → ON 時，執行下一步的程式。



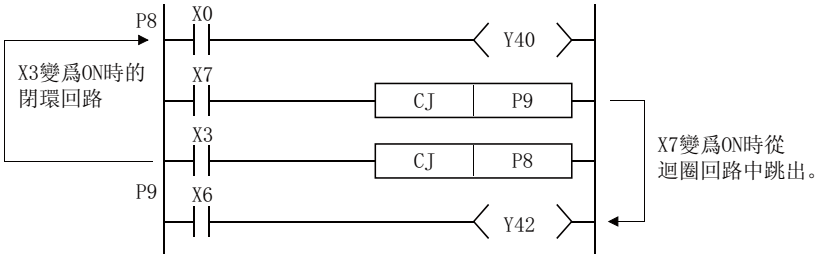
JMP

(1) 無條件地執行同一程式文件內的指定的指標號的程式。

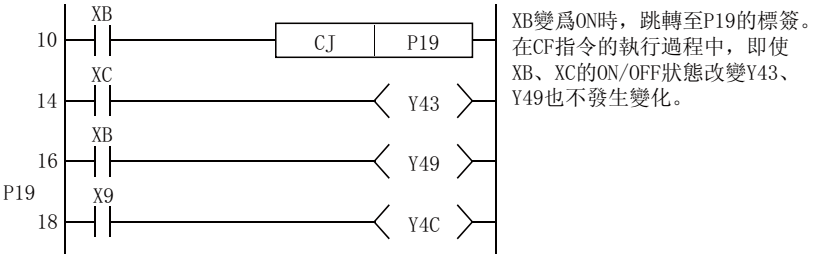
要 點

使用跳轉指令時的注意事項如下所示：

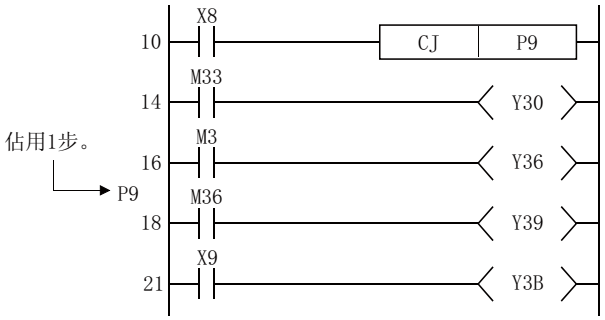
1. 在計時器線圈變爲 ON 後，如果使用 CJ、SCJ、JMP 指令對處於 ON 狀態的計時器進行跳轉，則無法進行正常計測。
2. 如果使用 CJ、SCJ、JMP 指令對 OUT 指令進行跳轉，則掃描時間將變短。
3. 如果使用 CJ、SCJ、JMP 指令強制跳轉到程式尾部，則掃描時間將變短。
4. CJ、SCJ、JMP 指令可用於跳轉到當前正執行的步之前的某一步。但是，需要考慮跳出該段迴圈回路的方法，以防止看門狗計時器超時。



5. 已通過 CJ、SCJ 、JMP 實現了的跳轉的軟元件不發生變化。



6. 標簽 (P*) 佔用 1 步。



7. 跳轉指令只能指定同一程式文件內的指標號。
8. 在跳轉運行過程中，如果跳轉到跳轉範圍內的某個指標號，則將執行跳轉目標指標號後面的程式。

出 錯

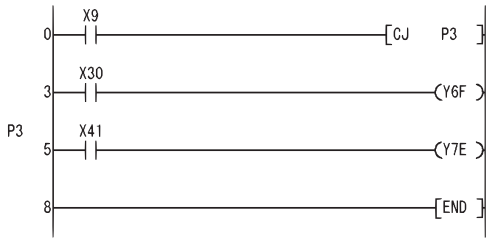
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4210	在 END 指令之前指定的指標號不存在時。 在同一程式中指定了未被作爲標簽使用的指標號時。 指定了一個位於其他程式中的公共指標時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X9 變爲 ON 時，跳轉至 P3 的程式。

[梯形圖模式]

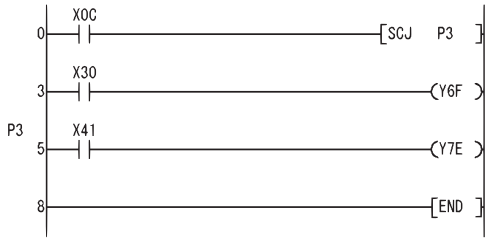


[列表模式]

步	指令	軟元件
0	LD	X9
1	CJ	P3
3	LD	X30
4	OUT	Y6F
5	P3	
6	LD	X41
7	OUT	Y7E
8	END	

(2) 以下爲 XC 變爲 ON 時，從下一個掃描跳轉至 P3 的程式。

[梯形圖模式]



[列表模式]

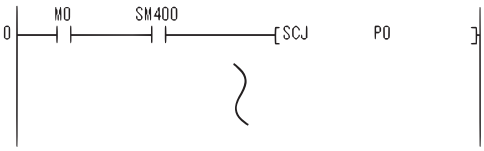
步	指令	軟元件
0	LD	XOC
1	SCJ	P3
3	LD	X30
4	OUT	Y6F
5	P3	
6	LD	X41
7	OUT	Y7E
8	END	

注意事項

(1) 在通用型 QCPU、LCPU 中，如果使用 SCJ 指令，需要在 SCJ 指令之前插入 AND SM400(或者 NOP 指令)。

[程式示例 1]

[梯形圖模式]

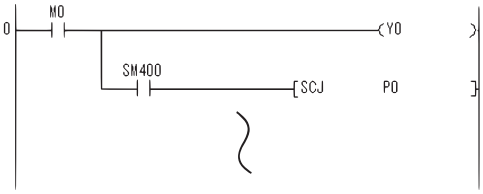


[列表模式]

步	指令	軟元件
0	LD	M0
1	AND	SM400
2	SCJ	P0

[程式示例 2]

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	M0
1	OUT	Y0
2	AND	SM400
3	SCJ	P0

6.5.2 GOEND

Basic | High performance | Process | Redundant | Universal | LCPU

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功　能

跳轉至同一程式中的 FEND 或者 END 指令。

出　錯

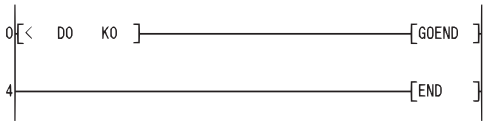
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4200	在執行 FOR 指令後執行 NEXT 指令前執行了 GOEND 指令時。	○	○	○	○	○	○
4211	在執行 CALL/ECALL 指令後執行 RET 指令前執行了 GOEND 指令時。	○	○	○	○	○	○
4221	在中斷程式執行過程中，在執行 IRET 指令前執行了 GOEND 指令時。	○	○	○	○	○	○
4230	在 CHKCIR ～ CHKEND 指令之間執行了 GOEND 指令時。	○	○	○	○	○	○
4231	在 IX ～ IXEND 指令之間執行了 GOEND 指令時。	○	○	○	○	○	○

程式示例

(1)　以下爲 D0 的值爲負數時，跳轉至 END 的程式。

[梯形圖模式]



[列表模式]

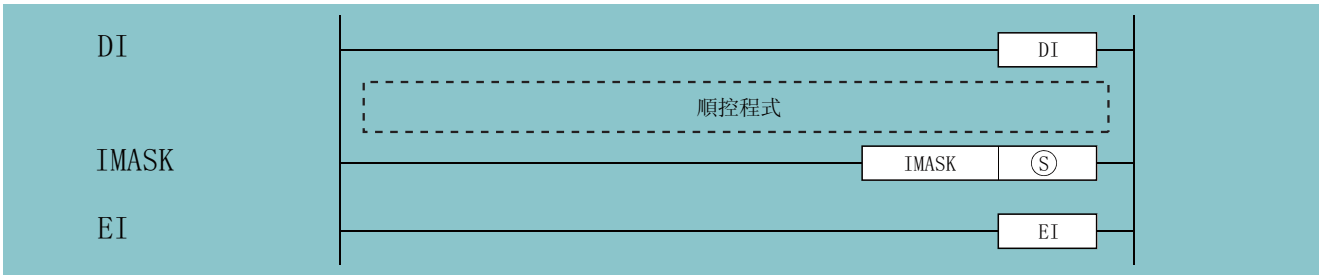
步	指令	軟元件	
0	LD<	D0	K0
3	GOEND		
4	END		

6.6 程式執行控制指令

6.6.1 DI、EI、IMASK

Basic | High performance | Process | Redundant | Universal | LCPU

1 使用基本型 QCPU 時



Ⓢ：中斷遮罩資料或者存儲中斷遮罩資料的軟元件的起始編號 (BIN16 位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ	---	○		---					

功 能

DI

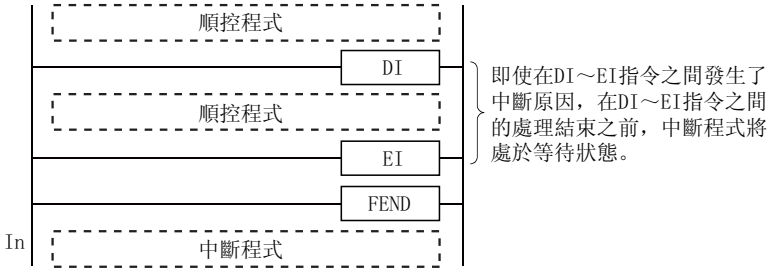
(1) 即使發生了中斷程式的啓動原因。也只能在執行了 EI 指令之後才允許執行中斷程式。

(2) 當電源接通或者 CPU 模組重定時，將變爲 DI 狀態。

EI

EI 指令用於解除 DI 指令執行時的中斷禁止狀態，並將通過 IMASK 指令設爲允許的中斷指標號的中斷程式置於可執行狀態。

未執行 IMASK 指令時，I32 ～ I47 處於中斷禁止狀態。



IMASK

(1) 根據Ⓢ中指定的軟元件開始的 8 點的位元模式，將指定中斷指標號的中斷程式置於執行允許狀態 / 執行禁止狀態。

· 1(ON) ..... 中斷程式的執行允許狀態

· 0(OFF) ..... 中斷程式的執行禁止狀態

(2) 對應於各個位的中斷指標號如下所示：

	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

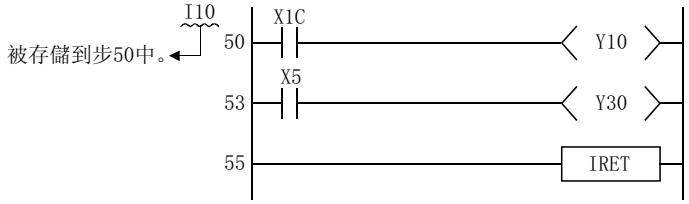
(3) 當電源接通或者 CPU 模組重定時，I0 ～ I31、I48 ～ I127 的中斷程式將變爲執行允許狀態，此外，I32 ～ I47 的中斷程式將變爲執行禁止狀態。

(4) Ⓢ、Ⓢ+1、Ⓢ+2、Ⓢ+3 ～Ⓢ+7 的軟元件狀態將被存儲到 SD715 ～ SD717、SD781 ～ SD785(IMASK 指令遮罩模式存儲區域) 中。

(5) 雖然特殊寄存器被分隔爲 SD715 ～ SD717、SD781 ～ SD785，但Ⓢ～Ⓢ+7 的軟元件號應設置爲連續的編號。

要 點

1. 中斷指標佔用 1 步。



2. 關於中斷條件，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。
3. 在中斷程式執行過程中是處於 DI(中斷禁止) 狀態的。不要在中斷程式中插入 EI 指令，不要在中斷程式執行過程中執行其他中斷程式，應避免進行多重中斷。
4. 如果在主控制中存在有 EI、DI 指令，則不管 MC 指令處於執行還是非執行狀態，都將執行 EI、DI 指令。

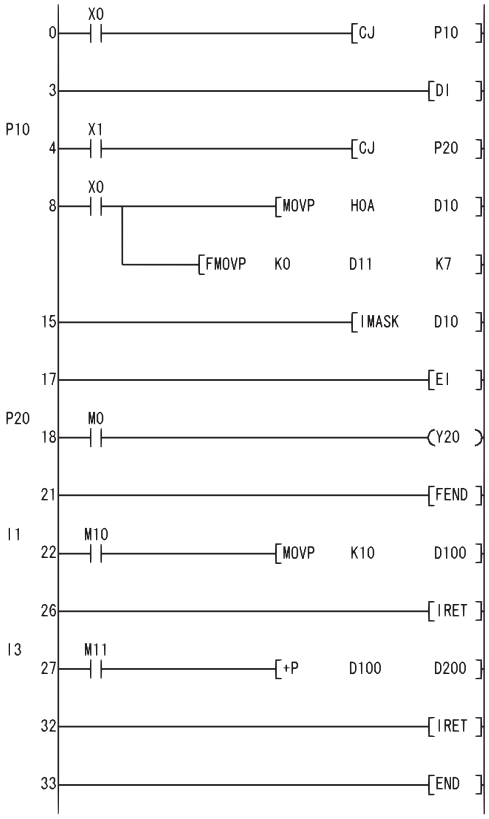
出 錯

(1) 在 DI、EI、IMASK 指令中無運算出錯。

程式示例

(1) 以下爲 X0 爲 ON 狀態時，僅將中斷指標號爲 I1、I3 的中斷程式置於執行允許狀態的程式。

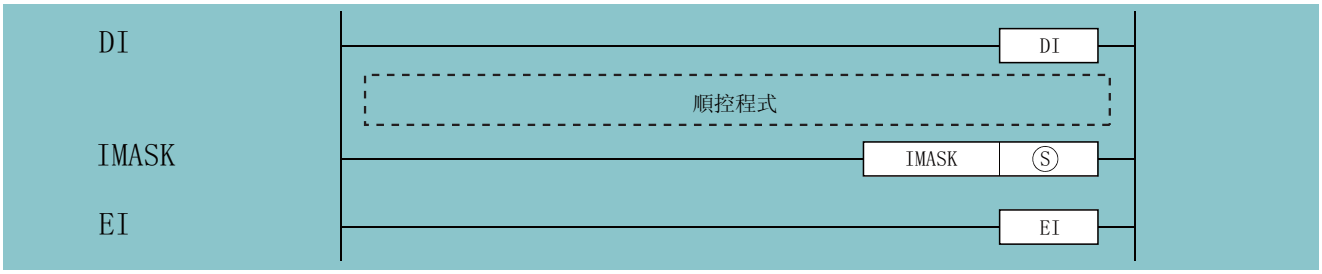
[梯形圖模式]



[列表模式]

步	指令	軟元件		
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/ 過程 CPU/ 冗餘 CPU/ 通用型 QCPU/LCPU 時



Ⓢ ：存儲中斷遮罩資料的軟元件的起始編號 (BIN16 位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ	---	○		---					

功　能

DI

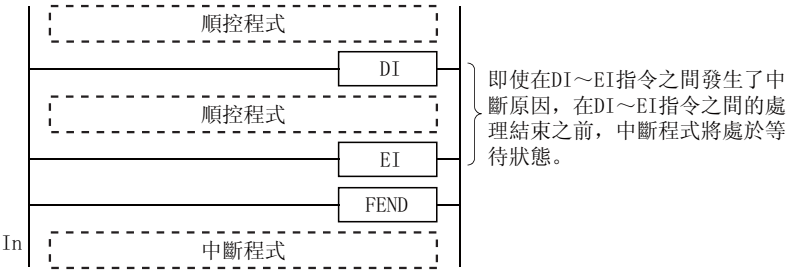
(1) 即使發生了中斷程式的啓動原因。也只能在執行了 EI 指令之後才允許執行中斷程式。

(2) 當電源接通或者 CPU 模組重定時，將變爲 DI 狀態。

EI

EI 指令用於解除 DI 指令執行時的中斷禁止狀態，並將通過 IMASK 指令設爲允許的中斷指標號的中斷程式及恒定周期執行型程式置於可執行狀態。

未執行 IMASK 指令時，I32 ～ I47 處於中斷禁止狀態。



IMASK

(1) 根據Ⓢ中指定的軟元件開始的 16 點的位元模式，將指定中斷指標號的中斷程式置於執行允許狀態 / 執行禁止狀態。

· 1(ON) ..... 中斷程式的執行允許狀態

· 0(OFF) .... 中斷程式的執行禁止狀態

(2) 對應於各個位的中斷指標號如下所示：

	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) 當電源接通或者 CPU 模組重定時，中斷程式將變爲下述狀態：

(a) 對於高性能型 QCPU、過程 CPU、冗餘 CPU

I0 ～ I31、I48 ～ I255 的中斷程式將變爲執行允許狀態，此外，I32 ～ I47 的中斷程式將變爲執行禁止狀態。

(b) 對於通用型 QCPU、LCPU

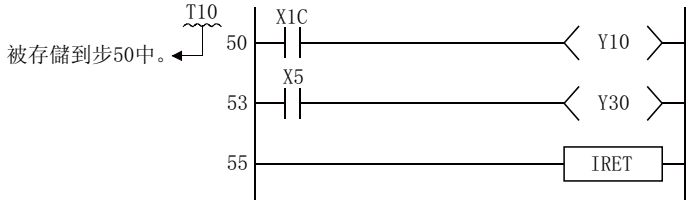
I0 ～ I31、I45 ～ I255 的中斷程式將變爲執行允許狀態，此外，I32 ～ I44 的中斷程式將變爲執行禁止狀態。

(4) Ⓢ、Ⓢ+1、Ⓢ+2、Ⓢ+3 ～Ⓢ+15 的軟元件狀態將被存儲到 SD715 ～ SD717、SD781 ～ SD793(IMASK 指令遮罩模式存儲區域) 中。

(5) 雖然特殊寄存器被分隔爲 SD715 ～ SD717、SD781 ～ SD793，但Ⓢ～Ⓢ+15 的軟元件號應設置爲連續的編號。

要 點

1. 中斷指標佔用 1 步。



2. 關於中斷條件，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。
3. 在中斷程式執行過程中是處於 DI (中斷禁止) 狀態的。不要在中斷程式中插入 EI 指令，不要在中斷程式執行過程中執行其他中斷程式，應避免進行多重中斷。
4. 如果在主控制中存在有 EI、DI 指令，則不管 MC 指令處於執行還是非執行狀態，都將執行 EI、DI 指令。

出 錯

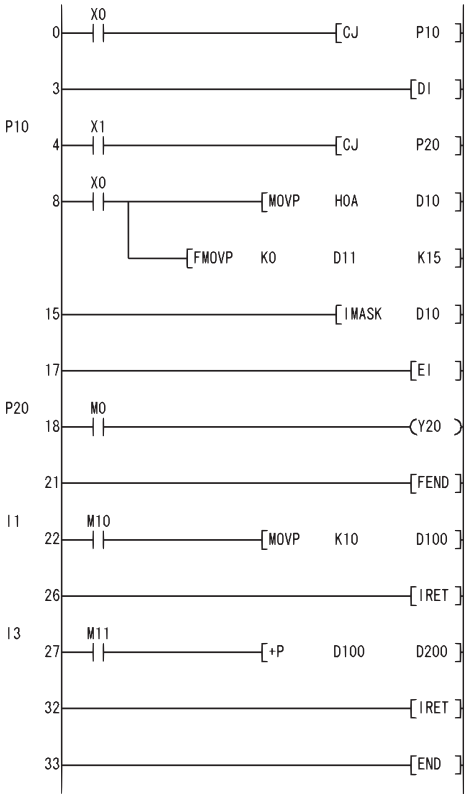
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓢ 中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 爲 ON 狀態時，將中斷指標號的中斷程式置於執行允許狀態的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X0		
1	CJ	P10		
3	DI			
4	P10			
5	LD	X1		
6	CJ	P20		
8	LD	X0		
9	MOVP	HOA	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			

6.6.2 IRET

Basic | High performance | Process | Redundant | Universal | LCPU

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功　能

(1) 表示中斷程式的處理結束。

(2) 執行 IRET 指令後，返回至順序程式處理。

出　錯

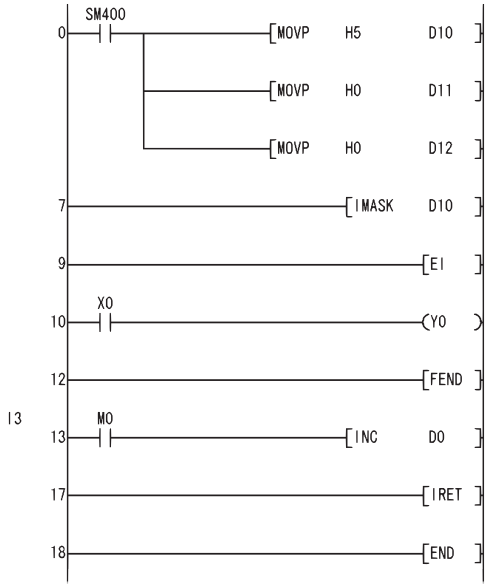
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4220	中斷號對應的指標不存在時。	○	○	○	○	○	○
4221	在發生中斷後執行 IRET 指令前執行了 END、FEND、GOEND、STOP 指令時。	○	○	○	○	○	○
4223	在執行中斷程式之前執行了 IRET 指令時。	○	○	○	○	○	○
4223	在恒定周期執行型程式中執行了 IRET 指令時。	---	---	---	---	○	○

程式示例

(1) 以下爲發生了 3 號中斷時，如果 M0 變爲 ON 則進行 D0+1 的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件	
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 RFS、RFSP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：刷新的軟元件的起始編號 (位)。
n ：刷新點數 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○ (僅 X,Y)	---							---
n	○	○							---

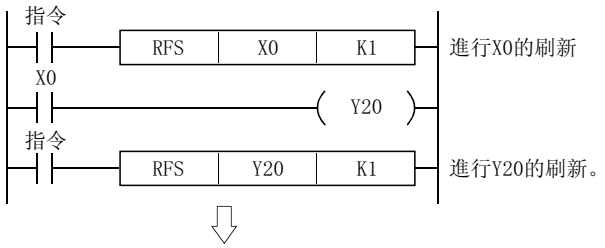
功　能

(1) 該功能是只對 1 個掃描中相應的軟元件進行刷新，並進行外部輸入的獲取或者至輸出模組的輸出的功能。

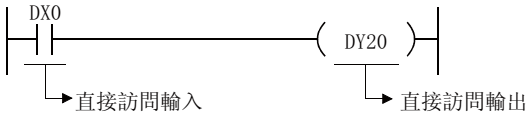
(2) 由於輸入的獲取及至外部的輸出只是在執行了程式的 END 指令之後批量地執行，因此不能在 1 個掃描中將脈衝信號輸出到外部。
執行 I/O 刷新指令時，由於在程式執行過程中對相應的輸入 (X) 或輸出 (Y) 進行強制刷新，因此可以在 1 個掃描中將脈衝信號輸出到外部。

(3) 以 1 點爲單元進行輸入 (X) 或輸出 (Y) 刷新時，應使用直接訪問輸入 (DX)、直接訪問輸出 (DY)。

[使用RFS指令的程式]



[使用直接訪問輸入、直接訪問輸出的程式]



出　錯

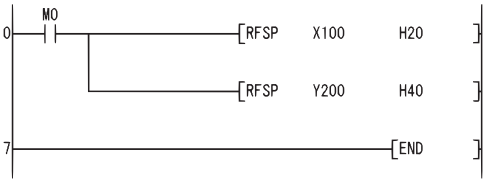
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓢ中指定的軟元件開始的 n 點的範圍超出了相鄰 I/O 的範圍時。	○	○	○	○	○	---

程式示例

(1) 以下爲 M0 變爲 ON 時，對 X100 ～ X11F、Y200 ～ Y23F 進行刷新的程式。

[梯形圖模式]



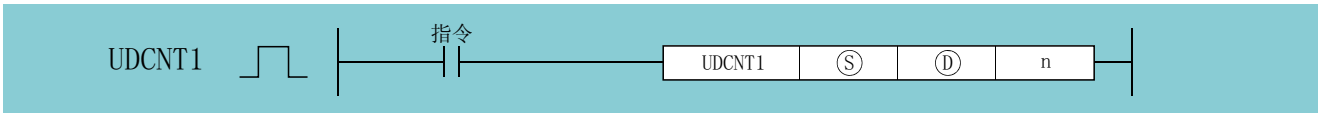
[列表模式]

步	指令	軟元件	
0	LD	M0	
1	RFSP	X100	H20
4	RFSP	Y200	H40
7	END		

6.8 其他方便的指令

6.8.1 UDCNT1

Basic High performance Process Redundant Universal LCPU



Ⓢ ：Ⓢ+0：用於計數輸入的輸入編號（位）。

Ⓢ+1：用於設置加法 / 減法計數（位）。

・OFF：加法計數（計數時數位增加）。

・ON ：減法計數（計數時數位減少）。

Ⓓ ：通過 UDCNT1 指令進行計數的計數器編號（軟元件名）。

n ：設置值 (BIN16 位）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○（僅 X）*1	---	---	---		---			---
Ⓓ	---	△*2（僅 C）	---	---		---			---
n	△*2	△*2	△*2	○					---

*1: Ⓢ只能使用 X 軟元件。
但是，X 軟元件只能在 I/O 點數（可進行實際 I/O 模組訪問的點數）的範圍內使用。

*2: 不能使用局部軟元件及各程式中設置的文件寄存器。

功 能

(1) Ⓢ中指定的輸入從 OFF 變爲 ON 時，對Ⓓ中指定的計數器的當前值進行更新。

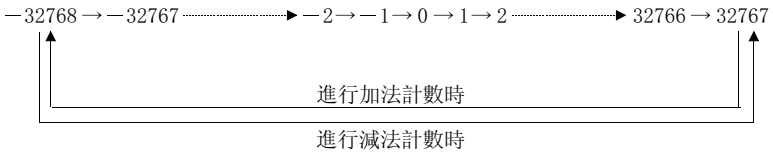
(2) 計數的方向取決於Ⓢ+1 中指定輸入的 ON/OFF 狀態。
・OFF：加法計數（以增加當前值的方式計數）
・ON：減法計數（以減少當前值的方式計數）

(3) 計數處理的執行方式如下：

・ 在加法計數過程中，當前值變爲與 n 中指定的設置值相等時，則Ⓓ中指定的計數器的觸點將變爲 ON。
但是，即使Ⓓ中指定的計數器的觸點變爲 ON 時，當前值的計數也將繼續。（參閱程式示例 (1)）

・ 在減法計數過程中，當前值變爲設置值 -1 時，則Ⓓ中指定的計數器的觸點將變爲 OFF。（參閱程式示例 (1)）

・ Ⓓ中指定的計數器是環型計數器。
在當前值爲 32767 時，如果正進行加法計數，則當前值將變爲 -32768。
此外，在當前值爲 -32768 時，如果正進行減法計數，則當前值將變爲 32767。
當前值的計數處理內容如下所示：



(4) 當執行指令從 OFF 變爲 ON 時，使用了 UDCNT1 指令的計數處理將開始計數；當執行指令從 ON 變爲 OFF 時，中止計數。
當執行指令再次從 OFF 變爲 ON 時，從計數中止時的值開始重新進行計數。

(5) Ⓓ中指定的計數器的當前值清除及觸點的 OFF 是通過 RST 指令進行的。

要 點

1. UDCNT1 指令將變數的軟元件資料登錄到 CPU 模組的工作區，而實際的計數動作是通過系統中斷進行處理。
   (當執行指令變爲 OFF 或者 CPU 模組 STOP → RUN 時，CPU 模組工作區中登錄的軟元件資料將被清除)。
   因此，可計數的脈衝的 ON 和 OFF 時間必須比 CPU 模組的中斷間隔長。
   CPU 模組的中斷間隔如下所示。

CPU 模組型號	中斷間隔
高性能型 QCPU、過程 CPU、通用型 QCPU、LCPU	1ms

2. 在使用了 UDCNT1 指令的計數過程中 (執行指令處於 ON 狀態時)，不能對設置值進行更改。
   如果要更改設置值，應先將執行指令置爲 OFF。
3. 其他指令不能使用 UDCNT1 指令中指定的計數器。
   如果這些計數器爲其他指令所使用，將不能進行正常計數。
4. 在所有正執行的程式中，UDCNT1 指令最多可以使用 6 次。
   第 7 個及以後的 UDCNT1 指令將執行無處理。

出 錯

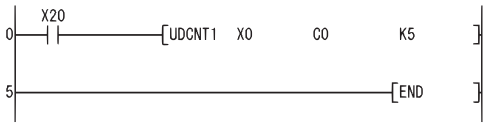
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	ⓢ 中指定的軟元件超出了相應軟元件的範圍時。	---	○	---	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 之後，將 X0 的 OFF → ON 次數通過 C0(加法計數器 / 減法計數器) 進行計數的程式。

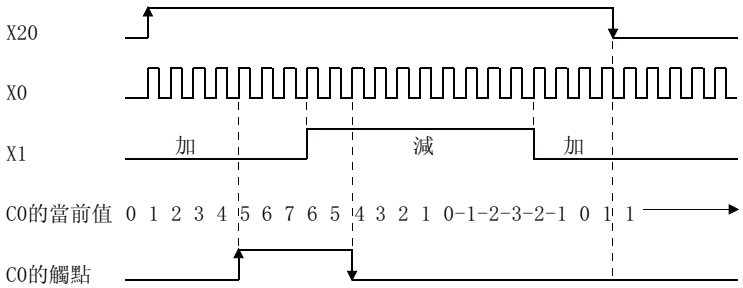
[梯形圖模式]



[列表模式]

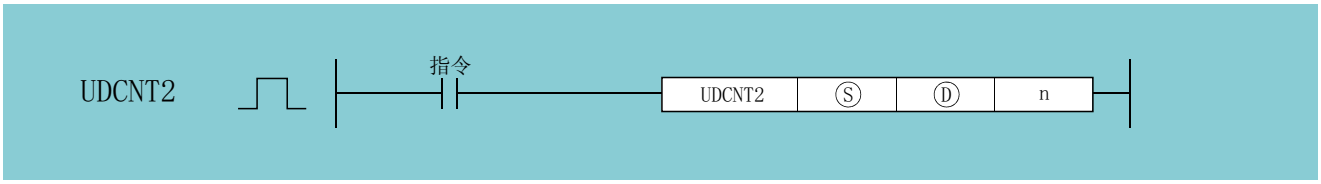
步	指令	軟元件		
0	LD	X20		
1	UDCNT1	X0	C0	K5
5	END			

[動作]



6.8.2 UDCNT2

Basic High performance Process Redundant Universal LCPU



Ⓢ : Ⓢ +0: 用於計數輸入的輸入編號 (A 相脈衝) (位)。
Ⓢ +1: 用於計數輸入的輸入編號 (B 相脈衝) (位)。

Ⓓ : 通過 UDCNT2 指令進行計數的計數器編號 (軟元件名)

n : 設置值 (BIN16 位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○ (僅 X) *1	---	---	---		---			---
Ⓓ	---	△ *2 (僅 C)	---	---		---			---
n	△ *2	△ *2	△ *2	○					---

*1: Ⓢ 只能使用 X 軟元件。
但是，X 軟元件只能在 I/O 點數 (可進行實際 I/O 模組訪問的點數) 的範圍內使用。

*2: 不能使用局部軟元件及各程式中設置的文件寄存器。

功 能

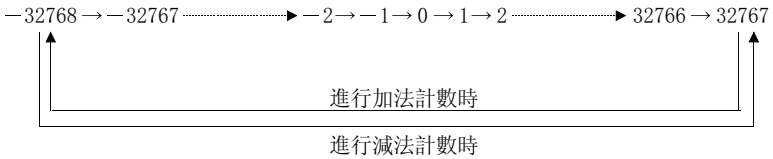
(1) 根據Ⓢ中指定的輸入 (A 相脈衝) 及Ⓢ+1 中指定的輸入 (B 相脈衝) 狀態，對Ⓓ中指定的計數器的當前值進行更新。

(2) 計數方向的確定如下所示。

- 當Ⓢ爲 ON 時，如果Ⓢ+1 從 OFF 變爲 ON，則執行加法計數 (計數器的當前值增加)。
- 當Ⓢ爲 ON 時，如果Ⓢ+1 從 ON 變爲 OFF，則執行減法計數運算 (計數器的當前值減少)。
- 如果Ⓢ为 OFF，則不執行計數。

(3) 計數處理的執行方式如下：

- 在加法計數過程中，當前值變爲與 n 中指定的設置值相等時，則Ⓓ中指定的計數器的觸點將變爲 ON。
  但是，即使Ⓓ中指定的計數器的觸點變爲 ON 時，當前值的計數也將繼續。(參閱程式示例 (1))
- 在減法計數過程中，當前值變爲設置值 -1 時，則Ⓓ中指定的計數器的觸點將變爲 OFF。(參閱程式示例 (1))
- Ⓓ中指定的計數器是環型計數器。
  在當前值爲 32767 時，如果正進行加法計數，則當前值將變爲 -32768。
  此外，在當前值爲 -32768 時，如果正進行減法計數，則當前值將變爲 32767。
  當前值的計數處理內容如下所示：



(4) 當執行指令從 OFF 變爲 ON 時，使用了 UDCNT2 指令的計數處理將開始計數；當執行指令從 ON 變爲 OFF 時，中止計數。
當執行指令再次從 OFF 變爲 ON 時，從計數中止時的值開始重新進行計數。

(5) Ⓓ中指定的計數器的當前值清除及觸點的 OFF 是通過 RST 指令進行的。

要 點

1. UDCNT2 指令將變數的軟元件資料登錄到 CPU 模組的工作區，而實際的計數動作是通過系統中斷進行處理。
   (當執行指令變爲 OFF 或者 CPU 模組 STOP → RUN 時，CPU 模組工作區中登錄的軟元件資料將被清除)。
   因此，可計數的脈衝的 ON 和 OFF 時間必須比 CPU 模組的中斷間隔長。
   CPU 模組的中斷間隔如下所示。

CPU 模組型號	中斷間隔
高性能型 QCPU、過程 CPU、通用型 QCPU、LCPU	1ms

2. 在使用了 UDCNT2 指令的計數過程中 (執行指令處於 ON 狀態時)，不能對設置值進行更改。
   如果要更改設置值，應先將執行指令置爲 OFF。
3. 其他指令不能使用 UDCNT2 指令中指定的計數器。
   如果這些計數器爲其他指令所使用，將不能進行正常計數。
4. 在所有正執行的程式中，UDCNT2 指令最多可以使用 5 次。
   第 6 個及以後的 UDCNT2 指令將執行無處理。

出 錯

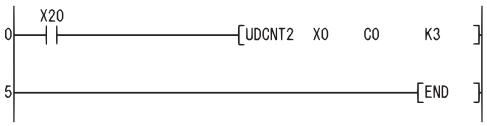
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓢ 中指定的軟元件超出了相應軟元件的範圍時。	---	○	---	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 之後，將 X0、X1 的狀態通過 C0(加法計數器 / 減法計數器) 進行計數的程式。

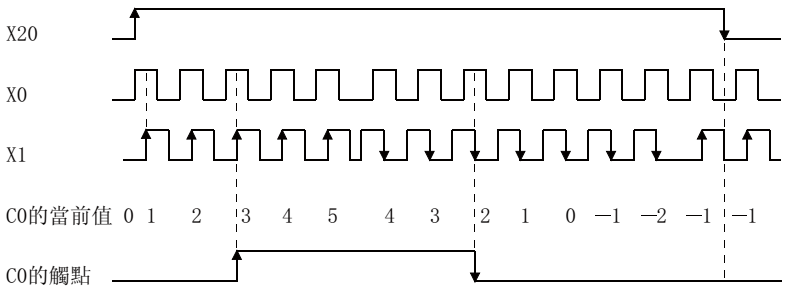
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X20		
1	UDCNT2	X0	C0	K3
5	END			

[動作]



6.8.3 TTMR

Basic High performance Process Redundant Universal LCPU

Ⓓ ：Ⓓ+0：存儲測量值的軟元件 (BIN16 位)。
Ⓓ+1：爲 CPU 模組的系統所用 (BIN16 位)。
n ：測量值的乘數 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓓ	---	○		---					---
n	---	○		○					---

功　能

(1) 對執行指令處於 ON 狀態的時間以秒爲單位進行測量，然後乘以 n 中指定的乘數，並將結果存儲到Ⓓ中指定的軟元件中。

(2) 當執行指令從 OFF 變爲 ON 時，對Ⓓ+0、Ⓓ+1 中指定的軟元件進行清除。

(3) n 中可指定的乘數如下所示。

n	乘數
0	1
1	10
2	100

要 點

1. 在執行 TTMR 指令時進行時間計測。
   如果使用 JMP 指令等對 TTMR 指令進行跳轉，將無法進行正確測量。
2. 在 TTMR 指令的執行過程中不要更改 n 中指定的乘數。
   更改 n 中指定的乘數會導致值不正確。
3. TTMR 指令也可用於低速執行型程式中。
4. 由於Ⓓ+1 中指定的軟元件爲 CPU 模組的系統所使用，因此用戶不要對該值進行變更。
   如果用戶更改了該值，則Ⓓ中指定的軟元件中存儲的值將爲不正確的值。

(4) 如果 n 中指定的值超出了 0 ～ 2 的範圍，則執行無處理。

出　錯

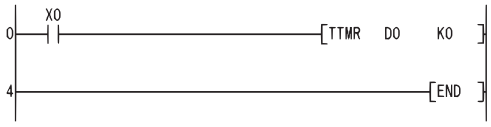
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 X0 處於 ON 狀態的時間存儲到 D0 中的程式。

[梯形圖模式]

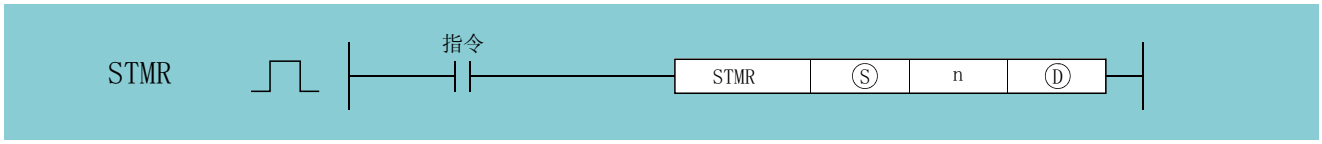


[列表模式]

步	指令	軟元件	
0	LD	X0	
1	TTMR	D0	K0
4	END		

6.8.4 STMR

Basic High performance Process Redundant Universal LCPU



Ⓢ ：計時器編號 (字)。
n ：設置值 (BIN16 位)。
Ⓓ ： Ⓓ+0：OFF 延遲計時器輸出 (位)。
Ⓓ+1：OFF 後一次計時器輸出 (位)。
Ⓓ+2：ON 後一次計時器輸出 (位)。
Ⓓ+3：ON 延遲 +OFF 延遲計時器 (位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	△*1	---	---		---			---
n	○	○	○	○					---
Ⓓ	○	---	---	---		---			---

*1: 僅計時器 (T) 可用。

功 能

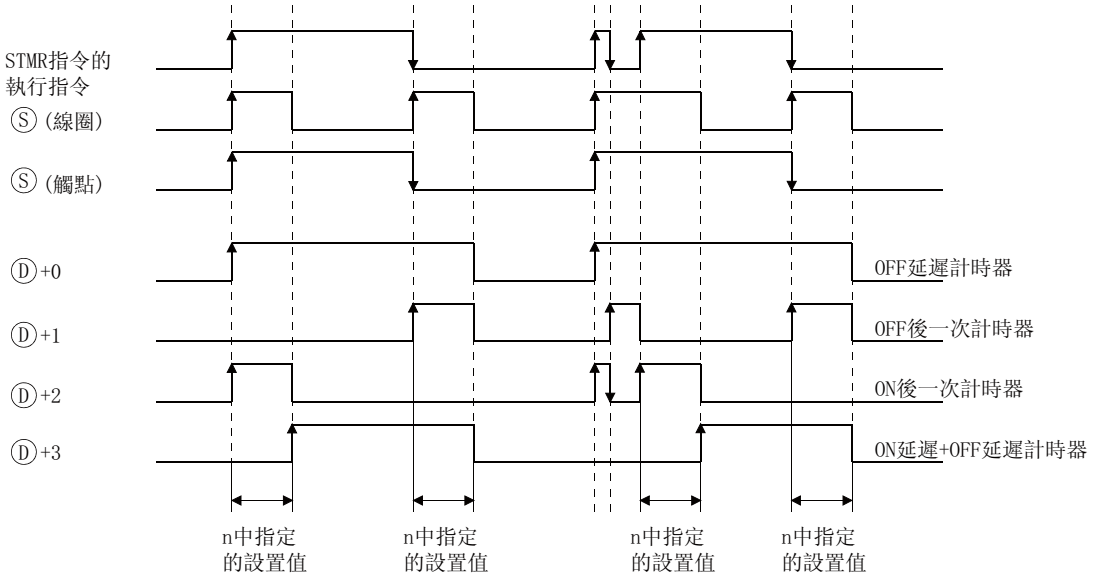
(1) 在 STMR 指令中，使用從Ⓓ中指定的軟元件開始的 4 點，進行 4 種類型的計時器輸出。

- OFF 延遲計時器輸出 (Ⓓ+0)
  在 STMR 指令的執行指令的上升沿變爲 ON，在執行指令的下降沿後經過了 n 中指定的時間之後變爲 OFF。
- OFF 後一次計時器輸出 (Ⓓ+1)
  在 STMR 指令的執行指令的下降沿變爲 ON，經過了 n 中指定的時間後變爲 OFF。
- ON 後一次計時器輸出 (Ⓓ+2)
  在 STMR 指令的執行指令的上升沿變爲 ON，經過了 n 中指定的時間後或者 STMR 指令的執行指令 OFF 時變爲 OFF。
- ON 延遲 +OFF 延遲計時器 (Ⓓ+3)
  在計時器線圈的下降沿變爲 ON，在執行指令的下降沿後經過了 n 中指定的時間之後變爲 OFF。

(2) Ⓢ中指定的計時器線圈通過 STMR 指令的執行指令的上升沿或下降沿變爲 ON 後，開始當前值的計測。

- 計時器線圈在到達 n 中指定的設置值之前進行計測，變爲 “ 時間到 ” 時變爲 OFF。
- 計時器線圈在到達 “ 時間到 ” 之前如果 STMR 指令的執行指令變爲 OFF，則保持爲 ON 狀態不變。
  此時計時器的計測將繼續進行。
  如果 STMR 指令再次變爲 ON，則在將當前值置爲 0 後，重新開始計測。

(3) 計時器觸點在 STMR 指令的執行指令的上升沿變爲 ON，在計時器線圈的下降沿之後，在 STMR 指令的執行指令的下降沿變爲 OFF。
計時器觸點爲 CPU 模組的系統所用，因此用戶不能使用。



(4) STMR 指令中指定的計時器的當前值的計測與 STMR 指令的執行指令的 ON/OFF 狀態無關。
通過 JMP 指令等跳過了 STMR 指令時，將無法進行正常計測。

(5) Ⓓ中指定的計時器的計測單位與低速計時器相同。

(6) n 的設置值的指定範圍爲 0 ～ 32767。
0 ～ 32767 以外時將執行無處理。

(7) Ⓢ中指定的計時器不能被用於 OUT 指令。
如果 STMR 指令與 OUT 指令使用了相同的計時器編號，將無法執行正常動作。

出　錯

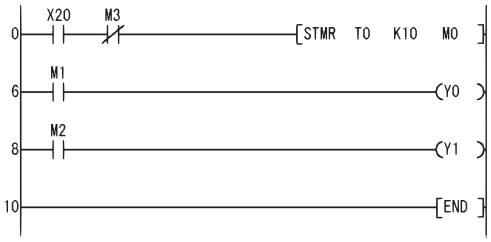
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，Y0、Y1 每隔 1 秒 ON/OFF(閃爍) 的程式。
(計時器使用 100ms 計時器)

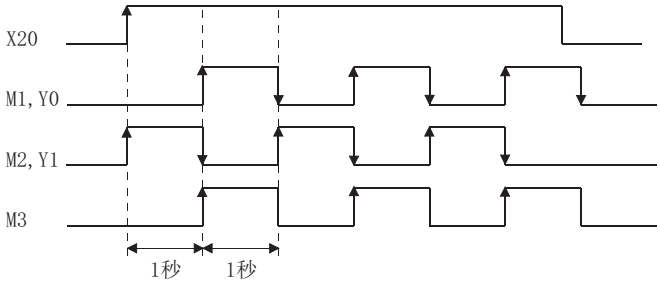
[梯形圖模式]



[列表模式]

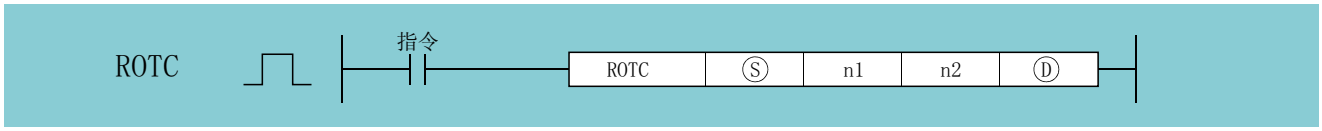
步	指令	軟元件		
0	LD	X20		
1	ANI	M3		
2	STMR	T0	K10	M0
6	LD	M1		
7	OUT	Y0		
8	LD	M2		
9	OUT	Y1		
10	END			

旋轉台就近控制 (ROTC)



6.8.5 ROTC

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ： Ⓢ+0：用於測量旋轉台旋轉次數。(系統用)(BIN16 位元)。
Ⓢ+1：調用窗口編號 (BIN16 位元)。
Ⓢ+2：調用部件編號 (BIN16 位)。
n1 ： 旋轉台的分區數 (2 ～ 32767)(BIN16 位)。
n2 ： 低速區間數 (0 ～ n1 以內的值)(BIN16 位)。
Ⓓ ： Ⓓ+0：A 相輸入信號 (位元)。
Ⓓ+1：B 相輸入信號 (位元)。
Ⓓ+2：0 點檢測輸入信號 (位元)。
Ⓓ+3：高速正轉輸出信號 (系統用)(位元)。
Ⓓ+4：低速正轉輸出信號 (系統用)(位元)。
Ⓓ+5：停止輸出信號 (系統用)(位元)。
Ⓓ+6：低速逆轉輸出信號 (系統用)(位元)。
Ⓓ+7：高速逆轉輸出信號 (系統用)(位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---					---
n1	○	○		○					---
n2	○	○		○					---
Ⓓ	○	---		---					---

功　能

(1) 該功能是爲了執行以下控制： 在以 n1 中指定的值進行了均等分區的旋轉臺上，爲了對Ⓢ+2 中指定編號的部件進行取放，使旋轉台就近旋轉到Ⓢ+1 中指定編號的窗口位置處。

(2) 部件編號以及視窗編號是基於對以逆時針旋轉方向分配的部件進行控制。

(3) Ⓢ+0 是系統用計數器，用於對 0 號視窗中的部件進行計數。
不要通過順控程式等對其資料進行改寫。
如果用戶進行了改寫，將無法進行正常控制。

(4) n2 的值應小於 n1 中指定的旋轉台的分區數。

(5) Ⓓ+0 及Ⓓ+1 是用於檢測旋轉台的正轉 / 逆轉的 A 相及 B 相輸入信號。
旋轉方向的判別是通過 A 相爲 ON 狀態時的 B 相的上升沿 / 下降沿來進行的。
・B 相上升沿時： 正轉 (順時針旋轉)
・B 相下降沿時： 逆轉 (逆時針旋轉)

(6) Ⓓ+2 是 0 號部件到達 0 號視窗時將變爲 ON 的 0 點檢測信號。
在 ROTC 指令的執行過程中，Ⓓ+2 中指定的軟元件變爲 ON 時，Ⓢ+0 將被清除。
應預先進行此清除操作之後，再通過 ROTC 指令開始就近控制。

(7) ⓓ+3 ～ⓓ+7 是用於進行旋轉台動作控制的輸出信號。

根據 ROTC 指令的執行結果，ⓓ+3 ～ⓓ+7 中的某個輸出信號將變爲 ON。

(8) ROTC 指令的執行指令變爲 OFF 時，不執行就近控制，將ⓓ+3 ～ⓓ+7 全部置爲 OFF。

(9) 在所有正在執行的程式中只能使用 1 次 ROTC 指令。
如果使用了 2 次或以上，將無法正常執行。

(10) Ⓢ+0 ～Ⓢ+2 或者 n2 的值大於 n1 的值時，將執行無處理。

出　錯

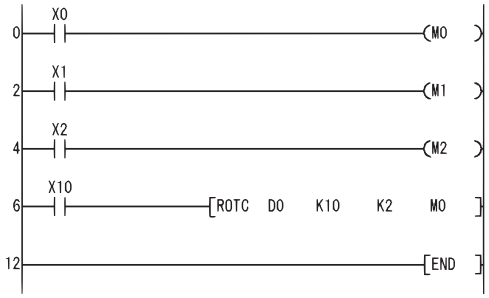
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓢ或ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

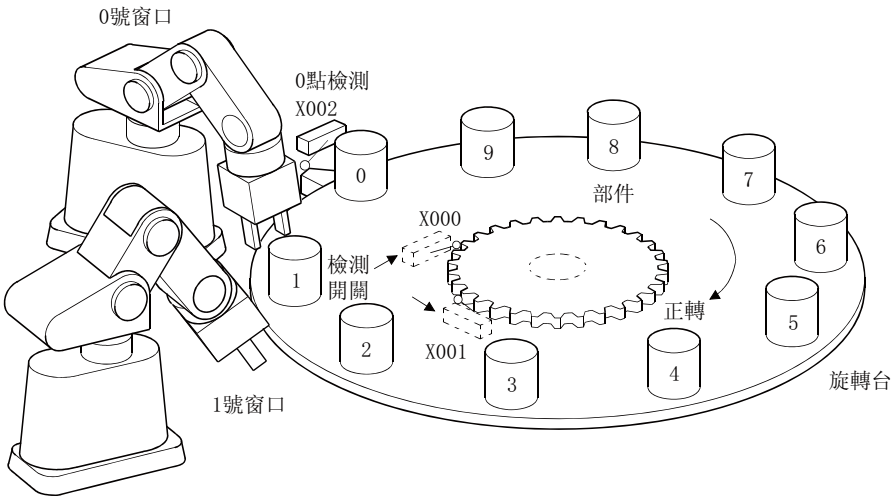
(1) 以下爲將放置在 10 等份的旋轉臺上的 D2 編號的部件在 D1 編號窗口處進行取放，使旋轉台在前後 2 個分區低速旋轉時，求出馬達的旋轉方向及控制速度的程式。

[梯形圖模式]



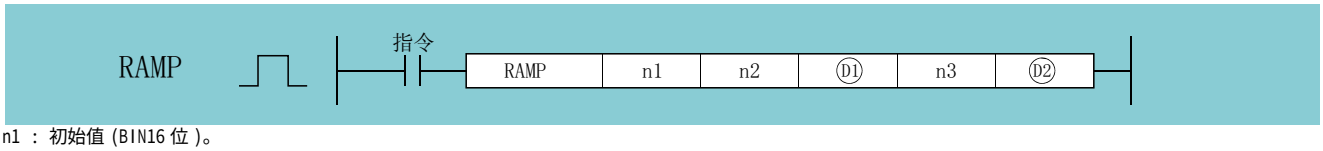
[列表模式]

步	指令	軟元件			
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 RAMP

Basic | High performance | Process | Redundant | Universal | LCPU



n1 : 初始值 (BIN16 位)。
n2 : 最終值 (BIN16 位)。
D1 : D1+0: 當前值 (BIN16 位)。
D1+1: 執行次數 (BIN16 位)。
n3 : 移位次數 (BIN16 位)。
D2 : D2+0: 結束軟元件 (位)。
D2+1: 結束時資料保持選擇位元 (位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n1	○	○						○	---
n2	○	○						○	---
D1	○	○						---	---
n3	○	○						○	---
D2	○	---						---	---

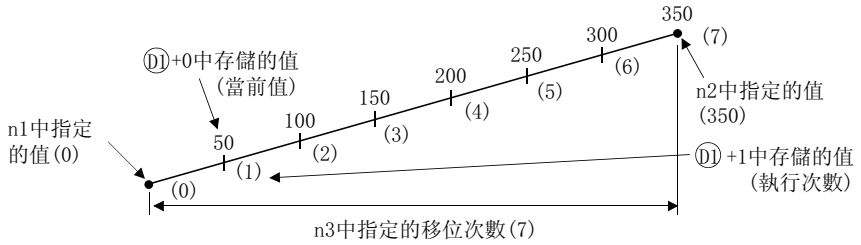
功 能

(1) 執行指令變爲 ON 時，進行以下處理：

- 從 n1 中指定的值開始至 n2 中指定的值爲止，按 n3 中指定的次數進行移位。
- 在 n3 中，對從 n1 處移位至 n2 處的掃描次數 (移位次數) 進行指定。
  如果不符合 0<n3<32768 的條件，則進行無處理。
- D1+1 爲系統所用，用於存儲本指令的執行次數。
- 1 次 (1 個掃描) 的變化值通過下列公式進行計算：

$$1\text{次的變化值}=\frac{(\text{n2中指定的值})-(\text{n1中指定的值})}{(\text{n3中指定的值})}$$

例 通過 7 個掃描從 0 變爲 350 時的示意圖如下所示。



如果使用算出的 1 次的變化值不能整除，則需要進行補償，以便按 n3 中指定的移位次數達到 n2 中指定的值。
因此有可能導致不能生成直線斜坡。

(2) 執行了 n3 中指定的移位次數的掃描後，D2+0 中指定的結束軟元件將變爲 ON。

結束軟元件的 ON/OFF 狀態以及D1+0 的值取決於D2+1 中指定的軟元件的 ON/OFF 狀態。

- D2+1 變爲 OFF 時，在下 1 個掃描中將D2+0 變爲 OFF 後，RAMP 指令將從初始值開始再次進行移位。
- D2+1 變爲 ON 時，D2+0 保持爲 ON 狀態不變，D1+0 的值不發生變化。

(3) 如果在本指令的執行過程中其執行指令變爲 OFF，則此後的D1+0 的值將不發生變化。
其執行指令再次變爲 ON 時，RAMP 指令將從初始值開始再次進行移位。

(4) 在Ⓓ2+0 中指定的結束軟元件變爲 ON 之前，不要對 n1 及 n2 的值進行變更。

由於每個掃描是以相同的計算公式算出Ⓓ1+1 中存儲的值，因此如果對 n1 及 n2 的值進行了變更，有可能導致急劇變化。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓓ1或Ⓓ2中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

注意事項

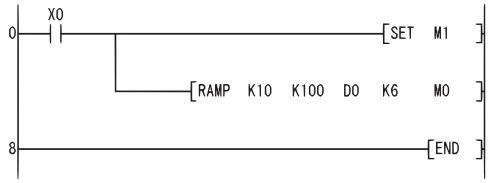
(1) 對Ⓓ1進行了位軟元件的位數指定時，只有在滿足以下條件的情況下其指定才有效。

· 位數的指定 ： K8

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D0 的值以 6 個掃描從 10 變爲 100，在移位結束時對 D0 的內容進行保持的程式。

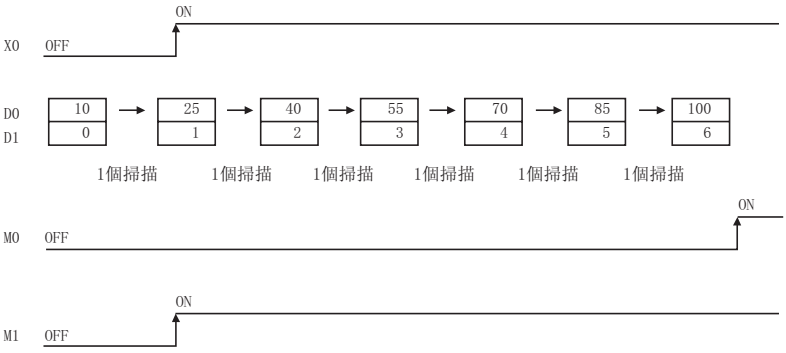
[梯形圖模式]



[列表模式]

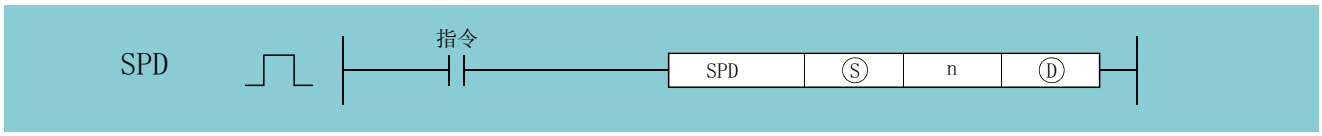
步	指令	軟元件				
0	LD	X0				
1	SET	M1				
2	RAMP	K10	K100	D0	K6	M0
8	END					

[時序圖]



6.8.7 SPD

Basic | High performance | Process | Redundant | Universal | LCPU



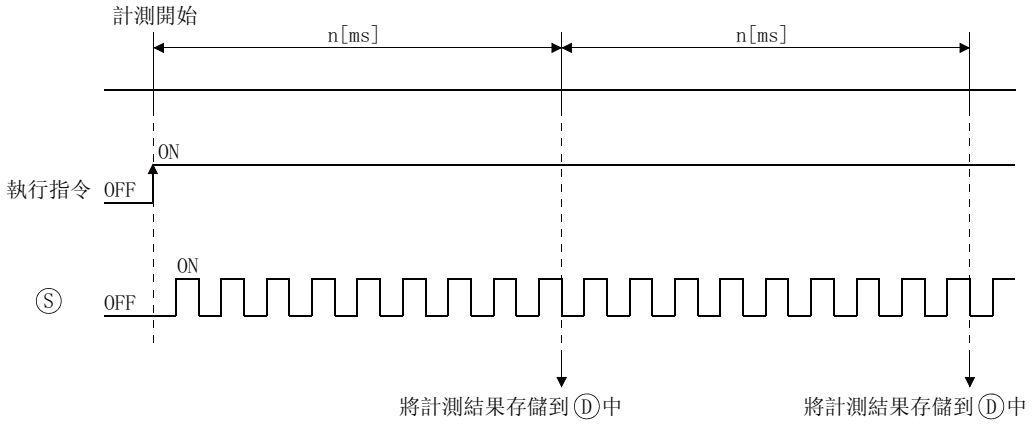
Ⓢ ：脈衝輸入 (位)。
n ：測定時間 (單位： ms)(BIN16 位)。
Ⓓ ：存儲測定結果的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○ (僅 X)	---		---					---
n	△ *1	△ *1		○					---
Ⓓ	---	△ *1		---					---

*1: 不能使用局部軟元件及各程式中設置的文件寄存器。

功 能

(1) 將Ⓢ中指定的軟元件輸入的 OFF → ON 的次數在 n 中指定的時間內進行計數，並將計數結果存儲到Ⓓ中指定的軟元件中。



(2) 如果通過 SPD 指令進行的計測結束，將再次從 0 開始進行計測。
中止通過 SPD 指令進行的計測時，應將執行指令置於 OFF。

要 點

1. SPD 指令將變數軟元件中的資料登錄到 CPU 模組的工作區後，實際的計數動作將通過系統中斷進行。
(將執行指令置於 OFF 或者 CPU 模組 STOP → RUN 時，CPU 模組的工作區中登錄的軟元件資料將被清除)。
因此，可計數的脈衝的 ON 和 OFF 時間必須比 CPU 模組的中斷間隔長。
CPU 模組的中斷間隔如下所示。

CPU 模組型號	中斷間隔
高性能型 QCPU、過程 CPU、通用型 QCPU、LCPU	1ms

2. 使用高性能型 QCPU 或過程 CPU 時：
當 n=0 時，執行無處理。
3. 在所有正執行的程式中，SPD 指令最多可以使用 6 次。
第 7 個以及以後的 SPD 指令將執行無處理。
4. 在通過 SPD 指令進行測量的過程中 (指令輸入處於 ON 狀態)，不能進行設置值的變更。
在進行設置值變更時，應先將指令輸入置於 OFF 之後再進行變更。

出　錯

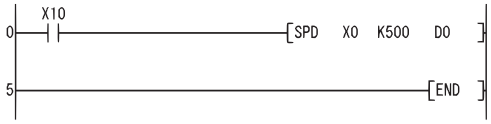
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓢ 中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X10 變爲 ON 時，在 500ms 的時間內對輸入到 X0 中的脈衝進行計測，並將結果存儲到 D0 中的程式。

[梯形圖模式]

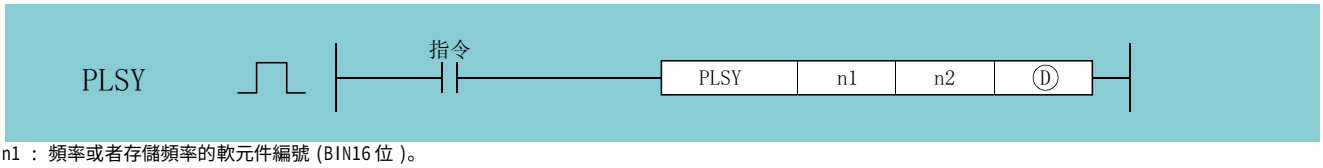


[列表模式]

步	指令	軟元件		
0	LD	X10		
1	SPD	X0	K500	D0
5	END			

6.8.8 PLSY

Basic　High performance　Process　Redundant　Universal　LCPU



n1 ： 頻率或者存儲頻率的軟元件編號 (BIN16 位)。
n2 ： 輸出次數或者存儲輸出次數的軟元件編號 (BIN16 位)。
Ⓓ ：脈衝輸出的軟元件編號 (位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n1	○				○				---
n2	○				○				---
Ⓓ	△ *1				---				---

*1:　僅輸出 (Y) 可用。

功　能

(1) 將 n1 中指定頻率的脈衝按 n2 中指定的次數存儲到Ⓓ中指定的輸出編號 (Y) 的輸出模組中。

(2) n1 的可設置頻率範圍爲 1Hz ～ 100Hz。
n1 的值超出了 1Hz ～ 100Hz 的範圍時，將執行無處理。

(3) n2 的輸出次數的可設置範圍爲 0 ～ 65535(0000_H ～ $FFFF_H$)。
n2 的值爲 0 時，將連續輸出脈衝。

(4) Ⓓ中指定的脈衝輸出只能指定與輸出模組相對應的輸出號。

(5) 通過 PLSY 指令的執行指令的上升沿開始脈衝輸出。
PLSY 指令的執行指令變爲 OFF 時，脈衝輸出將停止。

要 點

1. PLSY 指令將變數軟元件資料登錄到 CPU 模組的工作區後，實際的輸出動作將通過系統中斷進行處理。
   (將執行指令置於 OFF 或者 CPU 模組 STOP → RUN 時，CPU 模組的工作區中登錄的軟元件資料將被清除)。
   因此，可輸出的脈衝的 ON 和 OFF 時間必須比 CPU 模組的中斷間隔長。
   CPU 模組的中斷間隔如下所示。

CPU 模組型號	中斷間隔
高性能型 QCPU、過程 CPU、通用型 QCPU、LCPU	1 ms

2. 在通過 PLSY 指令進行的脈衝輸出過程中 (指令輸入處於 ON 狀態)，不要對 PLSY 指令的變數進行變更。
   進行變數變更時，應先將指令輸入置於 OFF 之後再進行變更。
3. 在 CPU 模組的正在執行的全部程式中，只能使用 1 次 PLSY 指令。
   第 2 次及以後的 PLSY 指令將被執行無處理。

出 錯

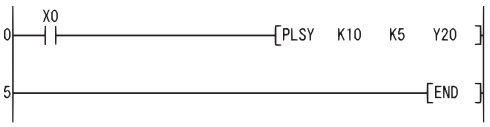
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 10Hz 的脈衝輸出 5 次到 Y20 中的程式。

[梯形圖模式]

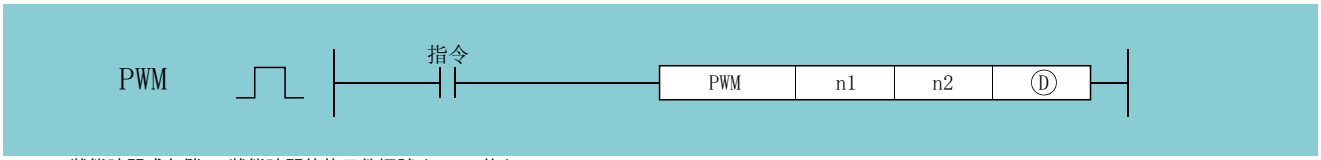


[列表模式]

步	指令	軟元件		
0	LD	X0		
1	PLSY	K10	K5	Y20
5	END			

6.8.9 PWM

Basic | High performance | Process | Redundant | Universal | LCPU



n1 : ON 狀態時間或存儲 ON 狀態時間的軟元件編號 (BIN16 位)。
n2 : 周期或存儲周期的軟元件編號 (BIN16 位)。

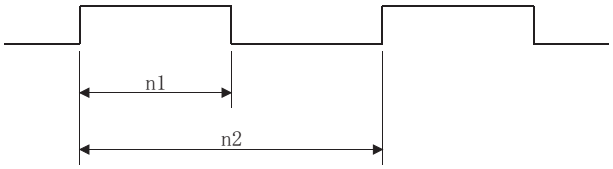
ⓓ : 脈衝輸出的軟元件編號 (位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n1	○	○							---
n2	○	○							---
ⓓ	△ *1	---							---

*1: 僅輸出 (Y) 可用。

功　能

(1) 將 n1 中指定的 ON 狀態時間及 n2 中指定的周期脈衝輸入到ⓓ中指定的輸出模組中。



(2) n1、n2 的設置時間如下所示：

CPU 模組型號	n1、n2 的設置範圍 [ms]*2
高性能型 QCPU、過程 CPU、通用型 QCPU、LCPU	1 ～ 65535(0001_H ～ $FFFF_H$)

*2:　n1 中指定的值應小於 n2 中指定的值。

要 點

1. PWM 指令將變數軟元件資料登錄到 CPU 模組的工作區後，實際的輸出動作將通過系統中斷進行處理。
   (將執行指令置於 OFF 或者 CPU 模組 STOP → RUN 時，CPU 模組的工作區中登錄的軟元件資料將被清除)。
   CPU 模組的中斷間隔如下所示。

CPU 模組型號	n1、n2 中斷間隔
高性能型 QCPU、過程 CPU、通用型 QCPU、LCPU	1ms

   因此，在 CPU 模組正在執行的全部程式中，只能使用 1 次 PWM 指令。
2. 在以下情況下將執行無處理：
   · n1、n2 爲 0 時。
   · n1 ≥ n2 時。
   · 執行了 2 次或以上的 PWM 指令時。
3. 在通過 PWM 指令進行的脈衝輸出過程中 (指令輸入處於 ON 狀態)，不要對 PWM 指令的變數進行變更。
   進行變數變更時，應先將執行指令置於 OFF 之後再進行變更。

出　錯

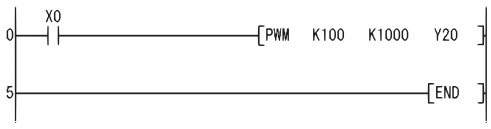
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 100ms 的脈衝每隔 1 秒輸出到 Y20 中的程式。

[梯形圖模式]

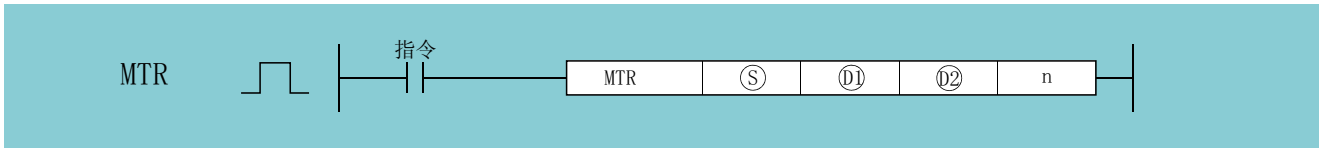


[列表模式]

步	指令	軟元件		
0	LD	X0		
1	PWM	K100	K1000	Y20
5	END			

6.8.10 MTR

Basic High performance Process Redundant Universal LCPU



Ⓢ ：輸入的起始編號 (位)。

Ⓓ1 ：輸出的起始編號 (位)。

Ⓓ2 ：存儲矩陣輸入資料的軟元件的起始編號 (位)。

n ：輸入列數 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○ (僅 X)	---							---
Ⓓ1	○ (僅 Y)	---							---
Ⓓ2	○	---							---
n	○	○							---

功　能

(1) 對Ⓢ中指定的輸入編號後面連接的 16 點× n 列的輸入依次進行讀取，將讀取的輸入資料存儲到Ⓓ2中指定的軟元件後面。

(2) 在 1 個掃描中可以讀取 1 列 (16 點)。

(3) 按從第 1 列至第 n 列的順序反復進行讀取。

(4) Ⓓ2中指定的軟元件後面，從起始開始的 16 點存儲第 1 列資料，下一個 16 點存儲第 2 列的資料。

因此，從Ⓓ2中指定的軟元件開始的 16 × n 點被 MTR 指令所佔用。

(5) Ⓓ1是用於選擇執行讀取的列的輸出，由系統自動地進行 ON/OFF。

該軟元件佔用從Ⓓ1中指定的軟元件開始的 n 點。

(6) Ⓢ、Ⓓ1、Ⓓ2中只能指定 16 的倍數的軟元件號。

(7) n 值的指定範圍爲 2 ～ 8。

(8) 在以下情況下將執行無處理：

- Ⓢ、Ⓓ1、Ⓓ2中指定的軟元件號不爲 16 的倍數時。
- Ⓢ中指定的軟元件超出了實際輸入範圍時。
- Ⓓ1指定的軟元件超出了實際輸出範圍時。
- Ⓓ2中指定的軟元件後面的 16 × n 點超出了相應軟元件範圍時。
- n 值超出了 2 ～ 8 的範圍時。

出　錯

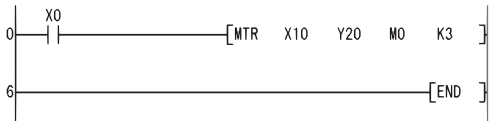
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓢ 中指定了除輸入 (X) 以外的軟元件時。 Ⓓ1 中指定了除輸出 (Y) 以外的軟元件時。	---	○	---	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時， 對 X10 後面連接的 16 點× 3 列的矩陣進行讀取後，將結果存儲到 M0 後面的程式。

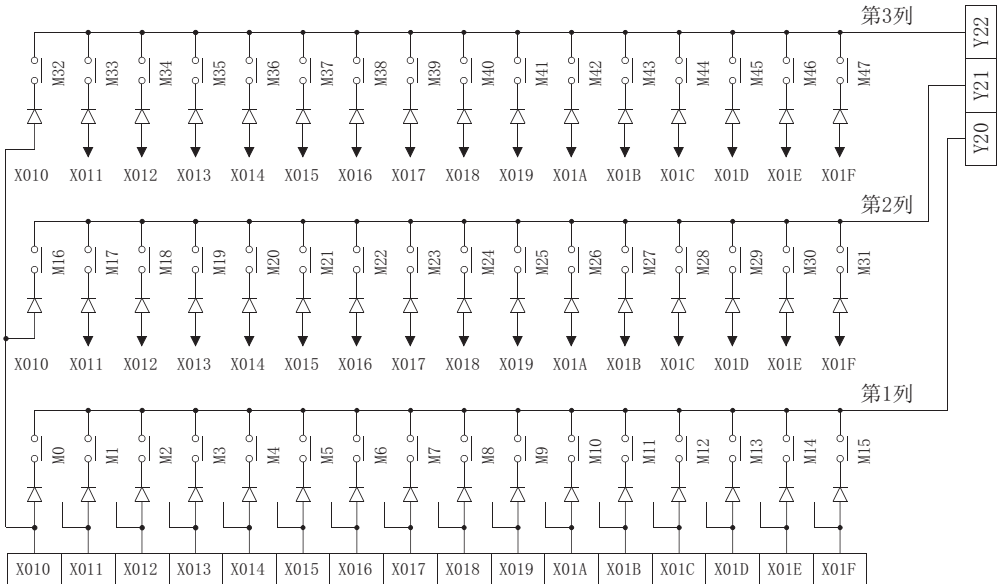
[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X0			
1	MTR	X10	Y20	M0	K3
6	END				

[動作]



注意事項

(1) 由於 MTR 指令直接對輸入輸出進行操作，因此應加以注意。

即使 MTR 指令變爲 OFF，通過 MTR 指令變爲 ON 的輸出Ⓓ1 也不會變爲 OFF。

應通過順控程式將Ⓓ1 中指定的輸出置於 OFF。

(2) MTR 指令的執行間隔必須長於輸入模組與輸出模組的回應時間的合計值。
如果 MTR 指令的執行間隔設置短於上述時間，則無法正常地讀取輸入。
如果順序程式的掃描時間過短，則應通過恒定掃描將掃描時間改爲長於回應時間的合計值。

第 7 章 應用指令

7.1 邏輯運算指令

(1) 邏輯運算指令以 1 位元爲單位進行邏輯和及邏輯積等邏輯運算。

分類	處理內容	運算公式	示例		
			A	B	Y
邏輯積 (AND)	只有當輸入 A 和輸入 B 都爲 1 時才爲 1，除此以外均爲 0。	$Y=A \cdot B$	0	0	0
			0	1	0
			1	0	0
			1	1	1
邏輯和 (OR)	只有當輸入 A 和輸入 B 都爲 0 時才爲 0，除此以外均爲 1。	$Y=A+B$	0	0	0
			0	1	1
			1	0	1
			1	1	1
排他邏輯和 (XOR)	如果輸入 A 和輸入 B 相等，則變爲 0，不相等時變爲 1。	$Y=\overline{A} \cdot B+A \cdot \overline{B}$	0	0	0
			0	1	1
			1	0	1
			1	1	0
否定排他邏輯和 (XNR)	如果輸入 A 和輸入 B 相等，則變爲 1，不相等時變爲 0。	$Y=(\overline{A}+B)(A+\overline{B})$	0	0	1
			0	1	0
			1	0	0
			1	1	1

7.1.1 WAND、WANDP、DAND、DANDP

Basic | High performance | Process | Redundant | Universal | LCPU

1 設置資料爲 2 個時 (Ⓓ∧Ⓢ→Ⓓ、(Ⓓ+1、Ⓓ) ∧ (Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ))

Ⓢ ：執行邏輯積的資料或者存儲資料的軟元件的起始編號 (BIN16/32 位)。

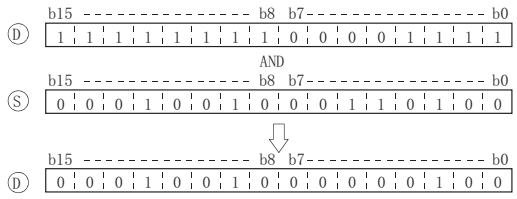
Ⓓ ：存儲邏輯積結果的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功 能

WAND

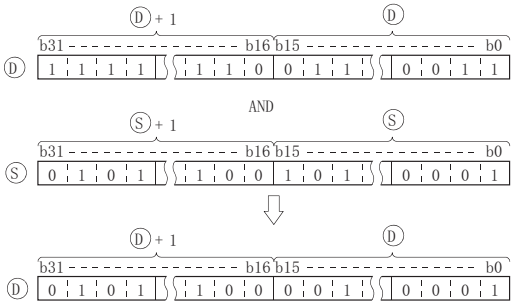
(1) 將Ⓓ中指定的軟元件的 16 位元資料與Ⓢ中指定的軟元件的 16 位元資料逐位元進行邏輯積運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。(參閱程式示例 (2))

DAND

(1) 將Ⓓ中指定的軟元件的 32 位元資料與Ⓢ中指定的軟元件的 32 位元資料逐位元進行邏輯積運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。(參閱程式示例 (2))

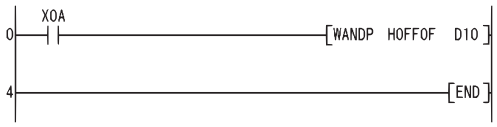
出 錯

(1) 在 WAND(P)、DAND(P) 指令中無運算出錯。

程式示例

(1) 以下爲 XA 變爲 ON 時，將 D10 的 BCD4 位數值中的十位（從低位起第 2 位）上的值遮罩爲 0 的程式。

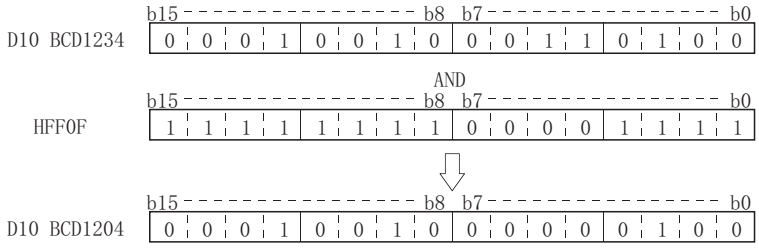
[梯形圖模式]



[列表模式]

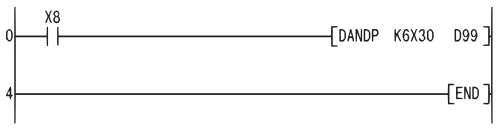
步	指令	軟元件
0	LD	X0A
1	WANDP	H0FF0F D10
4	END	

[動作]



(2) 以下爲 X8 變爲 ON 時，將 D99、D100 的資料與 X30 ～ X47 的 24 位元資料進行邏輯積運算，並將結果存儲到 D99、D100 中的程式。

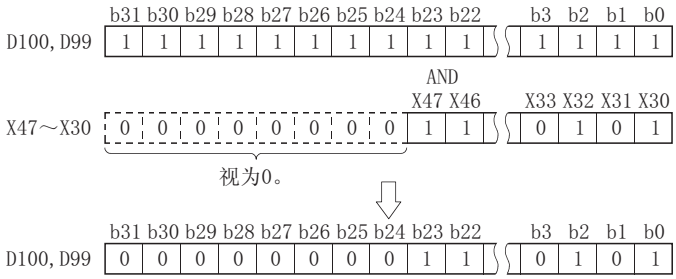
[梯形圖模式]



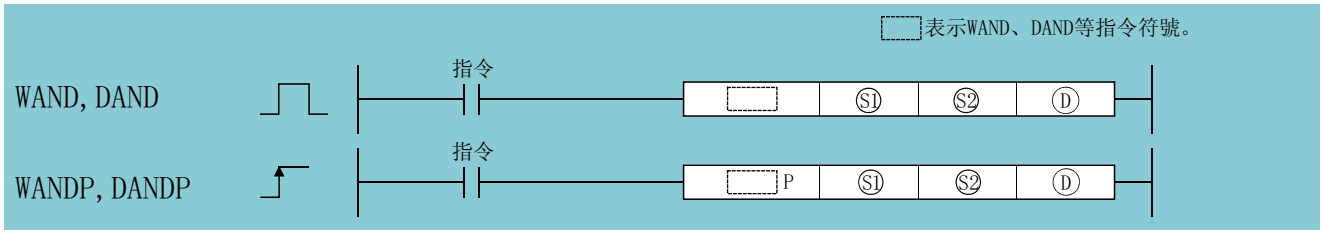
[列表模式]

步	指令	軟元件
0	LD	X8
1	DANDP	K6X30 D99
4	END	

[動作]



2 設置資料爲 3 個時 (Ⓢ1∧Ⓢ2→Ⓓ、(Ⓢ1+1、Ⓢ1) ∧ (Ⓢ2+1、Ⓢ2) → (Ⓓ+1、Ⓓ))



Ⓢ1、Ⓢ2：執行邏輯積的資料或者存儲資料的軟元件的起始編號 (BIN16/32 位)。

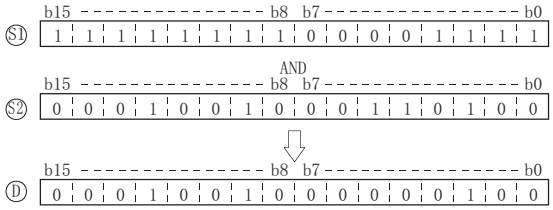
Ⓓ　：存儲邏輯積結果的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓢ1	○							○	---
Ⓢ2	○							○	---
Ⓓ	○							---	---

功　能

WAND

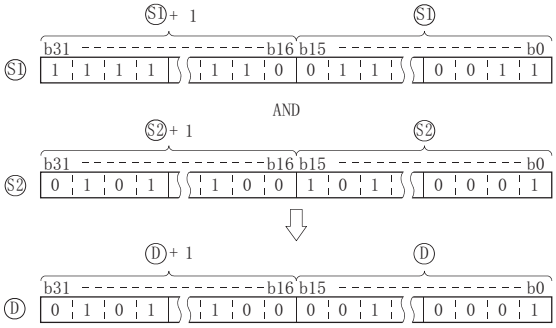
(1) 將Ⓢ1中指定的軟元件的 16 位元資料與Ⓢ2中指定的軟元件的 16 位元資料逐位元進行邏輯積運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。(參閱程式示例 (1),(2))

DAND

(1) 將Ⓢ1中指定的軟元件的 32 位元資料與Ⓢ2中指定的軟元件的 32 位元資料逐位元進行邏輯積運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。(參閱程式示例 (3))

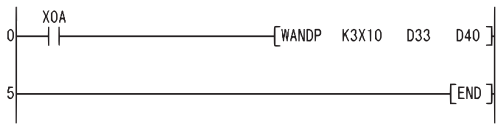
出　錯

(1) 在 WAND(P)、DAND(P) 指令中無運算出錯。

程式示例

(1) 以下爲 XA 變爲 ON 時，將 X10 ～ X1B 的資料與 D33 進行邏輯積運算，並將其結果存儲到 D40 中的程式。

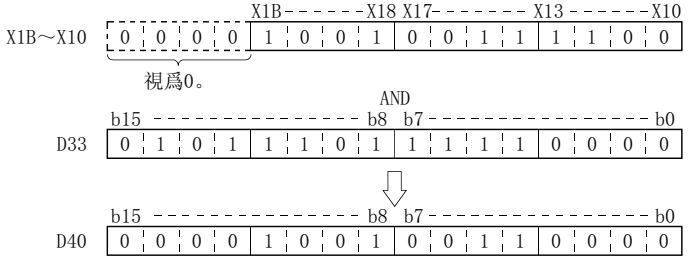
[梯形圖模式]



[列表模式]

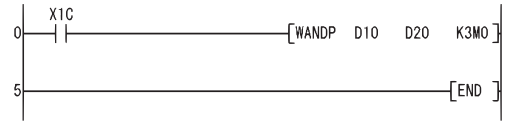
步	指令	軟元件		
0	LD	X0A		
1	WANDP	K3X10	D33	D40
5	END			

[動作]



(2) 以下爲 X1C 變爲 ON 時，將 D10 與 D20 的資料進行邏輯積運算，並將結果存儲到 M0 ～ M11 中的程式。

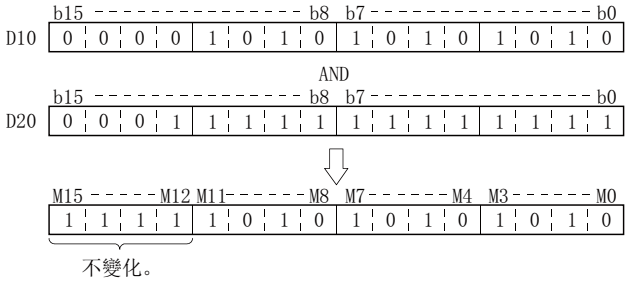
[梯形圖模式]



[列表模式]

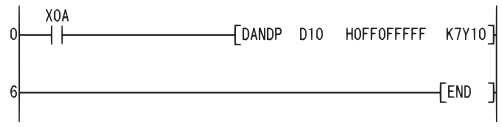
步	指令	軟元件		
0	LD	X1C		
1	WANDP	D10	D20	K3M0
5	END			

[動作]



(3) 以下爲 XA 變爲 ON 時，將 D10、D11 的 BCD8 位數值中的十萬位（從低位起第 6 位）上的值遮罩爲 0，並將運算結果存儲到 Y10 ～ Y2B 中的程式。

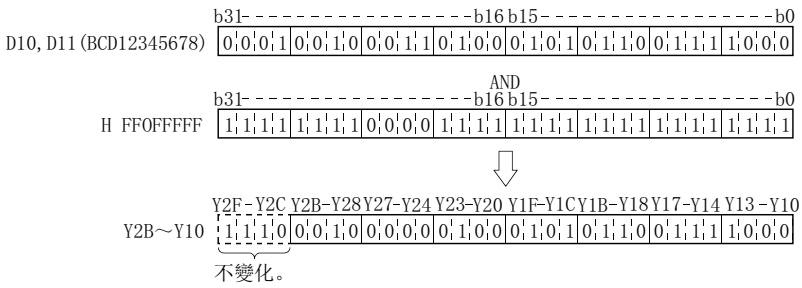
[梯形圖模式]



[列表模式]

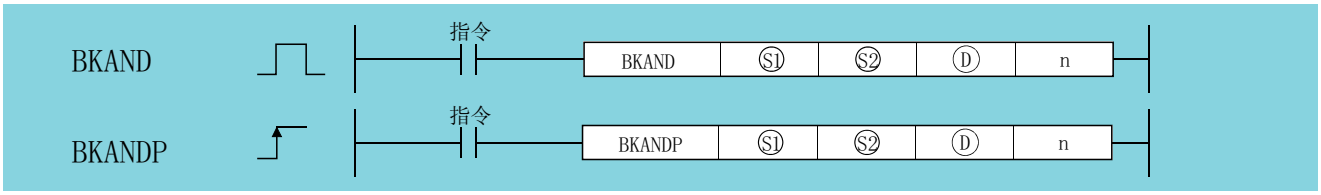
步	指令	軟元件		
0	LD	XOA		
1	DANDP	D10	H0FF0FFFFF	K7Y10
6	END			

[動作]



7.1.2 BKAND、BKANDP

Basic | High performance | Process | Redundant | Universal | LCPU



(S1)*1 ： 存儲執行邏輯積的資料的軟元件的起始編號 (BIN16 位)。

(S2)*1 ： 執行邏輯積的資料或者存儲資料的軟元件的起始編號 (BIN16 位)。

(D)*1 ： 存儲邏輯積結果的軟元件的起始編號 (BIN16 位)。

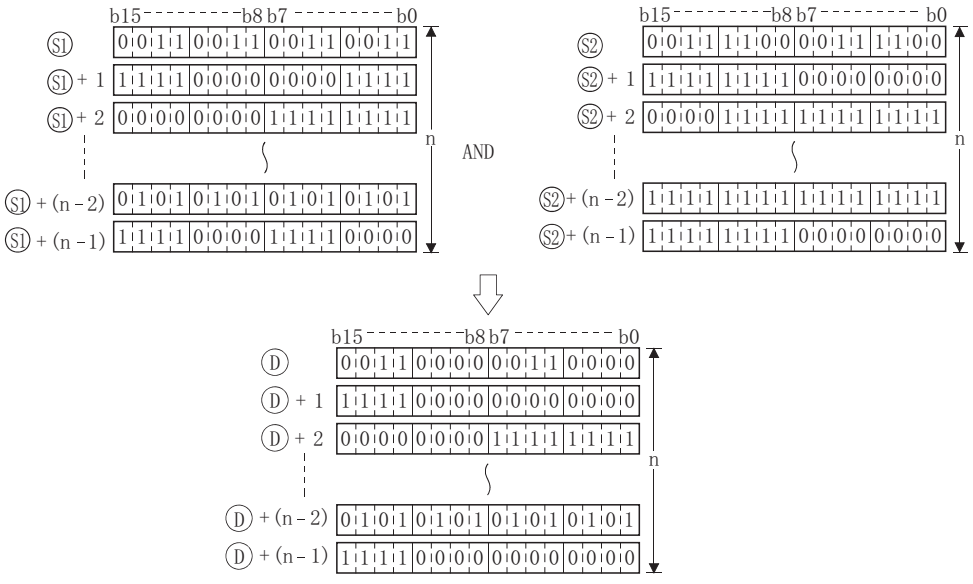
n ： 運算資料個數 (BIN16 位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
(S1)*1	---	○		---		---		---	---
(S2)*1	---	○		---		---		○	---
(D)*1	---	○		---		---		---	---
n	○	○		○				○	---

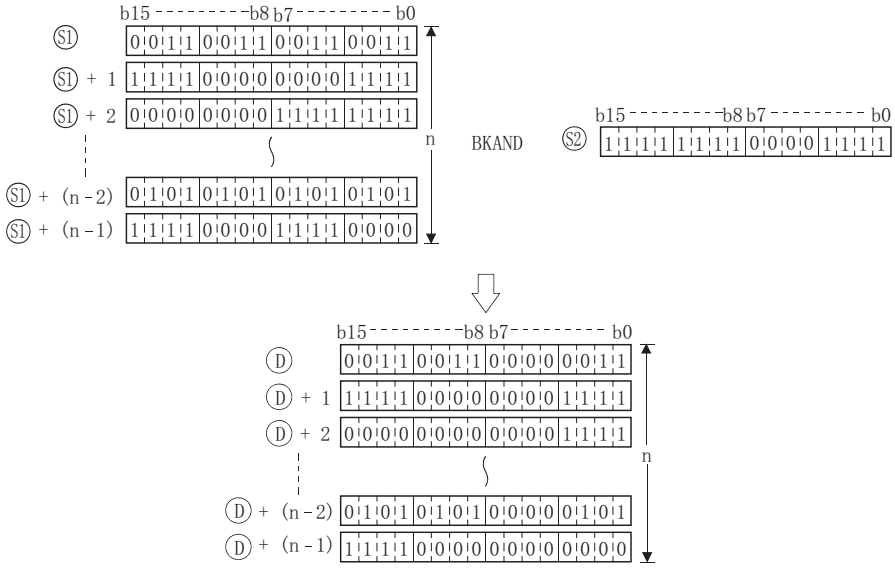
*1: (S1)與(D)或者(S2)與(D)可以指定爲相同的軟元件編號。

功 能

(1) 將從(S1)中指定的軟元件開始的 n 點的內容與從(S2)中指定的軟元件開始的 n 點的內容執行邏輯積運算，並將結果存儲到(D)中指定的軟元件後面。



(2) ⓢ2中可以指定爲 -32768 ～ 32767(BIN16 位) 的常數。



出　錯

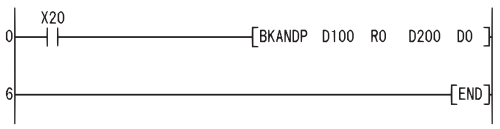
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從ⓢ1、ⓢ2、ⓓ的軟元件開始的 n 點的範圍超出了相應軟元件範圍時。 從ⓢ1開始至第 n 點爲止的軟元件範圍與從ⓓ開始的至第 n 點爲止的軟元件範圍有部分重復時。(ⓢ1與ⓓ中指定了同一個軟元件時除外。) 從ⓢ2開始至第 n 點爲止的軟元件範圍與從ⓓ開始的至第 n 點爲止的軟元件範圍相重復時。(ⓢ2與ⓓ中指定了同一個軟元件時除外。)	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D100 ～ D102 中存儲的資料值與 R0 ～ R2 中存儲的資料值進行邏輯積運算，並將其結果存儲到 D200 後面的程式。

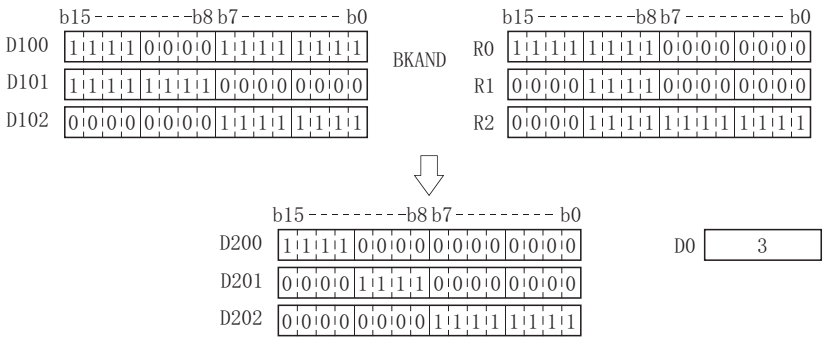
[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X20			
1	BKANDP	D100	R0	D200	D0
6	END				

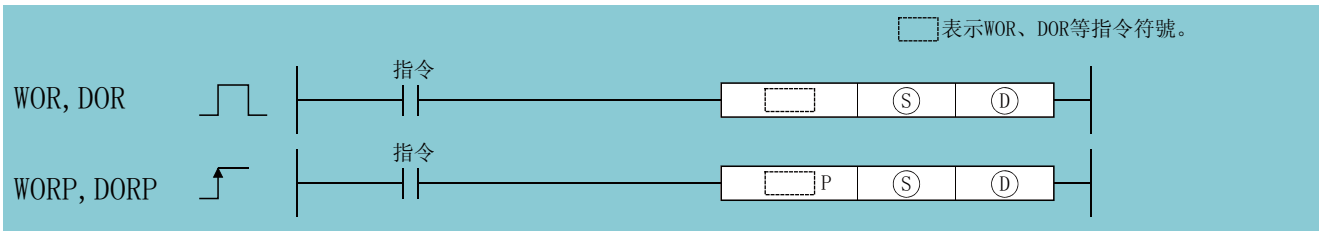
[動作]



7.1.3 WOR、WORP、DOR、DORP

Basic | High performance | Process | Redundant | Universal | LCPU

1 設置資料爲 2 個時 (Ⓓ∨Ⓢ→Ⓓ、(Ⓓ+1、Ⓓ) ∨ (Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ))



Ⓢ ：執行邏輯和的資料或者存儲資料的軟元件的起始編號 (BIN16/32 位)。

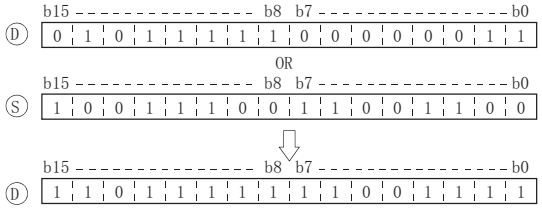
Ⓓ ：存儲邏輯和結果的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功　能

WOR

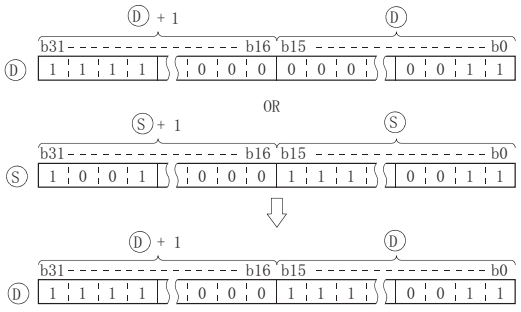
(1) 將Ⓓ中指定的軟元件的 16 位元資料與Ⓢ中指定的軟元件的 16 位元資料逐位元進行邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。

DOR

(1) 將Ⓓ中指定的軟元件的 32 位元資料與Ⓢ中指定的軟元件的 32 位元資料逐位元進行邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。

出　錯

(1) 在 WOR(P)、DOR(P) 指令中無運算出錯。

程式示例

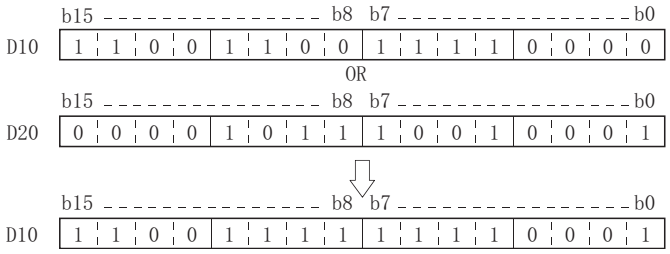
(1) 以下爲 XA 變爲 ON 時，將 D10 的資料與 D20 的資料執行邏輯和運算，並將其結果存儲到 D10 中的程式。

[梯形圖模式]

[列表模式]

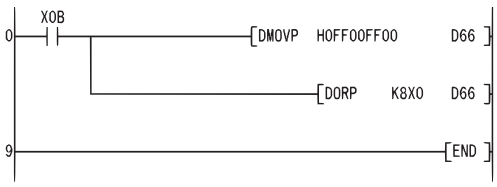
步	指令	軟元件	
0	LD	X0A	
1	WORP	D20	D10
4	END		

[動作]



(2) 以下爲 XB 變爲 ON 時，將 X0 ～ X1F 的 32 位元資料與 16 進制數 FF00FF00H 進行邏輯和運算，並將其結果存儲到 D66、D67 中的程式。

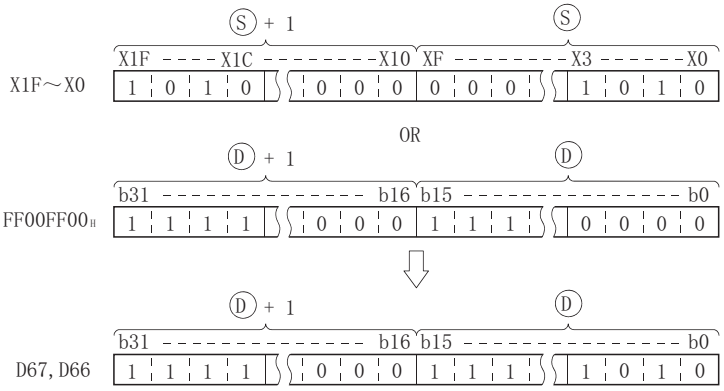
[梯形圖模式]



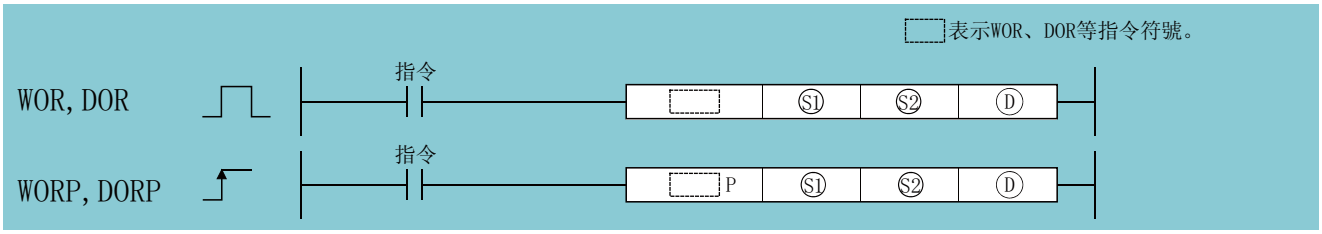
[列表模式]

步	指令	軟元件	
0	LD	X0B	
1	DMOVP	H0FF00FF00	D66
4	DORP	K8X0	D66
9	END		

[動作]



2 設置資料爲 3 個時 (Ⓢ1∨Ⓢ2→Ⓓ、(Ⓢ1+1、Ⓢ1) ∨ (Ⓢ2+1、Ⓢ2) → (Ⓓ+1、Ⓓ))



Ⓢ1、Ⓢ2：執行邏輯和的資料或者存儲資料的軟元件的起始編號 (BIN16/32 位)。

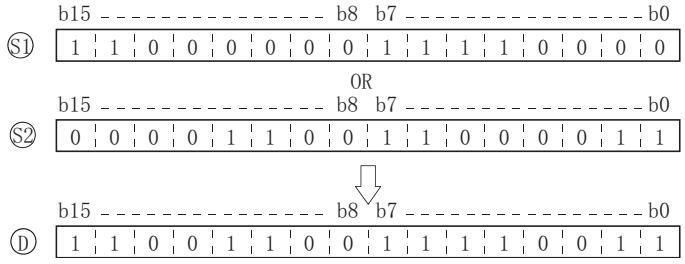
Ⓓ　：　存儲邏輯和結果的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	○							○	---
Ⓢ2	○							○	---
Ⓓ	○							---	---

功　能

WOR

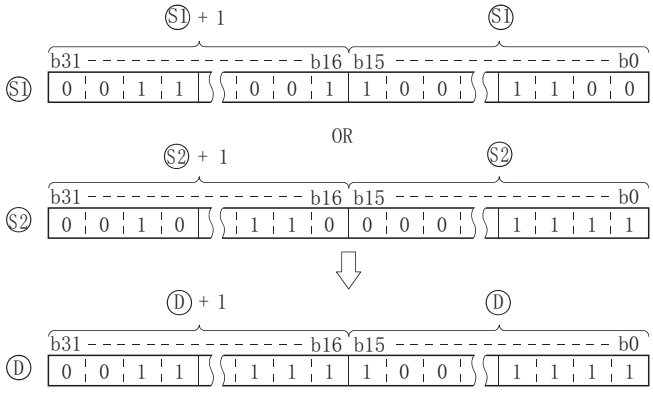
(1) 將Ⓢ1中指定的軟元件的 16 位元資料與Ⓢ2中指定的軟元件的 16 位元資料逐位元進行邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。(參閱程式示例 (1))

DOR

(1) 將Ⓢ1中指定的軟元件的 32 位元資料與Ⓢ2中指定的軟元件的 32 位元資料逐位元進行邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。(參閱程式示例 (2))

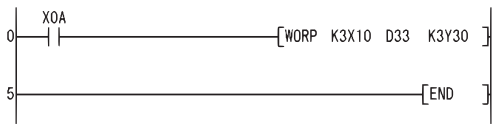
出　錯

(1) 在 WOR(P)、DOR(P) 指令中無運算出錯。

程式示例

(1) 以下爲 XA 變爲 ON 時，將 X10 ～ X1B 的資料與 D33 進行邏輯和運算，並將其結果存儲到 Y30 ～ Y3B 中的程式。

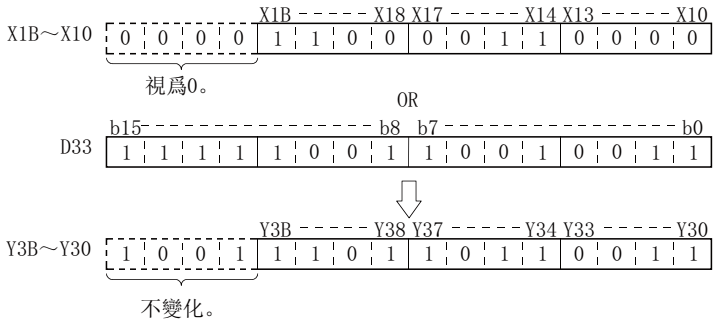
[梯形圖模式]



[列表模式]

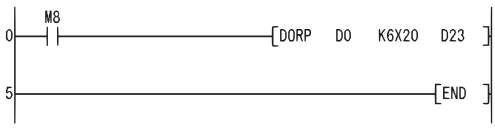
步	指令	軟元件		
0	LD	X0A		
1	WORP	K3X10	D33	K3Y30
5	END			

[動作]



(2) 以下爲 M8 變爲 ON 時，將 D0、D1 的 32 位元資料與 X20 ～ X37 的 24 位元資料進行邏輯和運算，並將結果存儲到 D23、D24 中的程式。

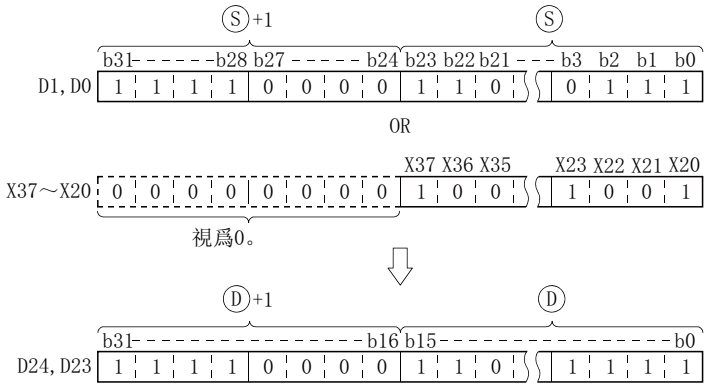
[梯形圖模式]



[列表模式]

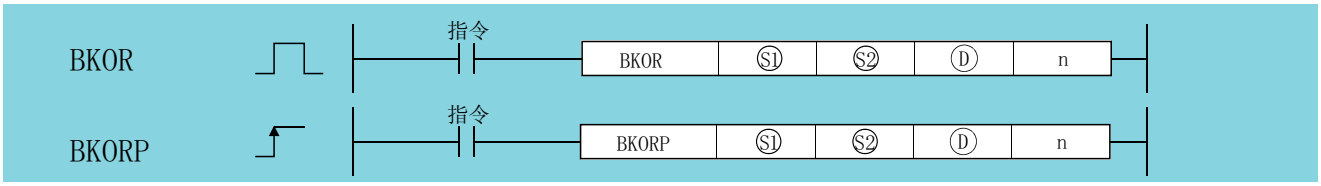
步	指令	軟元件		
0	LD	M8		
1	DORP	D0	K6X20	D23
5	END			

[動作]



7.1.4 BKOR、BKORP

Basic | High performance | Process | Redundant | Universal | LCPU



S1*1 ： 存儲執行邏輯和的資料的軟元件的起始編號 (BIN16 位)。

S2*1 ： 執行邏輯和的資料或者存儲資料的軟元件的起始編號 (BIN16 位)。

D*1 ： 存儲邏輯和結果的軟元件的起始編號 (BIN16 位)。

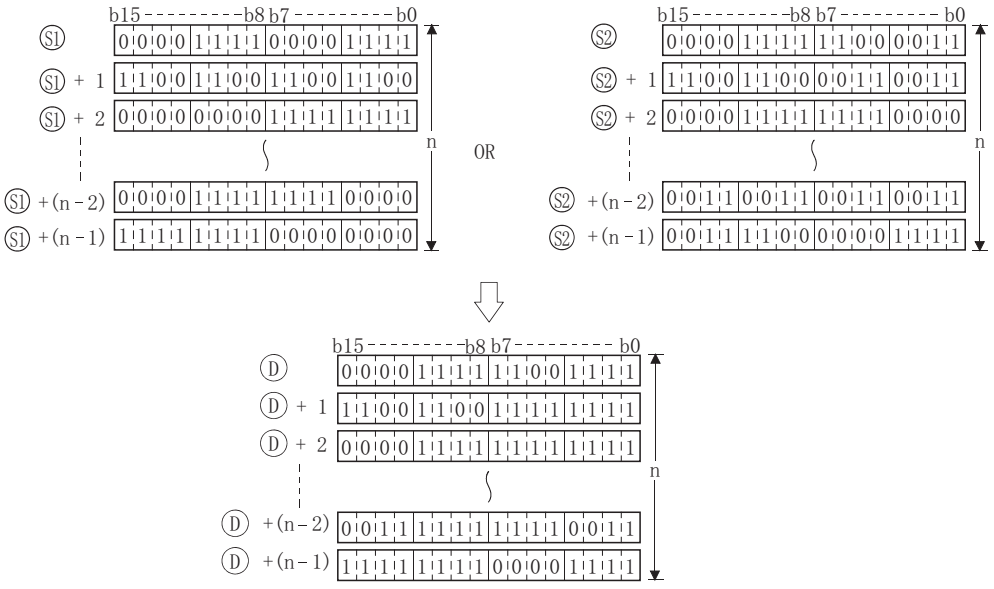
n ： 運算資料個數 (BIN16 位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
S1*1	---	○		---				---	---
S2*1	---	○		---				○	---
D*1	---	○		---				---	---
n	○	○		○				○	---

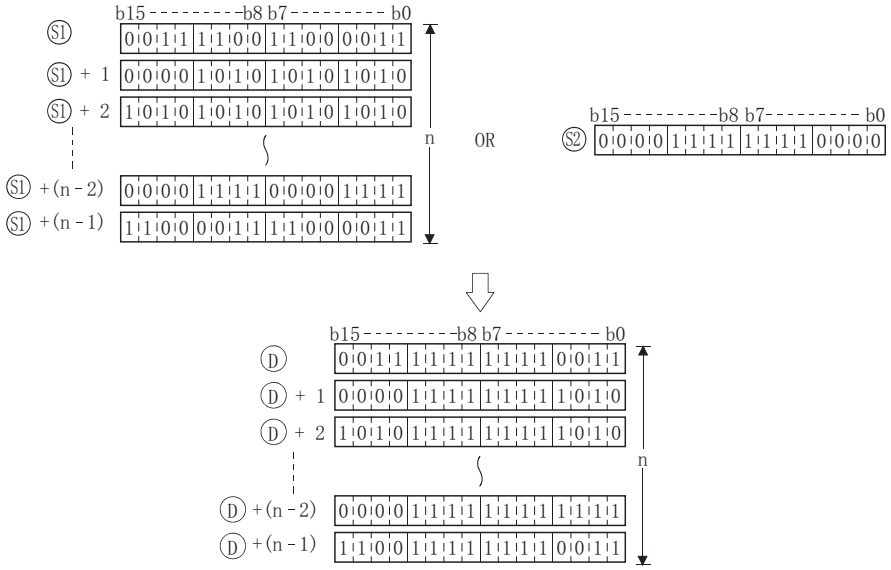
*1: S1與D 或者S2與D 可以指定爲相同的軟元件編號。

功 能

(1) 將從S1中指定的軟元件開始的 n 點的內容與從S2中指定的軟元件開始的 n 點的內容執行邏輯和運算，並將結果存儲到D中指定的軟元件後面。



(2) ⓢ2中可以指定爲 -32768 ～ 32767(BIN16 位) 的常數。



出　錯

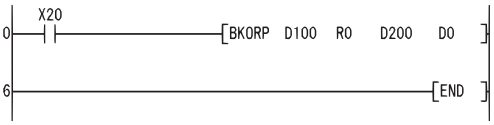
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從ⓢ1、ⓢ2、ⓓ的軟元件開始的 n 點的範圍超出了相應軟元件範圍時。 從ⓢ1開始至第 n 點爲止的軟元件範圍與從ⓓ開始的至第 n 點爲止的軟元件範圍有部分重復時。(ⓢ1與ⓓ中指定了同一個軟元件時除外。) 從ⓢ2開始至第 n 點爲止的軟元件範圍與從ⓓ開始的至第 n 點爲止的軟元件範圍相重復時。(ⓢ2與ⓓ中指定了同一個軟元件時除外。)	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D100 ～ D102 中存儲的資料值與 R0 ～ R2 中存儲的資料值進行邏輯和運算，並將其結果存儲到 D200 後面的程式。

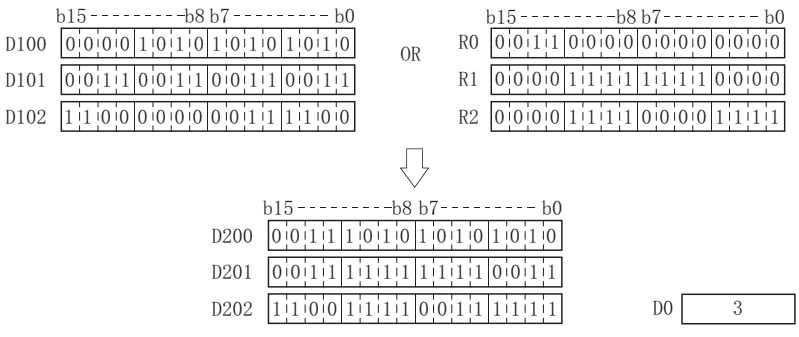
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X20
1	BKORP	D100 R0 D200 D0
6	END	

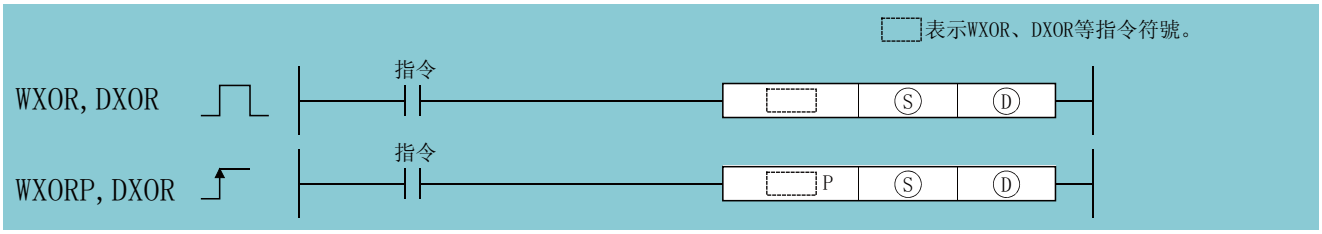
[動作]



7.1.5 WXOR、WXORP、DXOR、DXORP

Basic | High performance | Process | Redundant | Universal | LCPU

1 設置資料爲 2 個時 (Ⓓ ⩛ Ⓢ→Ⓓ、(Ⓓ+1、Ⓓ) ⩛ (Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ))



Ⓢ ：執行排他邏輯和的資料或者存儲資料的軟元件的起始編號 (BIN16/32 位)。

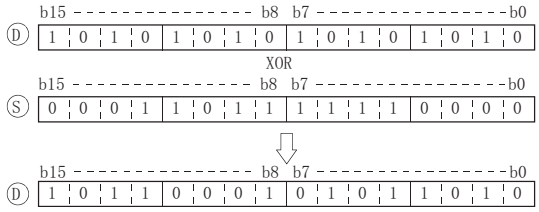
Ⓓ ：存儲排他邏輯和結果的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功　能

WXOR

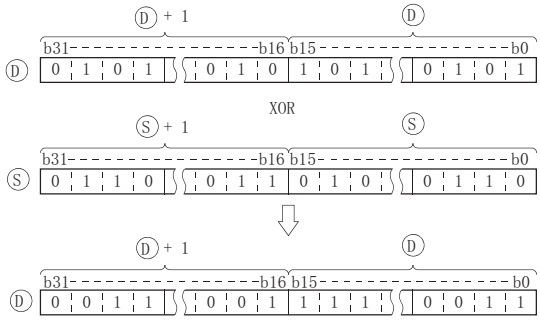
(1) 將Ⓓ中指定的軟元件的 16 位元資料與Ⓢ中指定的軟元件的 16 位元資料逐位元進行排他邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。

DXOR

(1) 將Ⓓ中指定的軟元件的 32 位元資料與Ⓢ中指定的軟元件的 32 位元資料逐位元進行排他邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。

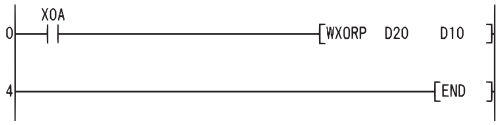
出　錯

(1) 在 WXOR(P)、DXOR(P) 指令中無運算出錯。

程式示例

(1) 以下爲 XA 變爲 ON 時，將 D10 的資料與 D20 的資料執行排他邏輯和運算，並將其結果存儲到 D10 中的程式。

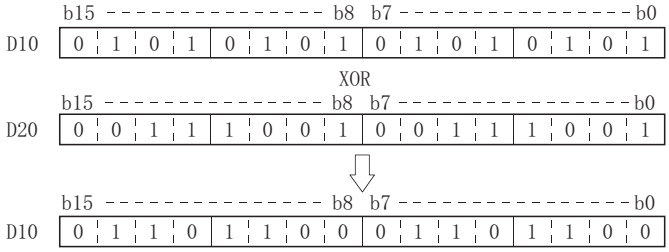
[梯形圖模式]



[列表模式]

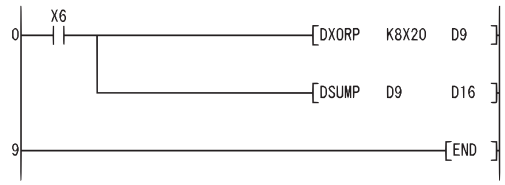
步	指令	軟元件	
0	LD	X0A	
1	WXORP	D20	D10
4	END		

[動作]



(2) 以下爲 X6 變爲 ON 時，將 X20 ～ X3F 的 32 位元資料與 D9、D10 的資料的位元模式進行比較，並將不同的位元數存儲到 D16 中的程式。

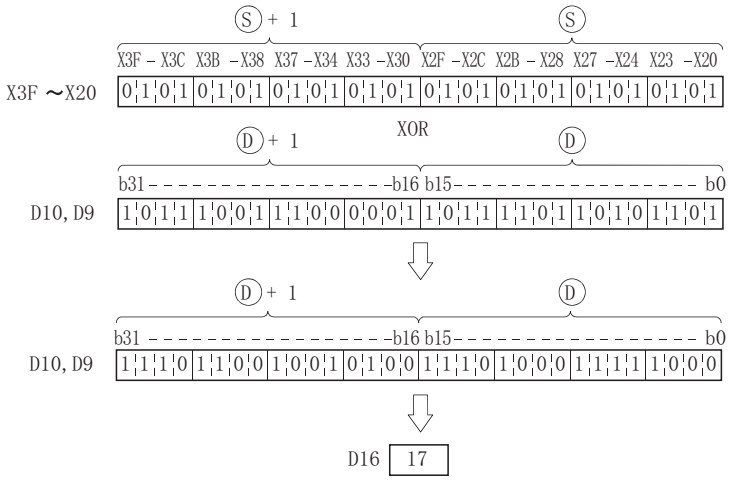
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X6		
1	DXORP	K8X20	D9	
6	DSUMP	D9	D16	
9	END			

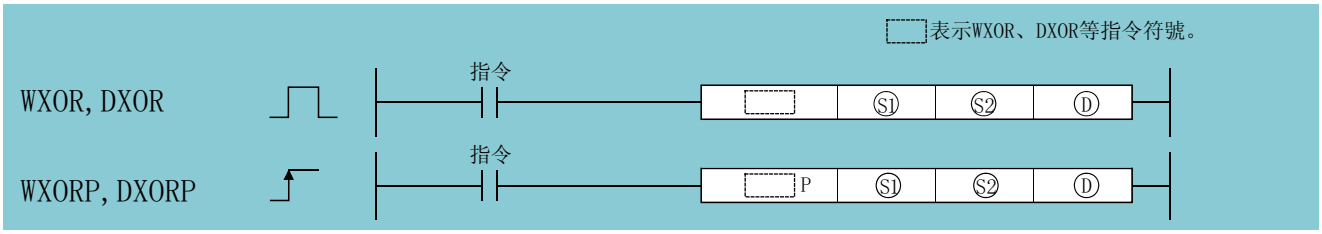
[動作]



備 註

關於 DSUMP 指令的有關內容，請參閱 345 頁 7.5.2 項。

2 設置資料爲 3 個時 (Ⓢ1 ∨ Ⓢ2→Ⓓ、(Ⓢ1+1、Ⓢ1) ∨ (Ⓢ2+1、Ⓢ2) → (Ⓓ+1、Ⓓ))



Ⓢ1、Ⓢ2：執行排他邏輯和的資料或者存儲資料的軟元件的起始編號 (BIN16/32 位)。

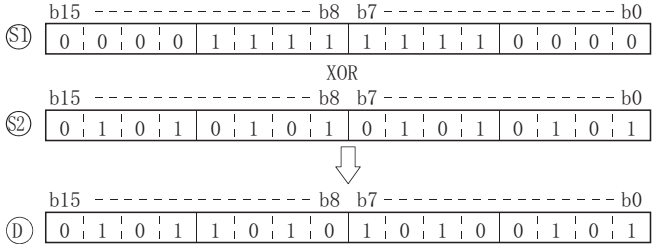
Ⓓ　：　存儲排他邏輯和結果的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	○							○	---
Ⓢ2	○							○	---
Ⓓ	○							---	---

功　能

WXOR

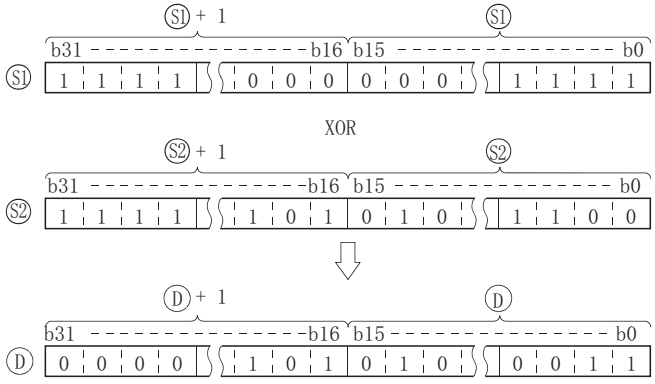
(1) 將Ⓢ1中指定的軟元件的 16 位元資料與Ⓢ2中指定的軟元件的 16 位元資料逐位元進行排他邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。(參閱程式示例 (1))

DXOR

(1) 將Ⓢ1中指定的軟元件的 32 位元資料與Ⓢ2中指定的軟元件的 32 位元資料逐位元進行排他邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。

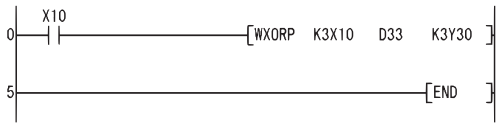
出　錯

(1) 在 WXOR(P)、DXOR(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X10 變爲 ON 時，將 X10 ～ X1B 的資料與 D33 的資料進行排他邏輯和運算，並將結果存儲到 Y30 ～ Y3B 中的程式。

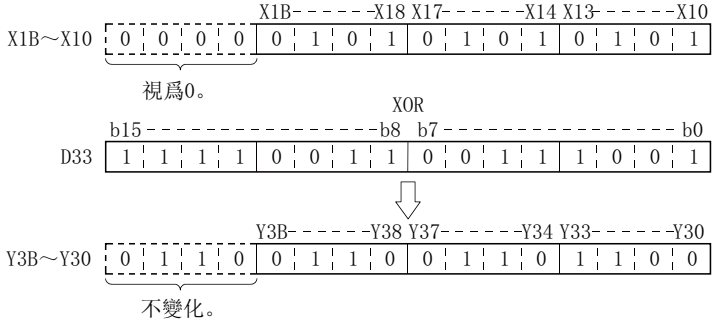
[梯形圖模式]



[列表模式]

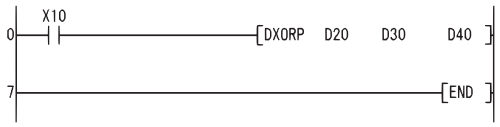
步	指令	軟元件		
0	LD	X10		
1	WXORP	K3X10	D33	K3Y30
5	END			

[動作]



(2) 以下爲 X10 變爲 ON 時，將 D20、D21 的資料與 D30、D31 的資料進行排他邏輯和運算，並將結果存儲到 D40、D41 中的程式。

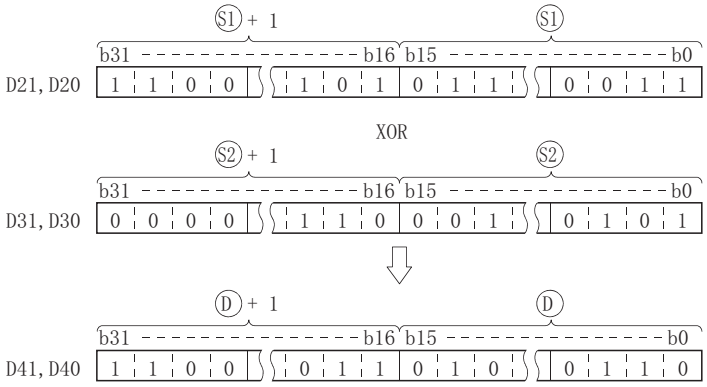
[梯形圖模式]



[列表模式]

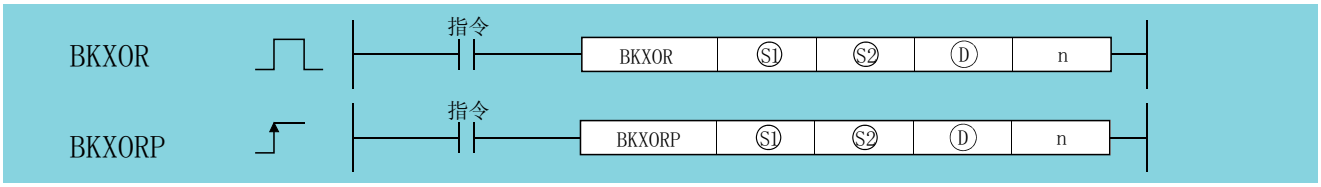
步	指令	軟元件		
0	LD	X10		
1	DXORP	D20	D30	D40
7	END			

[動作]



7.1.6 BKXOR、BKXORP

Basic | High performance | Process | Redundant | Universal | LCPU



(S1)*1 ： 存儲執行邏輯運算資料的軟元件的起始編號 (BIN16 位)。

(S2)*1 ： 執行邏輯運算的資料或者存儲執行邏輯運算資料的軟元件的起始編號 (BIN16 位)。

(D)*1 ： 存儲運算結果的軟元件的起始編號 (BIN16 位)。

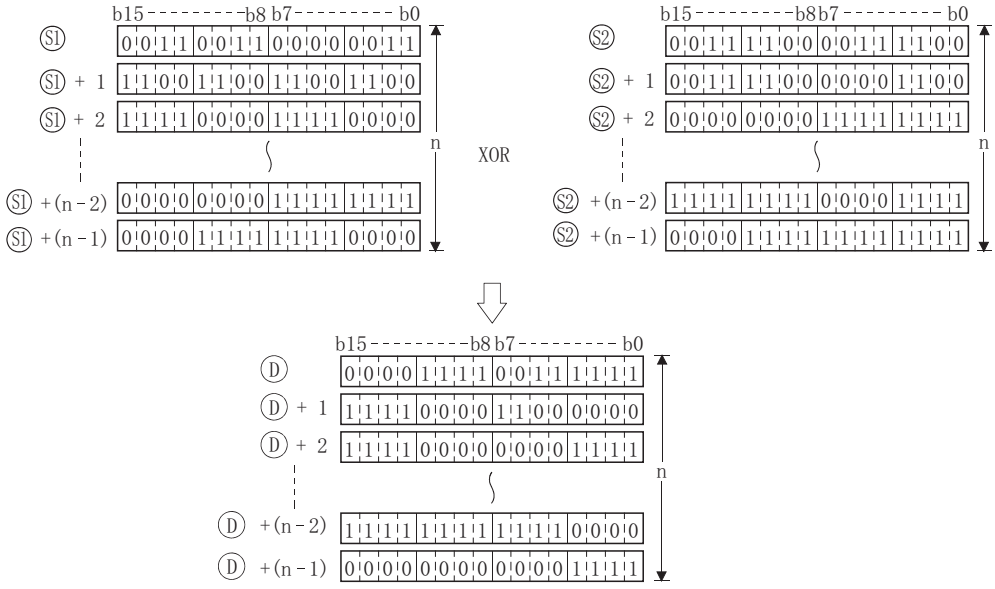
n ： 運算資料個數 (BIN16 位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
(S1)*1	---	○		---				---	---
(S2)*1	---	○		---				○	---
(D)*1	---	○		---				---	---
n	○	○		○				○	---

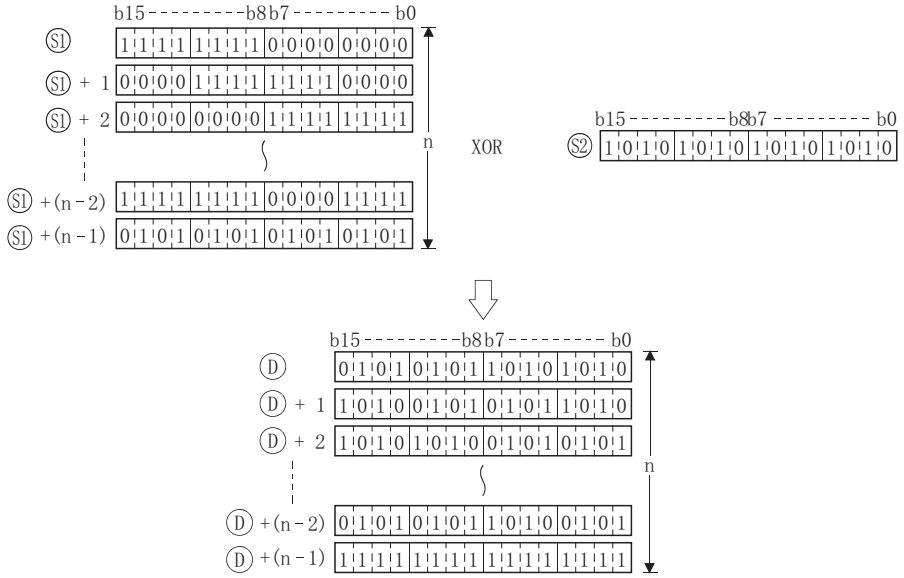
*1: (S1)與(D)或者(S2)與(D)可以指定爲相同的軟元件編號。

功 能

(1) 將從(S1)中指定的軟元件開始的 n 點的內容與從(S2)中指定的軟元件開始的 n 點的內容執行排他邏輯和運算，並將結果存儲到(D)中指定的軟元件後面。



(2) ⓢ2中可以指定爲 -32768 ～ 32767(BIN16 位) 的常數。



出　錯

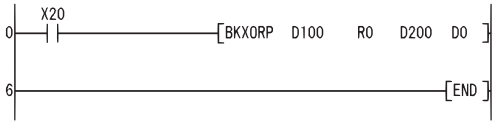
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從ⓢ1、ⓢ2、ⓓ的軟元件開始的 n 點的範圍超出了相應軟元件範圍時。 從ⓢ1開始至第 n 點爲止的軟元件範圍與從ⓓ開始的至第 n 點爲止的軟元件範圍有部分重復時。(ⓢ1與ⓓ中指定了同一個軟元件時除外。) 從ⓢ2開始至第 n 點爲止的軟元件範圍與從ⓓ開始的至第 n 點爲止的軟元件範圍相重復時。(ⓢ2與ⓓ中指定了同一個軟元件時除外。)	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D100 ～ D102 中存儲的資料值與 R0 ～ R2 中存儲的資料值進行排他邏輯和運算，並將其結果存儲到 D200 後面的程序。

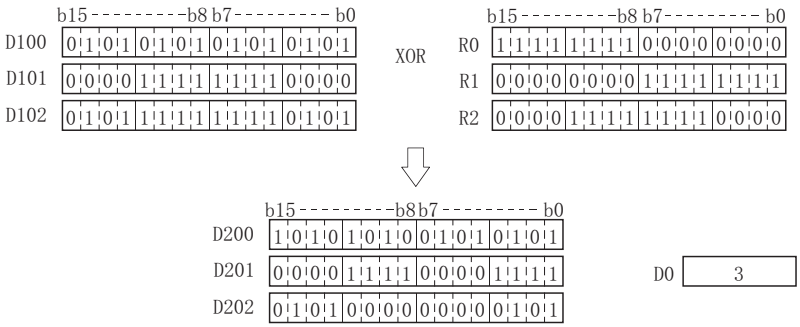
[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X20			
1	BKXORP	D100	R0	D200	D0
6	END				

[動作]



7.1.7 WXNR、WXNRP、DXNR、DXNRP

Basic | High performance | Process | Redundant | Universal | LCPU

1 設置資料爲 2 個時 (Ⓓ ⊻ Ⓢ→Ⓓ、(Ⓓ+1、Ⓓ) ⊻ (Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ))

Ⓢ ：執行否定排他邏輯和的資料或者存儲資料的軟元件的起始編號 (BIN16/32 位)。

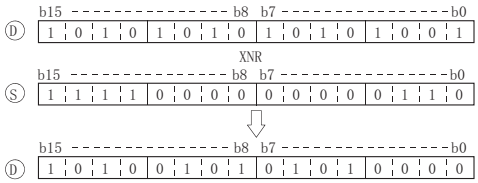
Ⓓ ：存儲否定排他邏輯和結果的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功　能

WXNR

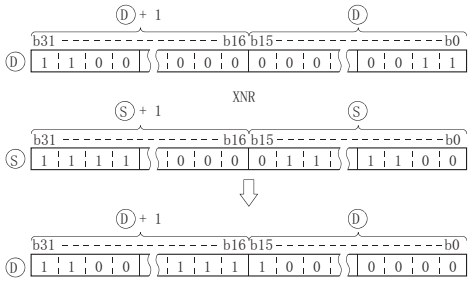
(1) 將Ⓓ中指定的軟元件的 16 位元資料與Ⓢ中指定的軟元件的 16 位元資料逐位元進行否定排他邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。

DXNR

(1) 將Ⓓ中指定的軟元件的 32 位元資料與Ⓢ中指定的軟元件的 32 位元資料逐位元進行否定排他邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。

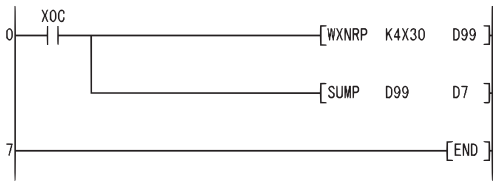
出　錯

(1) 在 WXNR(P)、DXNR(P) 指令中無運算出錯。

程式示例

(1) 以下爲 XC 變爲 ON 時，將 X30 ～ X3F 的 16 位元資料與 D99 的 16 位元資料執行比較，並將相同的位元模式數存儲到 D7 中的程式。

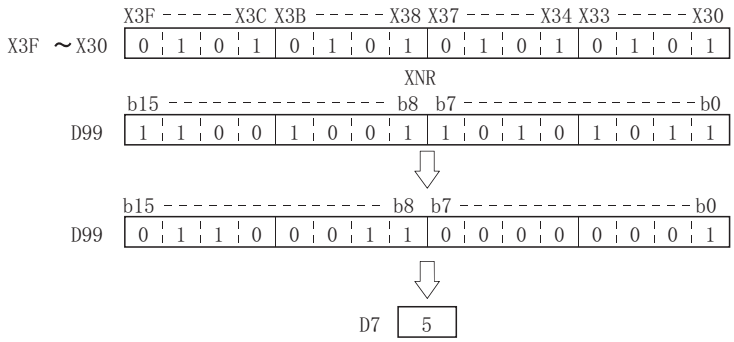
[梯形圖模式]



[列表模式]

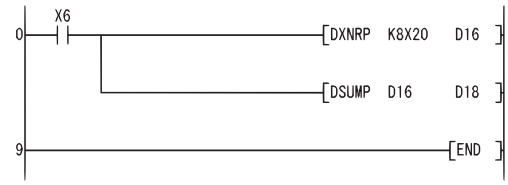
步	指令	軟元件	
0	LD	X0C	
1	WXNRP	K4X30	D99
4	SUMP	D99	D7
7	END		

[動作]



(2) 以下爲 X6 變爲 ON 時，將 X20 ～ X3F 的 32 位元資料與 D16、D17 的資料的位元模式進行比較，並將相同的位元模式數存儲到 D18 中的程式。

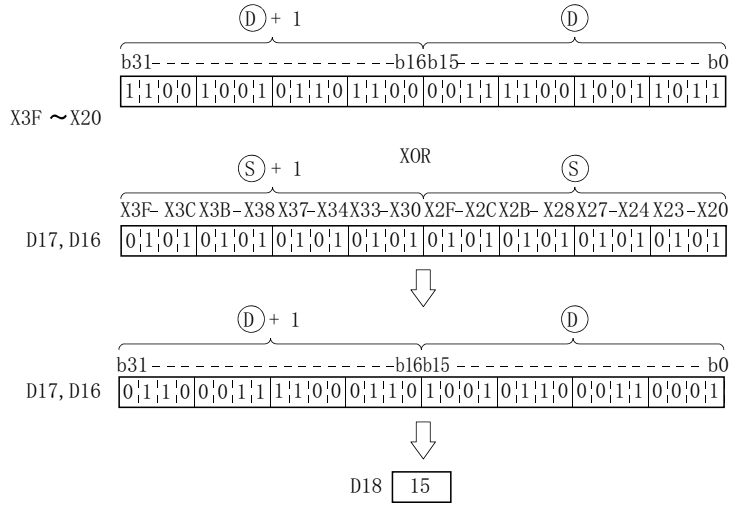
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X6	
1	DXNRP	K8X20	D16
6	DSUMP	D16	D18
9	END		

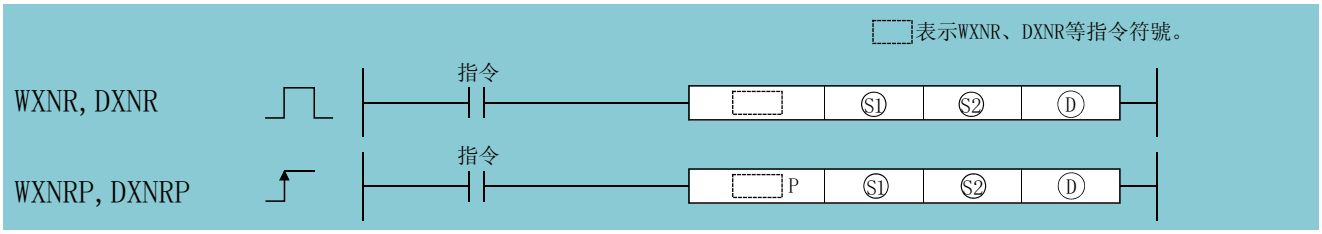
[動作]



備 註

關於 SUMP/DSUMP 指令的有關內容，請參閱 345 頁 7.5.2 項。

2 設置資料爲 3 個時 ($\overline{\text{Ⓢ1} \veebar \text{Ⓢ2}}$→Ⓓ、$\overline{(\text{Ⓢ1}+1、\text{Ⓢ1}) \veebar (\text{Ⓢ2}+1、\text{Ⓢ2})}$ → (Ⓓ+1、Ⓓ))



Ⓢ1、Ⓢ2：執行否定排他邏輯和的資料或者存儲資料的軟元件的起始編號 (BIN16/32 位)。

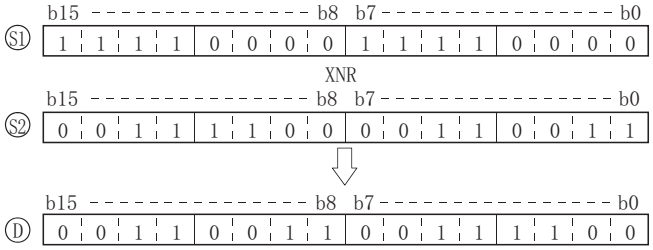
Ⓓ　：存儲否定排他邏輯和結果的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	○							○	---
Ⓢ2	○							○	---
Ⓓ	○							---	---

功　能

WXNR

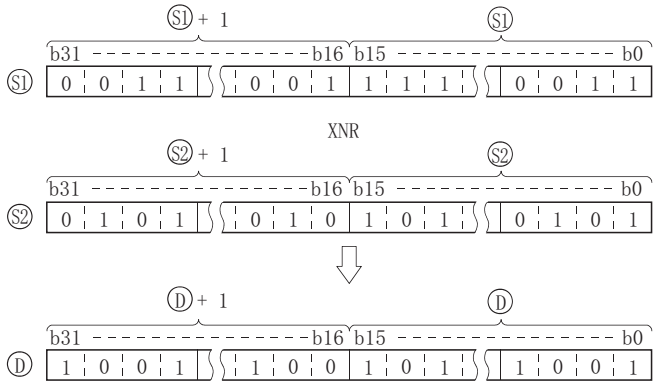
(1) 將Ⓢ1中指定的軟元件的 16 位元資料與Ⓢ2中指定的軟元件的 16 位元資料進行否定排他邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。

DXNR

(1) 將Ⓢ1中指定的軟元件的 32 位元資料與Ⓢ2中指定的軟元件的 32 位元資料進行否定排他邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件中。



(2) 位元軟元件的情況下，位元數指定點數以後的位軟元件將被作爲 0 進行運算。

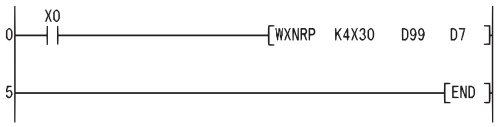
出　錯

(1) 在 WXNR(P)、DXNR(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X0 變爲 ON 時，將 X30 ～ X3F 的 16 位元資料與 D99 的資料進行否定排他邏輯和運算，並將結果存儲到 D7 中的程式。

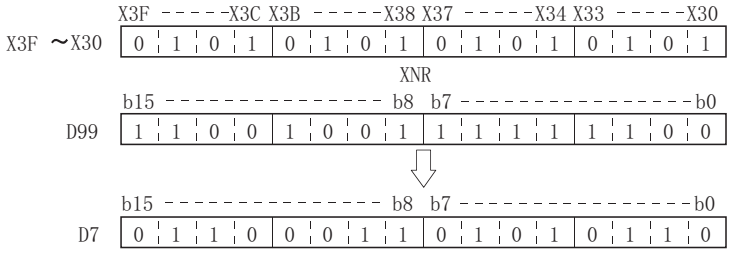
[梯形圖模式]



[列表模式]

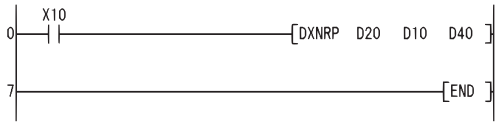
步	指令	軟元件		
0	LD	X0		
1	WXNRP	K4X30	D99	D7
5	END			

[動作]



(2) 以下爲 X10 變爲 ON 時，將 D20、D21 的 32 位元資料與 D10、D11 的資料進行否定排他邏輯和運算，並將結果存儲到 D40、D41 中的程式。

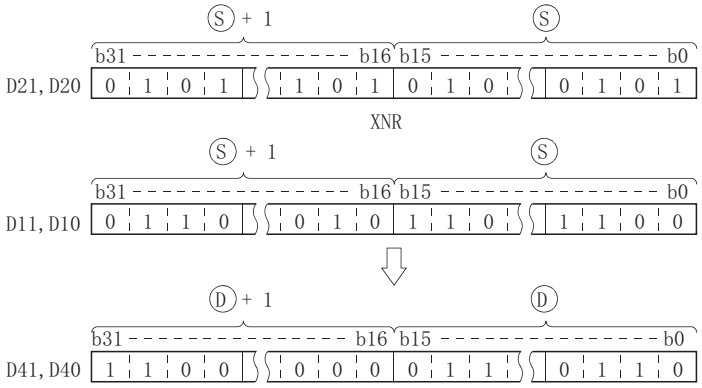
[梯形圖模式]



[列表模式]

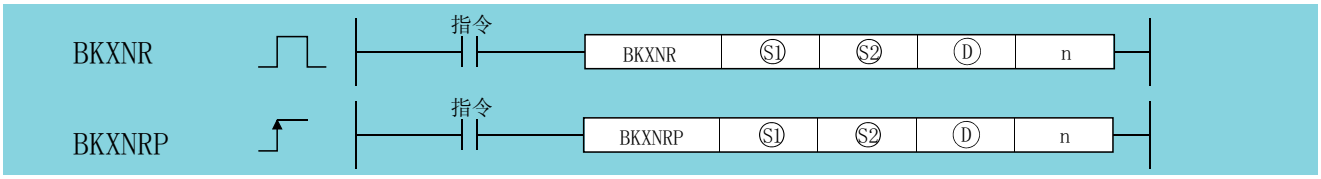
步	指令	軟元件		
0	LD	X10		
1	DXNRP	D20	D10	D40
7	END			

[動作]



7.1.8 BKXNR、BKXNRP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ1*1 ： 存儲執行邏輯運算資料的軟元件的起始編號 (BIN16 位)。

Ⓢ2*1 ： 執行邏輯運算的資料或者存儲執行邏輯運算資料的軟元件的起始編號 (BIN16 位)。

Ⓓ*1 ： 存儲運算結果的軟元件的起始編號 (BIN16 位)。

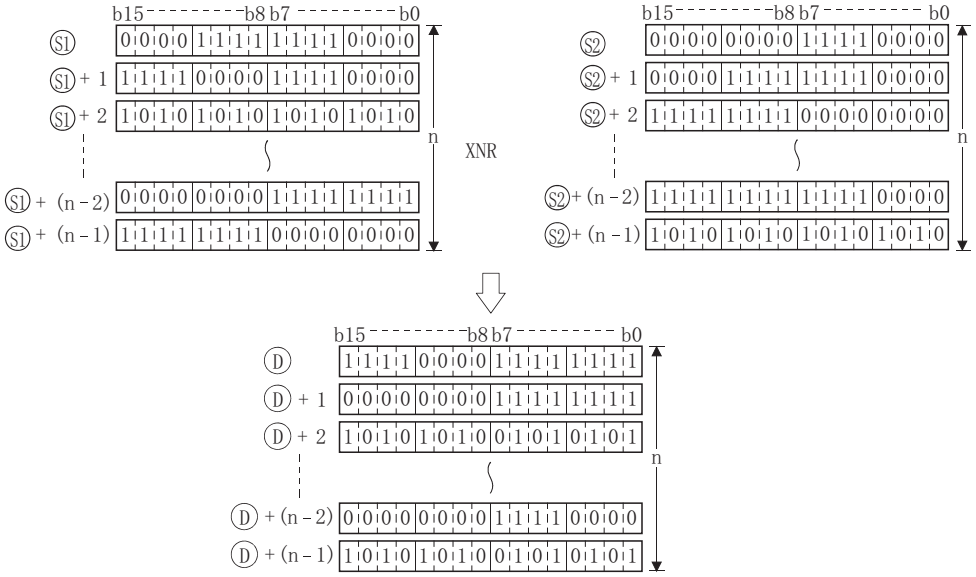
n ： 運算資料個數 (BIN16 位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1*1	---	○		---				---	---
Ⓢ2*1	---	○		---				○	---
Ⓓ*1	---	○		---				---	---
n	○	○		○				○	---

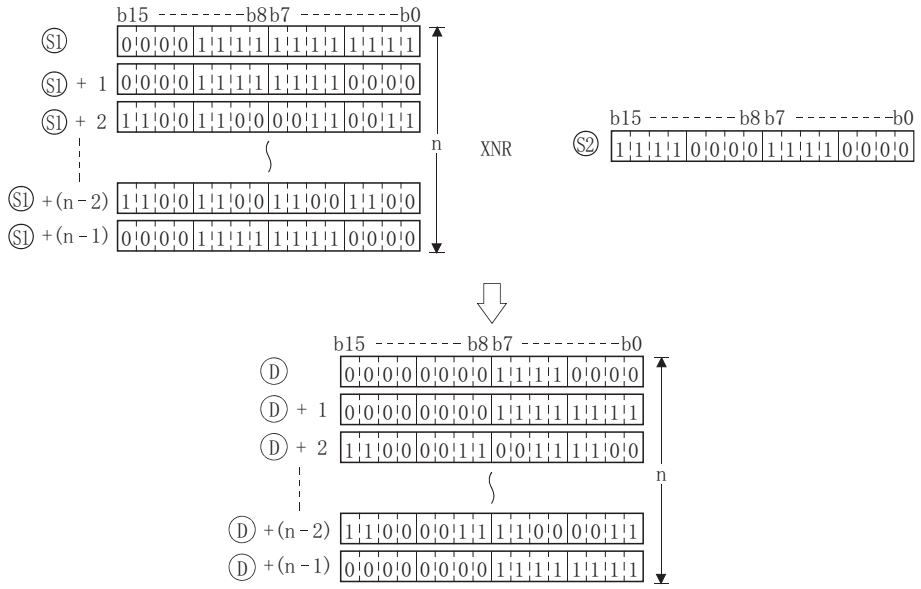
*1: Ⓢ1與Ⓓ或者Ⓢ2與Ⓓ可以指定爲相同的軟元件編號。

功 能

(1) 將從Ⓢ1中指定的軟元件開始的 n 點的內容與從Ⓢ2中指定的軟元件開始的 n 點的內容執行否定排他邏輯和運算，並將結果存儲到Ⓓ中指定的軟元件後面。



(2) ⓢ2中可以指定爲 -32768 ～ 32767(BIN16 位) 的常數。



出　錯

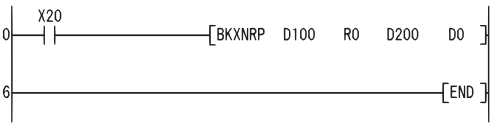
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從ⓢ1、ⓢ2、ⓓ的軟元件開始的 n 點的範圍超出了相應軟元件範圍時。 從ⓢ1開始至第 n 點爲止的軟元件範圍與從ⓓ開始的至第 n 點爲止的軟元件範圍有部分重復時。(ⓢ1與ⓓ中指定了同一個軟元件時除外。) 從ⓢ2開始至第 n 點爲止的軟元件範圍與從ⓓ開始的至第 n 點爲止的軟元件範圍相重復時。(ⓢ2與ⓓ中指定了同一個軟元件時除外。)	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D100 ～ D102 中存儲的資料值與 R0 ～ R2 中存儲的資料值進行否定排他邏輯和運算，並將其結果存儲到 D200 後面的程式。

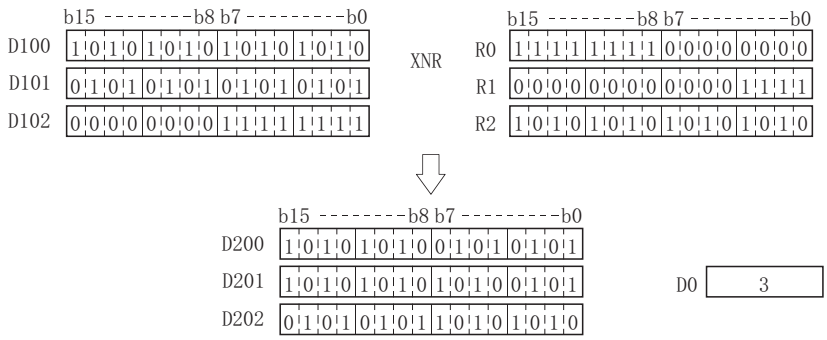
[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X20			
1	BKXNRP	D100	R0	D200	D0
6	END				

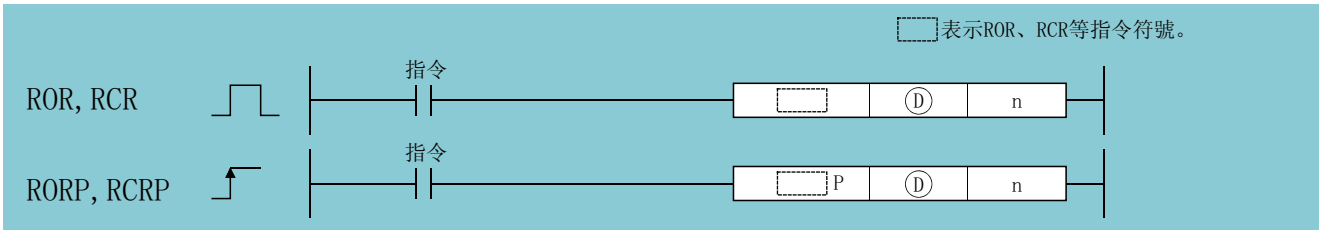
[動作]



7.2 旋轉指令

7.2.1 ROR、RORP、RCR、RCRP

Basic | High performance | Process | Redundant | Universal | LCPU



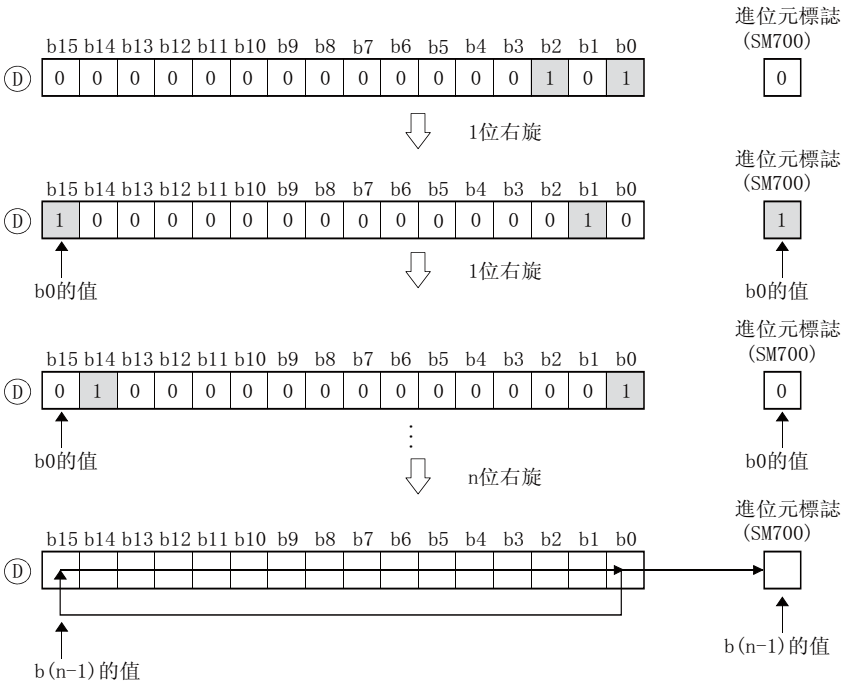
Ⓓ ：進行旋轉的軟元件的起始編號 (BIN16 位)。
n ：旋轉次數 (0 ～ 15)(BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓓ		○						---	---
n		○						○	---

功　能

ROR

(1) 將Ⓓ中指定的軟元件的 16 位元資料在不包含進位元標誌的狀態下進行 n 位元右旋轉。
進位元標誌根據 ROR 執行前的狀態而 ON/OFF。

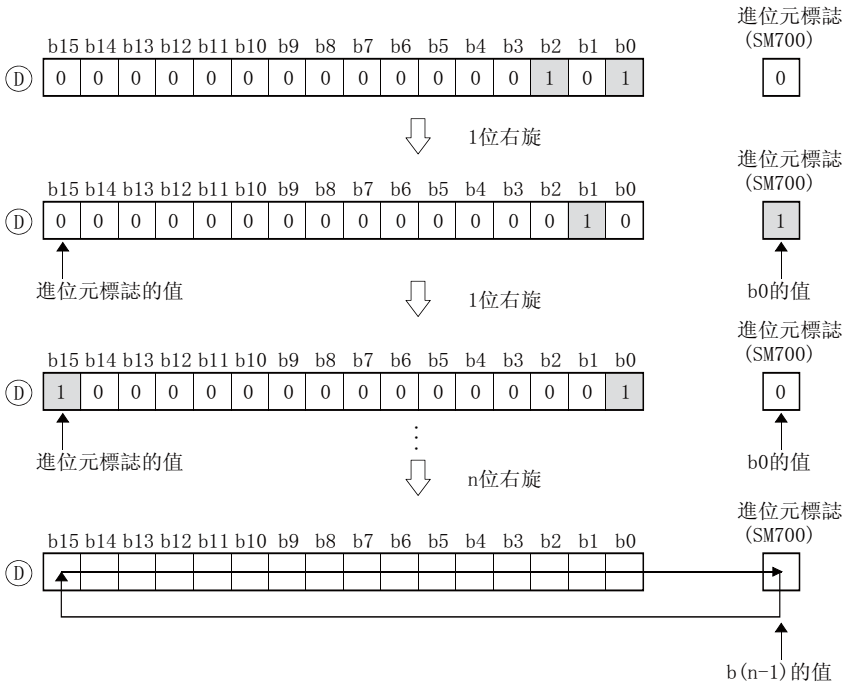


(2) Ⓓ中指定了軟元件的情況下，按位元數指定中指定的軟元件範圍進行旋轉。
此時，實際旋轉的位數爲 n ÷ (位數指定中指定的點數) 的餘數。
例如，n=15，(位數指定中指定的點數)=12 位時，15 ÷ 12=1 餘數爲 3，因此旋轉 3 位。

(3) n 的指定範圍爲 0 ～ 15。
n 中指定了 16 以上的值時，將按 n ÷ 16 的餘數值進行旋轉。
例如，n=18 時，18 ÷ 16=1 餘數爲 2，因此進行 2 位右旋。

RCR

(1) 將ⓓ中指定的軟元件的 16 位元資料在包含進位元標誌的狀態下進行 n 位元右旋轉。
進位元標誌根據 RCR 執行前的狀態而 ON/OFF。



(2) 在ⓓ中指定了軟元件的情況下，按位元數指定中指定的軟元件範圍進行旋轉。
此時，實際旋轉的位數爲 n ÷ (位數指定中指定的點數) 的餘數。
例如，n=15，(位數指定中指定的點數)=12 位時，15 ÷ 12=1 餘數爲 3，因此旋轉 3 位。

(3) n 的指定範圍爲 0 ～ 15。
n 中指定了 16 以上的值時，將按 n ÷ 16 的餘數值進行旋轉。
例如，n=18 時，18 ÷ 16=1 餘數爲 2，因此進行 2 位右旋。

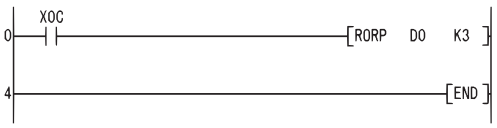
出　錯

(1) 在 ROR(P)、RCR(P) 指令中無運算出錯。

程式示例

(1) 以下爲 XC 變爲 ON 時，將 D0 的內容在不包含進位元標誌的狀態下進行 3 位元右旋轉的程式。

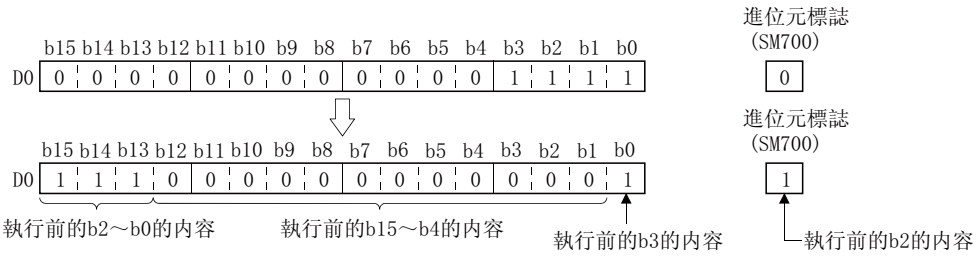
[梯形圖模式]



[列表模式]

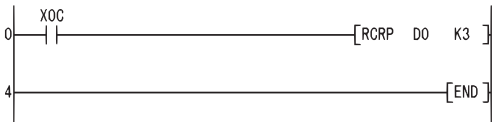
步	指令	軟元件	
0	LD	X0C	
1	RORP	D0	K3
4	END		

[動作]



(2) 以下爲 XC 變爲 ON 時，將 D0 的內容在包含進位元標誌的狀態下進行 3 位元右旋轉的程式。

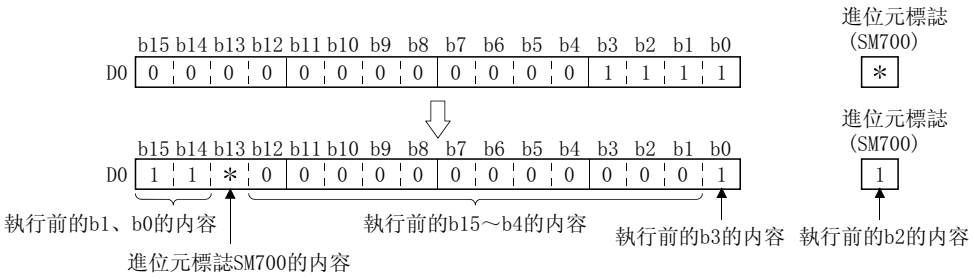
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	XOC	
1	RCRP	D0	K3
4	END		

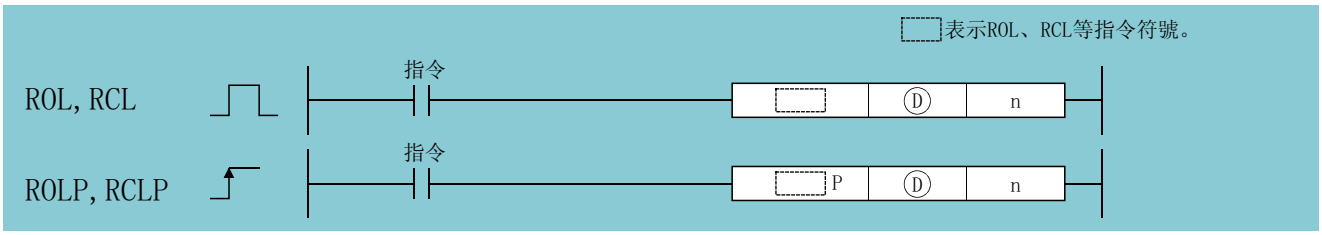
[動作]



*: 進位元標誌根據RCR執行前的狀態而ON/OFF。

7.2.2 ROL、ROLP、RCL、RCLP

Basic | High performance | Process | Redundant | Universal | LCPU



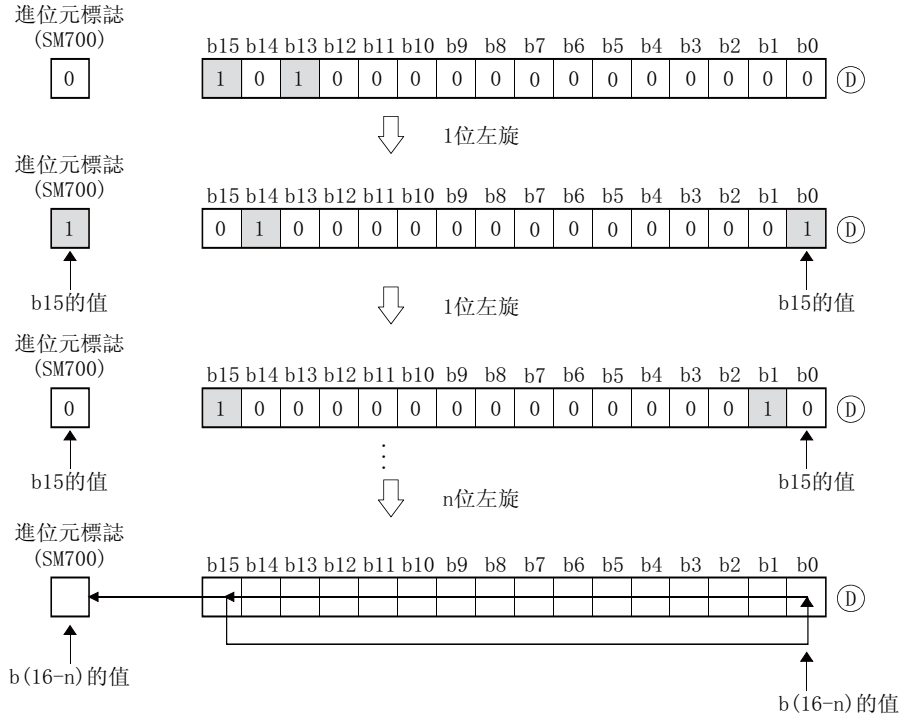
Ⓓ : 進行旋轉的軟元件的起始編號 (BIN16 位)。
n : 旋轉次數 (0 ～ 15)(BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓓ	○							---	---
n	○							○	---

功　能

ROL

(1) 將ⓓ中指定的軟元件的 16 位元資料在不包含進位元標誌的狀態下進行 n 位元左旋轉。
進位元標誌根據 ROL 執行前的狀態而 ON/OFF。

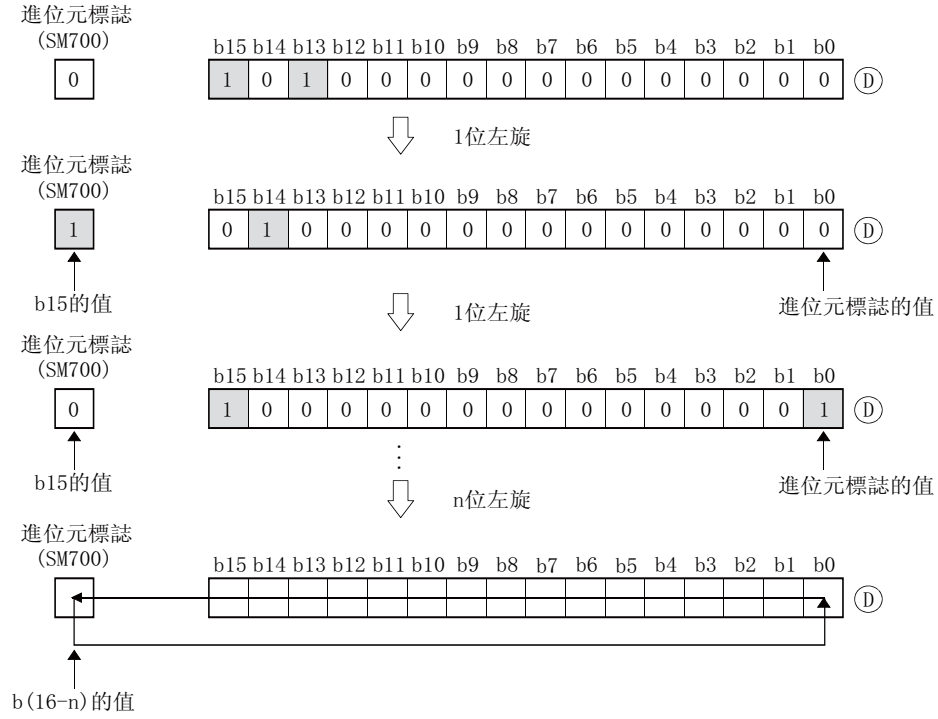


(2) 在ⓓ中指定了軟元件的情況下，按位元數指定中指定的軟元件範圍進行旋轉。
此時，實際旋轉的位數爲 n ÷（位數指定中指定的點數）的餘數。
例如，n=15，（位數指定中指定的點數)=12 位時，15 ÷ 12=1 餘數爲 3，因此旋轉 3 位。

(3) n 的指定範圍爲 0 ～ 15。
n 中指定了 16 以上的值時，將按 n ÷ 16 的餘數值進行旋轉。
例如，n=18 時，18 ÷ 16=1 餘數爲 2，因此進行 2 位左旋。

RCL

(1) 將ⓓ中指定的軟元件的 16 位元資料在包含進位元標誌的狀態下進行 n 位元左旋轉。
進位元標誌根據 RCL 執行前的狀態而 ON/OFF。



(2) 在ⓓ中指定了軟元件的情況下，按位元數指定中指定的軟元件範圍進行旋轉。
此時，實際旋轉的位數爲 n ÷ (位數指定中指定的點數) 的餘數。
例如，n=15，(位數指定中指定的點數)=12 位時，15 ÷ 12=1 餘數爲 3，因此旋轉 3 位。

(3) n 的指定範圍爲 0 ～ 15。
n 中指定了 16 以上的值時，將按 n ÷ 16 的餘數值進行旋轉。
例如，n=18 時，18 ÷ 16=1 餘數爲 2，因此進行 2 位左旋。

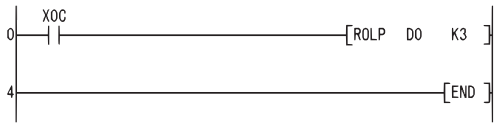
出　錯

(1) 在 ROL(P)、RCL(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X0C 變爲 ON 時，將 D0 的內容在不包含進位元標誌的狀態下進行 3 位元左旋轉的程式。

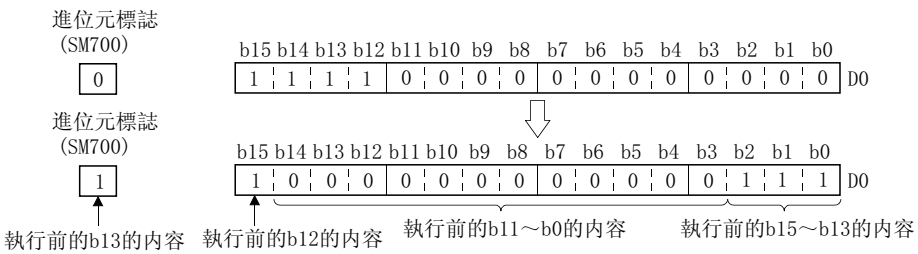
[梯形圖模式]



[列表模式]

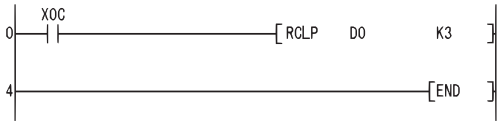
步	指令	軟元件	
0	LD	X0C	
1	ROLP	D0	K3
4	END		

[動作]



(2) 以下爲 XC 變爲 ON 時，將 D0 的內容在包含進位元標誌的狀態下進行 3 位元左旋轉的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X0C	
1	RCLP	D0	K3
4	END		

[動作]

*: 進位元標誌根據RCL執行前的狀態而ON/OFF。

7.2.3 DROR、DRORP、DRCR、DRCRP

Basic | High performance | Process | Redundant | Universal | LCPU

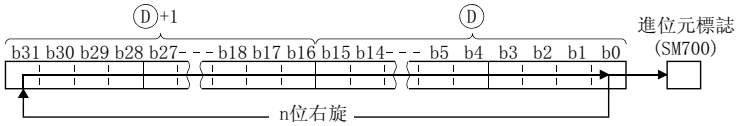
Ⓓ : 進行旋轉的軟元件的起始編號 (BIN32 位)。
n : 旋轉次數 (0～31)(BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓓ			○					---	---
n			○					○	---

功 能

DROR

(1) 將Ⓓ中指定的軟元件的 32 位元資料在不包含進位元標誌的狀態下進行 n 位元右旋轉。
進位元標誌根據 DROR 執行前的狀態而 ON/OFF。

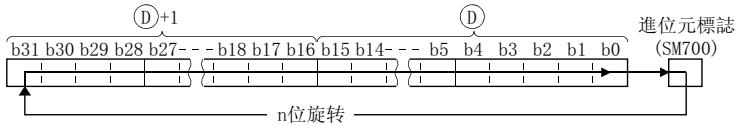


(2) 在Ⓓ中指定了軟元件的情況下，按位元數指定中指定的軟元件範圍進行旋轉。
此時，實際旋轉的位數爲 n ÷ (位數指定中指定的點數) 的餘數。
例如，n=31，(位數指定中指定的點數)=24 位時，31 ÷ 24=1 餘數爲 7，因此旋轉 7 位。

(3) n 的指定範圍爲 0 ～ 31。
n 中指定了 32 以上時，按 n ÷ 32 的餘數值進行旋轉。
例如，n=34 時，34 ÷ 32=1 餘數爲 2，因此進行 2 位右旋。

DRCR

(1) 將ⓓ中指定的軟元件的 32 位元資料在包含進位元標誌的狀態下進行 n 位元右旋轉。
進位元標誌根據 DRCR 執行前的狀態而 ON/OFF。



(2) 在ⓓ中指定了軟元件的情況下，按位元數指定中指定的軟元件範圍進行旋轉。
此時，實際旋轉的位數爲 n ÷ (位數指定中指定的點數) 的餘數。
例如，n=31，(位數指定中指定的點數)=24 位時，31 ÷ 24=1 餘數爲 7，因此旋轉 7 位。

(3) n 的指定範圍爲 0 ～ 31。
n 中指定了 32 以上的值時，將按 n ÷ 32 的餘數值進行旋轉。
例如，n=34 時，34 ÷ 32=1 餘數爲 2，因此進行 2 位右旋。

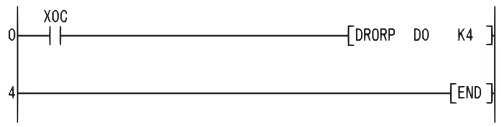
出　錯

(1) 在 DROR(P)、DRCR(P) 指令中無運算出錯。

程式示例

(1) 以下爲 XC 變爲 ON 時，將 D0、D1 的內容在不包含進位元標誌的狀態下進行 4 位元右旋轉的程式。

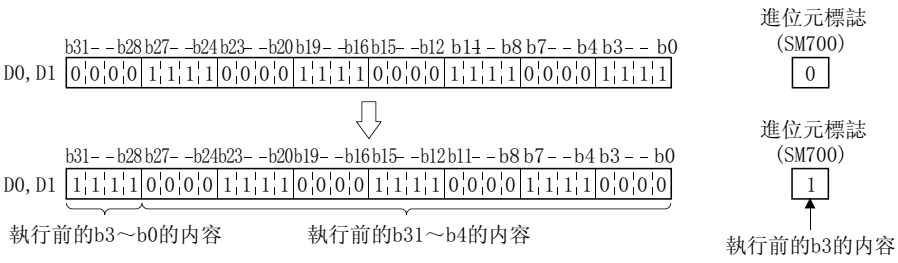
[梯形圖模式]



[列表模式]

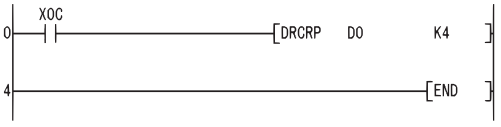
步	指令	軟元件	
0	LD	XOC	
1	DRORP	D0	K4
4	END		

[動作]



(2) 以下爲 XC 變爲 ON 時，將 D0、D1 的內容在包含進位元標誌的狀態下進行 4 位元右旋轉的程式。

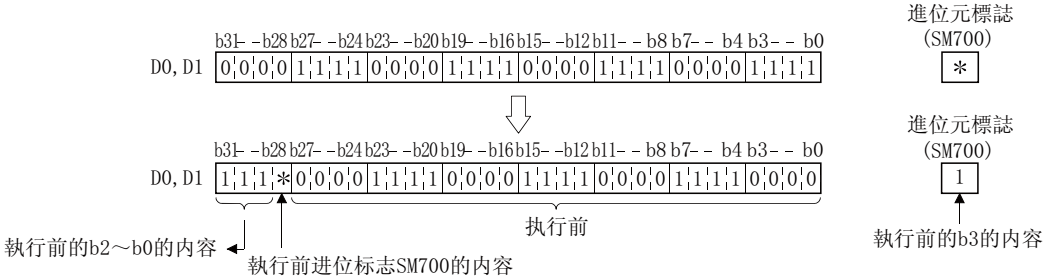
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	XOC	
1	DRCRP	D0	K4
4	END		

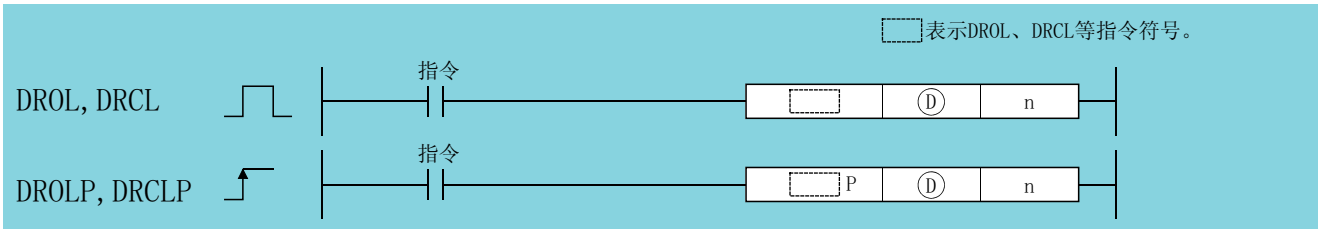
[動作]



*: 進位元標誌根據DRCR執行前的狀態而ON/OFF。

7.2.4 DROL、DROLP、DRCL、DRCLP

Basic | High performance | Process | Redundant | Universal | LCPU



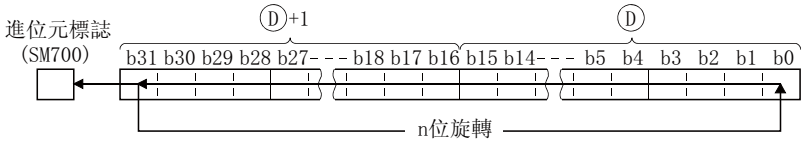
Ⓓ ：進行旋轉的軟元件的起始編號 (BIN32 位)。
n ：旋轉次數 (0～31)(BIN16 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓓ	○							---	---
n	○							○	---

功　能

DROL

(1) 將Ⓓ中指定的軟元件的 32 位元資料在不包含進位元標誌的狀態下進行 n 位元左旋轉。
進位元標誌根據 DROL 執行前的狀態而 ON/OFF。

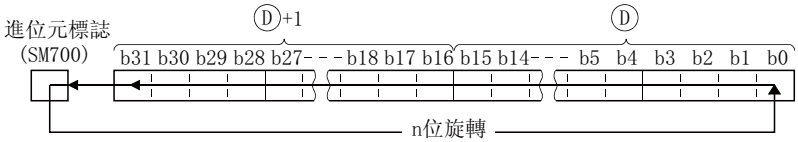


(2) 在Ⓓ中指定了軟元件的情況下，按位元數指定中指定的軟元件範圍進行旋轉。
此時，實際旋轉的位數爲 n ÷ (位數指定中指定的點數) 的餘數。
例如，n=31，(位數指定中指定的點數)=24 位時，31 ÷ 24=1 餘數爲 7，因此旋轉 7 位。

(3) n 的指定範圍爲 0 ～ 31。
n 中指定了 32 以上時，按 n ÷ 32 的餘數值進行旋轉。
例如，n=34 時，34 ÷ 32=1 餘數爲 2，因此進行 2 位左旋。

DRCL

(1) 將Ⓓ中指定的軟元件的 32 位元資料在包含進位元標誌的狀態下進行 n 位元左旋轉。
進位元標誌根據 DRCL 執行前的狀態而 ON/OFF。



(2) 在Ⓓ中指定了軟元件的情況下，按位元數指定中指定的軟元件範圍進行旋轉。
此時，實際旋轉的位數爲 n ÷ (位數指定中指定的點數) 的餘數。
例如，n=31，(位數指定中指定的點數)=24 位時，31 ÷ 24=1 餘數爲 7，因此旋轉 7 位。

(3) n 的指定範圍爲 0 ～ 31。
n 中指定了 32 以上的值時，將按 n ÷ 32 的餘數值進行旋轉。
例如，n=34 時，34 ÷ 32=1 餘數爲 2，因此進行 2 位左旋。

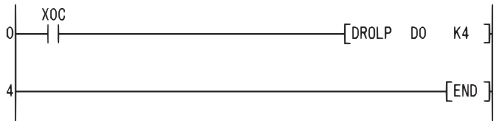
出　錯

(1) 在 DROL(P)、DRCL(P) 指令中無運算出錯。

程式示例

(1) 以下爲 XC 變爲 ON 時，將 D0、D1 的內容在不包含進位元標誌的狀態下進行 4 位元左旋轉的程式。

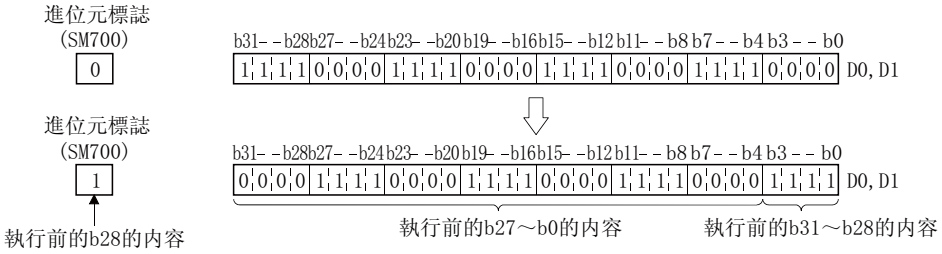
[梯形圖模式]



[列表模式]

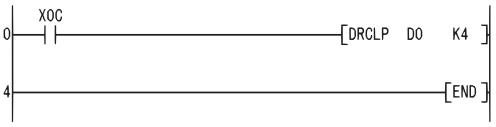
步	指令	軟元件	
0	LD	XOC	
1	DROLP	D0	K4
4	END		

[動作]



(2) 以下爲 XC 變爲 ON 時，將 D0、D1 的內容在包含進位元標誌的狀態下進行 4 位元左旋轉的程式。

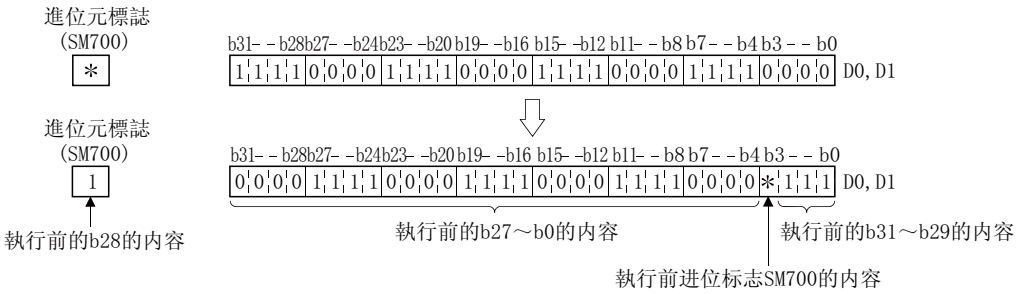
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	XOC	
1	DRCLP	D0	K4
4	END		

[動作]

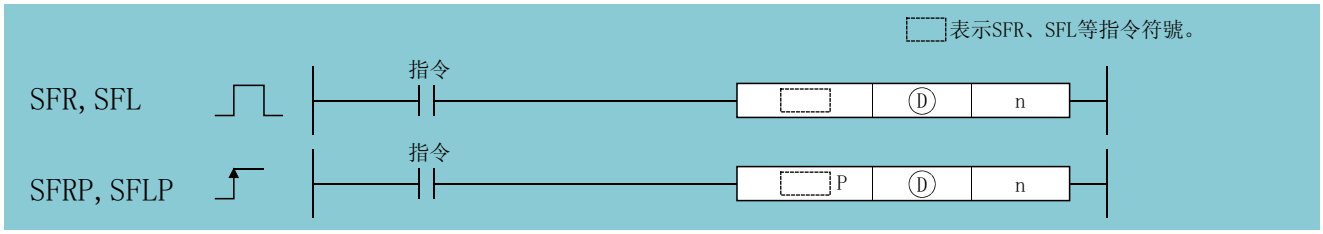


*: 進位元標誌根據DRCL執行前的狀態而ON/OFF。

7.3 移位元指令

7.3.1 SFR、SFRP、SFL、SFLP

Basic | High performance | Process | Redundant | Universal | LCPU



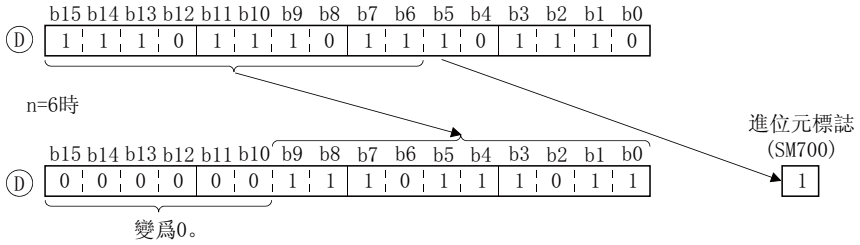
Ⓓ ：存儲移位元資料的軟元件的起始編號 (BIN16 位)。
n ：移位次數 (0 ～ 15)(BIN16 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓓ	○							---	---
n	○							○	---

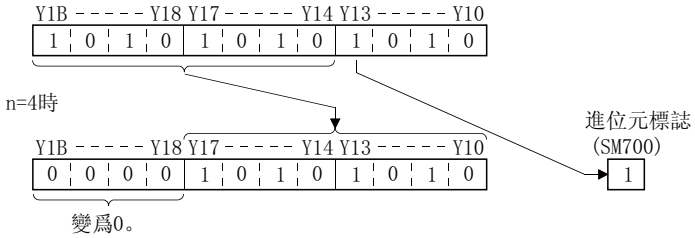
功 能

SFR

(1) 將Ⓓ中指定的軟元件的 16 位元資料右移 n 位元。
從最高位開始至 n 位爲止將變爲 0。



(2) Ⓓ中指定了位軟元件時，按位數指定中指定的軟元件範圍向右移位。

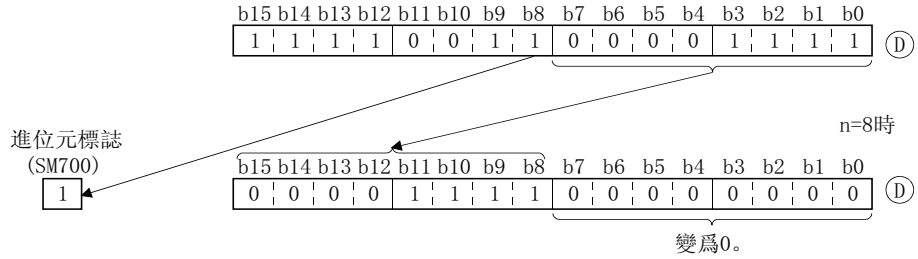


此時實際的移位位數爲 n ÷ (位數指定中指定的點數) 的餘數。
例如，n=15，(位數指定中指定的點數)=8 位時，15 ÷ 8=1 餘數爲 7，因此進行 7 位的移位。

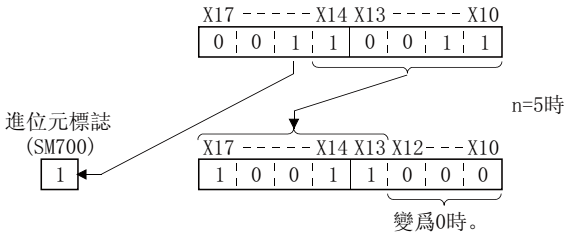
(3) n 的指定範圍爲 0 ～ 15。
在 n 中指定了 16 以上的值時，按 n ÷ 16 的餘數值進行右移。
例如，n=18 時，18 ÷ 16=1 餘數爲 2，因此右移 2 位。

SFL

(1) 將ⓓ中指定的軟元件的 16 位元資料左移 n 位元。
從最低位開始至 n 位爲止將變爲 0。



(2) ⓓ中指定了位軟元件時，按位數指定中指定的軟元件範圍向左移位。



此時實際的移位位數爲 n ÷ (位數指定中指定的點數) 的餘數。
例如，n=15，(位數指定中指定的點數)=8 位時，15 ÷ 8=1 餘數爲 7，因此進行 7 位的移位。

(3) n 的指定範圍爲 0 ～ 15。
在 n 中指定了 16 以上的值時，按 n ÷ 16 的余數值進行左移。
例如，n=18 時，18 ÷ 16=1 餘數爲 2，因此左移 2 位。

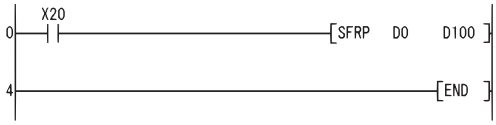
出　錯

(1) 在 SFR(P)、SFL(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D0 的內容按 D100 中指定的位元數右移的程式。

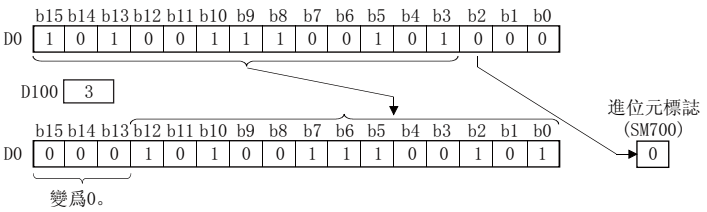
[梯形圖模式]



[列表模式]

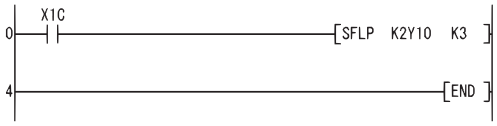
步	指令	軟元件	
0	LD	X20	
1	SFRP	D0	D100
4	END		

[動作]



(2) 以下爲 X1C 變爲 ON 時，將 X10 ～ X17 的內容左移 3 位元的程式。

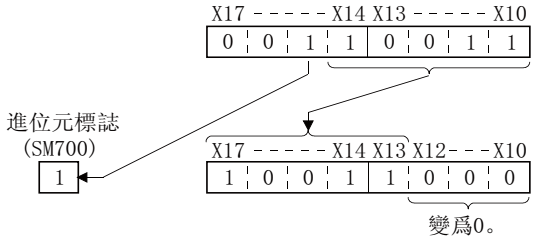
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X1C
1	SFLP	K2Y10 K3
4	END	

[動作]



7.3.2 BSFR、BSFRP、BSFL、BSFLP

Basic | High performance | Process | Redundant | Universal | LCPU

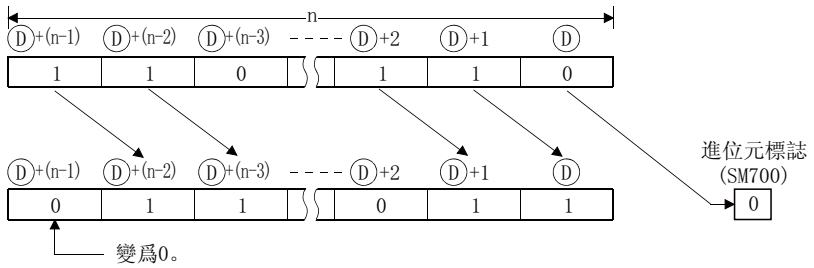
Ⓓ ：存儲移位元資料的軟元件的起始編號 (BIN16 位)。
n ：移位次數 (0 ～ 15)(BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓓ	○	---							---
n	○	○							---

功 能

BSFR

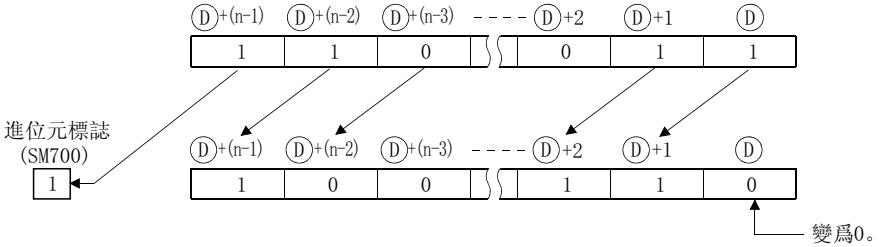
(1) 將Ⓓ中指定的軟元件開始的 n 點數據右移 1 位。



(2) Ⓓ+(n-1) 中指定的軟元件將變爲 0。

BSFL

(1) 將ⓓ中指定的軟元件開始的 n 點數據左移 1 位。



(2) ⓓ中指定的軟元件將變爲 0。

出　錯

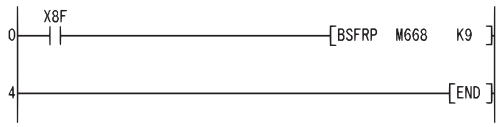
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從ⓓ的軟元件開始的 n 點範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X8F 變爲 ON 時，將 M668 ～ M676 的資料右移的程式。

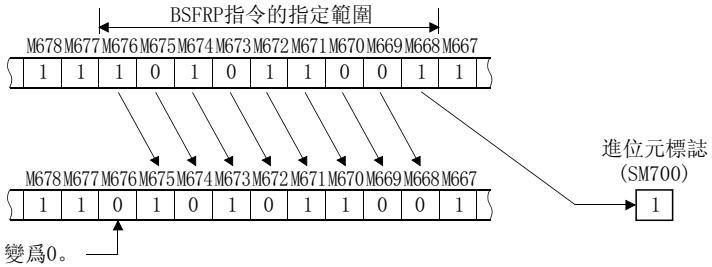
[梯形圖模式]



[列表模式]

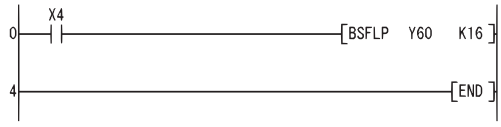
步	指令	軟元件	
0	LD	X8F	
1	BSFRP	M668	K9
4	END		

[動作]



(2) 以下爲 X4 變爲 ON 時，將 Y60 ～ Y6F 的資料左移的程式。

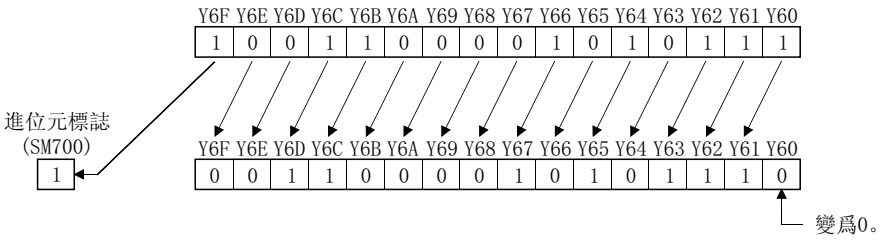
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X4	
1	BSFLP	Y60	K16
4	END		

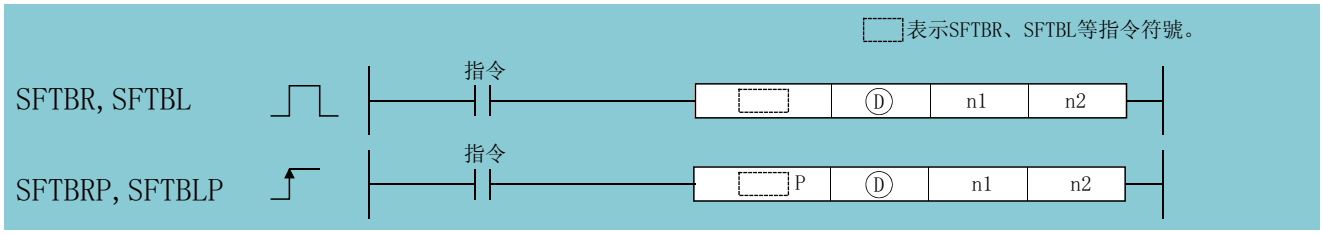
[動作]



Basic × High performance × Process × Redundant × Ver. Universal LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。

7.3.3 SFTBR、SFTBRP、SFTBL、SFTBLP



Ⓓ ：移位的軟元件的起始編號 (位)。
n1 ：進行移位的位數 (BIN16 位)。
n2 ：移位數 (BIN16 位)。

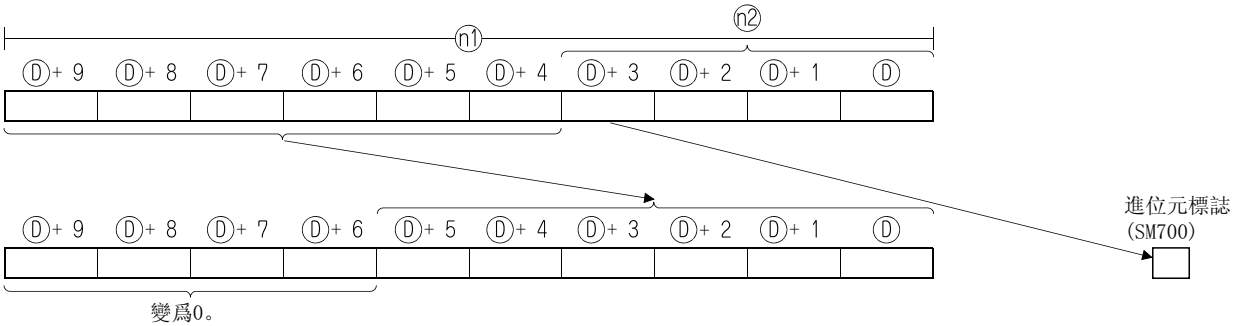
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓓ	○ *2	---	○	---					---
n1	---	○	○	○					---
n2	---	○	○	○					---

*2: T、C、ST、S 軟元件不能使用。

功 能

SFTBR(P)

(1) 將從Ⓓ中指定的軟元件開始至 n1 位元爲止的資料右移 n2 位元。
n1=10，n2=4 時



(2) n1、n2 的指定應滿足 n1>n2 的條件。n1 ≤ n2 的情況下，將按 n2 ÷ n1 的餘數值進行移位。
但是，n2 ÷ n1 的餘數值爲 0 時，將變爲無處理。

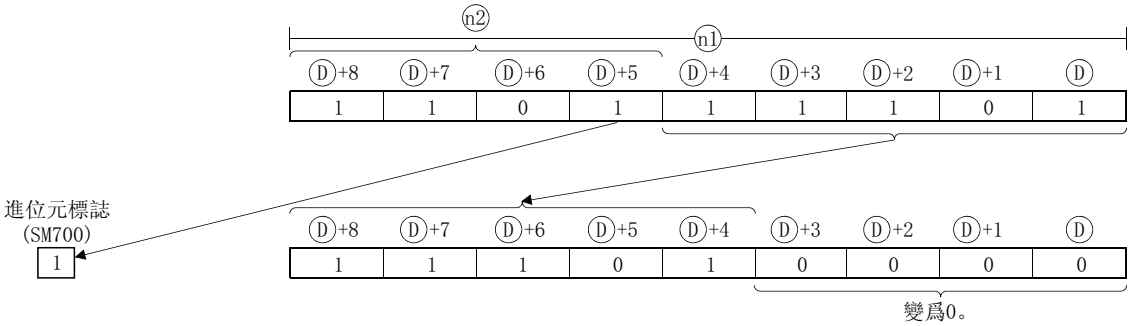
(3) n1 的設置範圍爲 1 ～ 64。

(4) 從最低位元開始至 n2 位元爲止的資料將變爲 0。n1<n2 的情況下，n2 ÷ n1 的餘數值部分將變爲 0。

(5) n1 或者 n2 中指定的值爲 0 時，將變爲無處理。

SFTBL(P)

(1) 將從ⓓ中指定的軟元件開始至 n1 位元爲止的資料左移 n2 位元。
n1=10，n2=4 時



(2) n1、n2 的指定應滿足 n1>n2 的條件。n1 ≤ n2 的情況下，將按 n2 ÷ n1 的餘數值進行移位。
但是，n2 ÷ n1 的餘數值爲 0 時，將變爲無處理。

(3) n1 的設置範圍爲 1 ～ 64。

(4) 從最低位元開始至 n2 位元爲止的資料將變爲 0。n1<n2 的情況下，n2 ÷ n1 的餘數值部分將變爲 0。

(5) n1 或者 n2 中指定的值爲 0 時，將變爲無處理。

出　錯

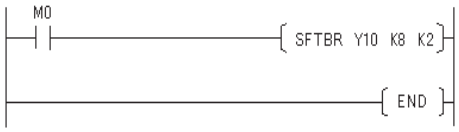
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n1 超出了 0 ～ 64 的範圍時。 n2 爲負數時。	○	○	○	○	○	○
4101	n1 中指定的軟元件點數超出了ⓓ中指定的軟元件範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 M0 變爲 ON 時，將ⓓ中指定的 Y10 ～ Y17(8 位元) 的內容右移 2(n2) 位元的程式。

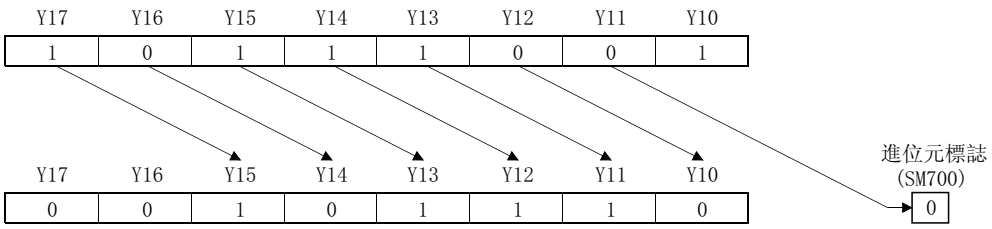
[梯形圖模式]



[列表模式]

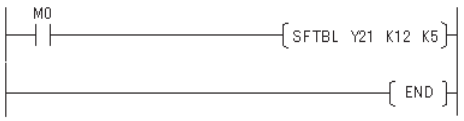
步	指令	軟元件
0	LD	M0
1	SFTBR	Y10 K8 K2
5	END	

[動作]



(2) 以下爲 M0 變爲 ON 時，將Ⓓ中指定的 Y21 ～ Y2C(12 位元) 的內容左移 5(n2) 位元的程式。

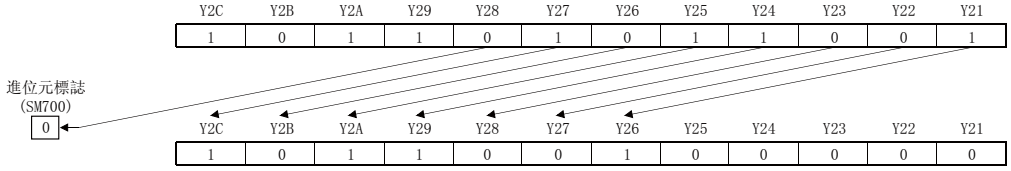
[梯形圖模式]



[列表模式]

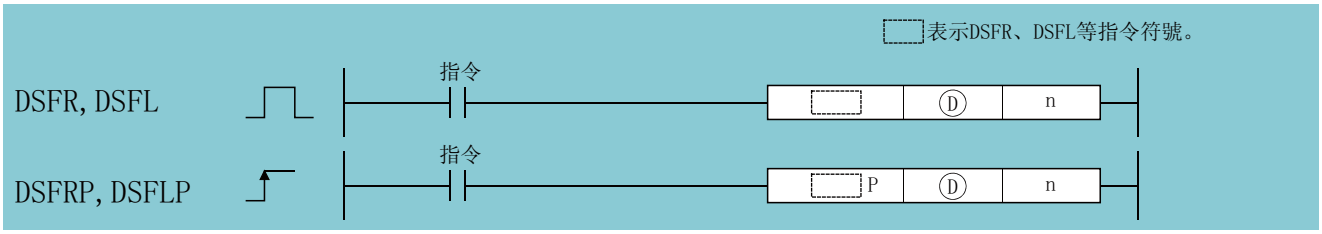
步	指令	軟元件
0	LD	M0
1	SFTBL	Y21 K12 K5
5	END	

[動作]



7.3.4 DSFR、DSFRP、DSFL、DSFLP

Basic | High performance | Process | Redundant | Universal | LCPU



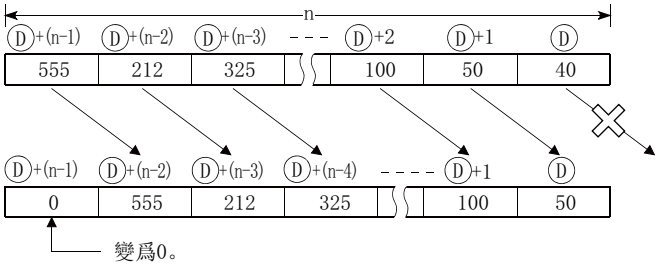
Ⓓ ：移位的軟元件的起始編號 (BIN16 位)。
n ：移位的軟元件的數 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓓ	---	○		---					---
n	○	○		○					---

功 能

DSFR

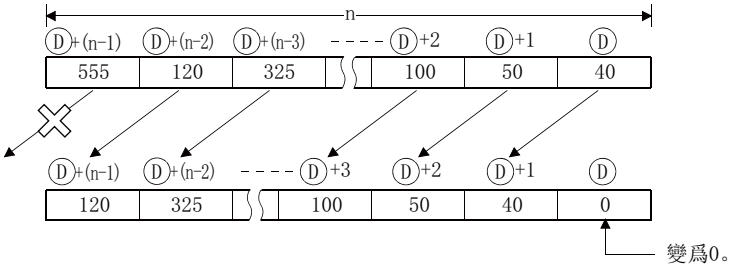
(1) 將從Ⓓ中指定的軟元件開始的 n 點數據右移 1 字。



(2) Ⓓ+(n-1) 中指定的軟元件將變爲 0。

DSFL

(1) 將從Ⓓ中指定的軟元件開始的 n 點數據左移 1 字。



(2) Ⓓ中指定的軟元件將變爲 0。

出　錯

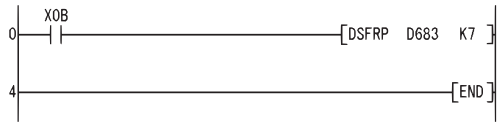
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從⑩的軟元件開始的 n 點的範圍超出了相應軟元件時。	○	○	○	○	○	○

程式示例

(1) 以下爲 XB 變爲 ON 時，將 D683 ～ D689 的內容右移的程式。

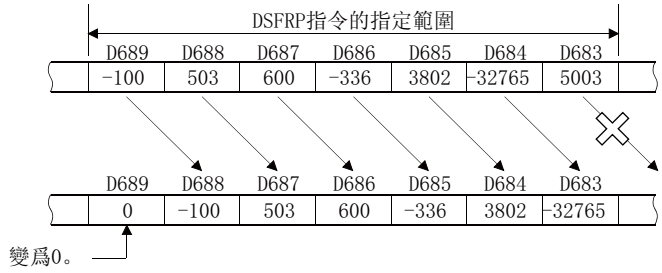
[梯形圖模式]



[列表模式]

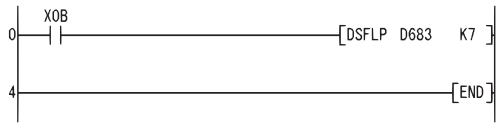
步	指令	軟元件	
0	LD	XOB	
1	DSFRP	D683	K7
4	END		

[動作]



(2) 以下爲 XB 變爲 ON 時，將 D683 ～ D689 的內容左移的程式。

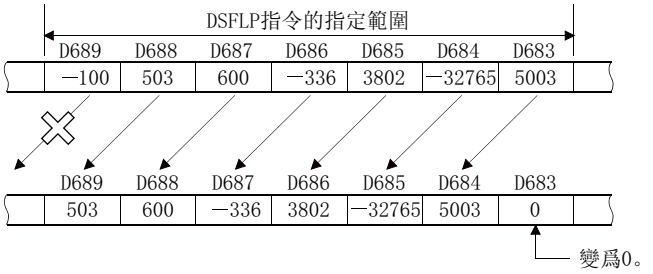
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	XOB	
1	DSFLP	D683	K7
4	END		

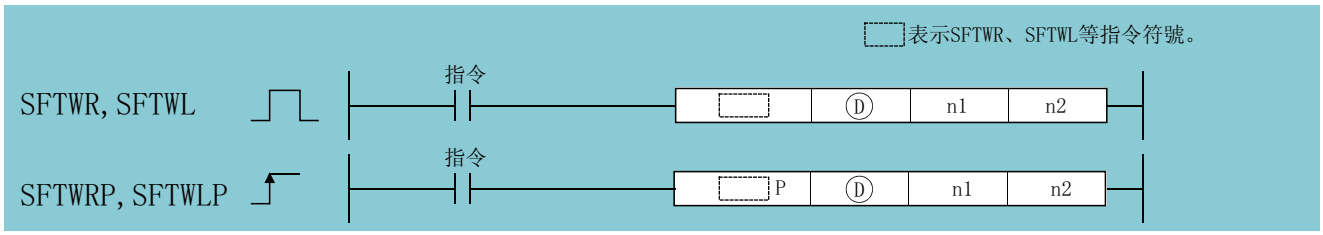
[動作]



7.3.5 SFTWR、SFTWRP、SFTWL、SFTWLP

Basic × | High performance × | Process × | Redundant × | Ver. Universal | LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。



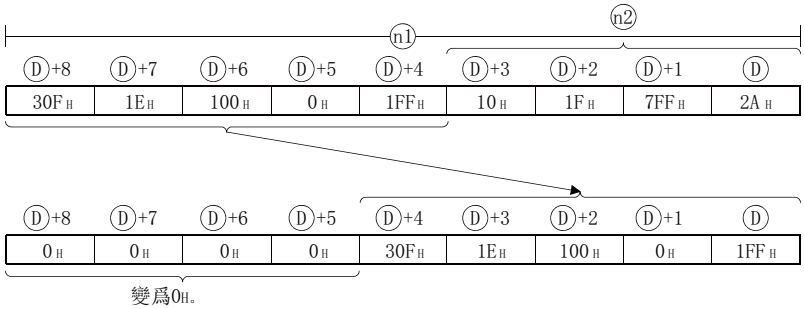
Ⓓ ：移位的軟元件的起始編號 (BIN16 位)。
n1 ：進行移位的字數 (BIN16 位)。
n2 ：移位數 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓓ	---	○	○	---					---
n1	---	○	○	○					---
n2	---	○	○	○					---

功 能

SFTWR(P)

(1) 將從Ⓓ中指定的軟元件開始至 n1 字爲止的資料右移 n2 字。
n1=9，n2=4 時



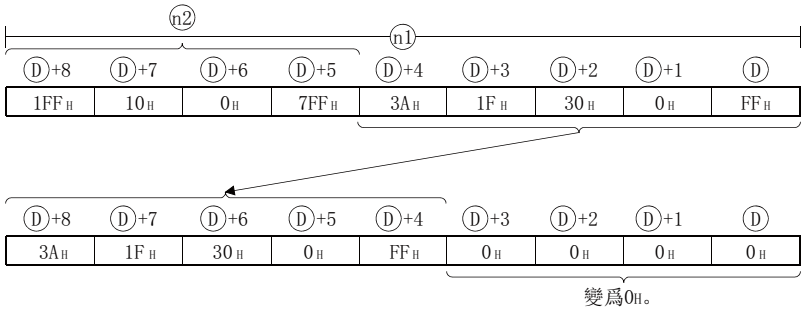
(2) 從最高位開始至 n2 字爲止將變爲 0。

(3) n1 或者 n2 中指定的值爲 0 時，將變爲無處理。

(4) n1 ≤ n2 的情況下，從Ⓓ中指定的軟元件開始至 n1 字爲止的資料將全部變爲 0。

SFTWL(P)

(1) 將從Ⓓ中指定的軟元件開始至 n1 字爲止的資料左移 n2 字。
n1=9，n2=4 時



(2) 從最低位開始至 n2 字爲止將變爲 0。

(3) n1 或者 n2 中指定的值爲 0 時，將變爲無處理。

(4) n1 ≤ n2 的情況下，從Ⓓ中指定的軟元件開始至 n1 字爲止的資料將全部變爲 0。

出　錯

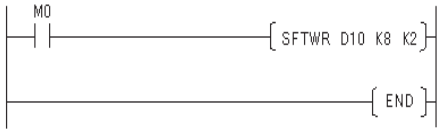
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n1 或者 n2 爲負數時。	---	---	---	---	○	○
4101	n1 中指定的軟元件點數超出了ⓓ中指定的軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 M0 變爲 ON 時，將ⓓ中指定的 D10 開始的 8(n1) 字的內容右移 2(n2) 字的程式。

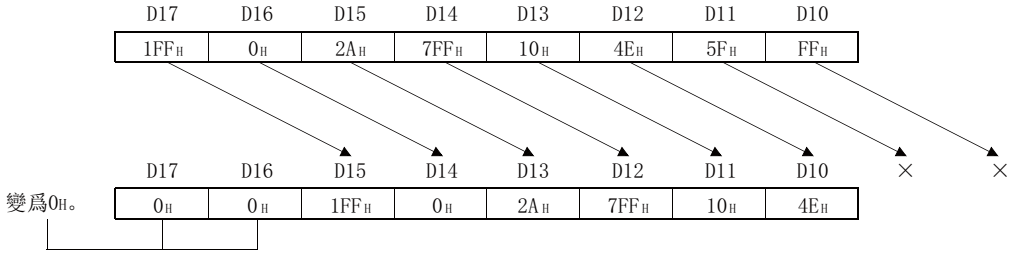
[梯形圖模式]



[列表模式]

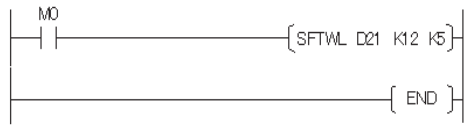
步	指令	軟元件
0	LD	M0
1	SFTWR	D10 K8 K2
5	END	

[動作]



(2) 以下爲 M0 變爲 ON 時，將ⓓ中指定的 D21 開始的 12(n1) 字的內容左移 5(n2) 字的程式。

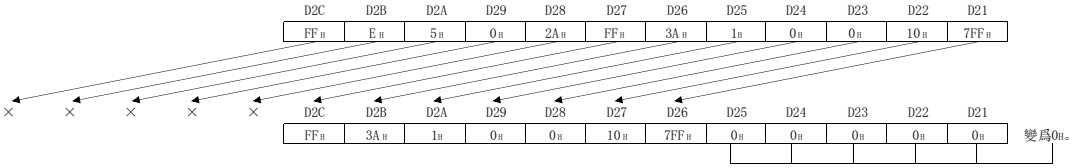
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	M0
1	SFTWL	D21 K12 K5
5	END	

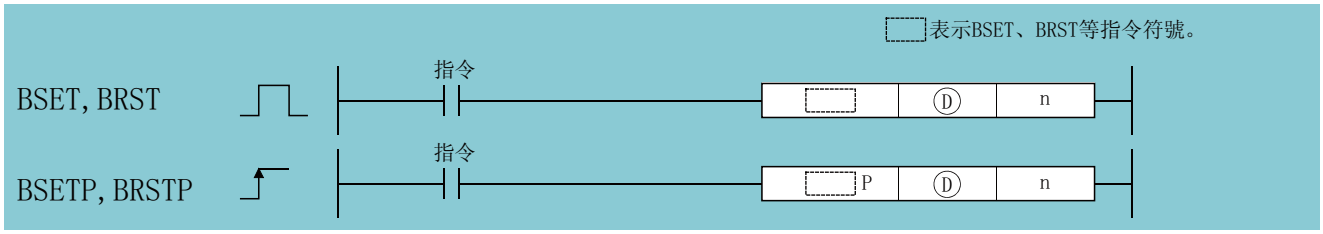
[動作]



7.4 位元處理指令

7.4.1 BSET、BSETP、BRST、BRSTP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓓ ：進行位設置、重定的軟元件編號 (BIN16 位)。
n ：進行位設置、重定的位數 (0 ～ 15)(BIN16 位)。

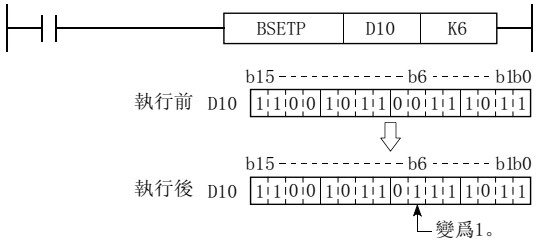
設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓓ	○							---	---
n	○							○	---

功　能

BSET

(1) 對Ⓓ中指定的字軟元件的第 n 位進行設置 (1)。

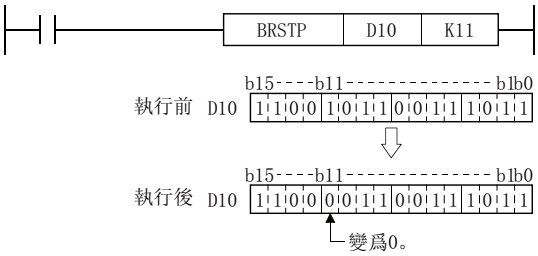
(2) n 中超過了 15 時，以低 4 位元的資料執行。



BRST

(1) 對Ⓓ中指定的字軟元件的第 n 位進行重定 (0)。

(2) n 中超過了 15 時，以低 4 位元的資料執行。



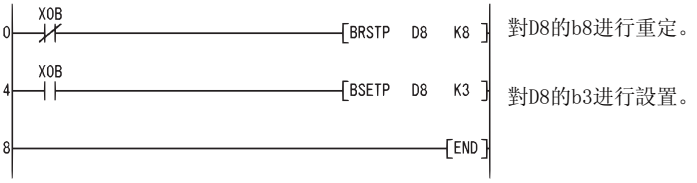
出　錯

(1) 在 BSET(P)、BRST(P) 指令中無運算出錯。

程式示例

(1) 以下爲 XB 變爲 OFF 時，對 D8 的第 8 位 (8) 進行重定 (0)，XB 變爲 ON 時，對 D8 的第 3 位 (b3) 進行設置 (1) 的程式。

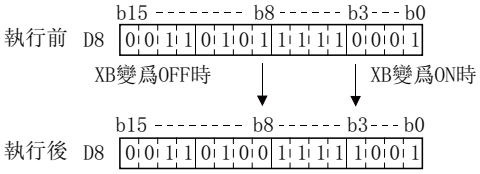
[梯形圖模式]



[列表模式]

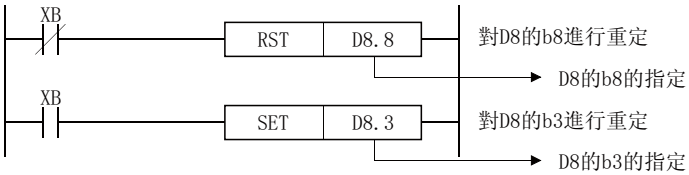
步	指令	軟元件	
0	LDI	X0B	
1	BRSTP	D8	K8
4	LD	X0B	
5	BSETP	D8	K3
8	END		

[動作]



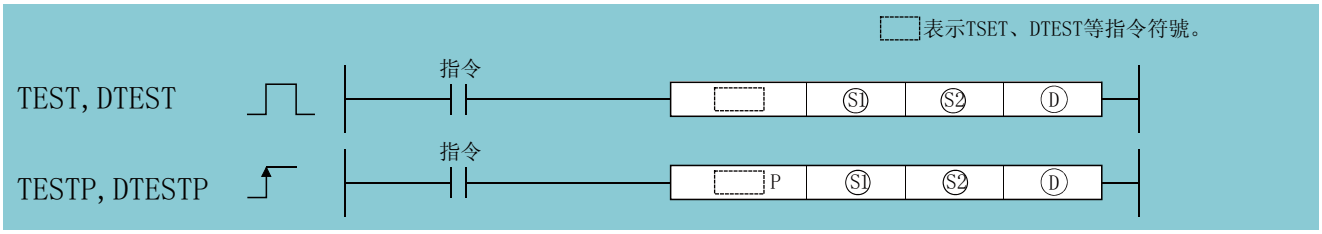
備 註

對於字軟元件的位設置 / 重定，也可以通過字軟元件的位指定來進行。
關於字軟元件的位指定，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。
如果使用了字軟元件的位元指定，程式示例 (1) 的處理過程如下所示。



7.4.2 TEST、TESTP、DTEST、DTESTP

Basic | High performance | Process | Redundant | Universal | LCPU



(S1) ：存儲被提取的位元資料的軟元件編號 (BIN16 位)。

(S2) ：被提取的位元資料的位置 (0 ～ 15(TEST)/0 ～ 31(DTEST))(BIN16/32 位)。

(D) ：存儲被提取的位元資料的位元軟元件編號 (位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
(S1)			○				○	---	---
(S2)			○				○	○	---
(D)			○				---	---	---

功　能

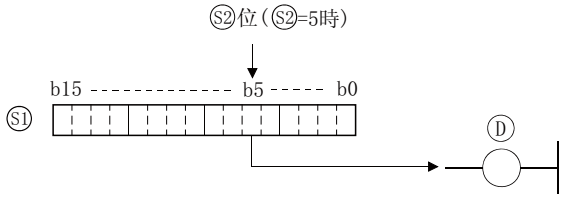
TEST

(1) 在Ⓢ1中指定的軟元件內，對Ⓢ2中指定位置的位元資料進行提取後，寫入到Ⓓ中指定的位軟元件中。

(2) 對於Ⓓ中指定的位軟元件，其相應位爲“0”時變爲 OFF，爲“1”時變爲 ON。

(3) Ⓢ2中指定的位置表示 1 字資料的各個位元位置 (0 ～ 15)。

在Ⓢ2中指定了 16 以上時，n ÷ 16 的餘數值位置的位元資料將成爲物件。
例如，n=18 時，18 ÷ 16=1 餘數爲 2，因此變爲 b2 的資料。



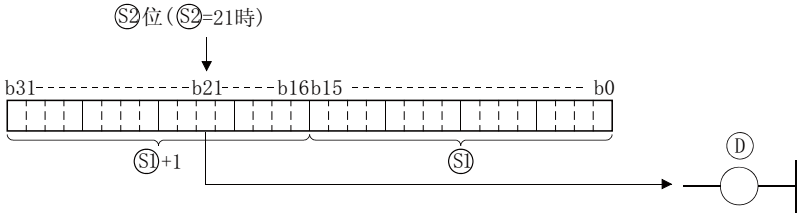
DTEST

(1) 在Ⓢ1、Ⓢ1+1 中指定的 2 字軟元件內，對Ⓢ2中指定位置的位元資料進行提取後，寫入到Ⓓ中指定的位軟元件中。

(2) 對於Ⓓ中指定的位軟元件，其相應位爲“0”時變爲 OFF，爲“1”時變爲 ON。

(3) Ⓢ2中指定的位置表示 2 字資料的各個位元位置 (0 ～ 31)。

在Ⓢ2中指定了 32 以上時，n ÷ 32 的餘數值位置的位元資料將成爲物件。
例如，n=34 時，34 ÷ 32=1 餘數爲 2，因此變爲 b2 的資料。



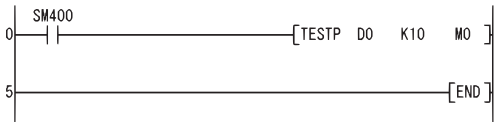
出　錯

(1) 在 TEST(P)、DTEST(P) 指令中無運算出錯。

程式示例

(1) 以下爲根據 1 字資料 (D0) 的第 10 位元的狀態，對 M0 進行 ON/OFF 的程式。

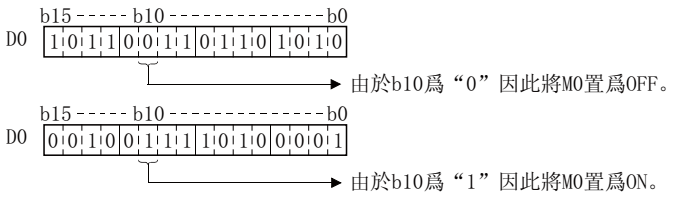
[梯形圖模式]



[列表模式]

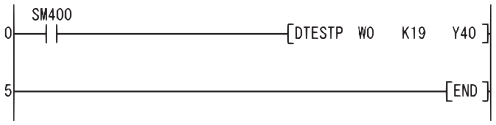
步	指令	軟元件		
0	LD	SM400		
1	TESTP	D0	K10	M0
5	END			

[動作]



(2) 以下爲根據 2 字資料 (W0、W1) 的第 19 位元的狀態，對 Y40 進行 ON/OFF 的程式。

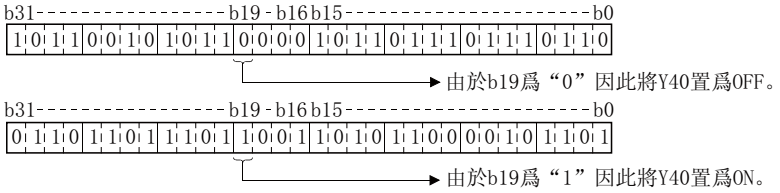
[梯形圖模式]



[列表模式]

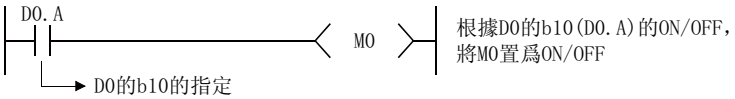
步	指令	軟元件		
0	LD	SM400		
1	DTESTP	W0	K19	Y40
5	END			

[動作]



備 註

對於使用位元測試指令的程式，可以用使用字軟元件的位元指定的程式進行替換。
對程式示例 (1) 的程式使用字軟元件的位元指定時的情況如下所示：



7.4.3 BKRST、BKRSTP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：進行重定的軟元件的起始編號 (位)。
n ：進行重定的軟元件數 (BIN16 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓓ		○			---				---
n		○			○				---

功 能

(1) 從Ⓓ中指定的位軟元件開始，對 n 點的位軟元件進行重定。

軟元件	狀態
報警器 (F)	·將Ⓓ中指定的報警器 (F) 號開始的 n 點軟元件置爲 OFF。 ·將 OFF 狀態的報警器號從 SD64 ～ SD79 中刪除後將剩餘的向前對齊。 ·將存儲在 SD64 ～ SD79 中的報警器數存儲到 SD63 中。
計時器 (T) 記數器 (C)	·將從Ⓓ中指定的計時器 (T) 或計數器 (C) 開始的 n 點的當前值置爲 0 後，將線圈觸點置爲 OFF
除上述以外的其他位軟元件	·將從Ⓓ中指定的軟元件開始的 n 點的線圈、觸點置爲 OFF。

(2) 指定的軟元件爲 OFF 時，軟元件的狀態不發生變化。

出　錯

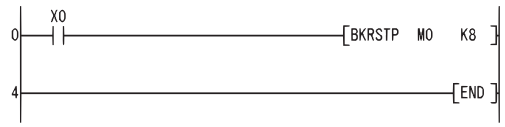
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從ⓓ的軟元件開始的 n 點的範圍超出了相應軟元件時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 M0 ～ M7 置爲 OFF 的程式。

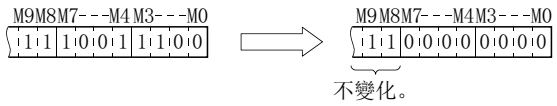
[梯形圖模式]



[列表模式]

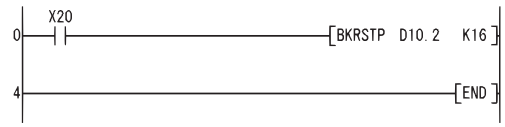
步	指令	軟元件	
0	LD	X0	
1	BKRSTP	M0	K8
4	END		

[動作]



(2) 以下爲 X20 變爲 ON 時，將 D10 的第 2 位 (b2) ～ D11 的第 1 位 (b1) 置爲 0 的程式。

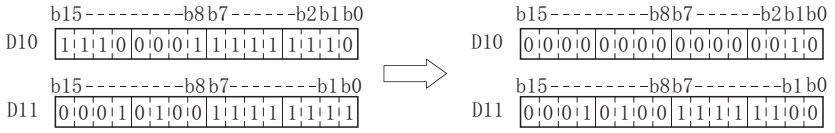
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X20	
1	BKRSTP	D10.2	K16
4	END		

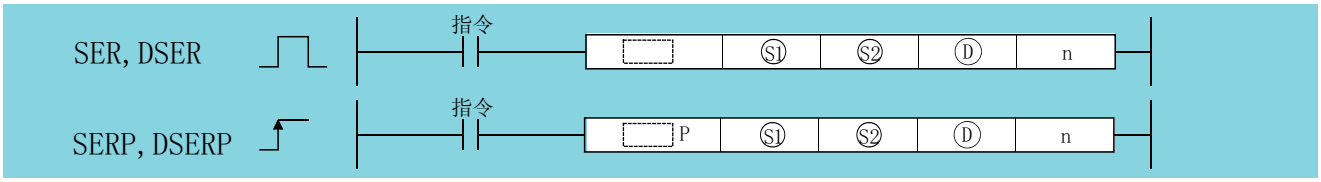
[動作]



7.5 資料處理指令

7.5.1 SER、SERP、DSER、DSERP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ1：搜索的資料或者存儲搜索資料的軟元件的起始編號 (BIN16/32 位)。

Ⓢ2：被搜索的資料或者存儲被搜索資料的軟元件的起始編號 (BIN16 位)。

Ⓓ：存儲搜索結果的軟元件的起始編號 (BIN16 位)。

n ：搜索數 (BIN16 位)。

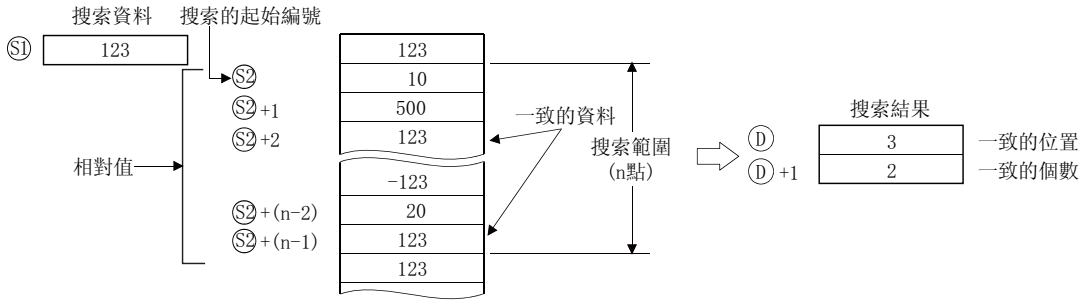
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	○	○		○	○			○	---
Ⓢ2	---	○		---	---			---	---
Ⓓ	---	○		---	○			---	---
n	○	○		○	○			○	---

功 能

SER

(1) 將Ⓢ1中指定的軟元件的 16 位元資料作爲關鍵字，從Ⓢ2中指定的軟元件的 16 位元資料開始至 n 點爲止進行搜索。

將與關鍵字一致的個數存儲到Ⓓ+1 中指定的軟元件中，將最先一致的軟元件號的Ⓢ2從算起的相對值存儲到Ⓓ中指定的軟元件中。



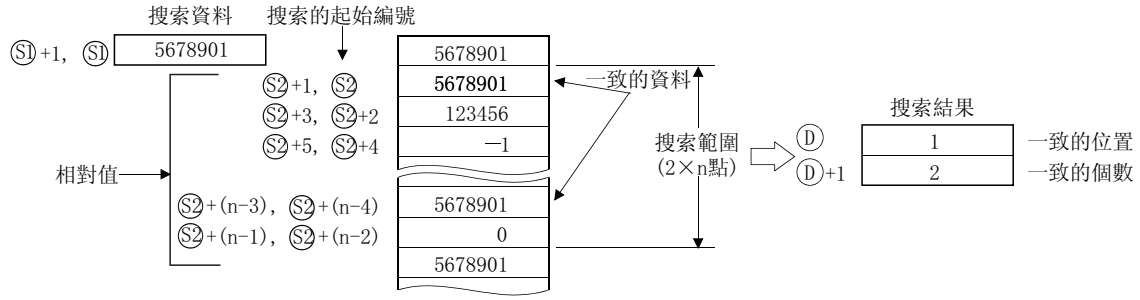
(2) n 爲 0 或者負數時，執行無處理。

(3) 未搜索到一致的資料時，Ⓓ、Ⓓ+1 中指定的軟元件將變爲 “ 0 ”。

DSER

(1) 將Ⓢ1+1,Ⓢ1中指定的軟元件的 32 位元資料作爲關鍵字，從Ⓢ2中指定的軟元件開始以 32 位元爲單位對 n 點（以 16 位元爲單位時爲 2 × n 點）的範圍進行搜索。

將與關鍵字一致的個數存儲到Ⓓ+1 中指定的軟元件中，將最先一致的軟元件號的從Ⓢ2算起的相對值存儲到ⒹⒹ中指定的軟元件中。



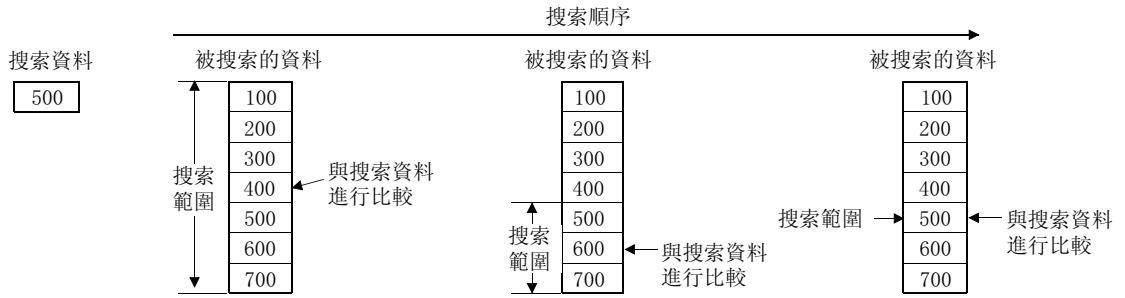
(2) n 爲 0 或者負數時，執行無處理。

(3) 未搜索到一致的資料時，Ⓓ、Ⓓ+1 中指定的軟元件將變爲 “ 0 ”。

要 點

在 SER、DSER 指令中，當被搜索的資料爲昇冪排序時，如果將 SM702*1 置爲 ON，通過二分搜索法進行搜索，可以加快搜索處理速度。
被搜索的資料未以昇冪排序時，如果將 SM702 置爲 ON，將無法獲得正常的搜索結果。

* 1: SM702 是用於設置搜索方法的特殊繼電器。
・SM702 爲 OFF 時：逐次搜索法（線性搜索法）
（從被搜索的資料的起始開始與搜索資料進行比較的方法。）
・SM702 爲 ON 時：二分搜索法
（該方法對於以昇冪排序的資料，找出位於搜索範圍中間的值，將該值與要搜索的值進行大小比較，以確定搜索方向，鎖定及縮小搜索範圍。如此反復執行此操作，找出要搜索的資料。）



出 錯

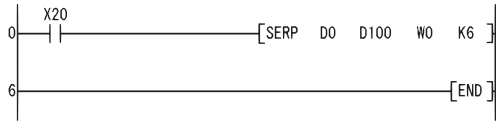
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓢ2的軟元件開始的 n 點的範圍超出了指定軟元件範圍時。	○	○	○	○	○	○
	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，在 D100 ～ D103 中對 D0 的內容進行搜索，並將搜索結果存儲到 W0、W1 中的程式。

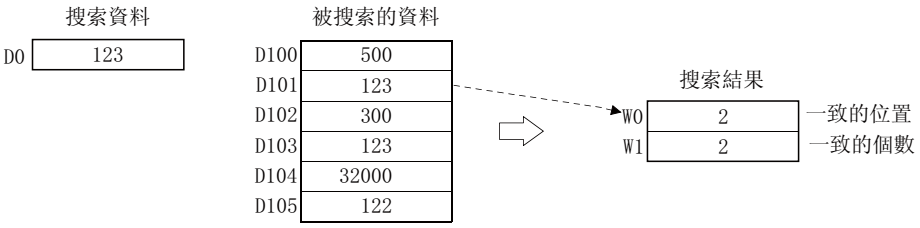
[梯形圖模式]



[列表模式]

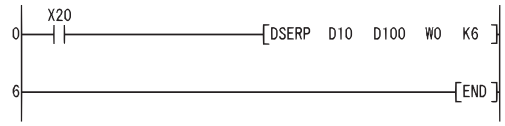
步	指令	軟元件
0	LD	X20
1	SERP	D0 D100 W0 K6
6	END	

[動作]



(2) 以下爲 X20 變爲 ON 時，在 D100 ～ D111 中對 D11、D10 的內容進行搜索，並將搜索結果存儲到 W0、W1 中的程式。

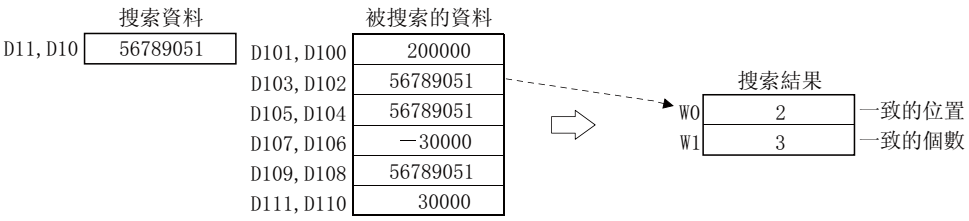
[梯形圖模式]



[列表模式]

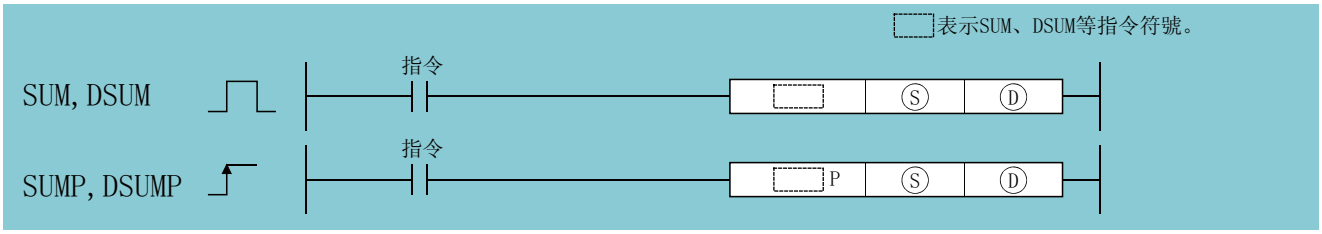
步	指令	軟元件
0	LD	X20
1	DSERP	D10 D100 W0 K6
6	END	

[動作]



7.5.2 SUM、SUMP、DSUM、DSUMP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ：對處於 1 狀態的位元的總數進行計數的軟元件的起始編號 (BIN16/32 位)。

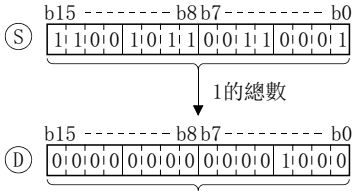
Ⓓ：存儲位元的總數的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功　能

SUM

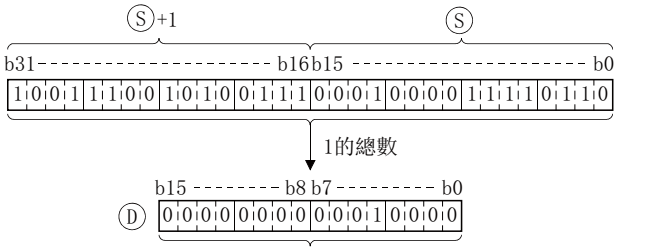
在Ⓢ中指定的軟元件的 16 位元資料中，將處於 1 狀態的位元的總數存儲到Ⓓ中指定的軟元件中。



將1的總數以BIN格式進行存儲。(在本例中爲8個。)

DSUM

在Ⓢ中指定的軟元件的 32 位元資料中，將處於 1 狀態的位元的總數存儲到Ⓓ中指定的軟元件中。



將1的總數以BIN格式進行存儲。(在本例中爲16個。)

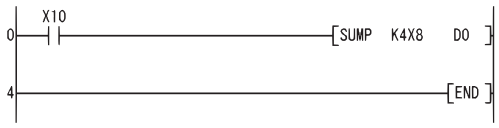
出　錯

(1) 在 SUM(P)、DSUM(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X10 變爲 ON 時，將 X8 ～ X17 中處於 ON 狀態的位元數存儲到 D0 中的程式。

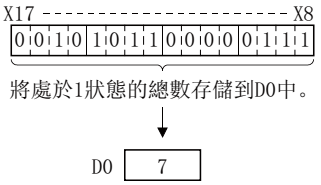
[梯形圖模式]



[列表模式]

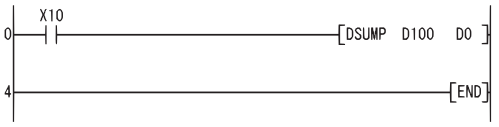
步	指令	軟元件	
0	LD	X10	
1	SUMP	K4X8	D0
4	END		

[動作]



(2) 以下爲 X10 變爲 ON 時，將 D100、D101 中處於 ON 狀態的位元數存儲到 D0 中的程式。

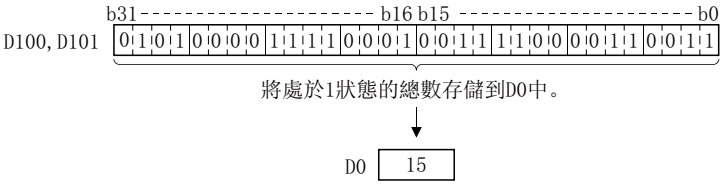
[梯形圖模式]



[列表模式]

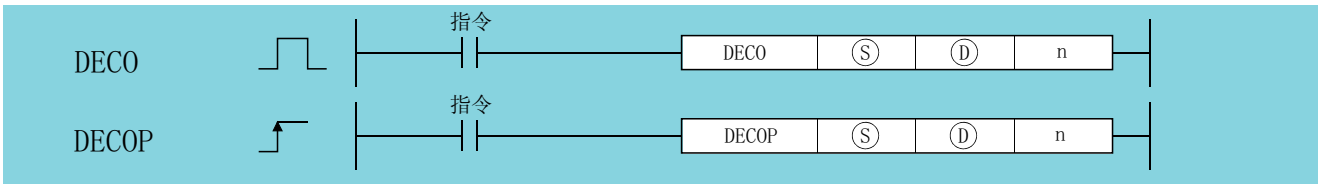
步	指令	軟元件	
0	LD	X10	
1	DSUMP	D100	D0
4	END		

[動作]



7.5.3 DECO、DECOP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ：解碼資料或者存儲解碼資料的軟元件編號 (BIN16 位)。

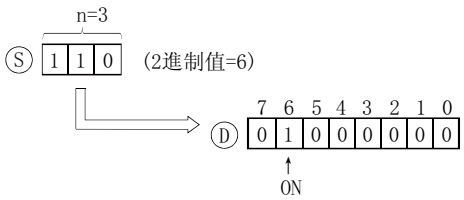
Ⓓ：存儲解碼結果的軟元件的起始編號 (軟元件名)。

n：有效位長 (1 ～ 8)，0 時無處理 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ		○			○			○	---
Ⓓ		○			---			---	---
n		○			○			○	---

功　能

(1) 將Ⓢ的低 n 位中指定的 2 進制值對應的Ⓓ的位的位置置爲 ON。



(2) n 的指定範圍爲 1 ～ 8。

(3) n=0 時執行無處理，Ⓓ中指定的軟元件的內容不發生變化。

(4) 位軟元件作爲 1 位處理，字軟元件作爲 16 位處理。

出　錯

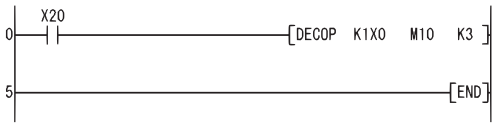
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 超出了 1 ～ 8 的範圍時。	○	○	○	○	○	○
4101	從Ⓓ開始的 2^n 位的範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，從 X0 開始進行 3 位解碼，並將結果存儲到 M10 後面的程式。

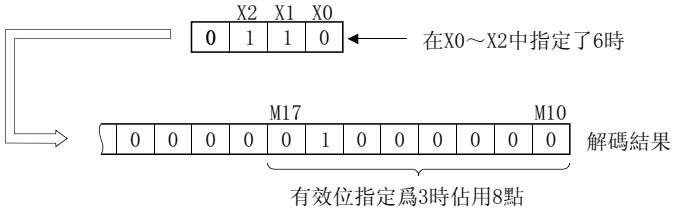
[梯形圖模式]



[列表模式]

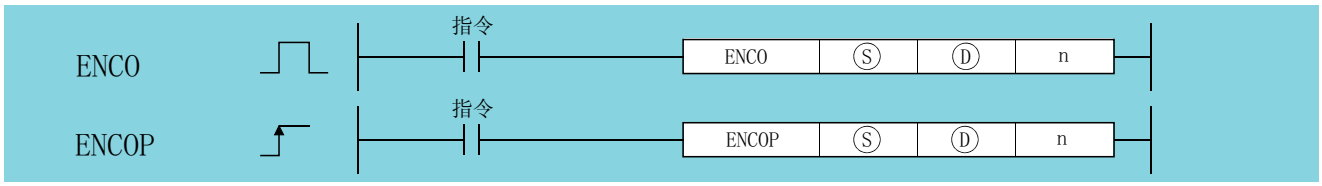
步	指令	軟元件		
0	LD	X20		
1	DECOP	K1X0	M10	K3
5	END			

[動作]



7.5.4 ENCO、ENCOP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ：存儲編碼資料的軟元件的起始編號 (軟元件名)。

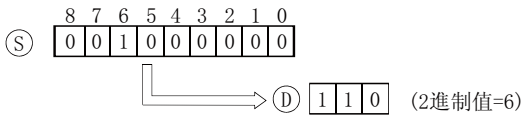
Ⓓ：存儲編碼結果的軟元件編號 (BIN16 位)。

n：有效位長 (1 ～ 8)，0 時無處理 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ		○		---				---	---
Ⓓ		○		○				---	---
n		○		○				○	---

功　能

(1) 從Ⓢ的 2^n 位元的資料開始，將處於 1 狀態的位元所對應的 2 進制值存儲到Ⓓ中。



(2) n 的指定範圍爲 1 ～ 8。

(3) n=0 時執行無處理，Ⓓ的內容不發生變化。

(4) 位軟元件作爲 1 位處理，字軟元件作爲 16 位處理。

(5) 多個位爲 1 時，按高位的位元的位置進行處理。

出　錯

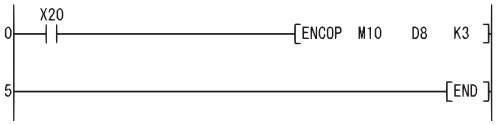
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 超出了 1 ～ 8 的範圍時。 從Ⓢ開始的 2^n 位元的資料全部爲 0 時。	○	○	○	○	○	○
4101	從Ⓢ開始的 2^n 位的範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，從 M10 開始進行 3 位元編碼，並將結果存儲到 D8 中的程式。

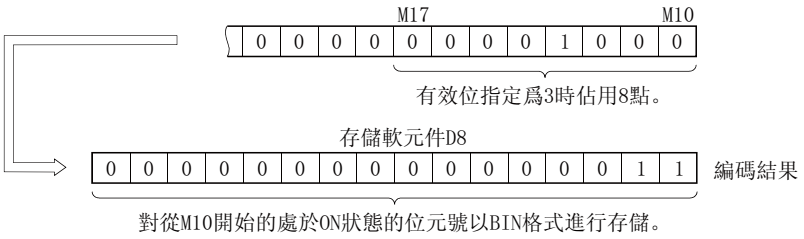
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X20		
1	ENCOP	M10	D8	K3
5	END			

[動作]



7.5.5 SEG、SEGP

Basic | High performance | Process | Redundant | Universal | LCPU

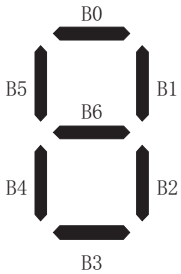
Ⓢ：解碼資料或者存儲編碼資料的軟元件的起始編號 (BIN16 位)。

Ⓓ：存儲解碼結果的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

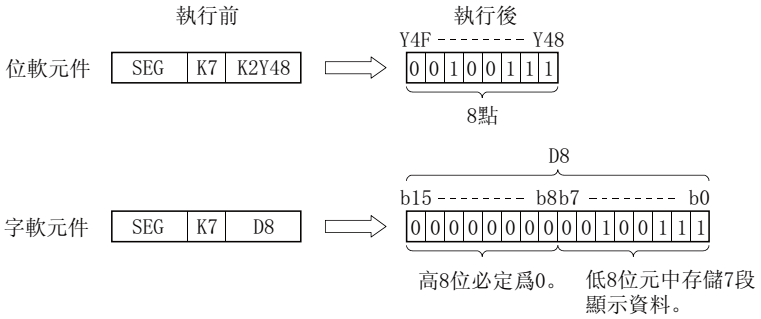
功　能

(1) 將Ⓢ的低 4 位中指定的 0 ～ F 的資料解碼爲 7 段顯示資料後，存儲到Ⓓ中。

Ⓢ		7 段的構成	Ⓓ								顯示資料
16 進制數	位元模式		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

位軟元件的起始
字軟元件的最低位

(2) 位元軟元件時，ⓓ表示存儲 7 段顯示資料的軟元件的起始編號。在字軟元件中，表示存儲的軟元件編號。



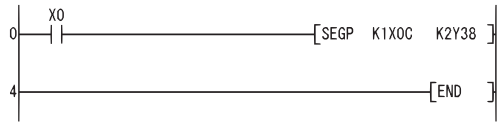
出　錯

(1) 在 SEG(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X0 變爲 ON 時，將 XC ～ XF 的資料轉換爲 7 段顯示資料後，輸出到 Y38 ～ Y3F 中的程式。

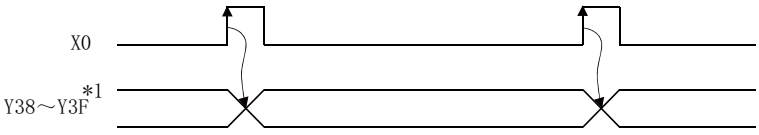
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X0	
1	SEGP	K1X0C	K2Y38
4	END		

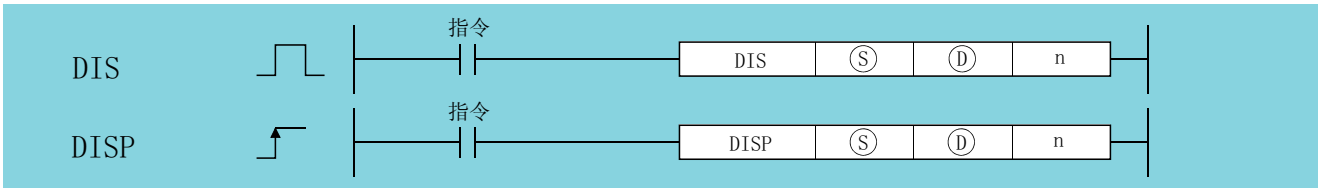
[時序圖]



*1: Y38～Y3F在下一個資料被輸出之前不發生變化。

7.5.6 DIS、DISP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ：存儲分離資料的軟元件的起始編號 (BIN16 位)。

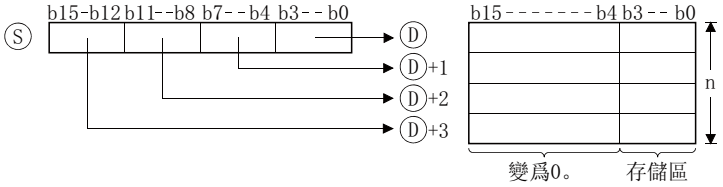
Ⓓ：存儲已被分離的資料的軟元件的起始編號 (BIN16 位)。

n：分離數 (1 ～ 4)，0 時無處理 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○		○					---
Ⓓ	---	○		---					---
n	○	○		○					---

功　能

(1) 將Ⓢ中指定的 16 位元資料的低 n 位元數 (1 位數 4 位元) 的資料，存儲到Ⓓ中指定的軟元件開始的 n 點的低 4 位中。



(2) 從Ⓢ中指定的軟元件開始的 n 點的高 12 位將變爲 0。

(3) n 的可指定範圍爲 1 ～ 4。

(4) n=0 時執行無處理，從Ⓓ的軟元件開始的 n 點的內容不發生變化。

出　錯

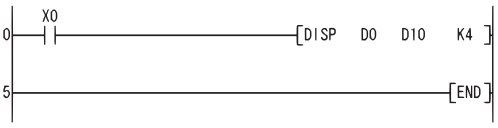
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 超出了 0 ～ 4 的範圍時。	○	○	○	○	○	○
4101	從Ⓓ開始的 n 點的範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

程式示例

(1) 在以下情況下將變爲運算出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

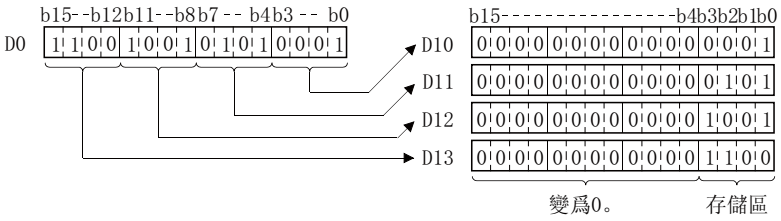
[梯形圖模式]



[列表模式]

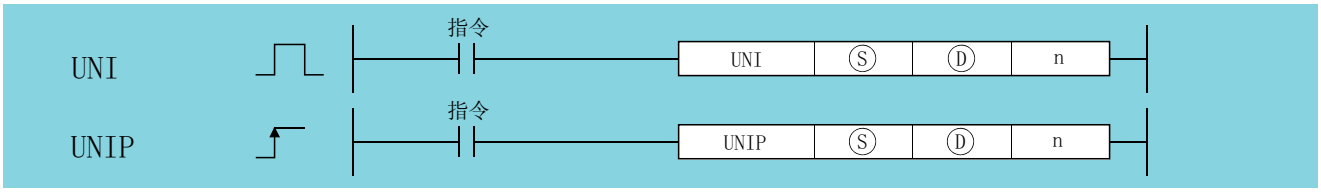
步	指令	軟元件		
0	LD	X0		
1	DISP	D0	D10	K4
5	END			

[動作]



7.5.7 UNI、UNIP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ： 存儲合併資料的軟元件的起始編號 (BIN16 位)。

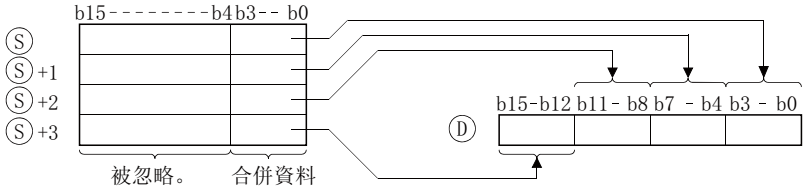
Ⓓ： 存儲被合併的資料的軟元件的起始編號 (BIN16 位)。

n： 合併數 (1 ～ 4)，0 時無處理 (BIN16 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓢ	---	○		---				---	---
Ⓓ	○	○		○				---	---
n	○	○		○				○	---

功　能

(1) 將Ⓢ中指定的軟元件開始的 n 點的 16 位元資料的低 4 位元，與Ⓓ指定的 16 位軟元件進行合併。



(2) Ⓓ中指定的軟元件的高位 (4-n) 的位數的位將變爲 0。

(3) n 的可指定範圍爲 1 ～ 4。

(4) n=0 時執行無處理，Ⓓ的軟元件的內容不發生變化。

出　錯

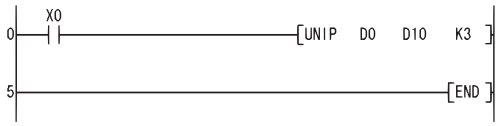
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 超出了 0 ～ 4 的範圍時。	○	○	○	○	○	○
4101	從Ⓢ開始的 n 點的範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D0 ～ D2 的低 4 位元資料進行合併，並將結果存儲到 D10 中的程式。

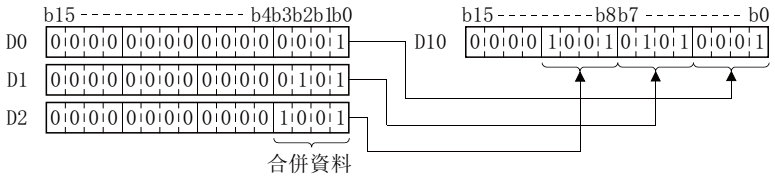
[梯形圖模式]



[列表模式]

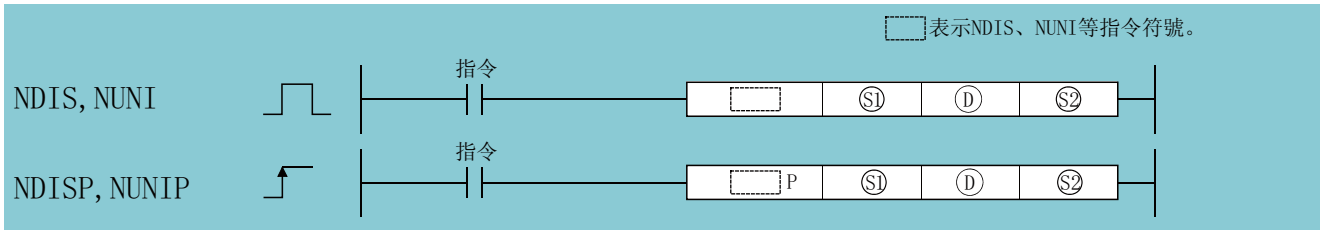
步	指令	軟元件		
0	LD	X0		
1	UNIP	D0	D10	K3
5	END			

[動作]



7.5.8 NDIS、NDISP、NUNI、NUNIP

Basic | High performance | Process | Redundant | Universal | LCPU



S1 ：存儲分離、合併資料的軟元件的起始編號 (BIN16 位)。

D ：存儲已被分離、合併的資料的軟元件的起始編號 (BIN16 位)。

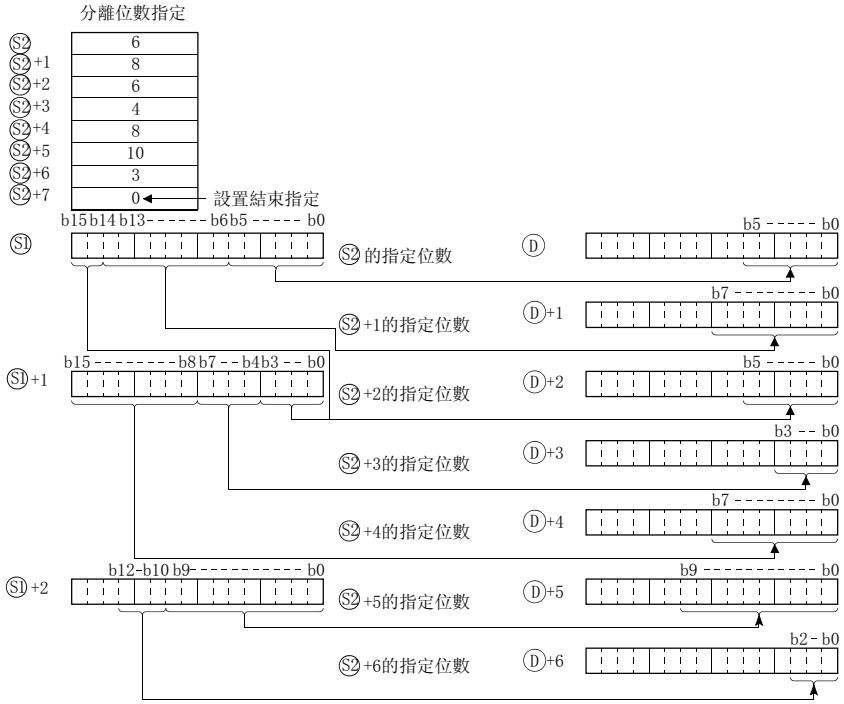
S2 ：存儲分離、合併單位的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
S1	---	○		---					
D	---	○		---					
S2	---	○		---					

功 能

NDIS

(1) 將Ⓢ1中指定的軟元件號後面存儲的資料的各個位元，分別分離成Ⓢ2中指定的位元數，並存儲到Ⓓ中指定的軟元件號後面。



(2) Ⓢ2中指定的分離位數可在 1 ～ 16 位的範圍內進行指定。

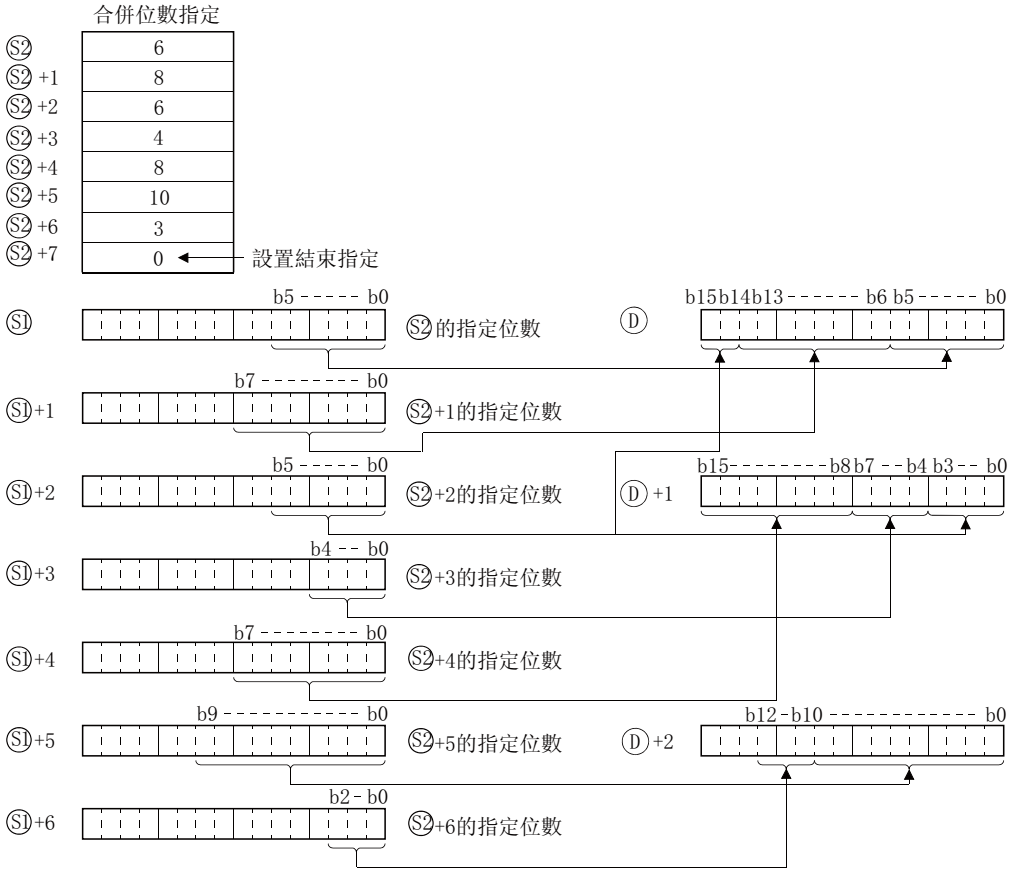
(3) 將從Ⓢ2中指定的軟元件號開始，至存儲了 “ 0 ” 的軟元件號爲止作爲分離位數進行處理。

(4) 應避免使分離資料的軟元件範圍 (Ⓢ1～Ⓢ1的結束範圍) 與存儲分離後的資料的軟元件範圍 (Ⓓ～Ⓓ的結束範圍) 重復。如果重復，將可能無法獲得正確的運算結果。

(5) 應避免使Ⓢ1、Ⓢ2、Ⓓ中指定的軟元件號重復。如果重復，將無法進行正常運算。

NUNI

(1) 將Ⓢ1中指定的軟元件號後面存儲的資料的各個位元，分別合併爲Ⓢ2中指定的位元數，並存儲到Ⓓ中指定的軟元件號後面。



(2) Ⓢ2中指定的合併位數可在 1 ～ 16 位的範圍內進行指定。

(3) 將從Ⓢ2中指定的軟元件號開始，至存儲了 “ 0 ” 的軟元件號爲止作爲合併位數進行處理。

(4) 應避免使合併資料的軟元件範圍 (Ⓢ1～Ⓢ1的結束範圍) 與存儲合併後的資料的軟元件範圍 (Ⓓ～Ⓓ的結束範圍) 重復。如果重復，將可能無法獲得正確的運算結果。

(5) 應避免使Ⓢ1、Ⓢ2、Ⓓ中指定的軟元件號重復。如果重復，將無法進行正常運算。

出　錯

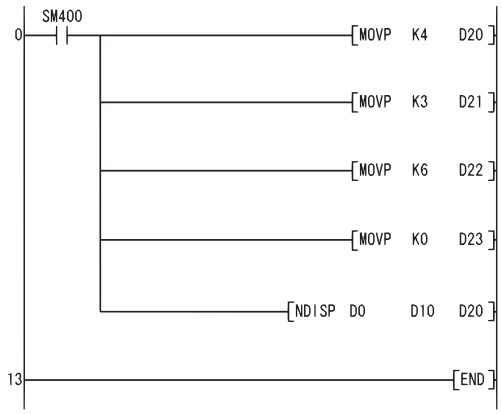
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ2中指定的分離位數指定超出了 1 ～ 16 位的範圍時。	○	○	○	○	○	○
4101	根據Ⓢ2中指定的分離位元數指定，Ⓢ1或者Ⓓ中指定的軟元件的使用範圍超出了各自的軟元件的最終軟元件號時。	○	○	○	○	○	○

程式示例

(1) 以下爲從 D0 的資料的低位元開始分別進行 4、3、6 位元分離後，存儲到 D10 ～ D12 中的程式。

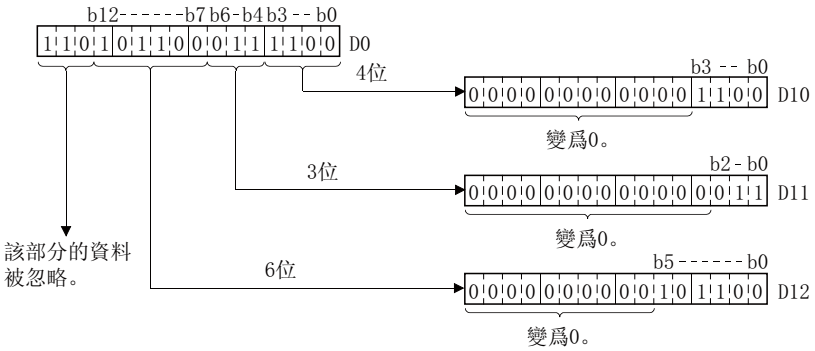
[梯形圖模式]



[列表模式]

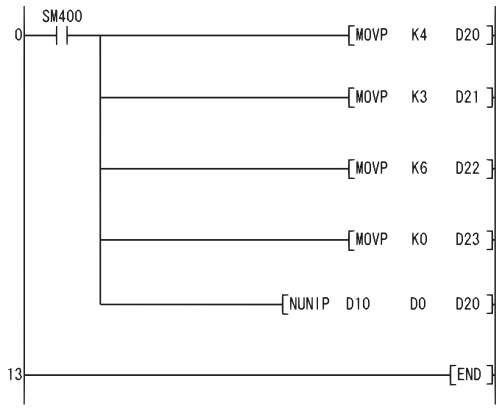
步	指令	軟元件		
0	LD	SM400		
1	MOVP	K4	D20	
3	MOVP	K3	D21	
5	MOVP	K6	D22	
7	MOVP	K0	D23	
9	NDISP	D0	D10	D20
13	END			

[動作]



(2) 以下爲將 D10 的資料的低 4 位元、D11 的資料的低 3 位元及 D12 的資料的低 6 位元進行合併後，存儲到 D0 中的程式。

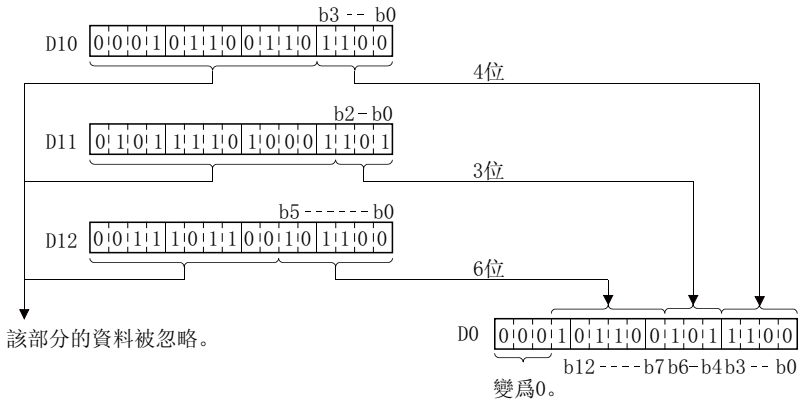
[梯形圖模式]



[列表模式]

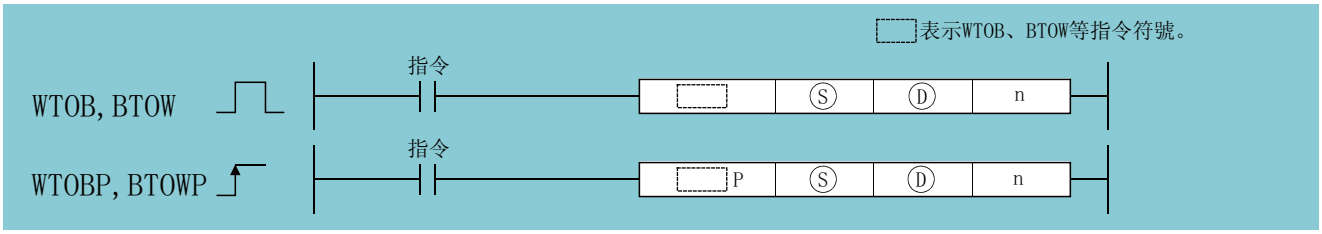
步	指令	軟元件		
0	LD	SM400		
1	MOVP	K4	D20	
3	MOVP	K3	D21	
5	MOVP	K6	D22	
7	MOVP	K0	D23	
9	NUNIP	D10	D0	D20
13	END			

[動作]



7.5.9 WTOB、WTOBP、BTOW、BTOWP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ： 存儲以位元組爲單位的分離、合併資料的軟元件的起始編號 (BIN16 位)。

Ⓓ： 存儲以位元組爲單位已被分離、合併的結果的軟元件的起始編號 (BIN16 位)。

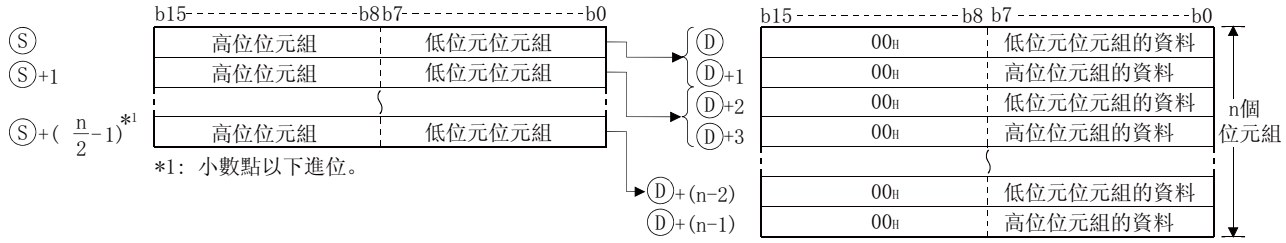
n： 分離、合併的位元組的個數 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---					---
Ⓓ	---	○		---					---
n	○	○		○					---

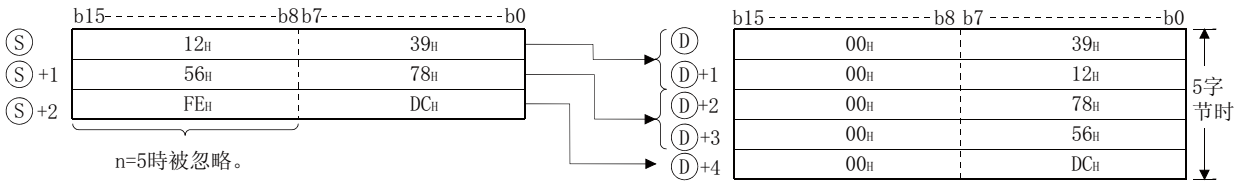
功　能

WTOB

(1) 將Ⓢ中指定的軟元件號後面存儲的 16 位元資料，分離爲 n 位元組後，存儲到Ⓓ中指定的軟元件號後面。



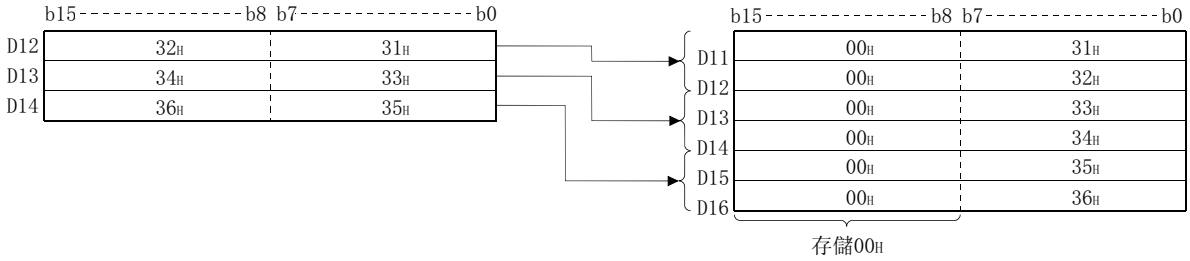
例如，n=5 時，將Ⓢ～ (Ⓢ+2) 的低 8 位元爲止的資料存儲到Ⓓ～ (Ⓓ+4) 中。



(2) 通過在 n 中設置位元組數，Ⓢ中指定的 16 位元資料的範圍以及存儲Ⓓ中指定的位元組資料的軟元件的範圍將被自動確定。

(3) n 中指定的位元組數爲 “ 0 ” 時，不執行處理。

(4) 存儲Ⓓ中指定的位元組資料的軟元件的高 8 位中，將自動地存儲 “ 00_H ”。



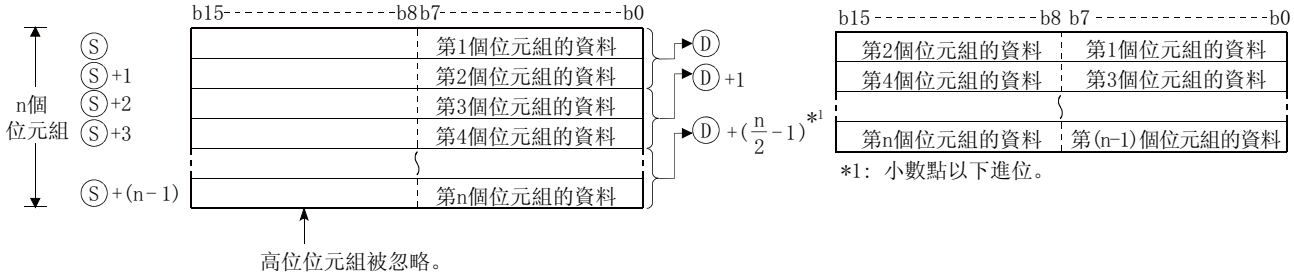
(5) 即使在存儲分離資料的軟元件範圍 (Ⓢ～Ⓢ+($\frac{n}{2}$-1)) 與存儲已分離資料的軟元件範圍 (Ⓓ～Ⓓ+(n-1)) 重復的情況下，也可正常執行處理。

BTOW

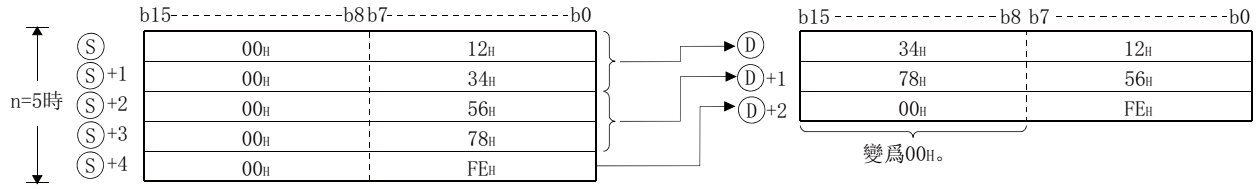
(1) 將Ⓢ中指定的軟元件號後面存儲的 n 字的 16 位元資料的低 8 位元，以字爲單位進行合併後，存儲到Ⓓ中指定的軟元件號後面。

Ⓢ中指定的軟元件號後面存儲的 n 字資料的高 8 位元將被忽略。

此外，n 爲奇數時，存儲了第 n 個位元組資料的軟元件的高 8 位元中將存儲 0。



例如，n=5 時，將Ⓢ～ (Ⓢ+4) 的低 8 位元的資料合併後，存儲到Ⓓ～ (Ⓓ+2) 中。



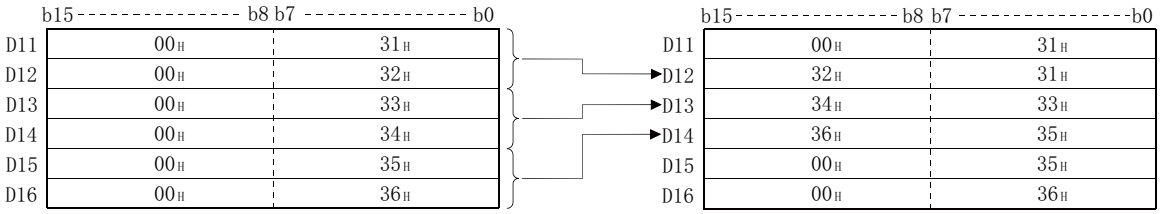
(2) 通過在 n 中設置位元組數，Ⓢ中指定的位元組資料的範圍以及存儲Ⓓ中指定的合併資料的軟元件的範圍將被自動確定。

(3) n 中指定的位元組數爲 “ 0 ” 時，不執行處理。

(4) 存儲Ⓢ中指定的位元組資料的軟元件的高 8 位將被忽略，低 8 位元將成爲物件。

(5) 即使在存儲合併資料的軟元件範圍 (Ⓢ～Ⓢ+(n-1)) 與存儲已合併資料的軟元件範圍 (Ⓓ～Ⓓ+($\frac{n}{2}$-1)) 重復的情況下，也可正常執行處理。

例如，將 D11 ～ D16 的低 8 位元存儲到 D12 ～ D14 中時的情況如下圖所示。



出　錯

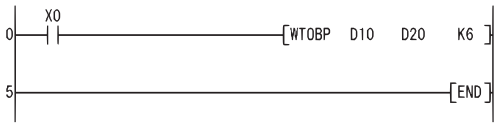
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓢ 中指定的軟元件號後面，n 中指定的位元組數的範圍超出了相應軟元件的範圍時。 Ⓓ 中指定的軟元件號後面，n 中指定的位元組數的範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲從 X0 變爲 ON 時，將 D10 ～ D12 的資料以位元組爲單位進行分離後，存儲到 D20 ～ D25 中的程式。

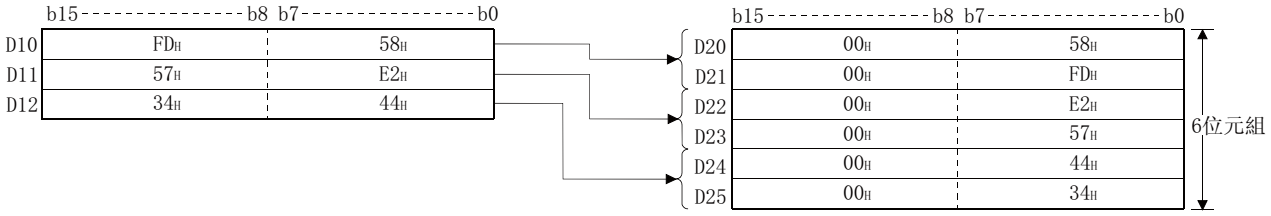
[梯形圖模式]



[列表模式]

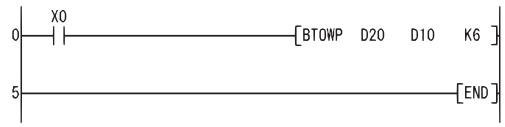
步	指令	軟元件		
0	LD	X0		
1	WTOBP	D10	D20	K6
5	END			

[動作]



(2) 以下爲從 X0 變爲 ON 時，將 D20 ～ D25 的低 8 位元資料合併後，存儲到 D10 ～ D12 中的程式。

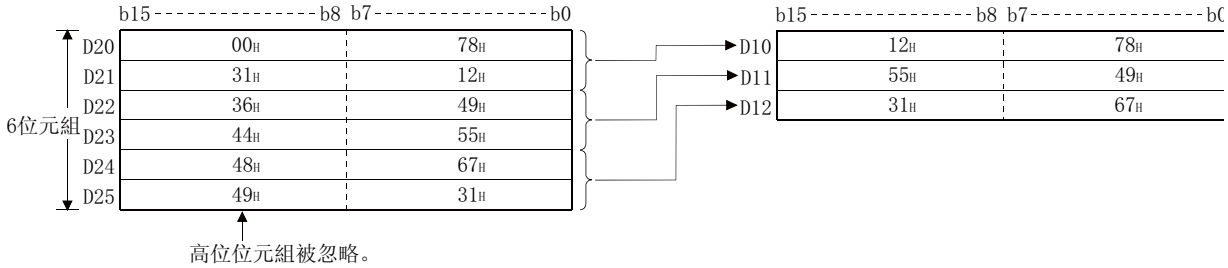
[梯形圖模式]



[列表模式]

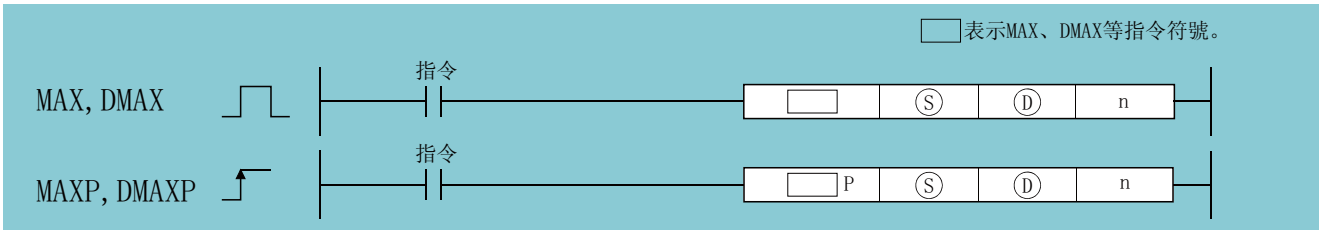
步	指令	軟元件		
0	LD	X0		
1	BTOWP	D20	D10	K6
5	END			

[動作]



7.5.10 MAX、MAXP、DMAX、DMAXP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ：搜索最大值的軟元件的起始編號 (BIN16/32 位)。

Ⓓ：存儲最大值搜索結果的軟元件的起始編號 (BIN16/32 位)。

n：搜索的資料數 (BIN16 位元)。

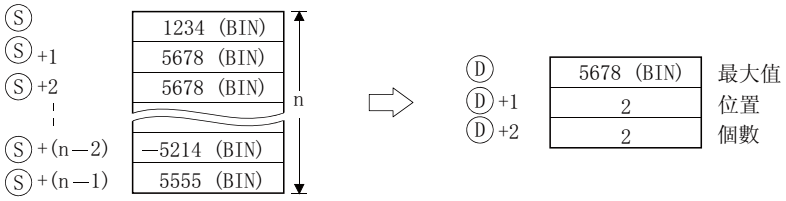
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---					---
Ⓓ	---	○		---					---
n	○	○		○					---

功　能

MAX

從Ⓢ中指定的軟元件開始，在 n 點的 16 位元 BIN 資料中搜索最大值，並將最大值存儲到Ⓓ中指定的軟元件中。

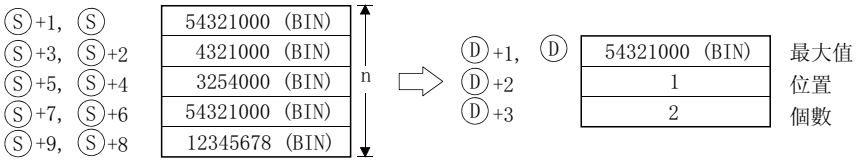
從Ⓢ中指定的軟元件開始搜索，將最先檢測出的存儲最大值的軟元件號從Ⓢ算起的點數存儲到Ⓓ+1 中，將最大值的個數存儲到Ⓓ+2 中。



DMAX

從Ⓢ中指定的軟元件開始，在 n 點的 32 位元 BIN 資料中搜索最大值，並將最大值存儲到、Ⓓ、Ⓓ+1 中指定的軟元件中。

從Ⓢ中指定的軟元件開始搜索，將最先檢測出的存儲最大值的軟元件號從Ⓢ算起的點數存儲到Ⓓ+2 中，將最大值的個數存儲到Ⓓ+3 中。



出　錯

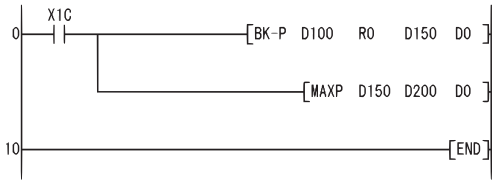
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓢ的軟元件開始的 n 點的範圍超出了相應軟元件範圍時。	○	○	○	○	○	○
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X1C 變爲 ON 時，將 D100 ～ D103 中存儲的資料值與 R0 ～ R3 中存儲的資料值進行減法運算，在其結果中搜索出最大值後，存儲到 D200 ～ D202 中的程式。

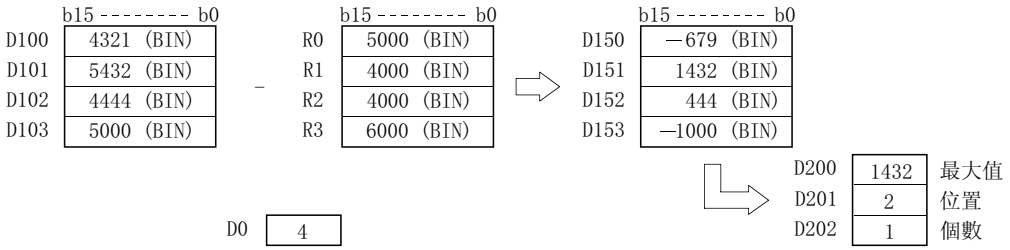
[梯形圖模式]



[列表模式]

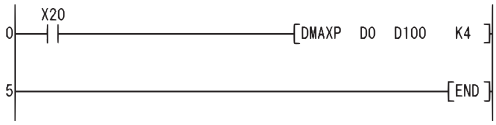
步	指令	軟元件			
0	LD	X1C			
1	BK-P	D100	R0	D150	D0
6	MAXP	D150	D200	D0	
10	END				

[動作]



(2) 以下爲 X20 變爲 ON 時，在 D0 ～ D7 的 32 位元資料中搜索最大值，並將結果存儲到 D100 ～ D103 中的程式。

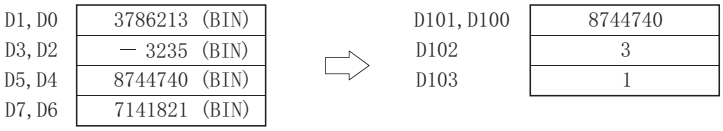
[梯形圖模式]



[列表模式]

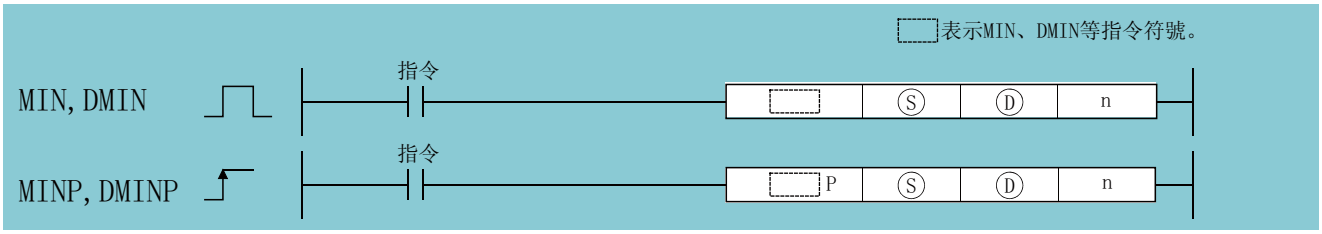
步	指令	軟元件			
0	LD	X20			
1	DMAXP	D0	D100	K4	
5	END				

[動作]



7.5.11 MIN、MINP、DMIN、DMINP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ： 搜索最小值的軟元件的起始編號 (BIN16/32 位)。

Ⓓ： 存儲最小值搜索結果的軟元件的起始編號 (BIN16/32 位)。

n： 搜索的資料數 (BIN16 位元)。

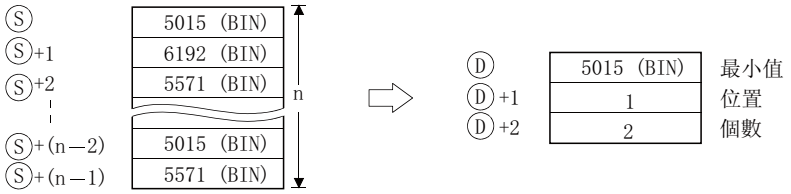
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---					---
Ⓓ	---	○		---					---
n	○	○		○					---

功　能

MIN

從Ⓢ中指定的軟元件開始，在 n 點的 16 位元 BIN 資料中搜索最小值，並將最小值存儲到Ⓓ中指定的軟元件中。

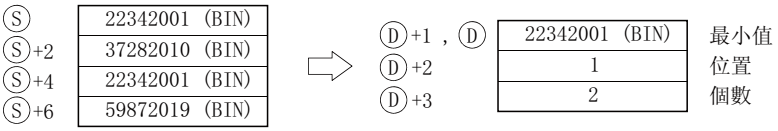
從Ⓢ中指定的軟元件開始搜索，將最先檢測出的存儲最小值的軟元件號從Ⓢ算起的點數存儲到Ⓓ+1 中，將最小值的個數存儲到Ⓓ+2 中。



DMIN

從Ⓢ中指定的軟元件開始，在 n 點的 32 位元 BIN 資料中搜索最小值，並將最小值存儲到Ⓓ、Ⓓ+1 中指定的軟元件中。

從Ⓢ中指定的軟元件開始搜索，將最先檢測出的存儲最小值的軟元件號從Ⓢ算起的點數存儲到Ⓓ+2 中，將最小值的個數存儲到Ⓓ+3 中。



出　錯

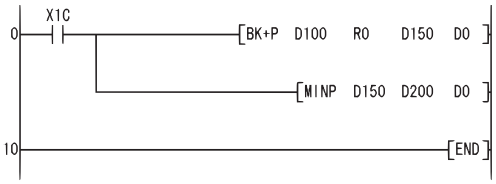
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓢ的軟元件開始的 n 點的範圍超出了相應軟元件範圍時。	---	---	---	---	○	○
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X1C 變爲 ON 時，將 D100 ～ D103 中存儲的資料值與 R0 ～ R3 中存儲的資料值進行加法運算，在其結果中搜索出最小值後，存儲到 D200 ～ D202 中的程式。

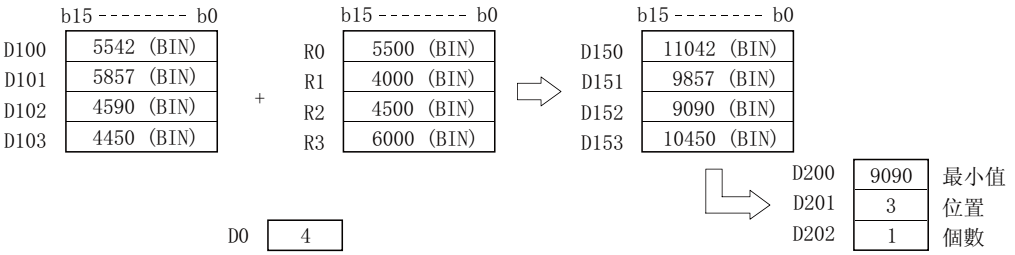
[梯形圖模式]



[列表模式]

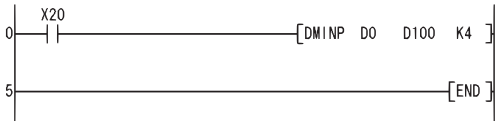
步	指令	軟元件			
0	LD	X1C			
1	BK+P	D100	R0	D150	D0
6	MINP	D150	D200	D0	
10	END				

[動作]



(2) 以下爲 X20 變爲 ON 時，在 D0 ～ D7 的 32 位元資料中搜索最小值，並將結果存儲到 D100 ～ D103 中的程式。

[梯形圖模式]



[列表模式]

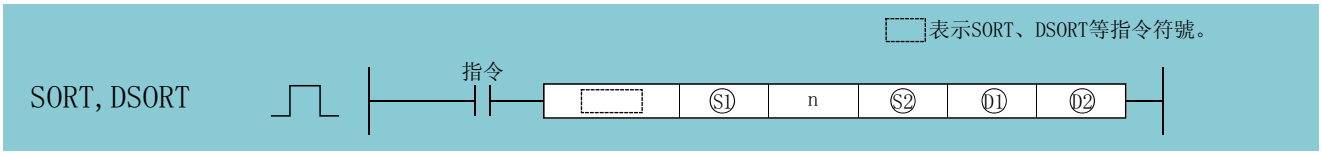
步	指令	軟元件		
0	LD	X20		
1	DMINP	D0	D100	K4
5	END			

[動作]

D1, D0	57020175 (BIN)	D101, D100	−69386
D3, D2	2070166 (BIN)	D102	4
D5, D4	3596045 (BIN)	D103	1
D7, D6	−69386 (BIN)		

7.5.12 SORT、DSORT

Basic | High performance | Process | Redundant | Universal | LCPU



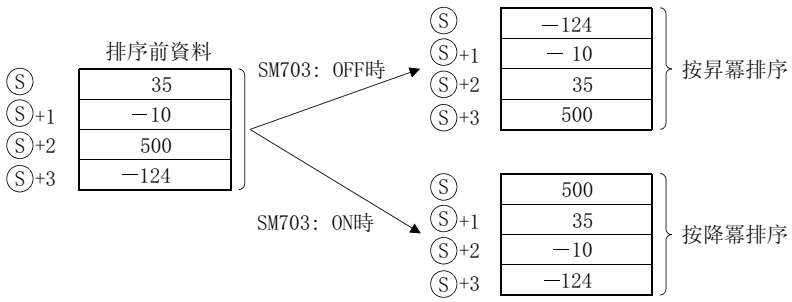
Ⓢ1： 排序的表格的起始軟元件編號 (BIN16/32 位)。
n： 排序的資料數 (BIN16 位元)。
Ⓢ2： 1 次執行中進行比較的資料數 (BIN16 位元)。
Ⓓ1： 排序結束時置爲 ON 的軟元件編號 (位)。
Ⓓ2： 系統用軟元件 (BIN16 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓢ1	---	○		---					---
n	○	○		○					---
Ⓢ2	○	○		○					---
Ⓓ1	○	---		---					---
Ⓓ2	---	○		---					---

功　能

SORT

(1) 將從Ⓢ1開始的 n 點的 BIN16 位元資料進行昇冪 / 降冪排序 (排列替換)。
排序指定是通過 SM703 的 ON/OFF 進行的。
・SM703 爲 OFF 時： 昇冪排序
・SM703 爲 OFF 時： 降冪排序



(2) 通過 SORT 指令進行排序時，需要數次掃描。

至執行結束爲止的掃描次數爲，將至排序執行結束爲止的最多執行次數，與Ⓢ2中指定的 1 次執行中進行比較的資料數相除後的值。(小數點以下進位。)

如果Ⓢ2的值較大，則至排序結束爲止的掃描次數將變少，但掃描時間將延長。

(3) 至排序執行結束爲止的最多執行次數應通過下述公式算出：
至排序執行結束爲止的最多執行次數 =(n) × (n-1) ÷ 2(次)

例 n=10 時，需要 10 × (10-1) ÷ 2=45(次)。

此時，如果設置爲Ⓢ2=2，則至排序結束爲止的掃描次數爲 45 ÷ 2=22.5 → 23(掃描)。

(4) Ⓓ1中指定的軟元件 (結束軟元件) 在 SORT 指令執行開始時變爲 OFF，在排序結束時變爲 ON。

排序結束後，Ⓓ1中指定的軟元件將保持爲 ON 狀態不變，因此應由用戶根據需要將其置爲 OFF。

(5) 從Ⓓ2中指定的軟元件開始的 2 點爲 SORT 指令執行時的系統所用。

用戶不要對從Ⓓ2中指定的軟元件開始的 2 點進行變更。
如果進行了變更，有可能導致出錯。(出錯代碼： 4100)

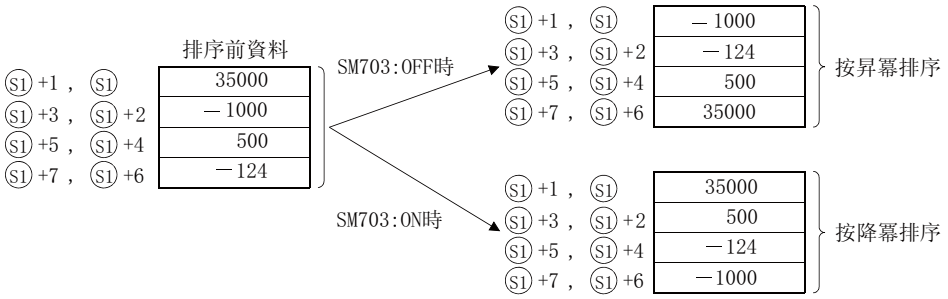
(6) 如果在排序執行過程中對 n 進行了變更，將按變更後的排序資料數進行排序。

(7) 如果在排序執行過程中將其執行指令置爲 OFF，則排序將被中斷。
再次將其執行指令置爲 ON 時，將從頭開始重新進行排序。

(8) 排序執行結束後，連續進行下次排序時，需要將執行指令置爲一次 OFF 後，再次將執行指令置爲 ON。

DSORT

(1) 將從Ⓢ1開始的 n 點的 BIN32 位元資料進行昇冪 / 降冪排序 (排列替換)。
排序指定是通過 SM703 的 ON/OFF 進行的。
· SM703 爲 OFF 時： 昇冪排序
· SM703 爲 ON 時： 降冪排序



(2) 通過 DSORT 指令進行排序時，需要數次掃描。

至執行結束爲止的掃描次數爲，將至排序執行結束爲止的最多執行次數，與Ⓢ2中指定的 1 次執行中進行比較的資料數相除後的值。(小數點以下進位。)

如果Ⓢ2的值較大，則至排序結束爲止的掃描次數將變少，但掃描時間將延長。

(3) 至排序執行結束爲止的最多執行次數應通過下述公式算出：
至排序執行結束爲止的最多執行次數 =(n) × (n-1) ÷ 2(次)

例 n=10 時，需要 10 × (10-1) ÷ 2=45(次)。

此時，如果設置爲Ⓢ2=2，則至排序結束爲止的掃描次數爲 45 ÷ 2=22.5 → 23(掃描)。

(4) Ⓓ1中指定的軟元件 (結束軟元件) 在 DSORT 指令執行開始時變爲 OFF，在排序結束時變爲 ON。

排序結束後，Ⓓ1中指定的軟元件將保持爲 ON 狀態不變，因此應由用戶根據需要將其置爲 OFF。

(5) 從Ⓓ2中指定的軟元件開始的 2 點爲 DSORT 指令執行時的系統所用。

用戶不要對從Ⓓ2中指定的軟元件開始的 2 點進行變更。
如果進行了變更，有可能導致出錯。(出錯代碼： 4100)

(6) 如果在排序執行過程中對 n 進行了變更，將按變更後的排序資料數進行排序。

(7) 如果在排序執行過程中將其執行指令置爲 OFF，則排序將被中斷。
再次將其執行指令置爲 ON 時，將從頭開始重新進行排序。

(8) 排序執行結束後，連續進行下次排序時，需要將執行指令置爲一次 OFF 後，再次將執行指令置爲 ON。

出　錯

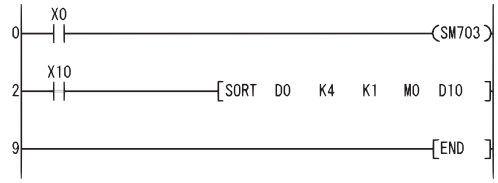
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	ⓢ2爲 0 或負數時。	○	○	○	○	○	○
4101	從ⓢ1開始的 n 點÷ 2 × n 點的軟元件範圍與從ⓓ2開始的 2 點的軟元件範圍重復時。	○	○	○	○	○	○
4101	SORT(P) 指令中從ⓢ1開始的 n 點的範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○
4101	DSORT(P) 指令中從ⓢ1開始的 2 × n 點的範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X10 變爲 ON 時，對從 D0 至 D3 的 BIN16 位元資料進行昇冪 / 降冪排序的程式。

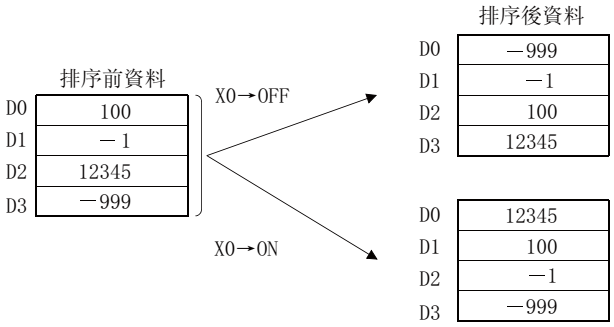
[梯形圖模式]



[列表模式]

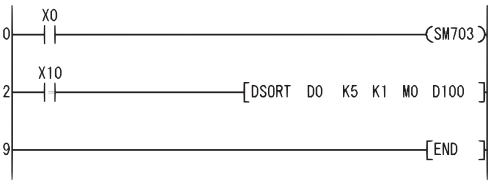
步	指令	軟元件
0	LD	X0
1	OUT	SM703
2	LD	X10
3	SORT	D0 K4 K1 M0 D10
9	END	

[動作]



(2) 以下爲 X10 變爲 ON 時，對從 D0 至 D9 的 BIN32 位元資料進行昇冪 / 降冪排序的程式。

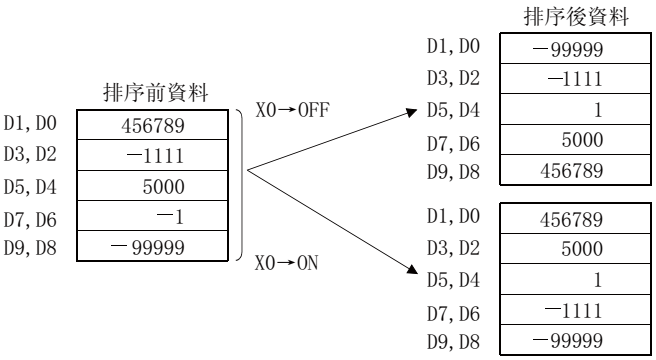
[梯形圖模式]



[列表模式]

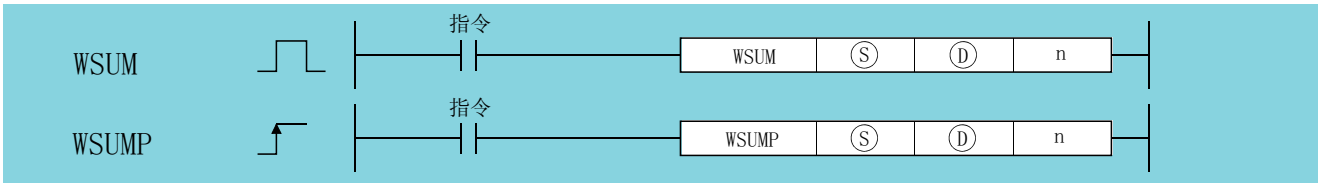
步	指令	軟元件
0	LD	X0
1	OUT	SM703
2	LD	X10
3	DSORT	D0 K5 K1 M0 D100
9	END	

[動作]



7.5.13 WSUM、WSUMP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ：存儲進行合計值計算資料的軟元件的起始編號 (BIN16 位)。

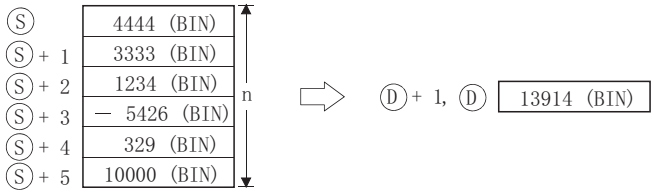
Ⓓ：存儲合計值的軟元件的起始編號 (BIN32 位)。

n：資料個數 (BIN16 位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---				---	---
Ⓓ	○	○		○				---	---
n	○	○		○				○	---

功　能

將Ⓢ中指定的軟元件開始的 n 點的 16 位元 BIN 資料全部進行加法運算後，將結果存儲到Ⓓ中指定的軟元件中。



出　錯

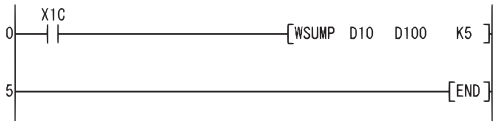
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓢ的軟元件開始的 n 點的範圍超出了相應軟元件時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X1C 變爲 ON 時，對 D0 ～ D14 的 16 位元 BIN 資料進行加法運算後，將結果存儲到 D100、D101 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X1C		
1	WSUMP	D10	D100	K5
5	END			

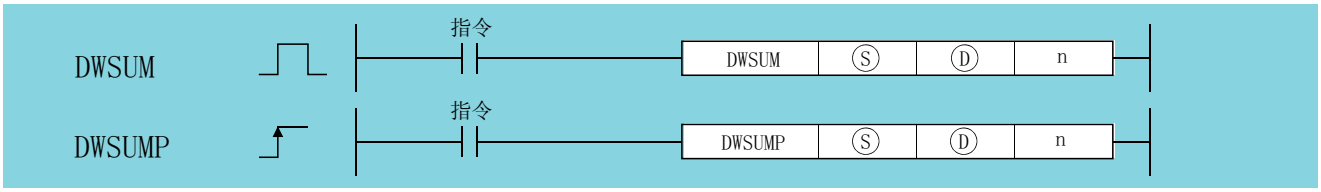
[動作]

D10	4500 (BIN)
D11	2500 (BIN)
D12	— 3276 (BIN)
D13	6780 (BIN)
D14	4444 (BIN)

⇨ D101, D100 | 14948 (BIN)

7.5.14 DWSUM、DWSUMP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ：存儲進行合計值計算資料的軟元件的起始編號 (BIN32 位)。

Ⓓ：存儲合計值的軟元件的起始編號 (BIN64 位)。

n：資料個數 (BIN16 位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---				---	---
Ⓓ	○	○		---				---	---
n	○	○		○				○	---

功　能

將Ⓢ中指定的軟元件開始的 n 點的 32 位元 BIN 資料全部進行加法運算後，將結果存儲到Ⓓ中指定的軟元件開始的 4 點 (4 字) 中。

Ⓢ+ 1, Ⓢ	32767000 (BIN)
Ⓢ+ 3, Ⓢ+ 2	6000 (BIN)
Ⓢ+ 5, Ⓢ+ 4	35392000 (BIN)
Ⓢ+ 7, Ⓢ+ 6	—11870000 (BIN)
Ⓢ+ 9, Ⓢ+ 8	12345000 (BIN)

n ⇨ Ⓓ+ 3 ～ Ⓓ | 68640000 (BIN)

出　錯

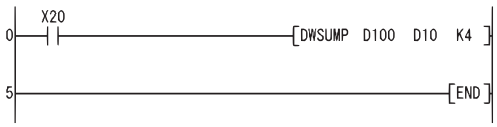
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓢ的軟元件開始的 n 點的範圍超出了相應軟元件時。	○	○	○	○	○	○
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，對 D100 ～ D107 的 32 位元 BIN 資料進行加法運算後，將結果存儲到 D10、D13 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X20		
1	DWSUMP	D100	D10	K4
5	END			

[動作]

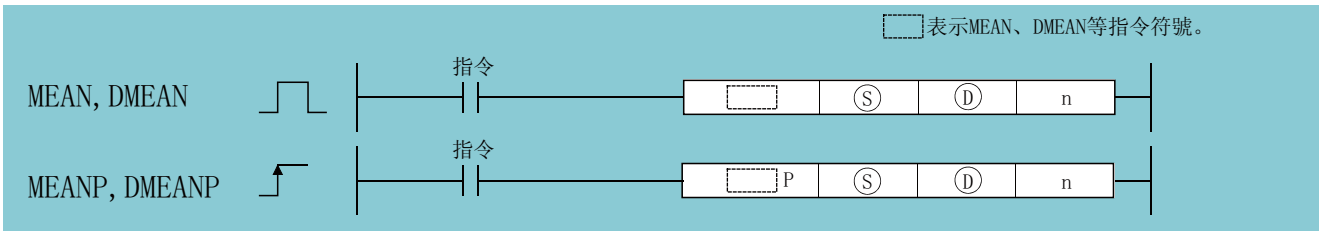
D101, D100	11245600 (BIN)
D103, D102	27543200 (BIN)
D105, D104	558800 (BIN)
D107, D106	— 15675000 (BIN)

⇨ D13 ～D10 | 23672600 (BIN)

7.5.15 MEAN、MEANP、DMEAN、DMEANP

Basic ✕ | High performance ✕ | Process ✕ | Redundant ✕ | Ver. Universal | LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。



Ⓢ： 存儲進行平均值計算資料的軟元件的起始編號 (BIN16/32 位)。

Ⓓ： 存儲平均值的軟元件的起始編號 (BIN16/32 位)。

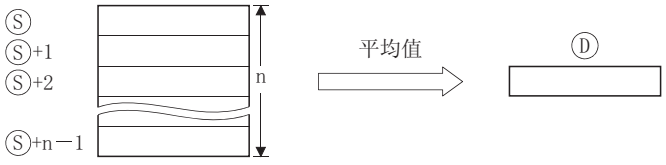
n： 資料個數或者存儲資料個數的軟元件編號 (設置範圍爲 1 ～ 32767) (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○	○	---				---	---
Ⓓ	---	○	○	---				---	---
n	---	○	○	○				○	---

功　能

MEAN(P)

(1) 將Ⓢ中指定的軟元件開始的 n 點 (16 位元 BIN 資料) 的平均值進行計算後，將結果存儲到Ⓓ中指定的軟元件中。

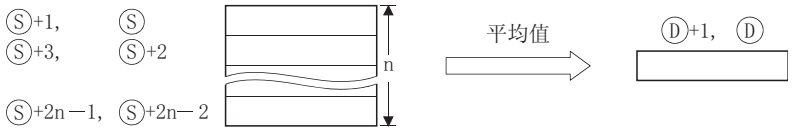


(2) 在計算結果不是整數值的情況下，小數點以下將進位。

(3) n 中指定的值爲 0 時執行無處理。

DMEAN(P)

(1) 將Ⓢ中指定的軟元件開始的 n 點 (32 位元 BIN 資料) 的平均值進行計算後，將結果存儲到Ⓓ中指定的軟元件中。



(2) 在計算結果不是整數值的情況下，小數點以下將進位。

(3) n 中指定的值爲 0 時執行無處理。

出　錯

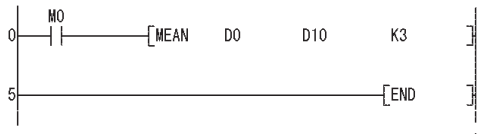
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 中指定的值超出了 0 ～ 32767 的範圍時。	---	---	---	---	○	○
4101	從Ⓢ中指定的軟元件開始的 n 點的範圍超出了指定軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 M0 變爲 ON 時，將 D0 ～ D2 的 16 位元資料的平均值存儲到 D10 中的程式。

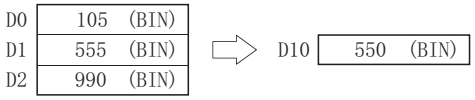
[梯形圖模式]



[列表模式]

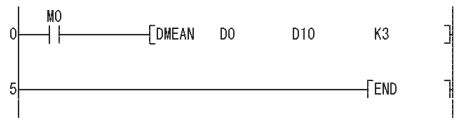
步	指令	軟元件		
0	LD	M0		
1	MEAN	D0	D10	K3
5	END			

[動作]



(2) 以下爲 M0 變爲 ON 時，將 D0 ～ D5 的 32 位元資料的平均值存儲到 D10、D11 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	M0		
1	DMEAN	D0	D10	K3
5	END			

[動作]

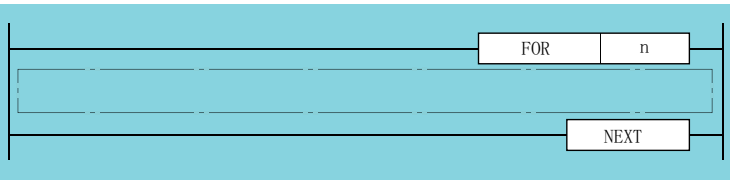
7.6 結構化指令

7.6.1 FOR、NEXT

Basic | High performance | Process | Redundant | Universal | LCPU

FOR

NEXT



n ： FOR ～ NEXT 之間的反復次數。(1 ～ 32767)(BIN16 位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n	○								---

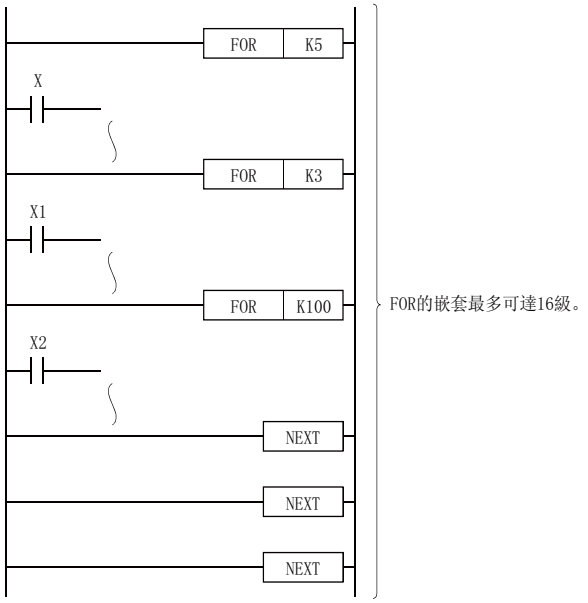
功 能

(1) 將 FOR ～ NEXT 指令之間的處理無條件地執行了 n 次時，執行 NNEXT 指令的下一步的處理。

(2) n 的可指定範圍爲 1 ～ 32767。如果指定了 -32768 ～ 0 的值，將執行等同于 n=1 時的處理。

(3) 不希望執行 FOR ～ NEXT 指令之間的處理時，應通過 CJ、SCJ 指令進行跳轉。

(4) FOR 的嵌套最多可達 16 級。



出 錯

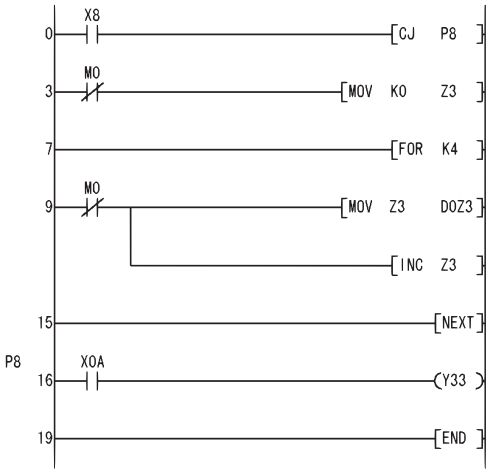
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4200	在執行 FOR 指令後，執行 NEXT 指令前執行了 END、FEND、GOEND 指令時。 在 FOR ～ NEXT 指令之間存在有 STOP 指令時。	○	○	○	○	○	○
4201	在執行 FOR 指令前執行了 NEXT 指令時。	○	○	○	○	○	○
4202	在執行 FOR 指令的嵌套的情況下，執行了第 17 級以上時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X8 變爲 OFF 時，執行 FOR ～ NEXT 指令，X8 變爲 ON 時，不執行 FOR ～ NEXT 指令的程式。

[梯形圖模式]

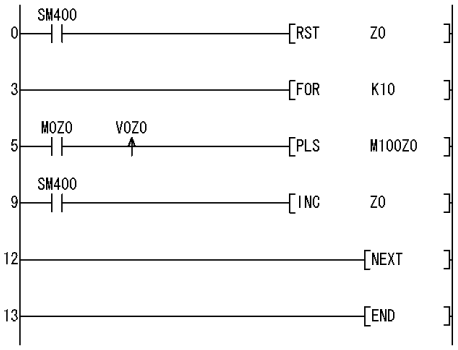


[列表模式]

步	指令	軟元件	
0	LD	X8	
1	CJ	P8	
3	LDI	M0	
4	MOV	K0	Z3
7	FOR	K4	
9	LDI	M0	
10	MOV	Z3	D0Z3
13	INC	Z3	
15	NEXT		
16	P8		
17	LD	X0A	
18	OUT	Y33	
19	END		

備 註

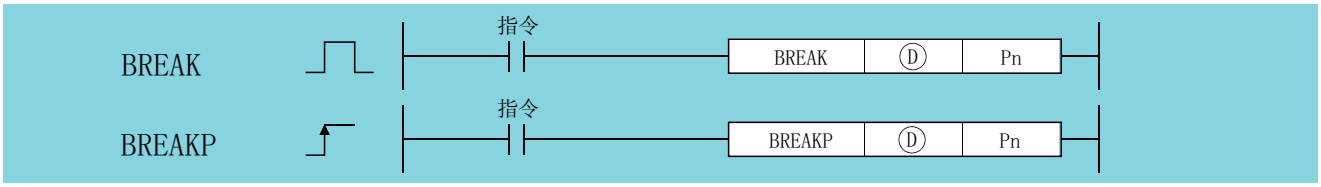
1. 在 FOR ～ NEXT 的反復執行過程中希望結束執行的情況下，應使用 BREAK 指令。
   關於 BREAK 指令的詳細內容，請參閱 372 頁 7.6.2 項。
2. 在 FOR ～ NEXT 之間執行變址修飾程式的脈衝運算時，應使用 EGP/EGF 指令。
   但是，動作輸出側不能使用上升沿指令、下降沿指令。
   關於 EGP/EGF 指令的詳細內容，請參閱 132 頁 5.2.5 項。其樣本程式如下所示：



3. 不能從 FOR ～ NEXT 的外面對 FOR ～ NEXT 指令內通過 JMP 指令等進行分支。

7.6.2 BREAK、BREAKP

Basic | High performance | Process | Redundant | Universal | LCPU

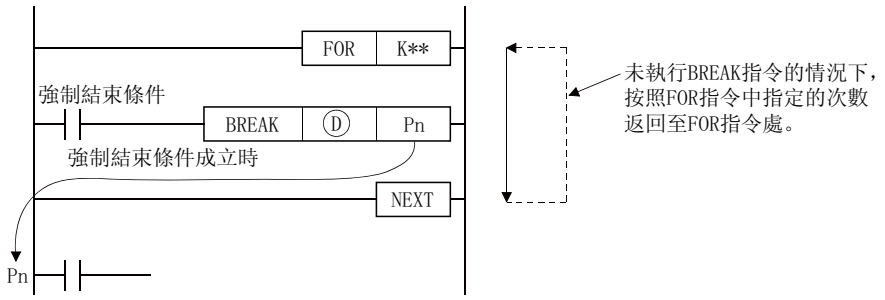


Ⓓ ： 存儲剩餘反復次數的軟元件編號 (BIN16 位)。
Pn ： 反復處理強制結束時的分支目標指標號 (軟元件名 (指標))。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數	其他 P
Ⓓ	○							---	---
Pn	---							---	○

功　能

(1) 對通過 FOR ～ NEXT 指令進行的反復處理執行強制結束，將執行切換到 Pn 中指定的指標處。
在 Pn 中只能指定同一個程式文件內的指標。
如果在 Pn 中指定了其他程式文件內的指標，將發生運算出錯。



(2) 強制結束時，將 FOR ～ NEXT 指令中剩餘的反復處理執行次數存儲到Ⓓ中。
但是，執行 BREAK 指令時的次數也包括在剩餘的反復處理次數中。

(3) BREAK 指令只能在 FOR ～ NEXT 指令之間才可使用。

(4) BREAK 指令只能用於 1 個嵌套。
對多重嵌套執行強制結束時，則應執行與嵌套級數相同數量的 BREAK 指令。

出　錯

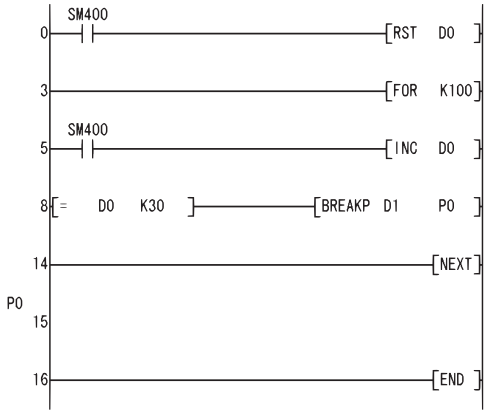
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4203	用於除 FOR ～ NEXT 指令以外時。	○	○	○	○	○	○
4210	Pn 中指定的指標的跳轉目標不存在時。 Pn 中指定了其他程式文件的指標時。	○	○	○	○	○	○

程式示例

(1) 以下爲 D0 變爲 30 時 (執行了 30 次 FOR ～ NEXT 時)，強制結束 FOR ～ NEXT 的程式。

[梯形圖模式]



[列表模式]

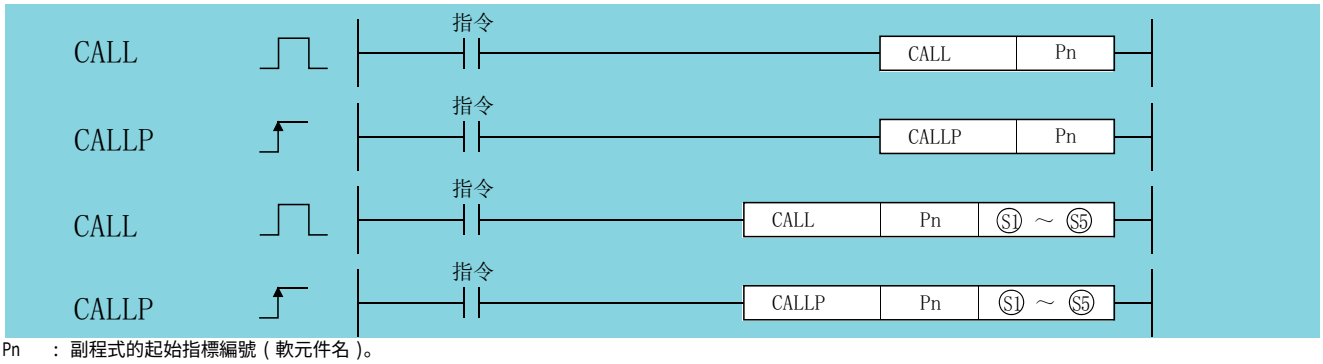
步	指令	軟元件	
0	LD	SM400	
1	RST	D0	
3	FOR	K100	
5	LD	SM400	
6	INC	D0	
8	LD=	D0	K30
11	BREAKP	D1	P0
14	NEXT		
15	P0		
16	END		

備 註

執行 BREAK 指令時，D1 中將存儲 71。

7.6.3 CALL、CALLP

Basic | High performance | Process | Redundant | Universal | LCPU



Pn ：副程式的起始指標編號（軟元件名）。

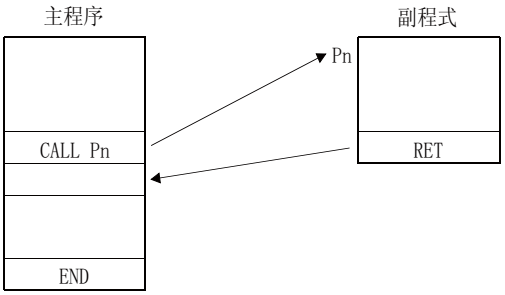
① ～ ⑤：作爲變數傳送到副程式中的軟元件編號（位、BIN16 位、BIN32 位）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他 P
	位	字		位	字				
Pn	---	---		---					○
①～⑤	○（除 F 以外）	○		○					---

功　能

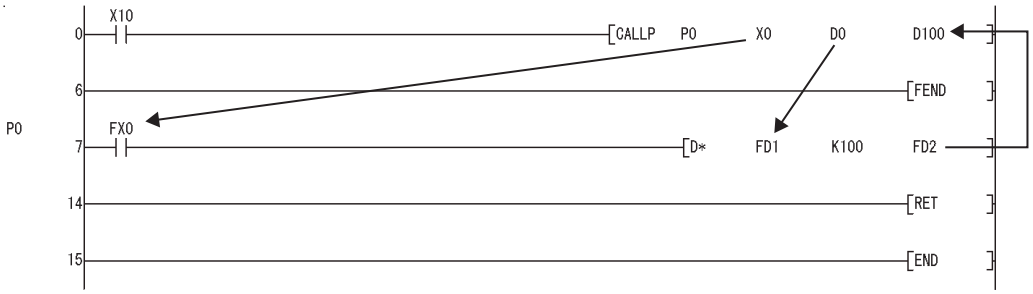
(1) 如果執行 CALL(P) 指令，將執行 Pn 中指定的指標的副程式。

CALL(P) 指令可以執行由同一個程式文件中的指標指定的副程式及由公共指標指定的副程式。



(2) 在副程式中使用功能軟元件 (FX、FY、FD) 時，在①～⑤中指定與功能軟元件對應的軟元件。

①～⑤中指定的軟元件的內容如下所示。



(a) 在副程式執行之前，將位元資料的內容傳送到 FX 中，將字資料的內容傳送到 FD 中。

(b) 在副程式執行之後，將 FY、FD 的內容傳送到對應的軟元件中。

(c) 功能軟元件的處理單位如下所示。

· FX,FY: 位元單位

· FD : 4 字單位

根據變數中指定的軟元件的種類，所使用的資料大小有所不同。

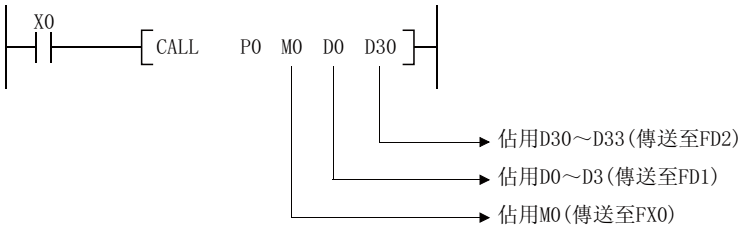
功能軟元件中指定的軟元件應預留出相當於該資料大小的容量。

未能預留出相當於該資料大小的容量時將會變爲出錯狀態。

功能軟元件	使用軟元件	資料大小	備註
· FX · FY	位軟元件	1 點	---
	字軟元件的位指定時	1 位	
· FD	位軟元件的位數指定時 *1	4 字	資料大小根據所使用的指令而有所不同。
	字軟元件	4 字	

*1: 位軟元件的位數指定時，即使在Ⓢ1～Ⓢ5中指定的軟元件號不是 16 的倍數時，也不會變爲出錯狀態。

[主程序]



(3) 在 CALL(P) 指令中，可使用範圍爲Ⓢ1～Ⓢ5。

(4) 副程式中使用的功能軟元件數必須與 CALL(P) 指令中的變數數相同。
此外，功能軟元件與 CALL(P) 的變數類型應該完全一致。

(5) CALL(P) 指令的變數中指定的軟元件號不應重復。
如果重復，有可能導致無法正常運算。

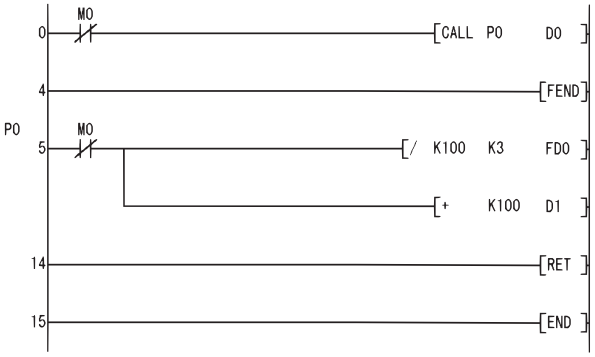
(6) CALL(P) 指令的變數中使用的軟元件不應在副程式中使用。
如果在副程式中使用了 CALL(P) 指令的變數中使用的軟元件，將無法正常進行運算。(參閱以下程式示例。)

(7) CALL(P) 指令的變數的軟元件中使用了計時器 / 計數器時，只進行當前值的接收發送。

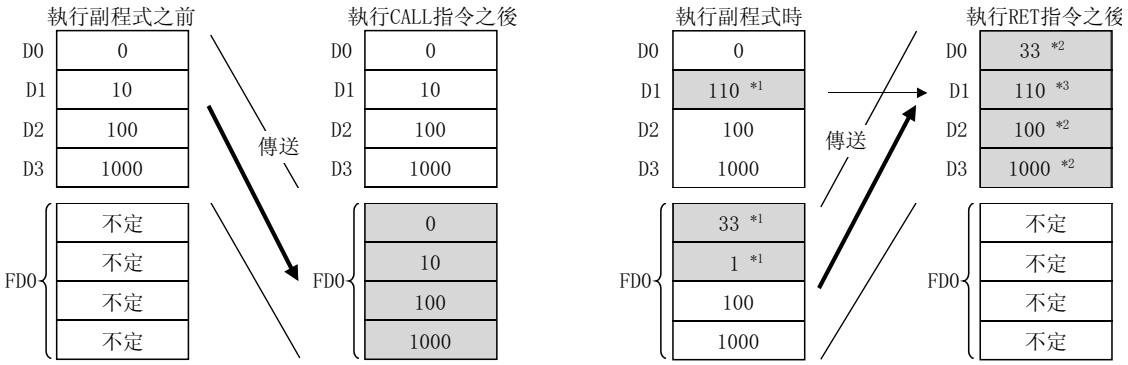
不正常的運算示例

將 D0 指定到副程式的 FD0 中，且在副程式中使用了 D1 時的動作如下所示。

[程式示例]



[執行了副程式後的動作]

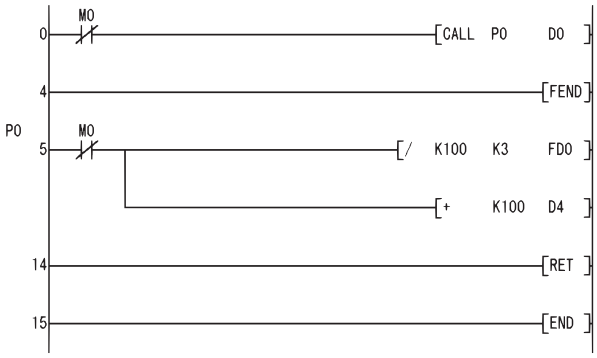


*1: 存儲副程式的執行結果。
*2: 替換爲功能軟元件的值。
*3: D1 中不能反映功能軟元件的值。

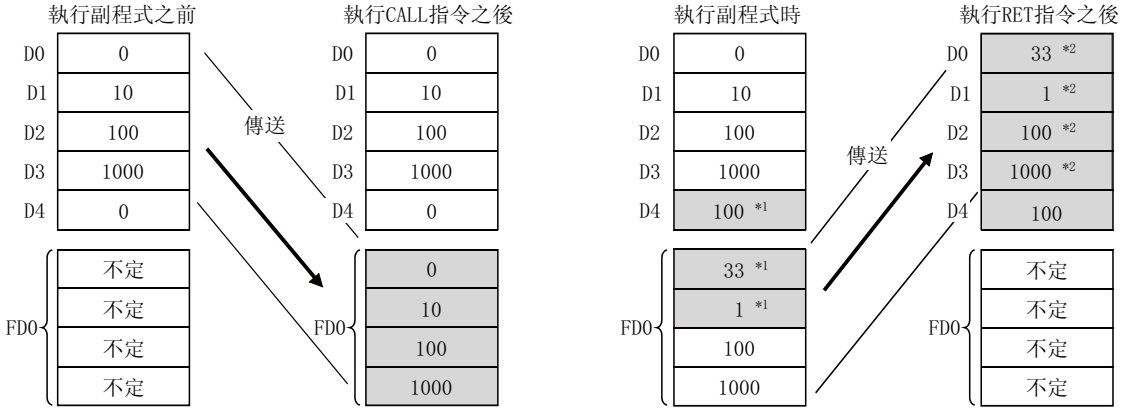
正常的運算示例

將 D0 指定到副程式的 FD0 中，且在副程式中使用了 D4 時的動作如下所示。

[程式示例]

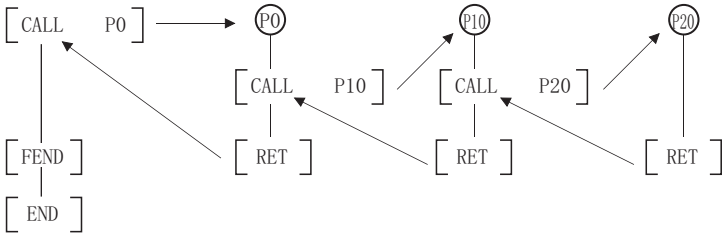


[執行了副程式後的動作]



*1: 存儲副程式的執行結果。
*2: 替換爲功能軟元件的值。

(8) CALL(P) 指令最多可以有 16 級嵌套。
然而，該 16 級是 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P)、XCALL 指令中嵌套的總級數。



(9) 對於在副程式內使其變爲 ON 的軟元件，即使副程式停止後其 ON 狀態也仍將被保持。
對於在執行副程式時變爲 ON 的軟元件，可以通過 FCALL(P) 指令使其變爲 OFF。

出　錯

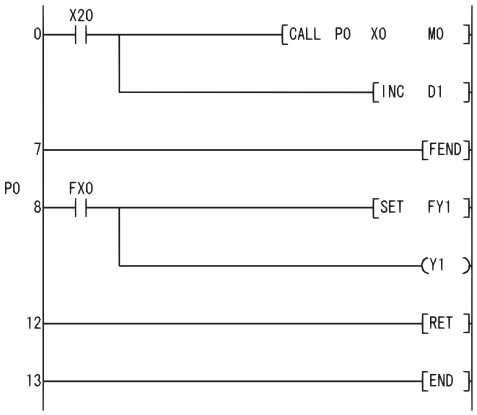
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	引數中指定的軟元件未能預留出相當於資料大小的容量時。	○	○	○	○	○	○
4210	CALL(P) 指令中指定的指標在副程式中不存在時。	○	○	○	○	○	○
4211	在執行 CALL(P) 指令後，執行 RET 指令前執行了 END、FEND、GOEND、STOP 指令時。	○	○	○	○	○	○
4212	在執行 CALL(P) 指令前執行了 RET 指令時。	○	○	○	○	○	○
4213	執行了第 17 級嵌套時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，執行帶變數副程式的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
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 RET

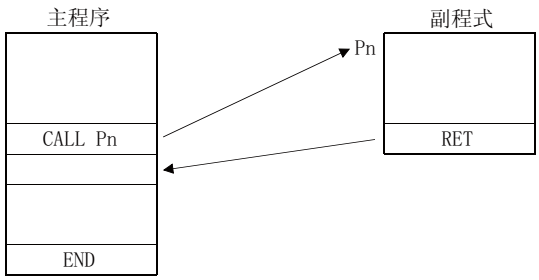
Basic　High performance　Process　Redundant　Universal　LCPU

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功　能

(1)　表示副程式的結束。

(2)　執行了 RET 指令時，調用了副程式的 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P)、XCALL 指令將返回至下一步。



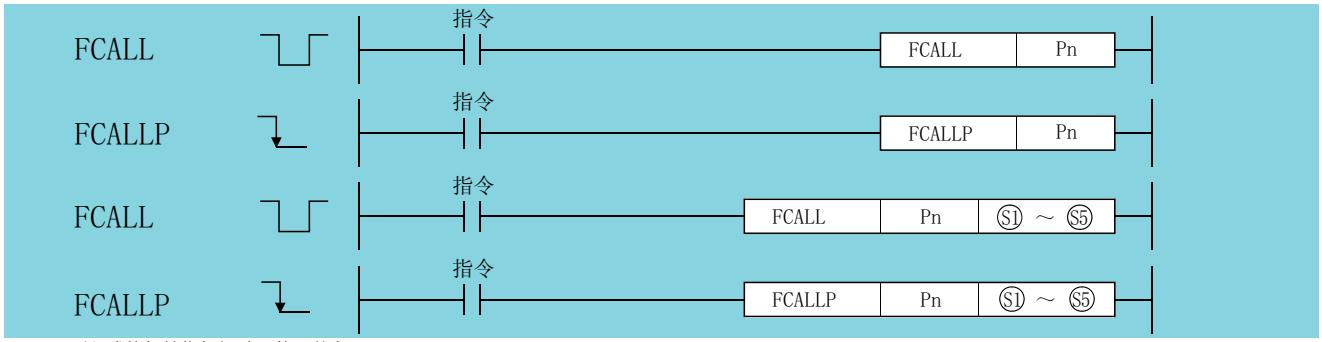
出　錯

(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4211	執行 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P)、XCALL 指令後，在執行 RET 指令之前執行了 END、FEND、GOEND、STOP 指令時。	○	○	○	○	○	○
4212	在執行 CALLL(P)、FCALL(P)、ECALL(P)、EFCALL(P)、XCALL 指令之前執行了 RET 指令時。	○	○	○	○	○	○

7.6.5　FCALL、FCALLP

Basic　High performance　Process　Redundant　Universal　LCPU



Pn　：副程式的起始指標編號 (軟元件名)。

S1 ～ S5：作爲變數傳送到副程式中的軟元件編號 (位、BIN16 位、BIN32 位)。

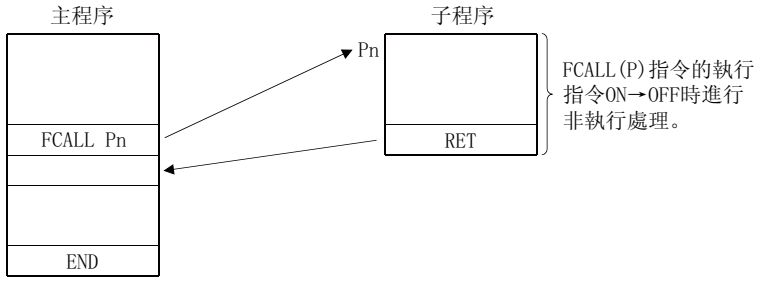
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				P
Pn	---	---		---					○
S1～S5	○ (除 F 以外)	○		○					---

功　能

(1)　如果執行 FCALL(P) 指令，將執行 Pn 中指定的指標的副程式的非執行處理。

[FCALL(P) 指令可以執行由同一個程式文件中的指標指定的副程式及由公共指標指定的副程式。]

(a) 非執行處理是指，對各線圈指令執行與條件設置處於 OFF 狀態時相同的處理。

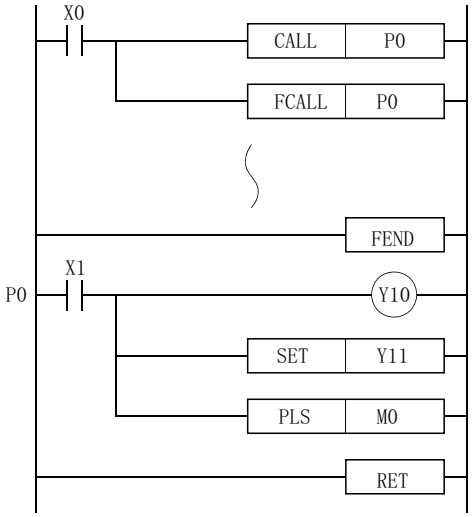


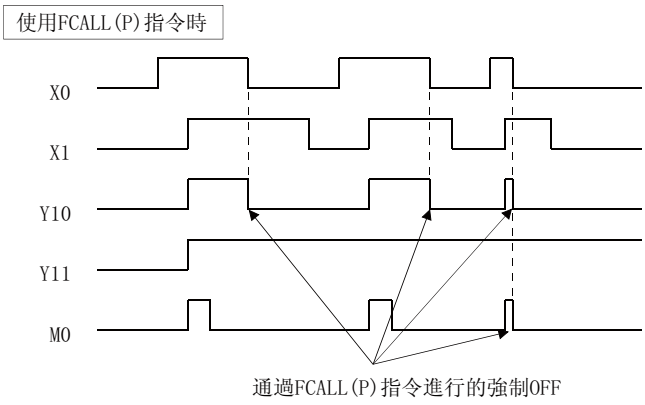
(b) 非執行處理後的各線圈指令的運算結果與條件觸點的 ON/OFF 狀態無關，其情況如下所示：

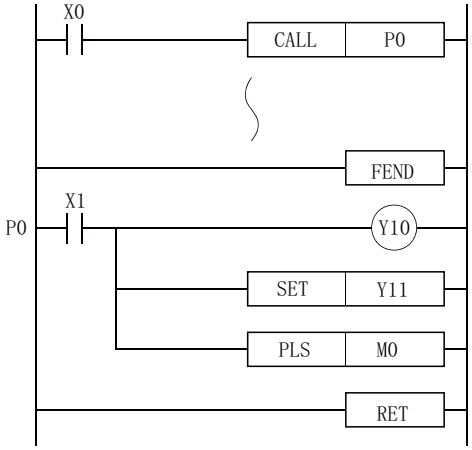
OUT 指令	.....	強制 OFF
SET 指令 RST 指令 SFT 指令 基本指令 應用指令	...	狀態保持
PLS 指令 脈衝化指令 (□ P)	...	與條件觸點 OFF 時相同的處理
低速 / 高速計時器的當前值	....	0
累計計時器的當前值 計數器的當前值	...	保持

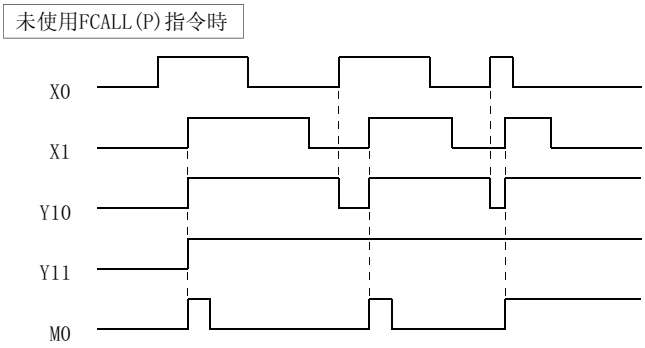
(2) FCALL(P) 指令是與 CALL(P) 指令組合使用。

(3) 將 FCALL(P) 指令與 CALL(P) 指令組合執行時，如果執行指令處於 OFF 狀態則進行副程式的非執行處理，因此可以對 OUT 指令、PLS 指令（包括 P 指令）進行強制 OFF。
未將 FCALL(P) 指令與 CALL(P) 指令組合執行時，即使執行指令處於 OFF 狀態也不進行副程式的非執行處理，因此各線圈指令的輸出狀態將被保持。

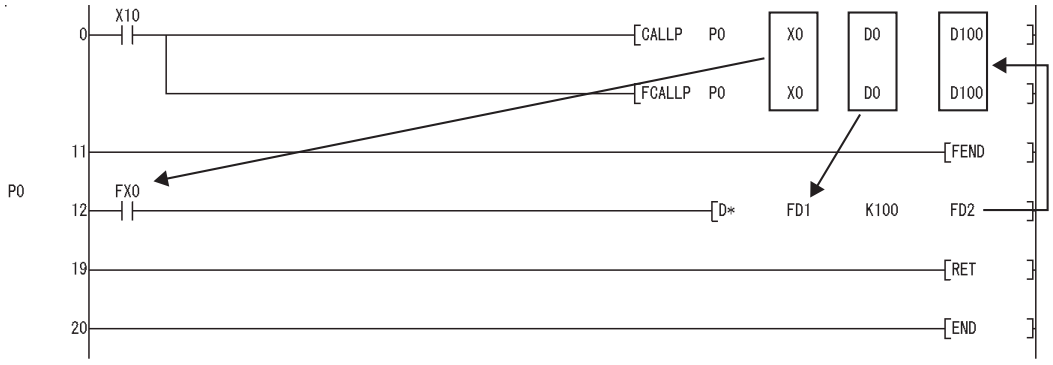








(4) 在副程式中使用功能軟元件 (FX、FY、FD) 時，在Ⓢ1～Ⓢ5中指定與功能軟元件對應的軟元件。
在Ⓢ1～Ⓢ5中指定的軟元件的內容如下所示。



(a) 在副程式執行之前，將位元資料的內容傳送到 FX 中，將字資料的內容傳送到 FD 中。
(b) 在副程式執行之後，將 FY、FD 的內容傳送到對應的軟元件中。

(c) 功能軟元件的處理單位如下所示。

· FX,FY : 位元單位

· FD : 4 字單位

根據變數中指定的軟元件的種類，所使用的資料大小有所不同。

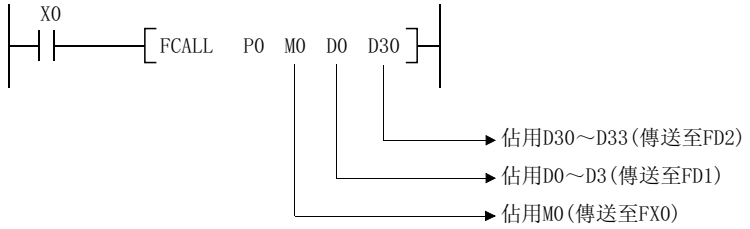
功能軟元件中指定的軟元件應預留出相當於該資料大小的容量。

未能預留出相當於該資料大小的容量時將會變爲出錯狀態。

功能軟元件	使用軟元件	資料大小	備註
· FX · FY	位軟元件	1 點	------
	字軟元件的位指定時	1 位	
· FD	位軟元件的位數指定時 *1	4 字	FD 的高位 2 字將變爲 0。
	字軟元件	4 字	------

*1: 位軟元件的位數指定時，即使在(S1)～(S5)中指定的軟元件號不是 16 的倍數時，也不會變爲出錯狀態。

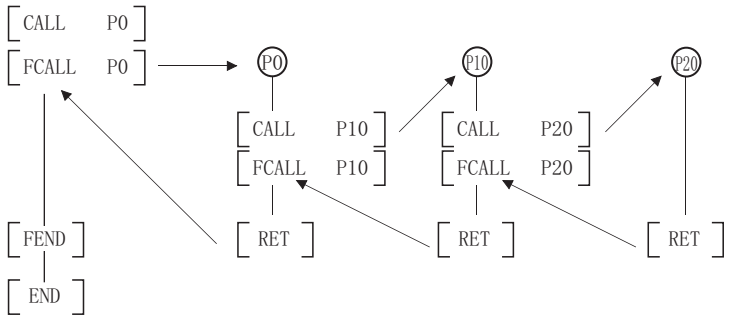
[主程序]



(5) 在 FCALL(P) 指令中，可使用範圍爲(S1)～(S5)。

(6) FCALL(P) 指令最多可以有 16 級嵌套。

但是，該 16 級是 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P)、XCALL 指令中嵌套的總級數。



出 錯

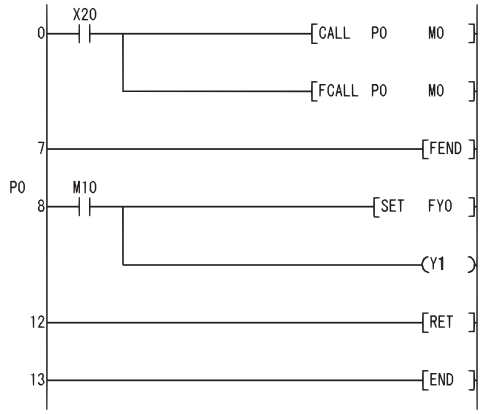
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	變數中指定的軟元件未能預留出相當於資料大小的容量時。	○	○	○	○	○	○
4210	FCALL(P) 指令中指定的指標在副程式中不存在時。	○	○	○	○	○	○
4211	在執行 FCALL(P) 指令後，執行 RET 指令前執行了END、FEND、GOEND、STOP 指令時。	○	○	○	○	○	○
4212	在執行 FCALL(P) 指令前執行了 RET 指令時。	○	○	○	○	○	○
4213	執行了第 17 級嵌套時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，執行帶變數副程式，X20 由 ON → OFF 時進行強制非執行處理的程式。

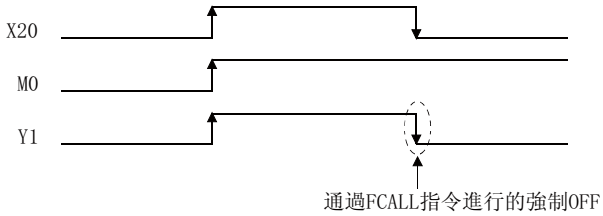
[梯形圖模式]



[列表模式]

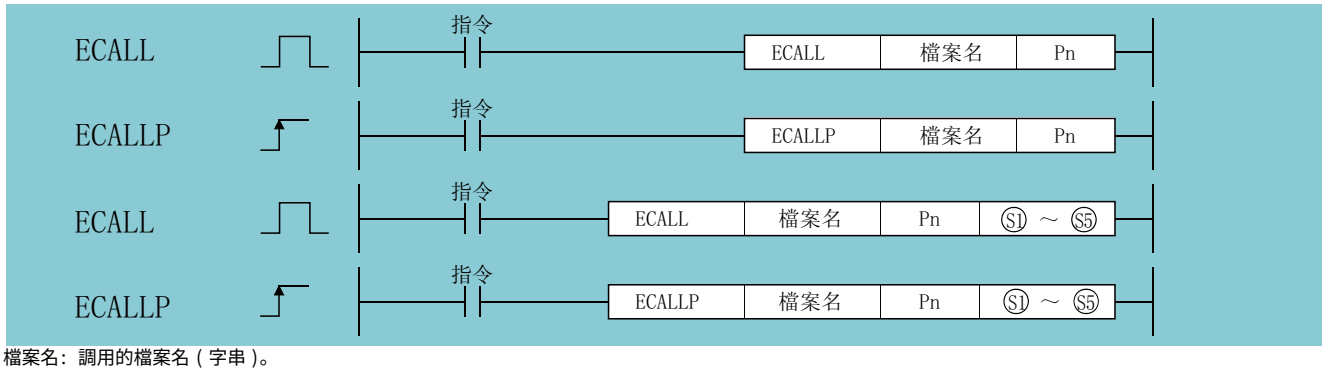
步	指令	軟元件	
0	LD	X20	
1	CALL	P0	M0
4	FCALL	P0	M0
7	FEND		
8	P0		
9	LD	M10	
10	SET	FY0	
11	OUT	Y1	
12	RET		
13	END		

[動作]



7.6.6 ECALL、ECALLP

Basic | High performance | Process | Redundant | Universal | LCPU



檔案名：調用的檔案名（字串）。

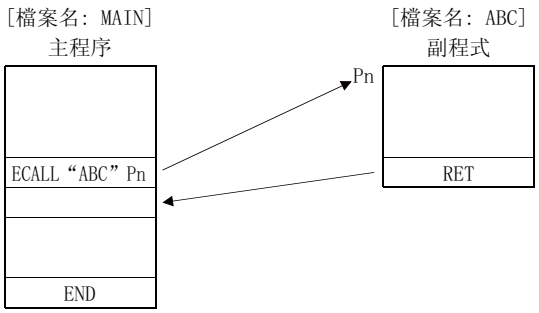
Pn　：副程式的起始指標編號（軟元件名）。

(S1) ～ (S5)：作爲變數傳送到副程式中的軟元件編號（位、BIN16 位、BIN32 位）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數		其他 P
	位	字		位	字			K、H	$	
檔案名	---	○		---					○	---
Pn	---	---		---					---	○
(S1)～(S5)	○（除 F 以外）	○		○					---	---

功 能

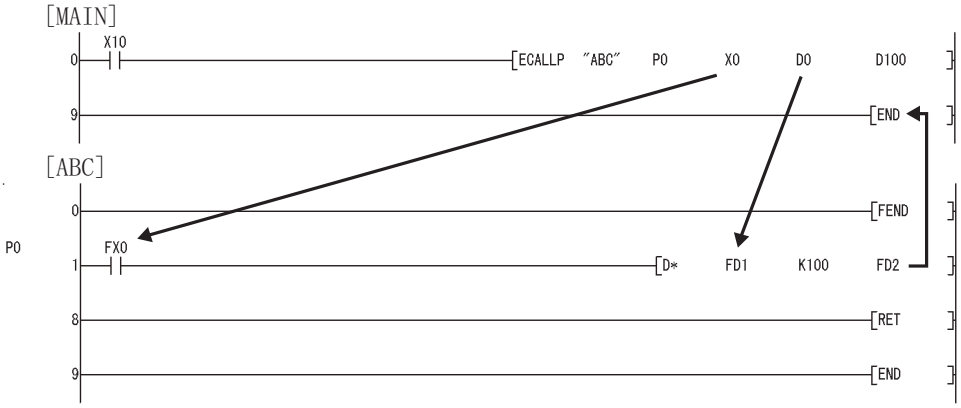
(1) 如果執行 ECALL(P) 指令，將執行指定程式檔案名的 Pn 中指定的指標的副程式。
使用 ECALL(P) 指令可從其他程式文件中調用使用局部指標的副程式。



(2) 檔案名只能指定驅動器 0(程式記憶體 / 內置 RAM) 中存儲的程式文件。

(3) 檔案名中無需指定副檔名 (.QPG)。
(僅以 .QPG 的文件爲物件。)

(4) 在副程式中使用功能軟元件 (FX、FY、FD) 時，在Ⓢ1～Ⓢ5中指定與功能軟元件對應的軟元件。
在Ⓢ1～Ⓢ5中指定的軟元件的內容如下所示。



(a) 在副程式執行之前，將位元資料的內容傳送到 FX 中，將字資料的內容傳送到 FD 中。

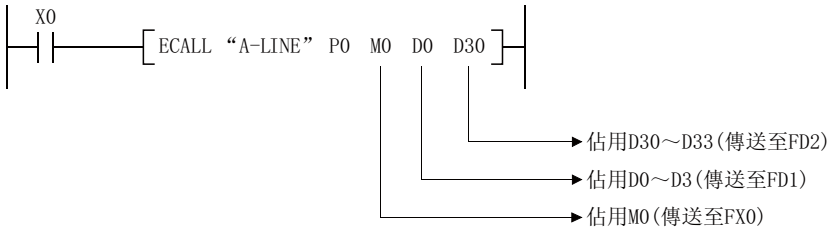
(b) 在副程式執行之後，將 FY、FD 的內容傳送到對應的軟元件中。

(c) 功能軟元件的處理單位如下所示。
· FX,FY : 位元單位
· FD : 4 字單位
根據變數中指定的軟元件的種類，所使用的資料大小有所不同。
功能軟元件中指定的軟元件應預留出相當於該資料大小的容量。
未能預留出相當於該資料大小的容量時將會變爲出錯狀態。

功能軟元件	使用軟元件	資料大小	備註
· FX · FY	位軟元件	1 點	------
	字軟元件的位指定時	1 位	
· FD	位軟元件的位數指定時 *1	4 字	資料大小根據所使用的指令而有所不同。
	字軟元件	4 字	

*1: 位軟元件的位數指定時，即使在Ⓢ1～Ⓢ5中指定的軟元件號不是 16 的倍數時，也不會變爲出錯狀態。

[主程序]



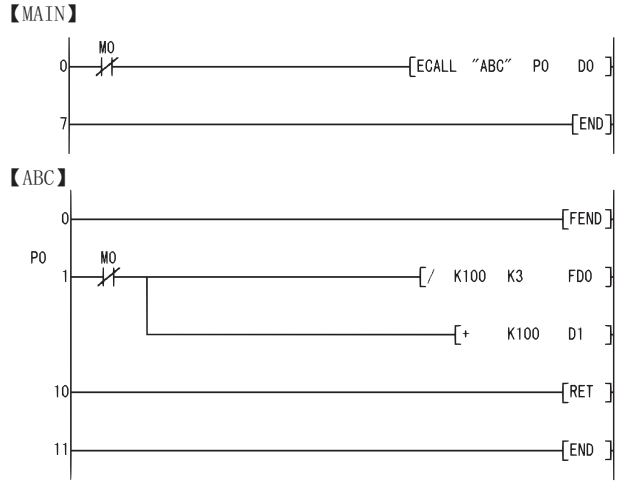
(5) 在 ECALL(P) 指令中，可使用範圍爲Ⓢ1～Ⓢ5。

(6) ECALL(P) 指令的變數中使用的軟元件不應在副程式中使用。
如果在副程式中使用了 ECALL(P) 指令的變數中使用的軟元件，將無法正常進行運算。(參閱以下程式示例。)

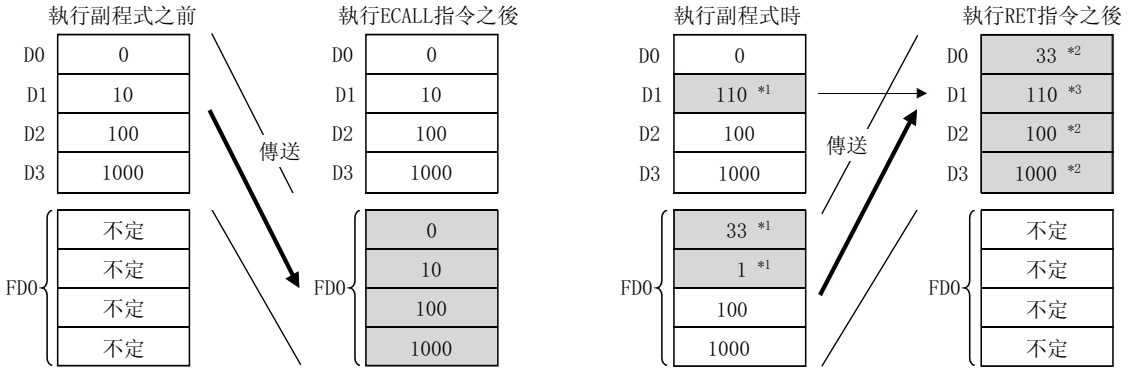
不正常的運算示例

將 D0 指定到副程式的 FD0 中，且在副程式中使用了 D1 時的動作如下所示。

[程式示例]



[執行了副程式後的動作]

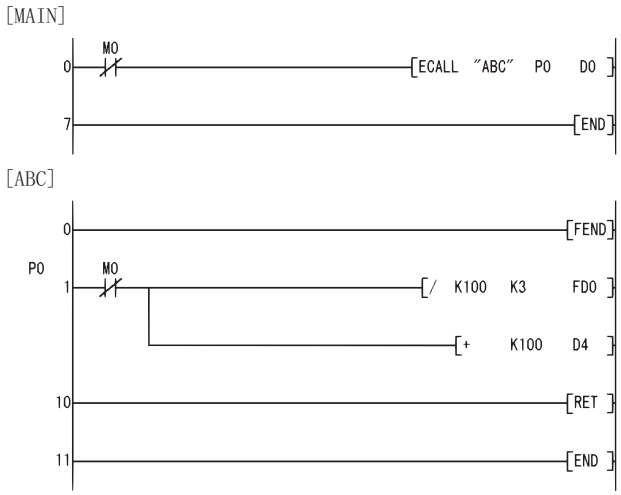


*1: 存儲副程式的執行結果。
*2: 替換爲功能軟元件的值。
*3: D1 中不能反映功能軟元件的值。

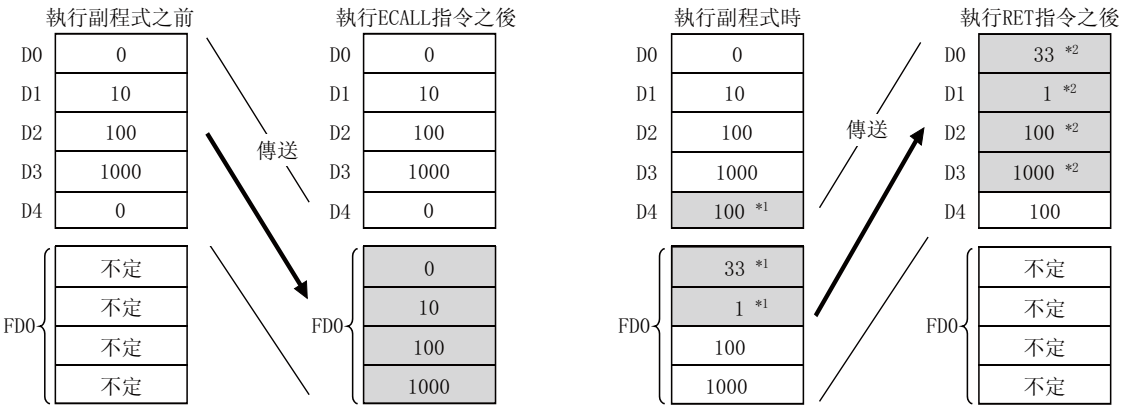
正常的運算示例

將 D0 指定到副程式的 FD0 中，且在副程式中使用了 D4 時的動作如下所示。

[程式示例]



[執行了副程式後的動作]



*1: 存儲副程式的執行結果。

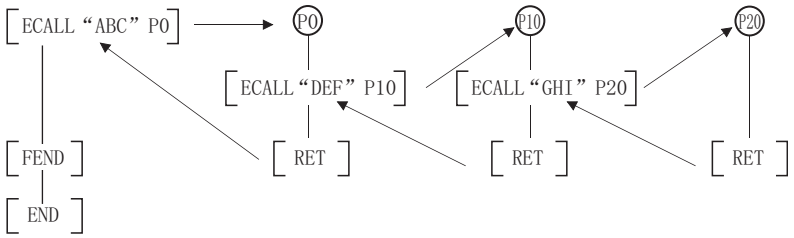
*2: 替換爲功能軟元件的值。

(7) ECALL(P) 指令的變數中指定的軟元件號不應重復。

如果重復，有可能導致無法正常運算。

(8) ECALL(P) 指令最多可以有 16 級嵌套。

但是，該 16 級是 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P)、XCALL 指令中嵌套的總級數。



(9) 對於在副程式內使其變爲 ON 的軟元件，即使副程式非執行時其 ON 狀態也仍將被保持。

對於在執行副程式時變爲 ON 的軟元件，可以通過 ECALL(P) 指令使其變爲 OFF。

出　錯

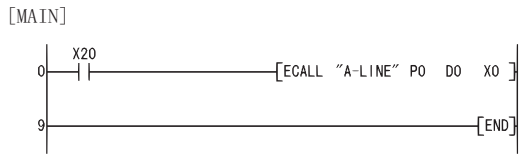
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	指定的文件不存在時。	○	○	○	○	○	○
2411	無法執行指定的文件時。	○	○	○	○	○	○
4101	變數中指定的軟元件未能預留出相當於資料大小的容量時。	○	○	○	○	○	○
4210	ECALL(P) 指令中指定的指標在副程式中不存在時。	○	○	○	○	○	○
4211	在執行 ECALL(P) 指令後，執行 RET 指令前執行了 END、FEND、GOEND、STOP 指令時。	○	○	○	○	○	○
4212	在執行 ECALL(P) 指令前執行了 RET 指令時。	○	○	○	○	○	○
4213	執行了第 17 級嵌套時。	○	○	○	○	○	○

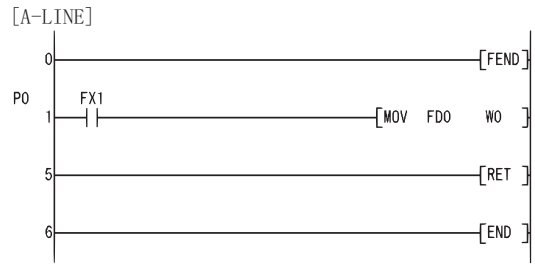
程式示例

(1)　以下爲 X20 變爲 ON 時，執行程式名爲 A-LINE 的 P0 的程式。

[梯形圖模式]　　[列表模式]



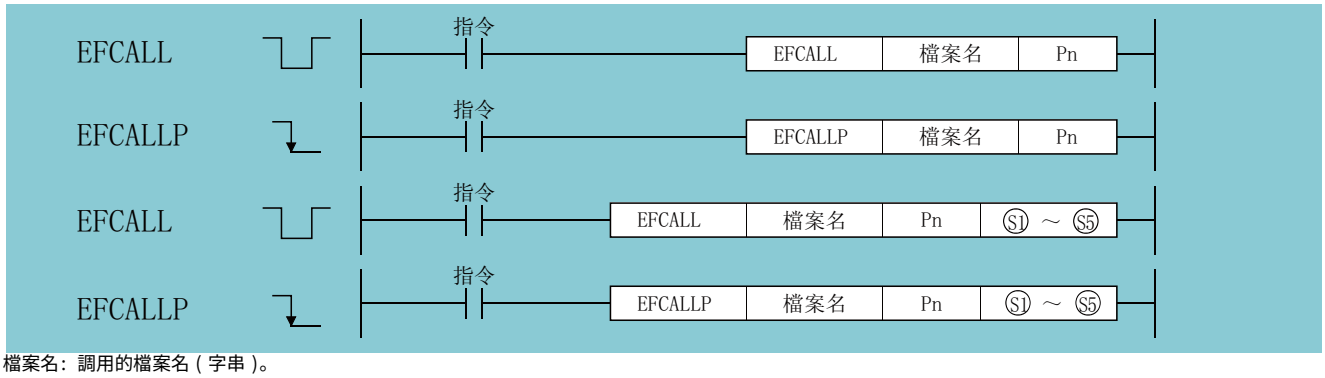
步	指令	軟元件
0	LD	X20
1	ECALL	"A-LINE" P0 D0 X0
9	END	



步	指令	軟元件
0	FEND	
1	P0	
2	LD	FX1
3	MOV	D0 W0
5	RET	
6	END	

7.6.7 EFCALL、EFCALLP

Basic High performance Process Redundant Universal LCPU



檔案名：調用的檔案名（字串）。
Pn　：　副程式的起始指標編號（軟元件名）。

(S1) ～ (S5)：作爲變數傳送到副程式中的軟元件編號（位、BIN16 位、BIN32 位）。

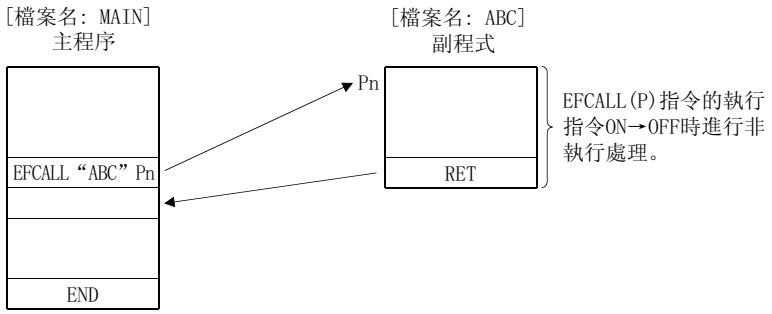
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數		其他 P
	位	字		位	字			K、H	$	
檔案名	---	○		---					○	---
Pn	---	---		---					---	○
(S1)～(S5)	○（除 F 以外）	○		○					---	---

功　能

(1) 如果執行 EFCALL(P) 指令，將執行指定 Pn 中指定的指標的副程式的非執行處理。

使用 EFCALL(P) 指令可從其他程式文件中調用使用局部指標的副程式。

(a) 非執行處理是指，對各線圈指令執行與條件設置處於 OFF 狀態時相同的處理。

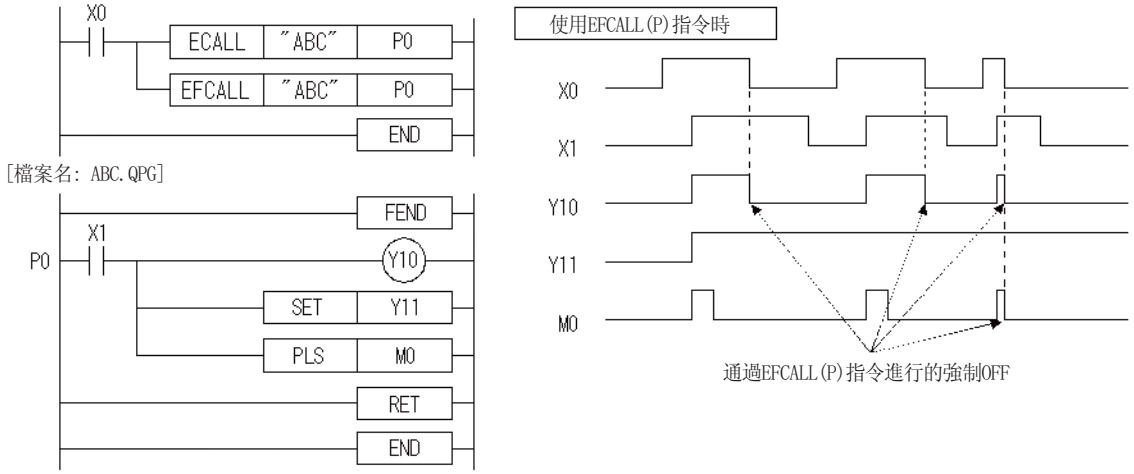


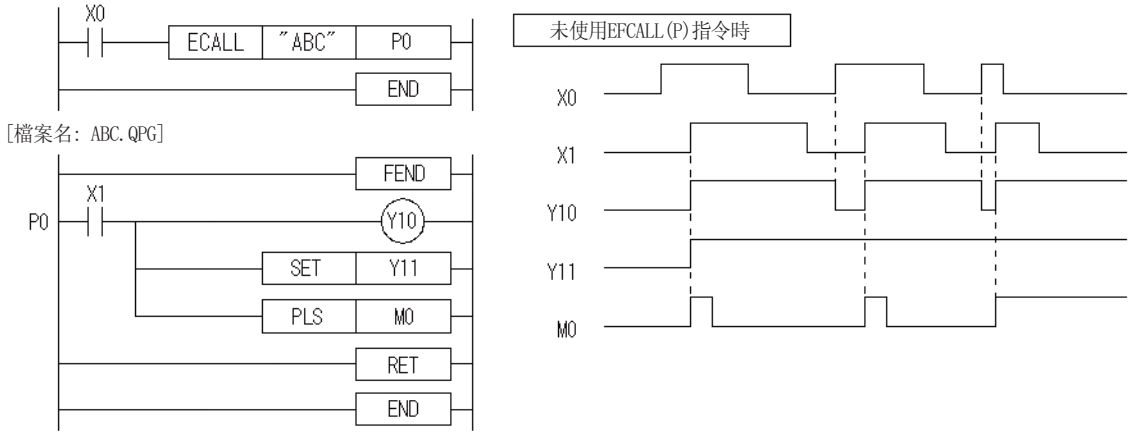
(b) 非執行處理後的各線圈指令的運算結果與條件觸點的 ON/OFF 狀態無關，其情況如下所示：

OUT 指令 ..... 強制 OFF
SET 指令
RST 指令
SFT 指令
基本指令
應用指令 ... 狀態保持
PLS 指令
脈衝化指令 (□ P) ... 與條件觸點 OFF 時相同的處理
低速 / 高速計時器的當前值 0
累計計時器的當前值
計數器的當前值 ... 保持

(2) EFCALL(P) 指令是與 ECALL(P) 指令組合使用。

(3) 將 EFCALL(P) 指令與 ECALL(P) 指令組合執行時，如果執行指令處於 OFF 狀態則進行副程式的非執行處理，因此可以對 OUT 指令、PLS 指令 (包括 P 指令) 進行強制 OFF。
未將 EFCALL(P) 指令與 ECALL(P) 指令組合執行時，即使執行指令處於 OFF 狀態也不進行副程式的非執行處理，因此各線圈指令的輸出狀態將被保持。

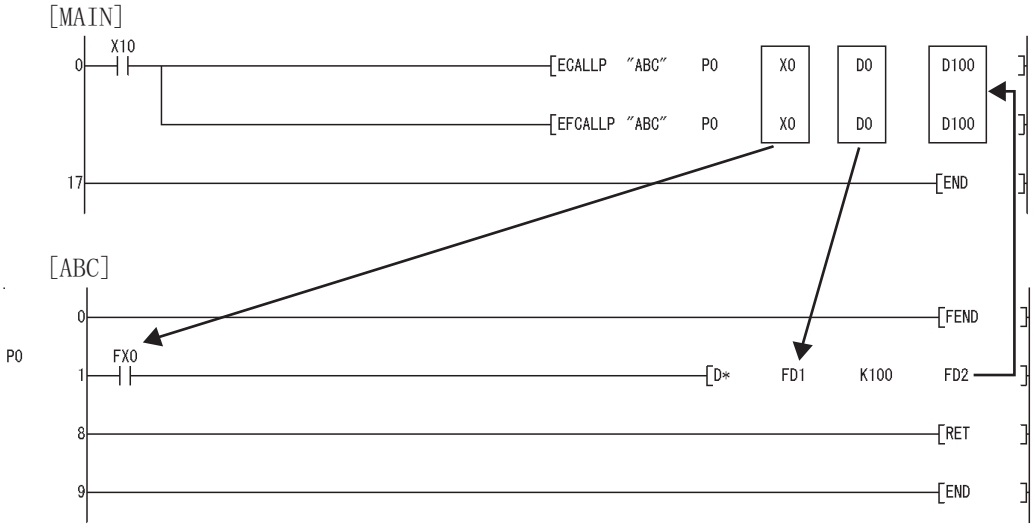




(4) 檔案名只能指定驅動器 0(程式記憶體 / 內置 RAM) 中存儲的程式文件。

(5) 檔案名中無需指定副檔名 (.QPG)。
(僅以 .QPG 的文件爲物件。)

(6) 在副程式中使用功能軟元件 (FX、FY、FD) 時，在Ⓢ1～Ⓢ5中指定與功能軟元件對應的軟元件。



(a) 在副程式執行之前，將位元資料的內容傳送到 FX 中，將字資料的內容傳送到 FD 中。

(b) 在副程式執行之後，將 FY、FD 的內容傳送到對應的軟元件中。

(c) 功能軟元件的處理單位如下所示。

· FX、FY ： 位元單位

· FD ： 4 字單位

根據變數中指定的軟元件的種類，所使用的資料大小有所不同。

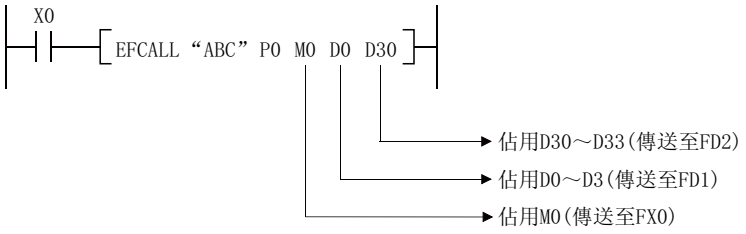
功能軟元件中指定的軟元件應預留出相當於該資料大小的容量。

未能預留出相當於該資料大小的容量時將會變爲出錯狀態。

功能軟元件	使用軟元件	資料大小	備註
· FX · FY	位軟元件	1 點	------
	字軟元件的位指定時	1 位	
· FD	位軟元件的位數指定時 *1	4 字	FD 的高位 2 字將變爲 0。
	字軟元件	4 字	------

*1: 位軟元件的位數指定時，即使在Ⓢ1～Ⓢ5中指定的軟元件號不是 16 的倍數時，也不會變爲出錯狀態。

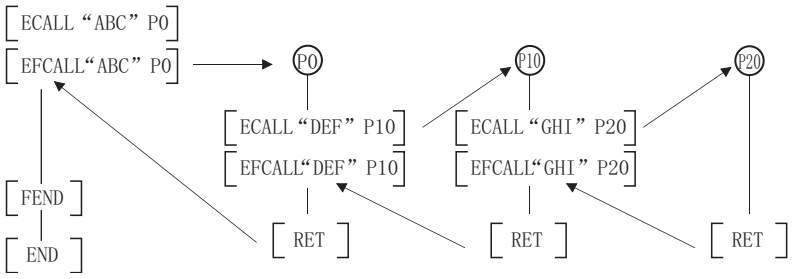
[主程序]



(7) 在 EFCALL(P) 指令中，可使用範圍爲Ⓢ1～Ⓢ5。

(8) 副程式中使用的功能軟元件數必須與 EFCALL(P) 指令中的變數數相同。
此外，功能軟元件與 EFCALL(P) 的變數類型應該完全一致。

(9) EFCALL(P) 指令最多可以有 16 級嵌套。
但是，該 16 級是 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P)、XCALL 指令中嵌套的總級數。



出　錯

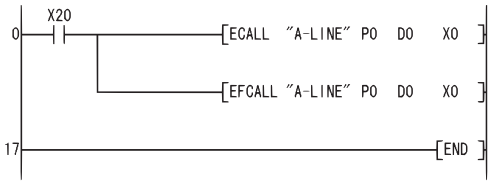
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2411	指定的文件無法執行時。	○	○	○	○	○	○
4101	引數中指定的軟元件未能預留出相當於資料大小的容量時。	○	○	○	○	○	○
4210	EFCALL(P) 指令中指定的指標在副程式中不存在時。 指定的文件不存在時。	○	○	○	○	○	○
4211	在執行 EFCALL(P) 指令後，執行 RET 指令前執行了END、FEND、GOEND、STOP 指令時。	○	○	○	○	○	○
4212	在執行 EFCALL(P) 指令前執行了 RET 指令時。	○	○	○	○	○	○
4213	執行了第 17 級嵌套時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，執行帶變數副程式，X20 由 ON → OFF 時進行強制非執行處理的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X20		
1	ECALL	"A-LINE" P0	D0	X0
9	EFCALL	"A-LINE" P0	D0	X0
17	END			

7.6.8 XCALL

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 " 04122 " 以後的基本型 QCPU 中可以使用。

Pn　：副程式的起始指標編號 (軟元件名)。

S1 ～ S5：作爲變數傳送到副程式中的軟元件編號 (位、BIN16 位、BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他 P
	位	字		位	字				
P	---	---		---					○
S1～S5	○ (除 F 以外)	○		○					---

功 能

(1) XCALL 指令是進行副程式的執行和非執行處理的指令。

(a) 副程式的執行

在運算過程中根據條件觸點的 ON/OFF 狀態執行各線圈指令。

(b) 副程式的非執行處理

對各線圈指令執行與條件觸點爲 OFF 狀態時相同的處理。

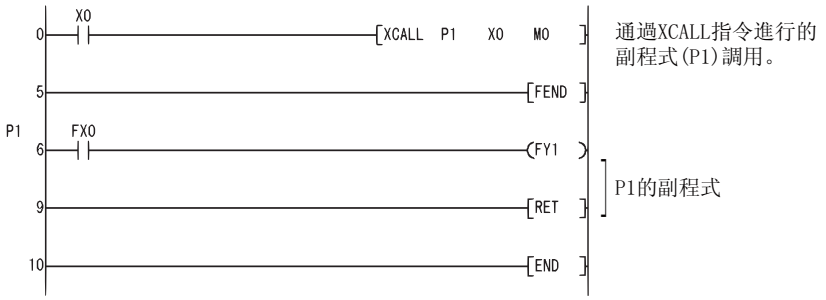
非執行處理之後的各線圈指令的運算結果與條件觸點的 ON/OFF 狀態無關，其情況如下所示。

OUT 指令	.....	強制 OFF
SET 指令 RST 指令 SFT 指令 基本指令 應用指令	...	狀態保持
PLS 指令 脈衝化指令 (□ P)	...	與條件觸點 OFF 時相同的處理
低速 / 高速計時器的當前值	....	0
累計計時器的當前值 計數器的當前值	...	保持

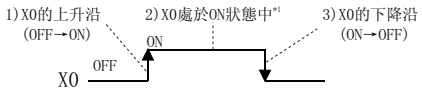
(2) XCALL 指令的動作根據 CPU 模組類型而有所不同。

各 CPU 模組的 XCALL 指令的動作如下圖的程式示例所示。

[程式示例]



[X0 的 ON/OFF 時機]

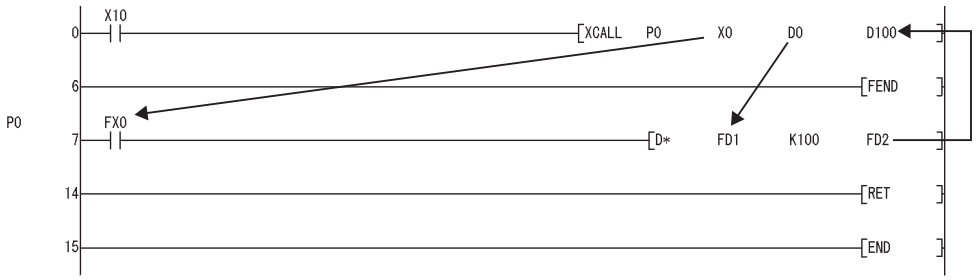


*1: X0 處於 ON 狀態中 (2)) 不包含 X0 的上升沿 (1))。

CPU 類型	XCALL 指令的動作
· 過程 CPU (序列號的前 5 位數： 07031 或以前) · 高性能型 QCPU (序列號的前 5 位數： 06081 或以前)	1) X0 的上升沿時 : 無處理 (不執行 “ P1 ” 的副程式。) 2) X0 處於 ON 狀態中 : 執行 “ P1 ” 的副程式。 3) X0 的下降沿時 : 執行 “ P1 ” 副程式的非執行處理。
· 高性能型 QCPU (序列號的前 5 位數： 06082 或以後。) · 過程 CPU (序列號的前 5 位數： 07032 或以後。)	1) 通過 SM734(XCALL 指令執行條件指定) 選擇 X0 的上升沿時的動作。 · SM734 爲 OFF 時 : 無處理 (不執行 “ P1 ” 的副程式。) · SM734 爲 ON 時 : 執行 “ P1 ” 的副程式。 2) X0 處於 ON 狀態中 : 執行 “ P1 ” 的副程式。 3) X0 的下降沿時 : 執行 “ P1 ” 副程式的非執行處理。
· 冗餘 CPU · 基本型 QCPU · 通用型 QCPU · LCPU	1) X0 的上升沿時 : 執行 “ P1 ” 的副程式。 2) X0 處於 ON 狀態中 : 執行 “ P1 ” 的副程式。 3) X0 的下降沿時 : 執行 “ P1 ” 副程式的非執行處理。

(3) 在副程式中使用功能軟元件 (FX、FY、FD) 時，在Ⓢ1～Ⓢ5中指定與功能軟元件對應的軟元件。

在Ⓢ1～Ⓢ5中指定的軟元件的內容如下所示。



(a) 在副程式執行之前，將位元資料的內容傳送到 FX 中，將字資料的內容傳送到 FD 中。

(b) 在副程式執行之後，將 FY、FD 的內容傳送到對應的軟元件中。

(c) 功能軟元件的處理單位如下所示。

・FX,FY ：位元單位

・FD ：4 字單位

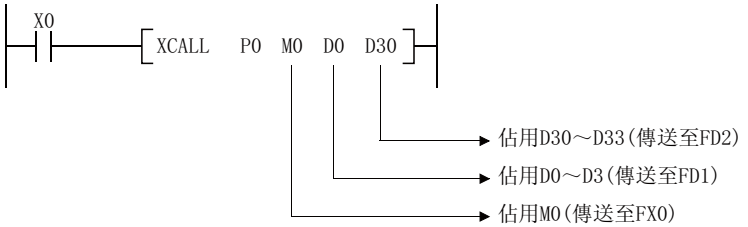
根據變數中指定的軟元件的種類，所使用的資料大小有所不同。

功能軟元件中指定的軟元件應預留出相當於該資料大小的容量。

未能預留出相當於該資料大小的容量時將會變爲出錯狀態。

功能軟元件	使用軟元件	資料大小	備註
・FX ・FY	位軟元件	1 點	------
	字軟元件的位指定時	1 位	
・FD	位軟元件的位數指定時 *2	4 字	資料大小根據所使用的指令而有所不同。
	字軟元件	4 字	

*2: 位軟元件的位數指定時，即使在Ⓢ1～Ⓢ5中指定的軟元件號不是 16 的倍數時，也不會變爲出錯狀態。



(4) 在 XCALL 指令中，可使用範圍爲Ⓢ1～Ⓢ5。

(5) 副程式中使用的功能軟元件數必須與 XCALL 指令中的變數數相同。

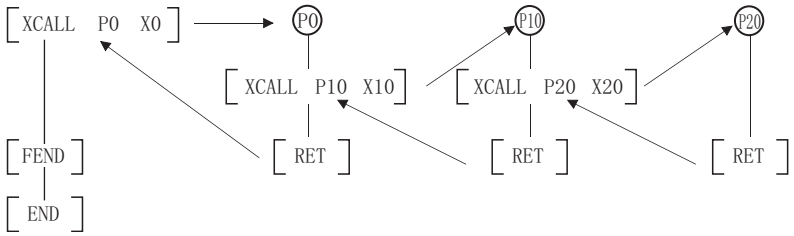
此外，功能軟元件與 XCALL 的變數類型應該完全一致。

(6) XCALL 指令的變數中指定的軟元件號不應重復。

如果重復，有可能導致無法正常運算。

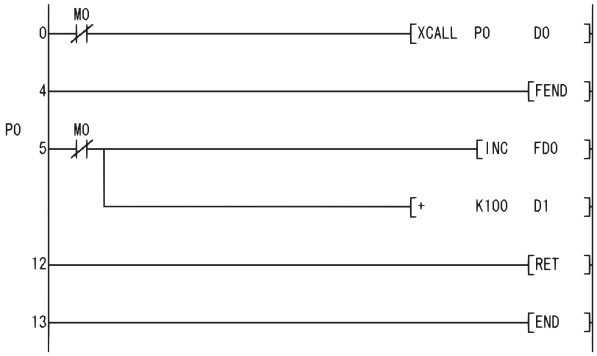
(7) XCALL 指令最多可以有 16 級嵌套。

但是，該 16 級是 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P)、XCALL 指令中嵌套的總級數。

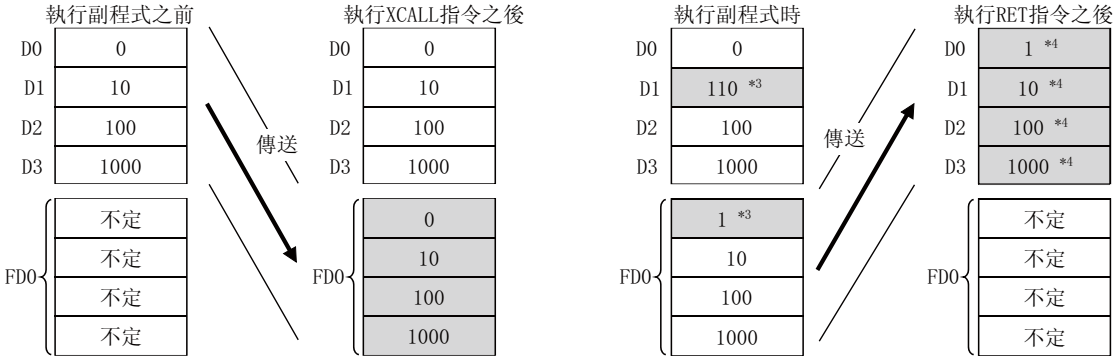


(8) XCALL 指令的變數中使用的軟元件不應在副程式中使用。
如果在副程式中使用了 XCALL 指令的變數中使用的軟元件，將無法正常進行運算。(參閱以下程式示例。)

不正常的運算示例
將 D0 指定到副程式的 FD0 中，且在副程式中使用了 D1 時的動作如下所示。
[程式示例]



[執行了副程式後的動作]

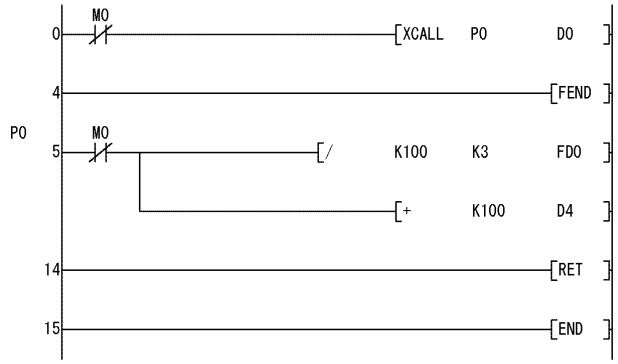


*3: 存儲副程式的執行結果。
*4: 替換爲功能軟元件的值。
*5: D1 中不能反映副程式中的運算結果。

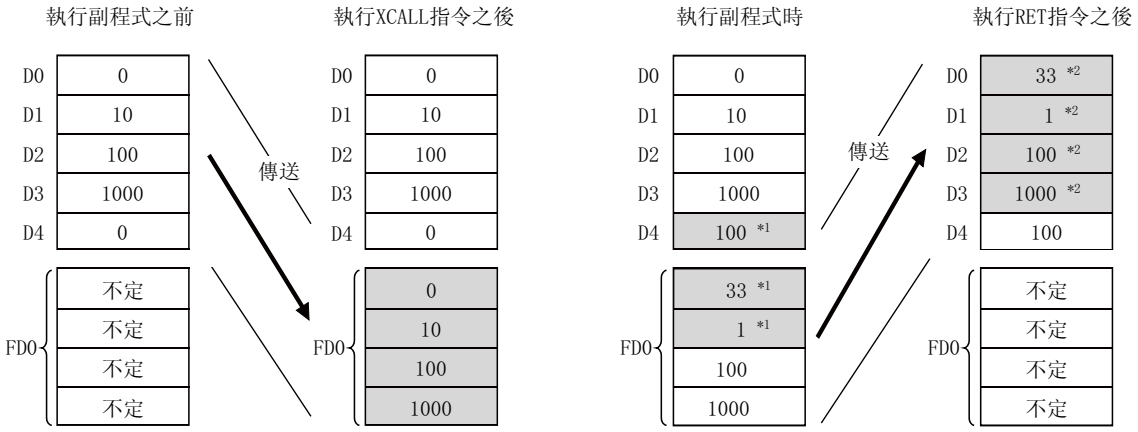
正常的運算示例

將 D0 指定到副程式的 FD0 中，且在副程式中使用了 D4 時的動作如下所示。

[程式示例]



[執行副程式後的動作]



*1: 存儲副程式的執行結果。

*2: 替換爲功能軟元件的值。

出　錯

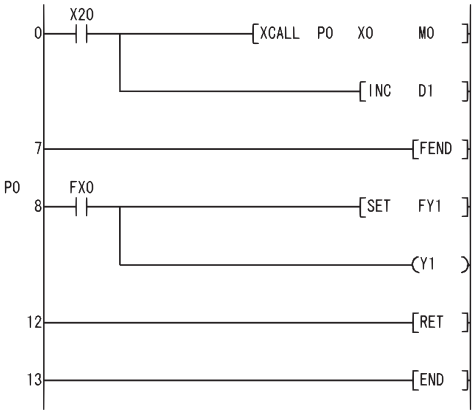
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	引數中指定的軟元件未能預留出相當於資料大小的容量時。	○	○	○	○	○	○
4210	在 XCALL 指令中指定的指標在副程式中不存在時。	○	○	○	○	○	○
4211	在執行 XCALL 指令後，執行 RET 指令前執行了 END、FEND、GOEND、STOP 指令時。	○	○	○	○	○	○
4212	在執行 XCALL 指令前執行了 RET 指令時。	○	○	○	○	○	○
4213	執行了第 17 級嵌套時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，執行帶變數副程式的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
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 COM

Basic | High performance | Process | Redundant | Universal | LCPU

以下 CPU 模組的 COM 指令請參閱 397 頁 7.6.10 項。

- 序列號爲 04122 或以後的基本型 QCPU
- 序列號爲 04012 或以後的高性能型 QCPU
- 序列號爲 07032 或以後的過程 CPU
- 冗餘 CPU
- 通用型 QCPU
- LCPU

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功 能

(1) COM 指令用於以下場合：

(a) 希望提高與遠端 I/O 站的發送 / 接收處理時機的速度時。

(b) 在資料鏈接執行過程中，希望與使用不同掃描時間的其他站之間進行可靠的資料發送 / 接收時。

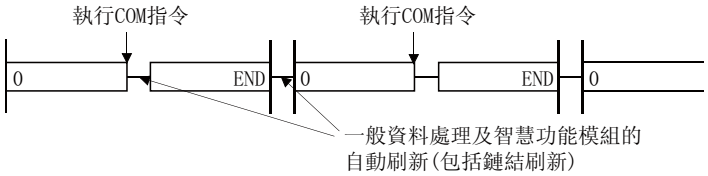
(2) 根據特殊繼電器 SM775 的 ON/OFF 狀態，COM 指令的處理有所不同。

- SM775 爲 OFF 時 ：進行自動刷新及與週邊設備的通信。 *1*2
- SM775 爲 ON 時 ：僅進行與週邊設備的通信。 *1

*1: 在與週邊設備的通信中也進行以下處理。
- 其他站監視處理。
- 通過串列通信模組對其他的智慧功能模組的緩衝記憶體進行讀取處理。

*2: 在自動刷新處理中進行以下處理。
- MELSECNET/10、MELSECNET/H 的刷新
- CC-Link 的刷新
- 智慧功能模組的自動刷新

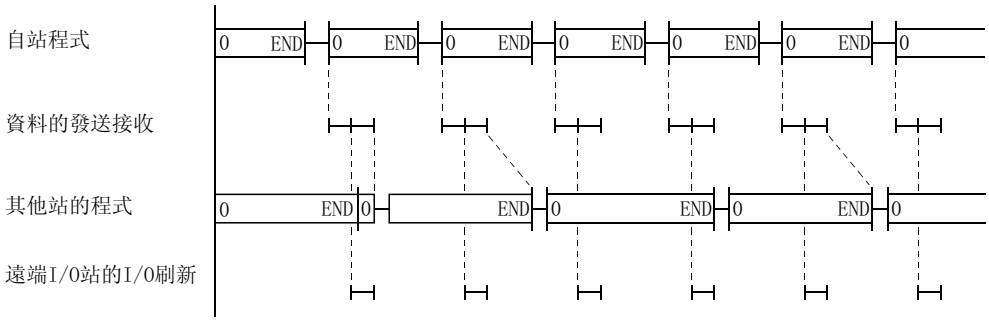
(3) 在執行了 COM 指令的時點，CPU 模組將暫停順控程式的處理，而進行與 END 處理時的一般資料處理及智慧功能模組的自動刷新（包括鏈結刷新）相同的處理。
但是，不進行 MELSECNET/10、MELSECNET/H 的低速迴圈刷新。



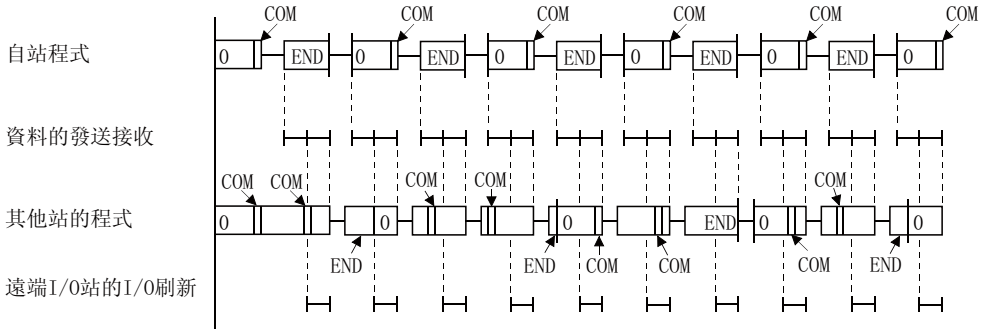
(4) COM 指令可在順控程式中使用任意次。
但是，由於與週邊設備的通信、智慧功能模組的自動刷新（包括鏈結刷新）需要耗費一定的時間，因此順控程式的掃描時間將會相應延長，應加以注意。

(5) 使用 COM 指令時的資料發送接收

(a) 未使用 COM 指令時的資料發送接收示例



(b) 使用了 COM 指令時的資料發送接收示例



1) 自站中使用了 COM 指令時，如上述 (b) 項所示，與遠端 I/O 站的資料發送接收次數可無條件增多，可以提高資料的發送接收速度。

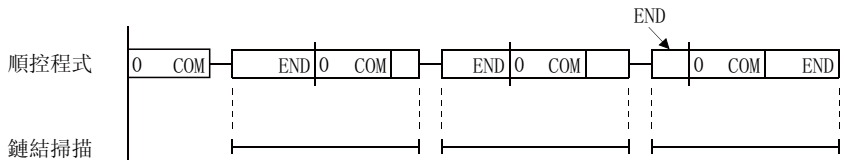
2) 在其他站的掃描時間長於自站的掃描時間的情況下，通過在其他站中使用 COM 指令，可以防止發生如上述 (a) 項所示的無法獲取資料的現象。

3) 如果對其他站使用 COM 指令，將在下述期間接收到來自於主站的指令時各進行 1 次鏈結刷新。

- 第0步 ～COM指令
- COM指令～COM指令
- COM指令～END指令

在此期間可各進行1次鏈結刷新

(6) 在鏈結掃描時間長於自站的順控程式的掃描時間的情況下，即使對自站指定 COM 指令，也不能加快資料的發送接收速度。



要 點

不能對以下程式使用 COM 指令：
・低速執行型程式
・中斷程式
・恒定周期執行型程式

出 錯

(1) 在 COM 指令中無運算出錯。

Basic Ver. / High performance Ver. / Process Ver. / Redundant / Universal

・在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。
・在序列號的前 5 位數爲 “ 04012 ” 以後高性能型 QCPU 中可以使用。
・在序列號的前 5 位數爲 “ 07032 ” 以後過程 CPU 中可以使用。

7.6.10 COM

關於下述 CPU 模組的 COM 指令請參閱 395 頁 7.6.9 項。

・序列號爲 04121 以前的基本型 QCPU
・序列號爲 04011 以前的高性能型 QCPU
・序列號爲 07031 以前的過程 CPU

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功 能

(1) 在順控程式執行過程中的任意的時機實施 I/O 刷新等的情況下使用 COM 指令。

(2) 執行 COM 指令時，可以執行下述刷新。

刷新專案	QCPU	LCPU
I/O 刷新	○	○
CC-link 的刷新	○	○
CC-link IE 控制網路的刷新	○	×
CC-link IE 現場網路的刷新	○ *1	○ *2
MELSECNET/H 的刷新	○	×
智慧功能模組的自動刷新	○	○
使用了多 CPU 系統的 QCPU 標準區域的自動刷新	○	×
多 CPU 系統的組外的輸入 / 輸出的獲取	○	×
使用了多 CPU 系統的多 CPU 高速通信區域的自動刷新	○	×
與顯示模組的通信	×	○
服務處理 (與編程工具、GOT 或者其他外部設備的通信)	○	○

*1: 以序列號的前 5 位數爲 “ 12012 ” 以後的 QCPU 爲物件。
*2: 以序列號的前 5 位數爲 “ 13012 ” 以後的 LCPU 爲物件。

備 註

在服務處理中還執行以下處理：
・其他站監視處理
・在串列通信模組中，對其他智慧功能模組的緩衝記憶體的讀取處理

(3) 將 SM755 置爲 OFF 時，進行除 I/O 刷新以外的處理。

(4) 選擇處理時

(a) 通過 SD778 選擇處理後，將 SM775 置爲 ON。

SM775 的 ON/OFF、SD778 中可指定的處理如下表所示：

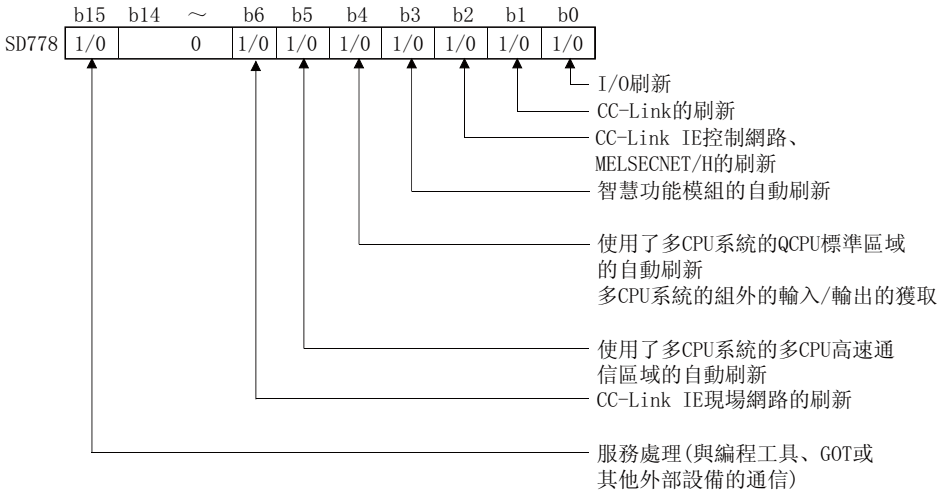
處理內容	QCPU		LCPU	
	SM775 爲 OFF 時	SM775 爲 ON 時	SM775 爲 OFF 時	SM775 爲 ON 時
I/O 刷新	非執行	可以選擇執行 / 非執行	非執行	可以選擇執行 / 非執行
CC-Link 的刷新	執行		執行	
CC-Link IE 控制網路的刷新			-	-
CC-Link IE 現場網路的刷新			執行	可以選擇執行 / 非執行
MELSECNET/H 的刷新			-	-
智慧功能模組的自動刷新			執行	可以選擇執行 / 非執行
使用了多 CPU 系統的 QCPU 標準區域的自動刷新			-	-
多 CPU 系統的組外的輸入 / 輸出的獲取			-	-
使用了多 CPU 系統的多 CPU 高速通信區域的自動刷新			-	-
與顯示模組的通信	-	-	執行	可以選擇執行 / 非執行
服務處理 (與編程工具、GOT 或其他外部設備的通信)	執行	可以選擇執行 / 非執行		

(b) 通過 SD778 選擇處理的執行 / 非執行

對 SD778 的各個位的執行 / 非執行按下表方式進行指定。

[對於 QCPU]

SD778 的位	執行	非執行
b0 ～ b6	1	0
b15	0	1



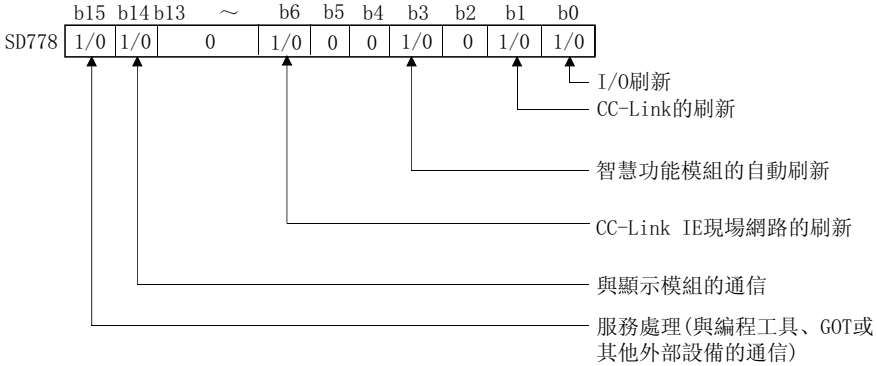
例 僅希望加快與遠端 I/O 站的發送接收處理的情況下，僅指定 MELSECNET/H 的刷新。(僅 SD778 的 b2 及 b15 寫入 1(SD778:8004_H))

要 點

通過 COM 指令進行的多 CPU 刷新在以下情況下執行：
・來自於其他站的接收動作： SD778 的 b4(CPU 共用記憶體的自動刷新) 爲 1 時
・來自於本站的發送動作： SD778 的 b15(與外部設備的通信的執行 / 非執行) 爲 0 時

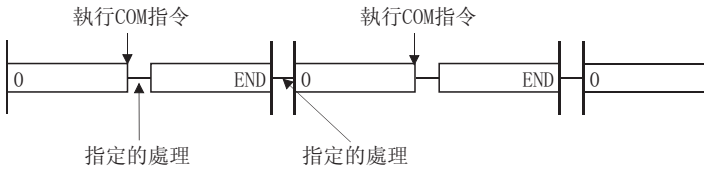
[對於 LCPU]

SD778 的位	執行	非執行
b0、b1、b3、b6、b14	1	0
b15	0	1



例 僅希望加快顯示模組的處理的情況下，僅指定與顯示模組的通信。(SD778 的 b14 及 b15 中寫入 1($SD778:C000_H$))

(5) 在執行了 COM 指令的時點 CPU 模組暫時中斷順控程式的處理，進行指定的處理。



(6) COM 指令在順控程式中的使用次數無限制。
但是，由於與通過 SD778 選擇的處理需要耗費時間，因此順控程式的掃描時間將相應延長，應加以注意。

(7) 對於通用型 QCPU 及 LCPU，在 COM 指令執行過程中將變爲中斷允許狀態。但是，通過中斷程式等使用刷新資料的情況下，有可能會發生資料背離，應加以注意。

(8) 在乙太網埠內置 QCPU、LCPU 中，CPU 內置乙太網埠上連接了乙太網的狀態下，通過 COM 指令執行了服務處理的情況下，處理時間有可能會延長。

要 點

1. 不能使用 COM 指令的程式如下所示：
・低速執行型程式
・中斷程式
・恒定周期執行型程式
2. 在冗餘 CPU 中，使用 COM 指令時有限制。有關詳細內容請參閱下述手冊：
・QnPRHCPU 用戶手冊 (冗餘系統篇)

出 錯

(1) 在 COM 指令中無運算出錯。

7.6.11 CCOM、CCOMP

Basic × | High performance × | Process × | Redundant × | Universal Ver. | LCPU

・在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---		---			---			

功　能

關於功能，請參閱 397 頁 7.6.10 項。

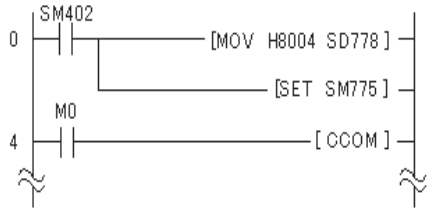
出　錯

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	在序列號的前 5 位數爲 “ 10101 ” 以前的 QnUD(H)CPU 中執行了 CCOM(P) 指令時。	---	---	---	---	○	○

程式示例

(1) 以下爲根據 M0 的 ON/OFF，可以對選擇刷新的執行 / 非執行進行切換的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	SM402
1	MOV	H8004 SD778
3	SET	SM775
4	LD	M0
5	CCOM	

7.6.12 IX、IXEND

Basic × | High performance | Process | Redundant | Universal × | LCPU ×

Ⓢ ：存儲變址修飾資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ	---	○		---					

功 能

(1) 使用在變址修飾表中設置的變址修飾值，對梯形圖內從 IX 指令起到 IXEND 指令爲止的所有軟元件進行軟元件號的變址修飾。

關於變址修飾表的創建方法，請參閱 403 頁 7.6.13 項。

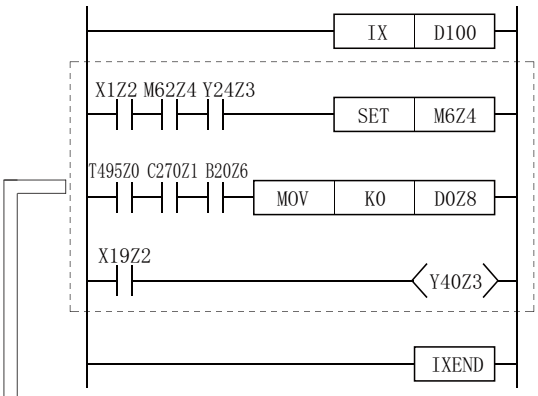
變址修改表的構成以及相應的變址寄存器編號如下所示：

	軟元件名	變址寄存器編號
Ⓢ	計時器(T)的修飾值	Z0
Ⓢ +1	計數器(C)的修飾值	Z1
Ⓢ +2	輸入(X)的修飾值	Z2
Ⓢ +3	輸出(Y)的修飾值	Z3
Ⓢ +4	內部繼電器(M)的修飾值	Z4
Ⓢ +5	鎖存繼電器(L)的修飾值	Z5
Ⓢ +6	鏈結繼電器(B)的修飾值	Z6
Ⓢ +7	變址繼電器(V)的修飾值	Z7

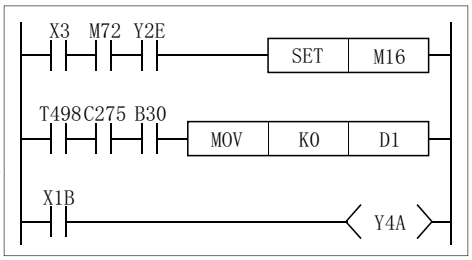
	軟元件名	變址寄存器編號	
Ⓢ +8	資料寄存器(D)的修飾值	Z8	
Ⓢ +9	鏈結寄存器(W)的修飾值	Z9	
Ⓢ +10	文件寄存器(R)的修飾值	Z10	
Ⓢ +11	緩衝寄存器I/O號(U)的修飾值	Z11	
Ⓢ +12	緩衝寄存器(G)的修飾值	Z12	
Ⓢ +13	鏈結直接軟元件網路號(J)的修飾值	Z13	*1
Ⓢ +14	文件寄存器(ZR)的修飾值	Z14	
Ⓢ +15	指標(P)的修飾值	Z15	

* 1: 使用基本型 QCPU 時，不支援 Z10 以後的變址寄存器。

(2) 進行軟元件號的變址修飾時，通過對各個軟元件預先設置修飾，對 IX ～ IXEND 指令之間的梯形圖中所使用的全部軟元件進行修飾值的加法運算，並根據加法運算後的軟元件號執行程式處理。



	修飾值		
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)



- 對X1、X19進行“2”的加法運算 → 作爲X3、X1B進行處理。
- 對Y24、Y40進行“10”(AH)的加法運算 → 作爲Y2E、Y4A進行處理。
- 對M6、M62進行“10”的加法運算 → 作爲M16、M72進行處理。
- 對B20進行“16”(10H)的加法運算 → 作爲B30進行處理。
- 對T495進行“8”的加法運算 → 作爲T498進行處理。
- 對C270進行“5”的加法運算 → 作爲C275進行處理。
- 對D0進行“1”的加法運算 → 作爲D1進行處理。

(3) 對於 PLS、PLF、□□□P 的指令之類的，通過輸入條件的上升沿僅執行 1 次的指令，不能通過 IX ～ IXEND 指令進行變址修飾。

(4) 由於修飾值的加法運算，軟元件號超出了軟元件範圍時，將不能進行正常處理。

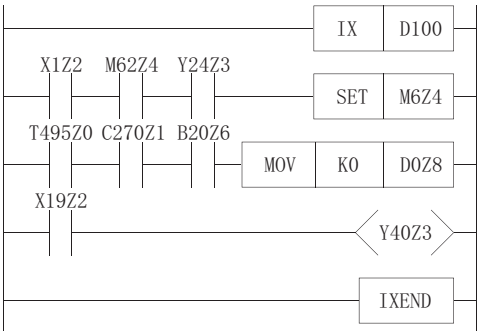
(5) 在進行順控程式的 RUN 中寫入時，不要執行 IX 指令、IXEND 指令。否則將不能進行正常處理。

(6) 將修飾值以 BIN 值格式預先設置到任意的字軟元件中後，在Ⓢ中對已設置了修飾值的軟元件的起始編號進行指定。

(7) 在 IX ～ IXEND 指令中，不要同時執行掃描執行型程式與中斷程式。

(8) 由於不能進行程式展開，因此需要由用戶創建程式。

對於通過 IX、IXEND 指令進行變址修飾的梯形圖，應由用戶進行變址寄存器的附加。*2



*2: 在執行 IXEND 指令之後，將 Zn 的值恢復爲執行 IX 指令之前的原來的值。

要 點

1. 將 IX ～ IXEND 指令用於普通的順控程式及中斷程式中時，應設置互鎖以防止同時執行。互鎖是指，將普通順控程式的 IX ～ IXEND 指令之間設置 DI 以執行中斷禁止。
2. 可以使用 IXDEV ～ IXSET 指令進行修飾值指定。
有關詳細內容請參閱 403 頁 7.6.13 項。

出 錯

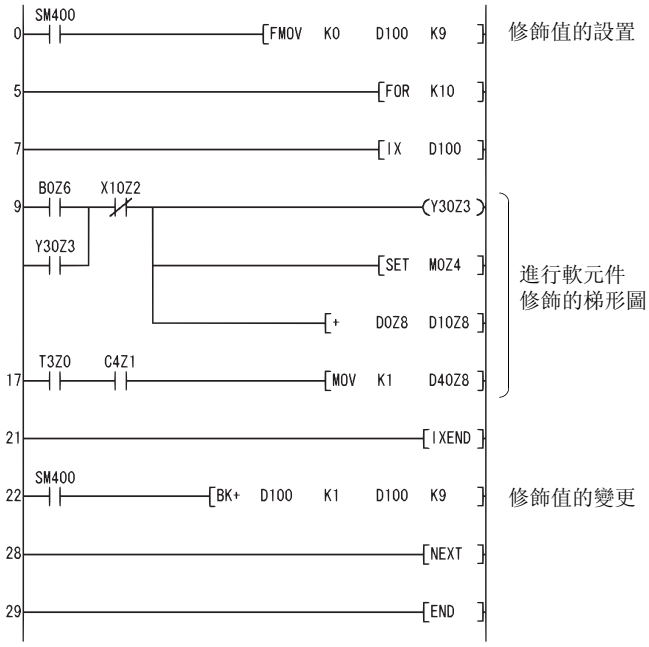
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4231	IXDEV 與 IXSET 指令未成對使用。 在執行 IX 指令後，執行 IXEND 指令之前執行了 END、FEND、GOEND、STOP 指令時。	○	○	○	○	○	○

程式示例

(1) 以下爲在變更軟元件號的同時將同一梯形圖執行 10 次的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件			
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				

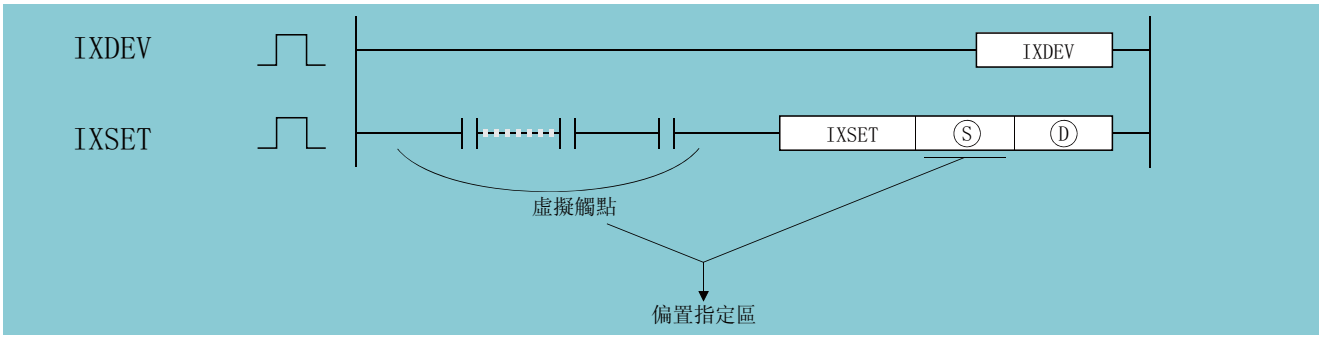
[動作]

	修飾值
D100	T的修飾值
D101	C的修飾值
D102	X的修飾值
D103	Y的修飾值
D104	M的修飾值
D105	L的修飾值
D106	B的修飾值
D107	V的修飾值
D108	D的修飾值

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

7.6.13 IXDEV、IXSET

Basic | High performance | Process | Redundant | Universal ✕ | LCPU ✕



Ⓢ ：存儲變址修飾資料的軟元件的起始編號（僅指標）P□（指標）。

Ⓓ ：存儲變址修飾資料的軟元件的起始編號（除指標以外）(BIN16 位）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				P
Ⓢ	---	---		---					○
Ⓓ	---	○		---					---

功　能

(1) 該指令是用於創建 IX、IXEND 指令中使用的變址修飾表的指令。

(2) 將通過偏置指定區指定的軟元件偏置值設置到Ⓓ中指定的變址修飾表中。

(3) 未指定的情況下將輸入 0。

(4) 字軟元件也以觸點（字軟元件的位指定）表示。
資料寄存器 10(D10) 是通過 D10.0 指定。
（位號可使用 0 ～ F 中的任意值。）

(5) 指定方法如下所示。*1(□爲偏置值。XX 爲任意值。)

軟元件	T	C	X	Y	M	L	V	B
指定方法	T□	C□	X□	Y□	M□	L□	V□	B□

軟元件	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

軟元件	P
指定方法	IXSET Ⓢ Ⓓ *3

*1: 使用基本型 QCPU 時，不能使用 R、U/G、J、ZR、P 的軟元件。

*2: 將 J□\後面的軟元件指定爲 B、W、X、Y 後，與其對應的偏置值也將被設置。

*3: 使用基本型 QCPU 時，應指定虛擬的軟元件號。

Ⓢ變爲 P□。

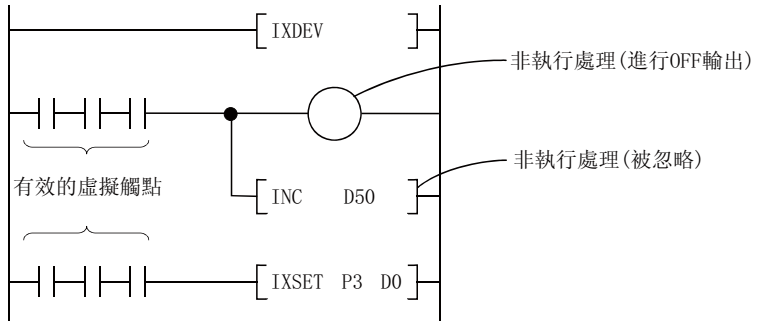
(6) 在偏置指定區記述了相同類型的軟元件時，最後設置的軟元件有效。

(7) 應將 IXDEV 與 IXSET 指令成對使用。

(8) ZR 時的有效範圍爲 0 ～ 32767。(將指定的軟元件號除以 32768 所得的餘數作爲偏置值。)

(9) 在偏置指定區的虛擬觸點隻對於 IXDEV 指令～ IXSET 指令範圍內的 LD 或者 AND 有效。如果記述了其他指令，IXDEV 指令～ IXSET 指令將不會執行。

例



出 錯

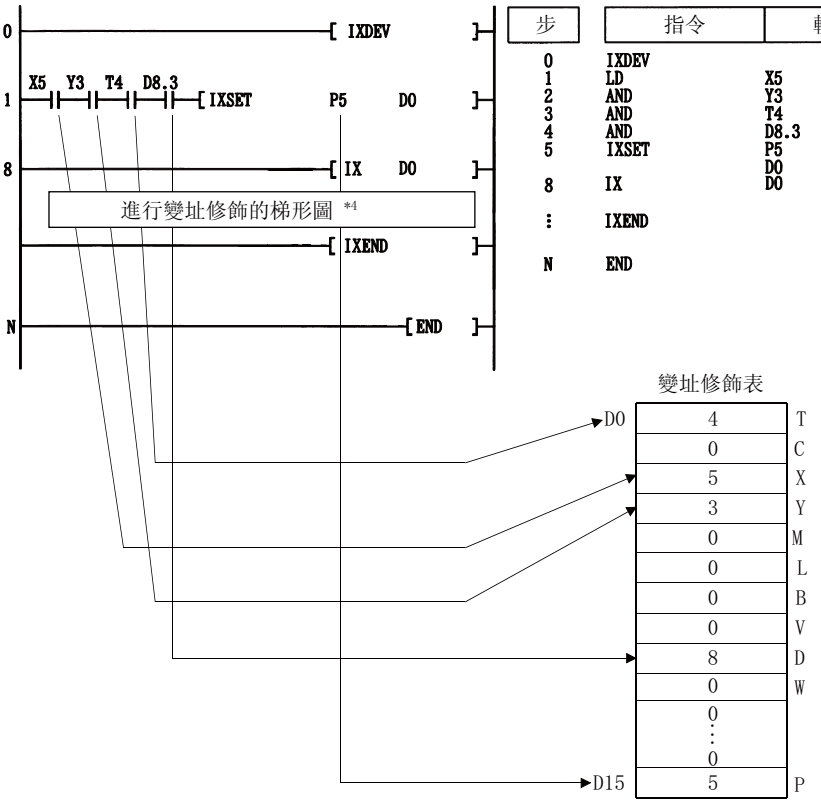
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4231	IXDEV 與 IXSET 指令未成對使用。	○	○	○	○	○	○

程式示例

(1) 以下爲對輸入 (X)、輸出 (Y)、資料寄存器 (D)、指標 (P) 的修飾值進行變更的程式。
使用基本型 QCPU 時，不能使用 R、U/G、J、ZR、P 的軟元件。

[梯形圖模式]



[列表模式]

步	指令	軟元件
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: 關於使用了 IX ～ IXEND 指令變址修飾，請參閱 400 頁 7.6.12 項。

7.7 資料表操作指令

7.7.1 FIFW、FIFWP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：寫入表的資料或者存儲資料的軟元件編號 (BIN16 位)。

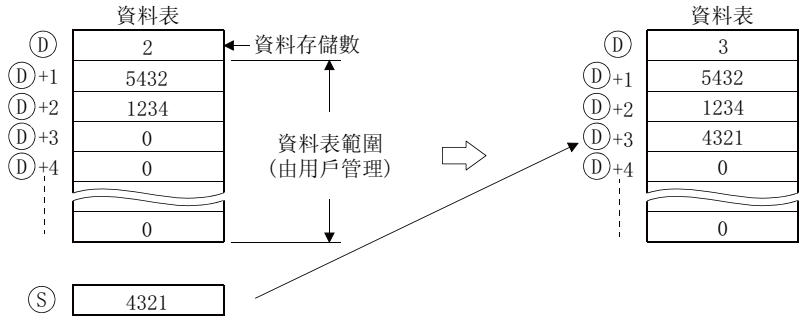
Ⓓ ：表的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○		○				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 將Ⓢ中指定的 16 位元資料存儲到Ⓓ中指定的資料表中。

將表中存儲的資料數量存儲到Ⓓ中，將Ⓢ中指定的資料依次存儲到Ⓓ+1 的後面。



(2) 初次執行 FIFW 指令時，應預先清除Ⓓ中指定的軟元件的值。

(3) 寫入到資料表中的資料數量及資料表範圍應由用戶管理。

(參閱程式示例 (2))

出　錯

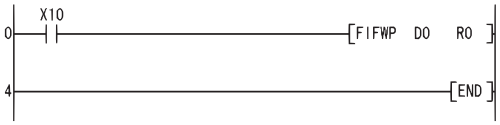
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	執行了 FIFW 指令時，資料表範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X10 變爲 ON 時，將 D0 的資料存儲到 R0 後面的資料表中的程式。

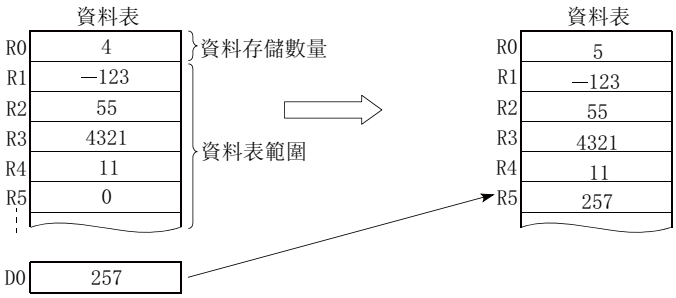
[梯形圖模式]



[列表模式]

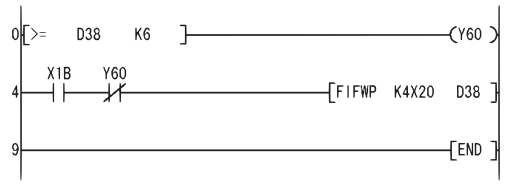
步	指令	軟元件	
0	LD	X10	
1	FIFWP	D0	R0
4	END		

[動作]



(2) 以下爲 X1B 變爲 ON 時，將 X20 ～ X2F 的資料存儲到 D38 ～ D44 的資料表中，資料存儲數量超過了 6 時，將 Y60 置爲 ON 使 FIFW 指令變爲禁用狀態的程式。

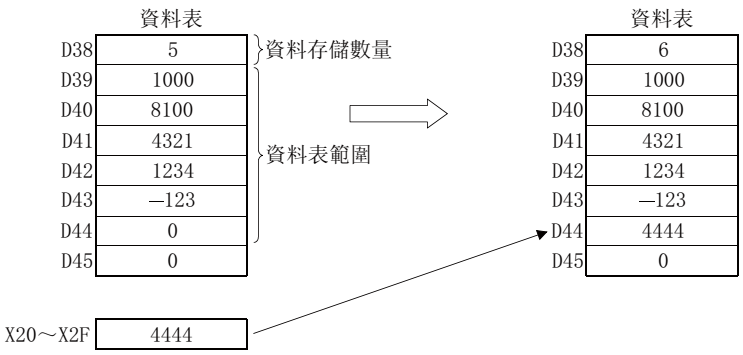
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD>=	D38	K6
3	OUT	Y60	
4	LD	X1B	
5	ANI	Y60	
6	FIFWP	K4X20	D38
9	END		

[動作]



7.7.2 FIFR、FIFRP

Basic | High performance | Process | Redundant | Universal | LCPU

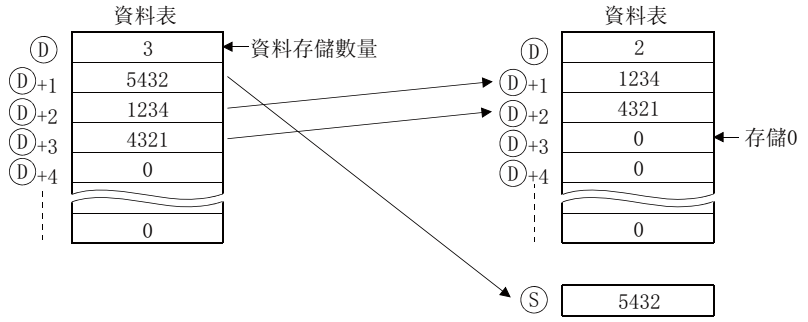
Ⓢ ：存儲從表中讀取的資料的軟元件的起始編號 (BIN16 位)。

Ⓓ ：表的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ	○	○		○				---	
Ⓓ	---	○		---				---	

功　能

(1) 將ⓓ中指定的表的最舊資料 (ⓓ+1) 存儲到Ⓢ中指定的軟元件中。
執行 FIFR 指令後資料表的資料逐個向前填充對齊。



(2) ⓓ中存儲的值爲 0 時，應由用戶設置互鎖以防止對其執行 FIFR 指令。
(參閱程式示例 (1))

出　錯

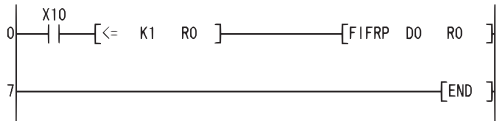
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	ⓓ中的值爲 0 的情況下執行了 FIFR 指令時。	○	○	○	○	○	○
4101	執行了 FIFR 指令時，資料表範圍超出了相應軟元件範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X10 變爲 ON 時，從 R0 ～ R7 的資料表中將 R1 的資料存儲到 D0 中的程式。

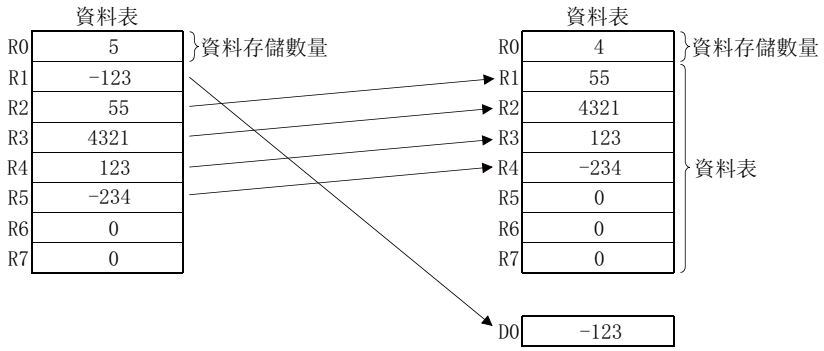
[梯形圖模式]



[列表模式]

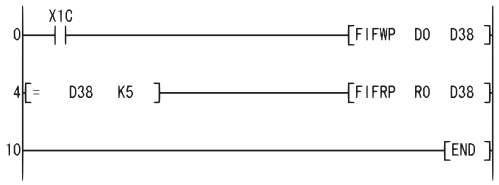
步	指令	軟元件	
0	LD	X10	
1	AND<=	K1	R0
4	FIFRP	D0	R0
7	END		

[動作]



(2) 以下爲 X1C 變爲 ON 時，將 D0 的資料存儲到 D38 ～ D43 的資料表中，資料存儲數量達到 5 時，將資料表 D39 的資料存儲到 R0 中的程式。

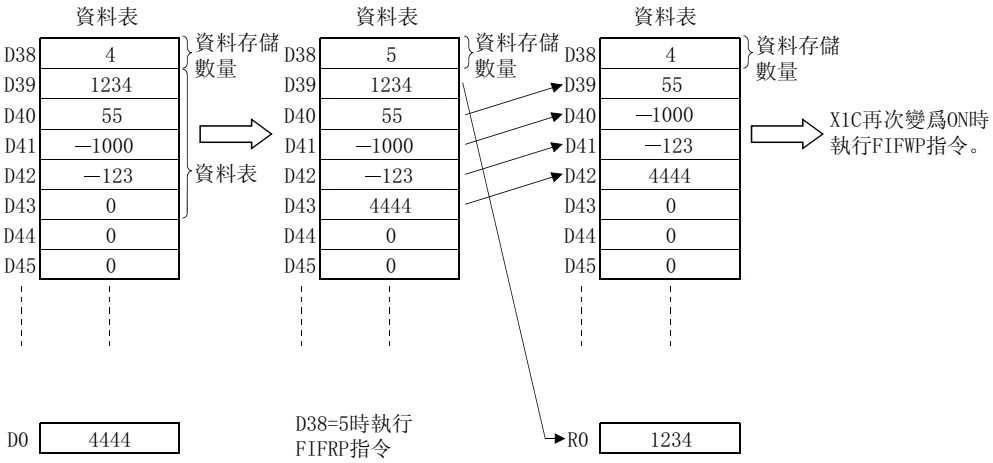
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X1C	
1	FIFWP	D0	D38
4	LD=	D38	K5
7	FIFRP	R0	D38
10	END		

[動作]



7.7.3 FP0P、FP0PP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：存儲從表中讀取的資料的軟元件的起始編號 (BIN16 位)。

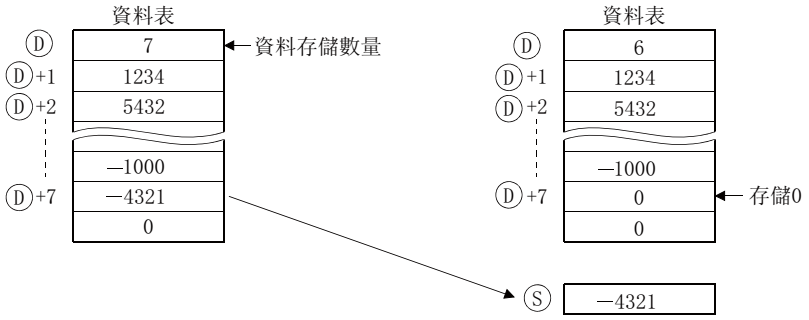
Ⓓ ：表的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ	○	○		○				---	
Ⓓ	---	○		---				---	

功　能

(1) 將Ⓓ中指定的表的最新資料存儲到Ⓢ中指定的軟元件中。

執行 FP0P 指令後，存儲通過 FP0P 指令讀取的資料的軟元件將變爲 0。



(2) ⓓ中存儲的值爲 0 時，應由用戶設置互鎖以防止對其執行 FP0P 指令。(參閱程式示例 (1))

出　錯

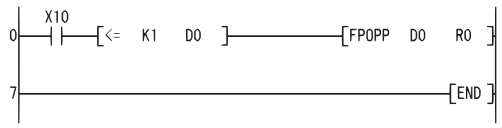
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	ⓓ中的值爲 0 的情況下執行了 FP0P 指令時。	○	○	○	○	○	○
4101	執行了 FP0P 指令時，資料表範圍超出了相應軟元件範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X10 變爲 ON 時，從 R0 ～ R7 的資料表中將最後存儲的資料存儲到 D0 中的程式。

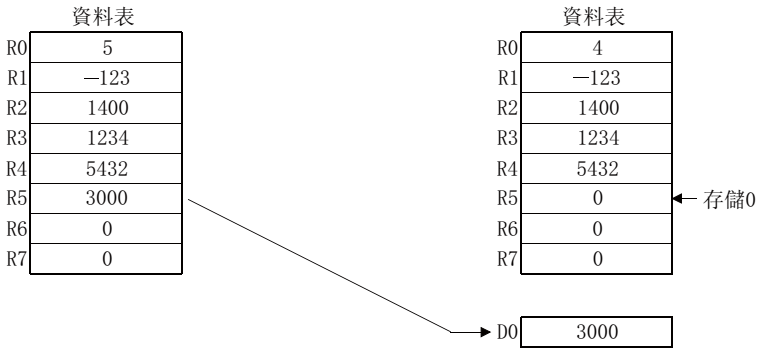
[梯形圖模式]



[列表模式]

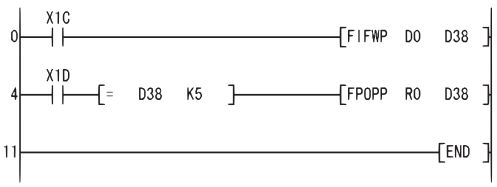
步	指令	軟元件	
0	LD	X10	
1	AND<=	K1	D0
4	FP0PP	D0	R0
7	END		

[動作]



(2) 以下爲 X1C 變爲 ON 時，將 D0 的資料存儲到 D38 ～ D43 的資料表中，資料存儲數量達到 5 時將 X1D 置爲 ON，將資料表中最後存儲的資料存儲到 R0 中的程式。

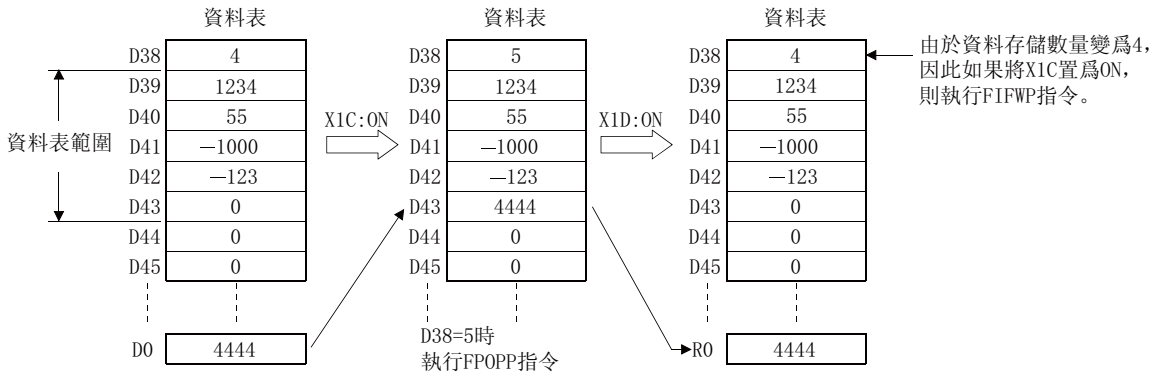
[梯形圖模式]



[列表模式]

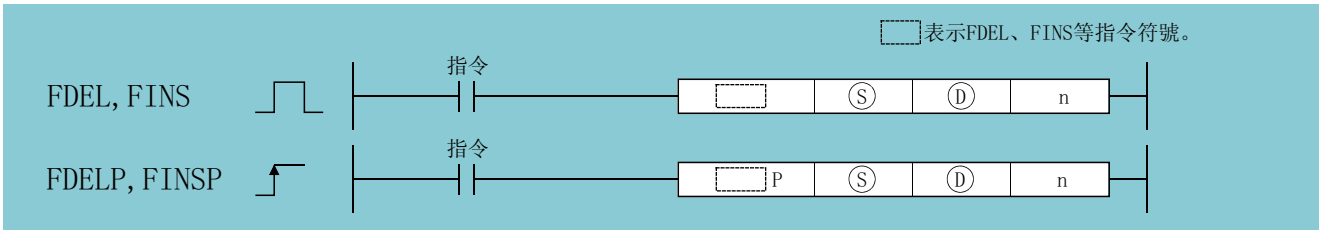
步	指令	軟元件	
0	LD	X1C	
1	FIFWP	D0	D38
4	LD	X1D	
5	AND=	D38	K5
8	FP0PP	R0	D38
11	END		

[動作]



7.7.4 FDEL、FDELP、FINS、FINSP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ：存儲插入資料的軟元件的起始編號 (BIN16 位)。
存儲刪除資料的軟元件的起始編號 (BIN16 位)。

Ⓓ：表的起始編號 (BIN16 位)。
n：進行插入 / 刪除的表位置 (BIN16 位)。

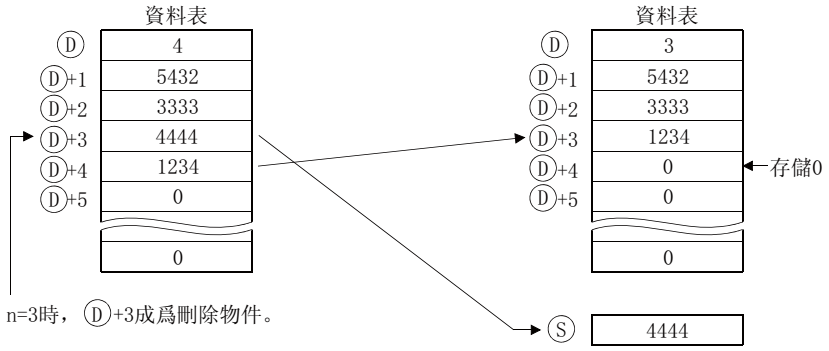
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○		○				---	---
Ⓓ	---	○		---				---	---
n	○	○		○				○	---

功　能

FDEL

將Ⓓ中指定的資料表的第 n 個資料刪除後，存儲到Ⓢ中指定的軟元件中。

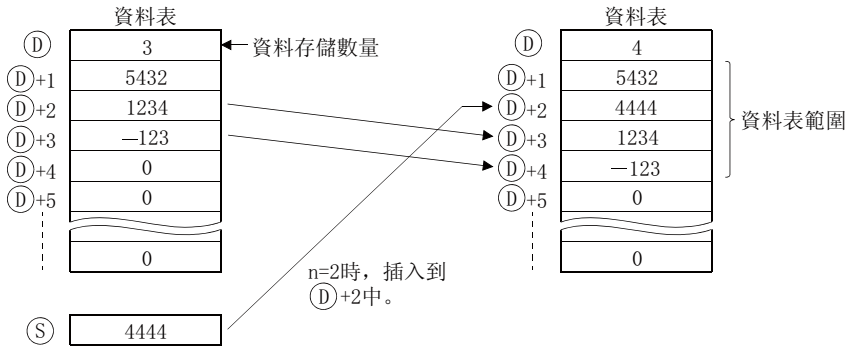
執行 FDEL 指令後，資料表的第 n+1 後面的資料將逐個向前填充對齊。



FINS

將Ⓢ中指定的 16 位元資料插入到Ⓓ中指定的資料表的第 n 號中。

執行 FINS 指令後，資料表的第 n 號開始的資料將逐個向下順延。



出　錯

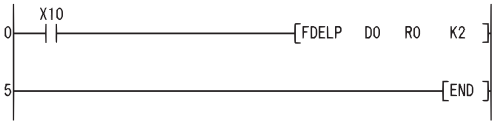
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	在 n=0 的情況下執行了 FDEL、FINS 指令時。 ⓓ 的值爲 0 的情況下執行了 FDEL 指令時。	○	○	○	○	○	○
4101	使用 FDEL 指令時，從ⓓ開始的第 n 號的位置大於資料存儲數量時。 使用 FDEL 指令時，從ⓓ開始的第 n 號的位置大於資料存儲數量 +1 時。 使用 FDEL、FINS 指令時，n 的值超出了ⓓ的表的軟元件範圍時。 執行了 FDEL、FINS 指令時，資料表範圍超出了相應軟元件範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X10 變爲 ON 時，從 R0 ～ R7 的資料表中將第 2 個資料刪除後，存儲到 D0 中的程式。

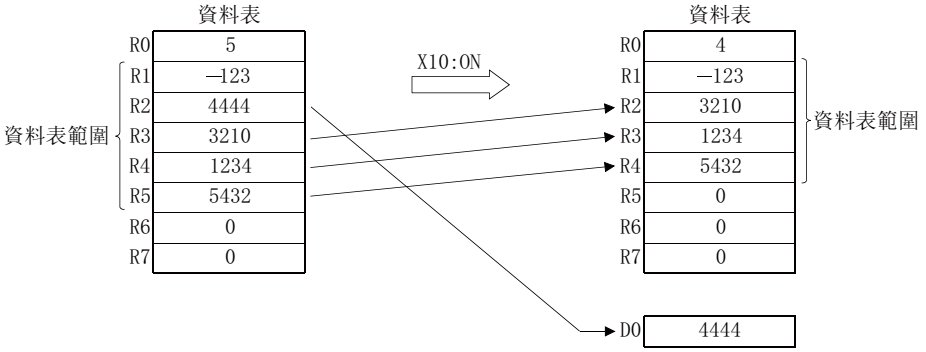
[梯形圖模式]



[列表模式]

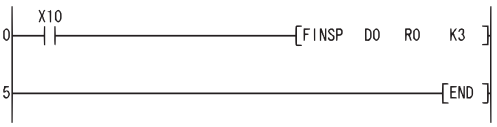
步	指令	軟元件		
0	LD	X10		
1	FDELP	D0	R0	K2
5	END			

[動作]



(2) 以下爲 X10 變爲 ON 時，將 D0 的資料插入到 R0 ～ R7 的資料表的第 3 號中的程式。

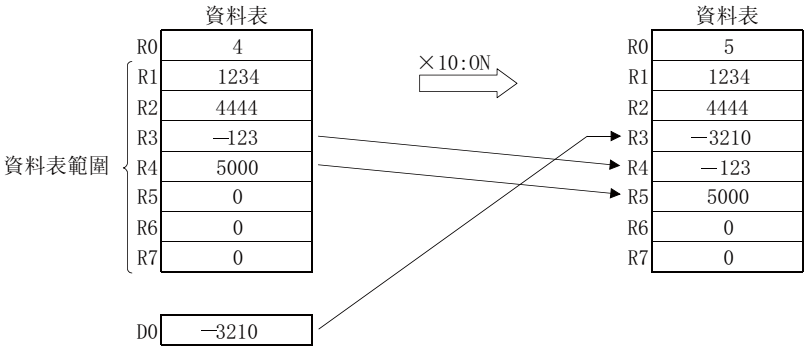
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X10		
1	FINSP	D0	R0	K3
5	END			

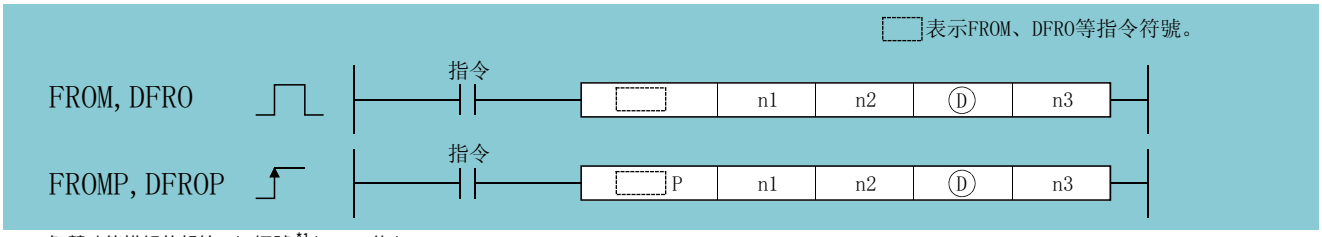
[動作]



7.8 緩衝記憶體訪問指令

7.8.1 FROM、FROMP、DFRO、DFROP

Basic | High performance | Process | Redundant | Universal | LCPU



n1 ： 智慧功能模組的起始 I/O 編號 *1(BIN16 位)
n2 ： 存儲讀取的資料的緩衝記憶體的起始位址 (BIN16 位)。
Ⓓ ： 存儲讀取的資料的軟元件起始編號 (BIN16/32 位)。
n3 ： 讀取的資料數 (BIN16 位)。

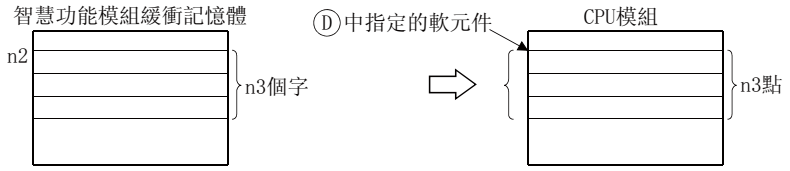
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他 U
	位	字		位	字				
n1		○				○			○
n2		○				○			---
Ⓓ		○				---			---
n3		○				○			---

*1: 通過將起始 I/O 編號以 4 位的 16 進制數表示時的高 3 位進行指定。

功 能

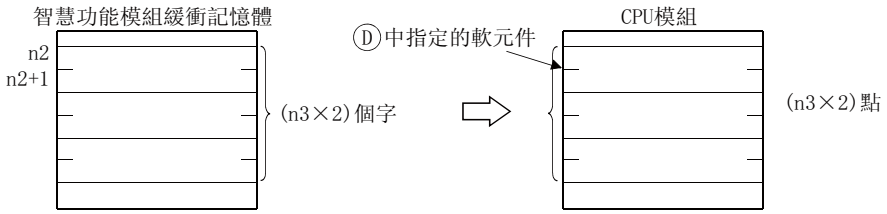
FROM

(1) 從 n1 中指定的智慧功能模組的 n2 中指定的緩衝記憶體位址開始，讀取 n3 個字的資料後，存儲到Ⓓ中指定的軟元件後面。



DFRO

(1) 從 n1 中指定的智慧功能模組的 n2 中指定的緩衝記憶體地址開始，讀取 (n3 × 2) 個字的資料後，存儲到Ⓓ中指定的軟元件後面。



要 點

智慧功能模組的資料讀取也可使用智慧功能模組軟元件進行。
關於智慧功能模組軟元件，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

出　錯

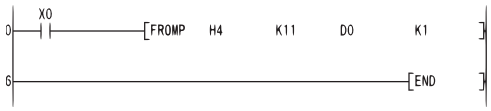
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
1402	執行指令時檢測出智慧功能模組異常時。	○	○	○	○	○	○
1412	執行指令時無法與智慧功能模組進行通信時。	○	○	○	○	○	○
2110	n1 中指定的 I/O 號不是智慧功能模組時。	○	○	○	○	○	○
4101	從ⓓ中指定的軟元件開始的 n3 點 (DFRO 時爲 2 × n3 點) 超出了指定軟元件範圍時。 n2 中指定的地址超出緩衝記憶體範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，在 I/O 號爲 040 ～ 05F 的 Q68ADV 中，將 CH1 的數位值讀取到 D0 中的程式。(從緩衝記憶體的位址 10 開始讀取 1 個字的資料。)

[梯形圖模式]

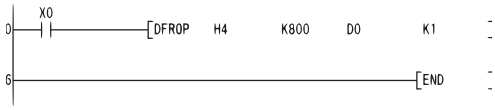


[列表模式]

步	指令	軟元件			
0	LD	X0			
1	FROMP	H4	K11	D0	K1
6	END				

(2) 以下爲 X0 變爲 ON 時，在 I/O 號爲 040 ～ 05F 的 QD75P4 中，將軸 1 的當前值讀取到 D0、D1 中的程式。(從緩衝記憶體的位址 800，801 開始讀取 2 個字的資料。)

[梯形圖模式]



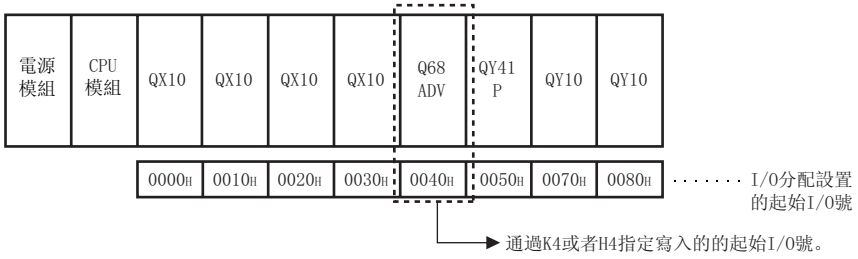
[列表模式]

步	指令	軟元件			
0	LD	X0			
1	DFROP	H4	K800	D0	K1
6	END				

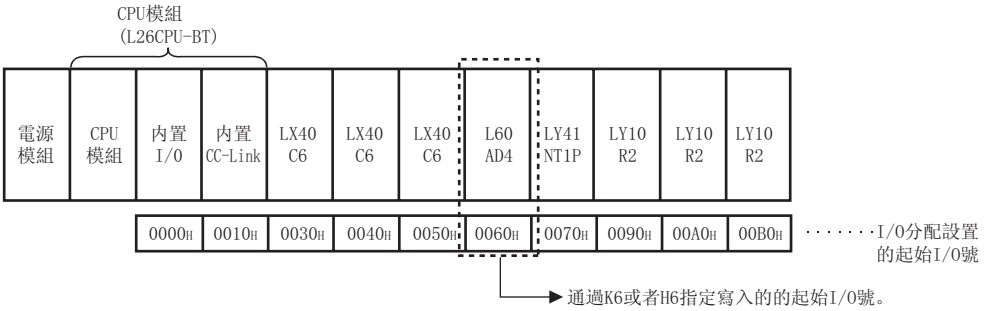
備 註

1. 當以 16 進制 4 位數表示安裝智慧功能模組的插槽的起始 I/O 號時，n1 的值由高 3 位指定。

QCPU



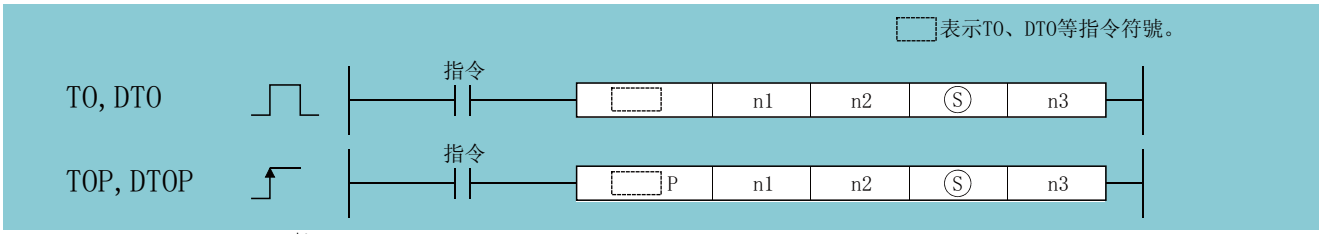
LCPU



2. 在 QCPU、LCPU 中建立了 FROM/DFRO 指令的自動互鎖。

7.8.2 TO、TOP、DTO、DTOP

Basic | High performance | Process | Redundant | Universal | LCPU



n1 ： 智慧功能模組的起始 I/O 編號 *1(BIN16 位)。
n2 ： 用於資料寫入的起始位址 (BIN16 位)。
Ⓢ ： 寫入資料或者存儲寫入資料的軟元件的起始編號 (BIN16/32 位)。
n3 ： 寫入的資料數 (BIN16 位)。

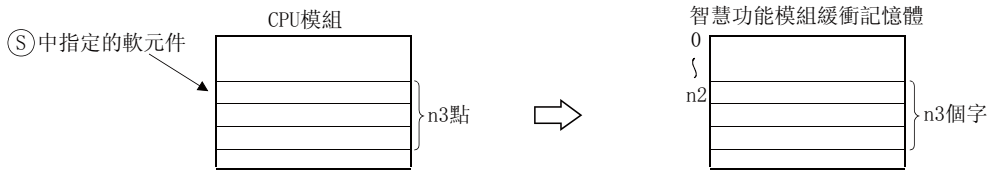
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他 U
	位	字		位	字				
n1		○			○			○	○
n2		○			○			○	---
Ⓢ		○			---			○	---
n3		○			○			○	---

*1: 通過將起始 I/O 編號以 4 位的 16 進制數表示時的高 3 位進行指定。

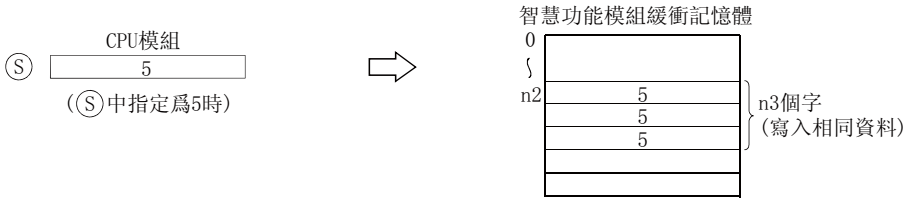
功 能

TO

將從Ⓢ中指定的軟元件開始的 n3 點的資料，寫入到 n1 中指定的智慧功能模組的 n2 中指定的緩衝記憶體地址的後面。

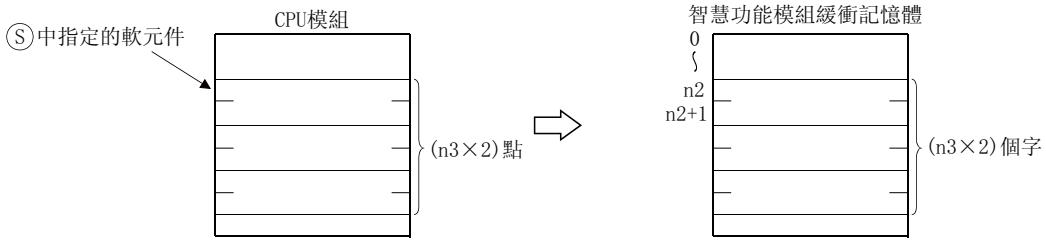


在Ⓢ中指定了常數時，將相同資料 (Ⓢ中指定的值) 寫入到從指定緩衝記憶體開始的 n3 個字中。(Ⓢ中的可指定範圍爲 -32768 ～ 32767 或者 0_H ～ $FFFF_H$。)



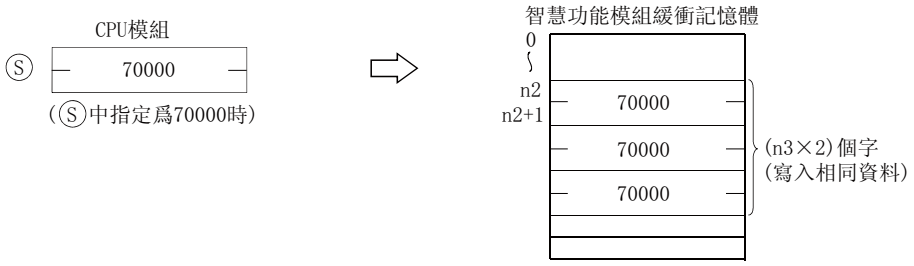
DTO

將從Ⓢ中指定的軟元件開始的 (n3 × 2) 點的資料，寫入到 n1 中指定的智慧功能模組的 n2 中指定的緩衝記憶體地址的後面。



在Ⓢ中指定了常數時，將相同資料 (Ⓢ中指定的值) 寫入到從指定緩衝記憶體開始的 (n3 × 2) 個字中。

(Ⓢ中的可指定範圍爲 -2147483648 ～ 2147483647 或者 0_H ～ $FFFFFFFF_H$。)



要 點

智慧功能模組的資料寫入也可使用智慧功能模組軟元件進行。
關於智慧功能模組軟元件，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

出　錯

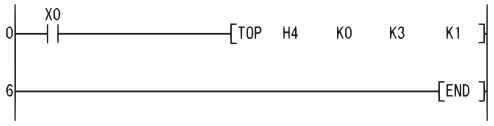
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
1402	執行指令時檢測出智慧功能模組異常時。	○	○	○	○	○	○
1412	執行指令時無法與智慧功能模組進行通信時。	○	○	○	○	○	○
2110	n1 中指定的 I/O 號不是智慧功能模組時。	○	○	○	○	○	○
4101	從Ⓢ中指定的軟元件開始的 n3 點 (DTO 時爲 2 × n3 點) 超出了指定軟元件範圍時。 n2 中指定的地址超出緩衝記憶體範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，在 I/O 號爲 040 ～ 04F 的 D68ADV 中，將 CH1 及 CH2 設置爲 “ 禁止 A/D 轉換 ” 的程式。(在緩衝記憶體的地址 0 中寫入 “ 3 ”。)

[梯形圖模式]

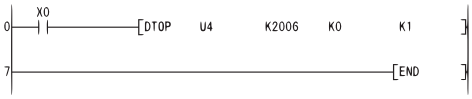


[列表模式]

步	指令	軟元件
0	LD	X0
1	TOP	H4 K0 K3 K1
6	END	

(2) 以下爲 X0 變爲 ON 時，在 I/O 號爲 040 ～ 05F 的 QD75P4 中，將軸 1 的定位地址 / 移動量置爲 0 的程式。(緩衝記憶體的地址 2006、2007 中寫入 0。)

[梯形圖模式]



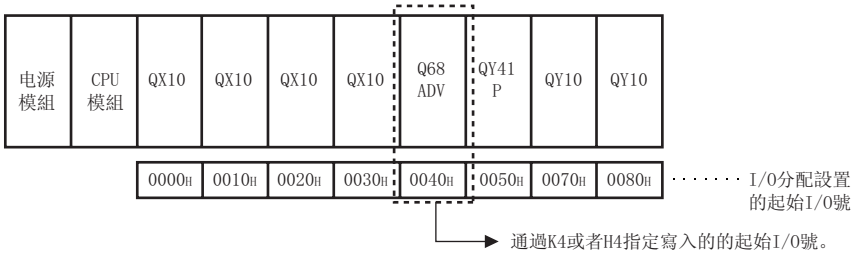
[列表模式]

步	指令	軟元件
0	LD	X0
1	DTOP	U4 K2006 K0 K1
7	END	

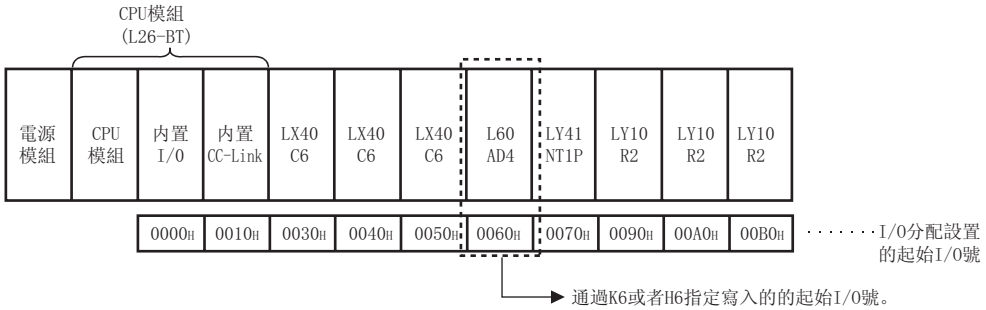
備 註

1. 當以 16 進制 4 位數表示安裝智慧功能模組的起始 I/O 號時，n1 的值由高 3 位指定。

QCPU



LCPU



2. 在 QCPU、LCPU 中建立了 TO/DTO 指令的自動互鎖。

7.9 顯示指令

7.9.1 PR

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：ASCII 碼或者存儲 ASCII 碼的軟元件的起始編號（字串）。

Ⓓ ：輸出 ASCII 碼的輸出模組的起始編號（位元）。

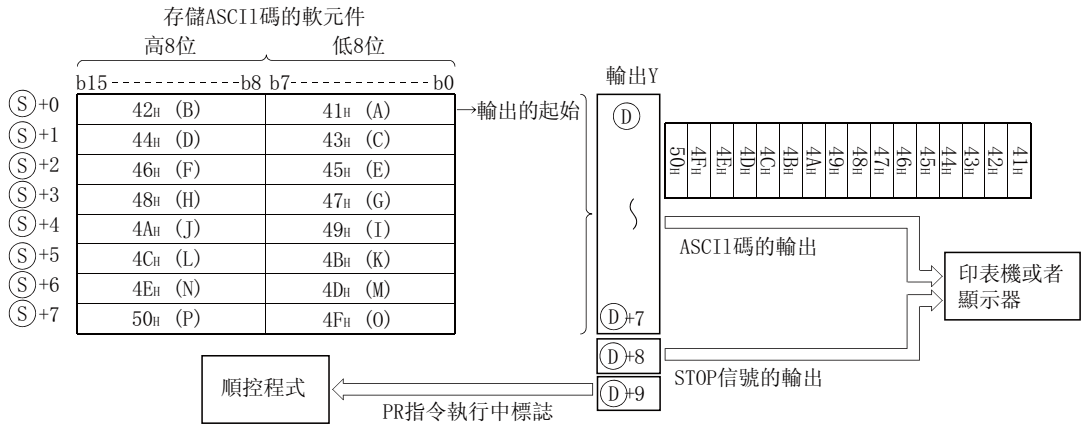
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	△ *1		---			○	○	---
Ⓓ	○（僅 Y）	---		---			○	---	---

*1: 局部軟元件以及各程式中設置的文件寄存器不能使用。

功 能

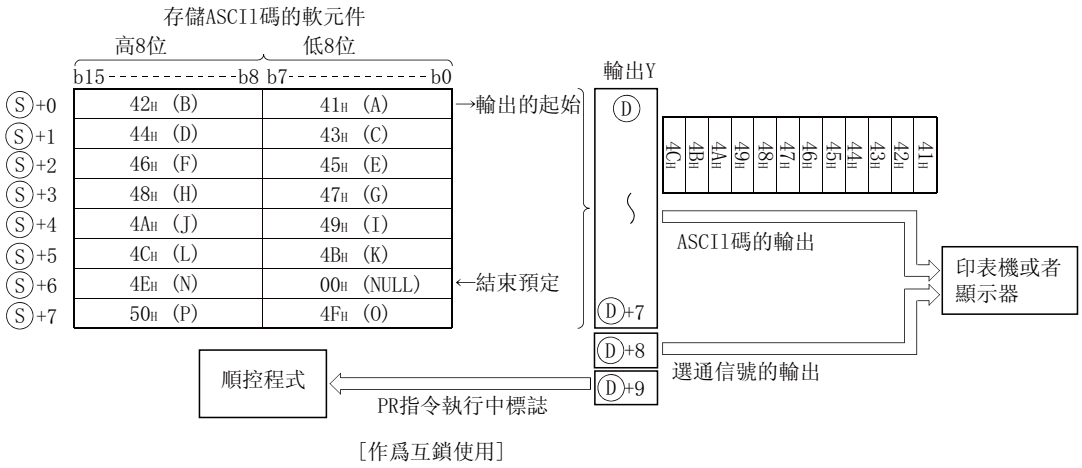
(1) 將Ⓢ中指定的 ASCII 碼或者存儲在軟元件號後面的 ASCII 碼輸出到Ⓓ中指定的輸出模組中。
輸出的字元數根據 SM701（輸出字元數切換）的 ON/OFF 狀態而不同。

(a) SM701 爲 ON 時，從Ⓢ中指定的軟元件開始的 8 點（16 個字元）將成爲輸出物件。



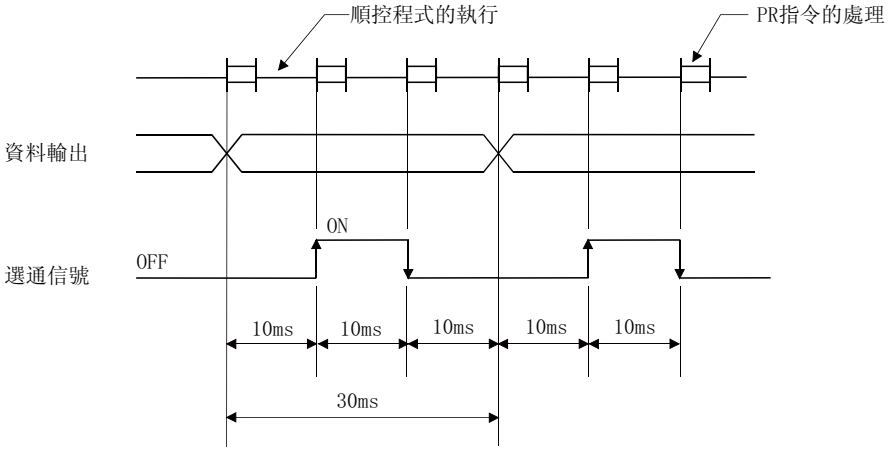
［作爲互鎖使用］

(b) SM701 爲 OFF 時，從Ⓢ中指定的軟元件開始至 00_H 碼爲止將成爲輸出物件。



(2) 輸出模組中使用的點數爲Ⓓ中指定的 Y 位址號開始的 10 點。

(3) 來自於輸出模組的輸出信號以每 30ms 一個字元的速率進行發送。
因此，完成 (n) 個指定字元數的傳送所需要的時間爲 30ms × n(ms)。
PR 指令通過 10ms 中斷，執行資料輸出→選通信號 ON →選通信號 OFF。在以上處理之間的時間，繼續執行其他指令。



(4) 除通過輸出模組輸出 ASCII 碼以外，還通過Ⓓ+8 的軟元件輸出選通信號 (10ms ON，20ms OFF)。

(5) 執行 PR 指令後，PR 指令執行中標誌 (Ⓓ+9 的軟元件) 將變爲 ON 狀態，直至指定字元數的 ASCII 碼發送完畢。

(6) 雖然可以多次使用 PR、PRC 指令，但應通過 PR 指令執行中標誌 (Ⓓ+9 的軟元件) 採取互鎖，以防止二個指令同時變爲 ON。

(7) 如果在 ASCII 碼的輸出過程中更改了存儲 ASCII 碼的軟元件的內容，則將輸出更改後的資料。

出　錯

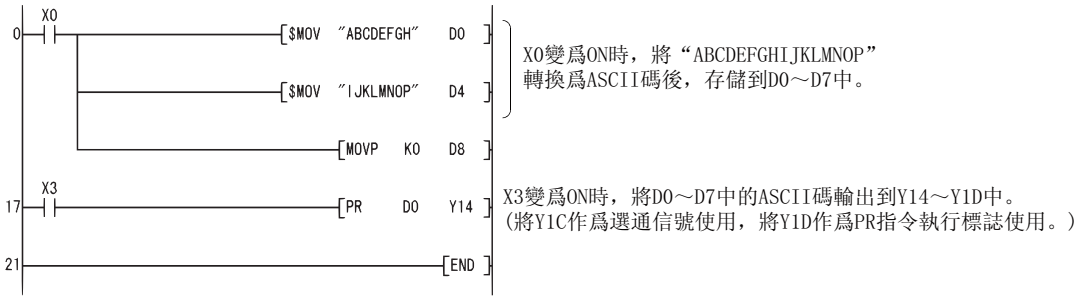
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	SM701 爲 OFF 時，Ⓢ 中指定的軟元件的範圍內沒有 00_H 碼時。	---	○	○	---	---	---

程式示例

(1) 以下爲 X0 變爲 ON 時，將 “ ABCDEFGHIJKLMNOP ” 轉換爲 ASCII 碼後存儲到 D0 ～ D7 中，X3 變爲 ON 時，將 D0 ～ D7 中的 ASCII 碼輸出到 Y14 ～ Y1D 中的程式。(將 Y1C 作爲選通信號使用，將 Y1D 作爲 PR 指令執行標誌使用。)

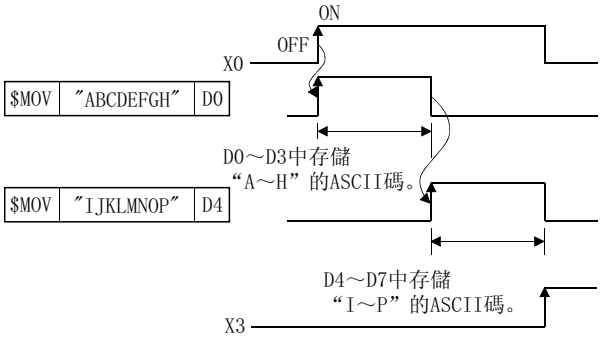
[梯形圖模式]

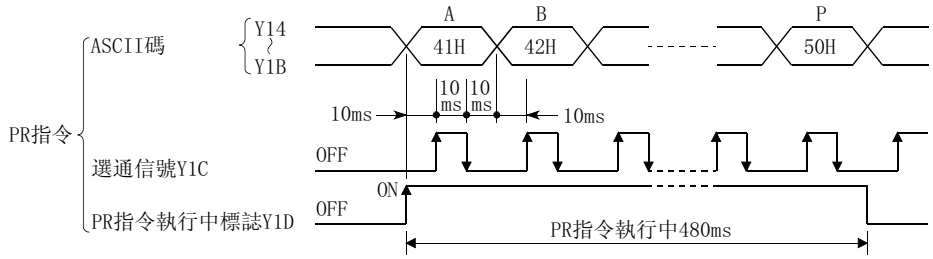


[列表模式]

步	指令	軟元件	
0	LD	X0	
1	$MOV	"ABCDEFGH"	D0
8	$MOV	"IJKLMNOP"	D4
15	MOVP	K0	D8
17	LD	X3	
18	PR	D0	Y14
21	END		

[時序圖]





7.9.2 PRC

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：進行注釋列印的軟元件的起始編號 (軟元件名)。

Ⓓ ：進行注釋輸出的輸出模組的起始編號 (位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				P,I,J,U
Ⓢ	○	○		○			---	---	○
Ⓓ	○ (僅 Y)	---		---			---	---	---

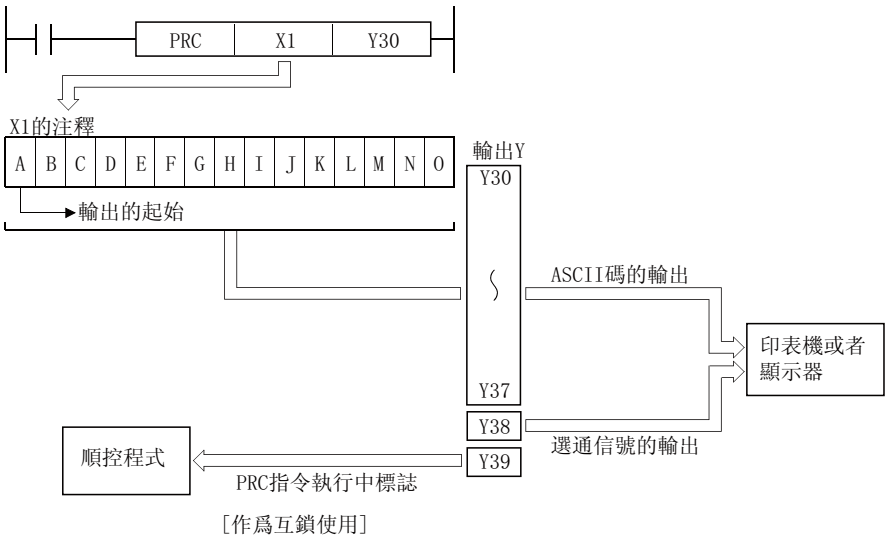
功　能

(1) 將Ⓢ中指定的軟元件注釋 (ASCII 碼) 輸出到Ⓓ中指定的輸出模組中。

輸出的字元數根據 SM701 的 ON/OFF 狀態而不同。

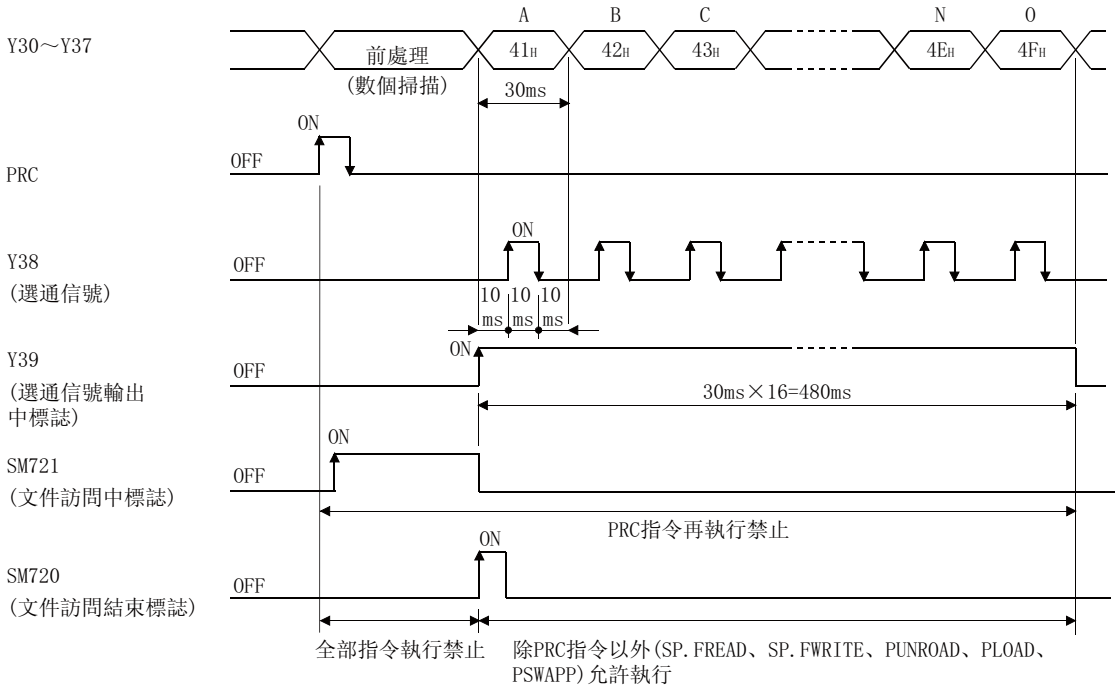
- SM701 爲 OFF 時　：32 個字元的注釋
- SM701 爲 ON 時　：注釋的高位 16 個字元

輸出模組中使用的點數爲Ⓓ中指定的 Y 位址號開始的 10 點。

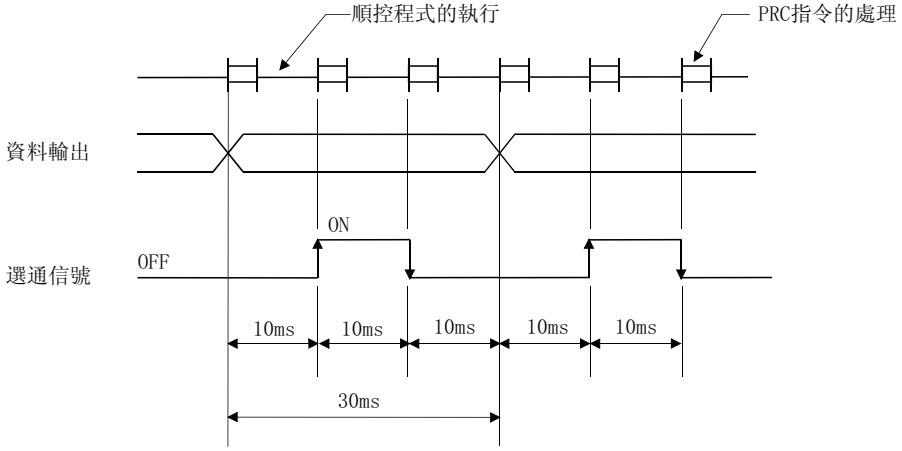


[作爲互鎖使用]

[時序圖]



(2) 來自於輸出模組的輸出信號以每 30ms 一個字元的速率進行發送。
因此，完成指定字元數的傳送所需要的時間爲 30ms × n(ms)。
PRC 指令通過 10ms 中斷，執行資料輸出→選通信號 ON →選通信號 OFF。在以上處理之間的時間，繼續執行其他指令。



(3) 除通過輸出模組輸出 ASCII 碼以外，還通過Ⓓ+8 的軟元件輸出選通信號 (10ms ON，20ms OFF)。

(4) 執行 PRC 指令後，PRC 指令執行中標誌 (Ⓓ+9 的軟元件) 將變爲 ON 狀態，直至指定字元數的 ASCII 碼發送完畢。

(5) 雖然可以多次使用 PRC 指令，但應通過 PRC 指令執行中標誌 (Ⓓ+9 的軟元件) 採取互鎖，以防止指令同時變爲 ON。

(6) 如果未將注釋登錄到Ⓢ中指定的軟元件中，將不執行處理。

(7) 讀取注釋時，指令結束後 SM720 將 ON 1 個掃描。
此外，在指令執行過程中 SM721 將變爲 ON。
在 SM721 處於 ON 狀態時，不能執行 PRC 指令。即使執行也將變爲無處理。

要 點

1.PRC 指令中使用的軟元件注釋使用存儲在標準 ROM 或存儲卡中的注釋文件。
2.PRC 指令中使用的注釋文件是在可程式控制器參數的 “ 可程式控制器文件設置 ” 中進行設置。
如果未在可程式控制器文件設置對所使用的注釋文件進行設置，將無法通過 PRC 指令進行軟元件注釋輸出。
3.不要在中斷程式中執行 PRC 指令。
如果執行將會導致誤動作。

出 錯

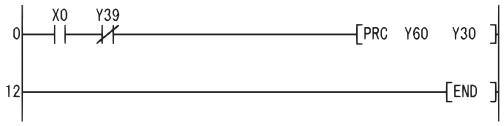
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	在對軟元件注釋進行 RUN 中寫入的過程中，執行了 PRC 指令時。	---	○	○	---	---	---

程式示例

(1) 以下爲 X0 變爲 ON 時，將 Y60 的注釋輸出到 Y30 ～ Y39 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0
1	ANI	Y39
2	PRC	Y60 Y30
5		<:y60="aa">
12	END	

7.9.3 LEDR

Basic (×) | High performance | Process | Redundant | Universal | LCPU

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功 能

對 CPU 模組的報警器顯示及繼續運行型自診斷出錯顯示進行重定。

在一次指令執行中，只能對出錯顯示或者報警器二者之一進行重定。

(1) 發生自診斷出錯時的動作

(a) 發生了繼續運行型自診斷出錯時

在 CPU 模組處於繼續運行型自診斷出錯的出錯顯示時，如果執行了 LEDR 指令，則只對 CPU 模組前面的 “ ERROR/ERR. ” LED 或者出錯顯示進行重定。此時 SM0、SM1、SD0 的內容不能被重定，因此應由用戶通過程式進行重定。
此外，此時出錯顯示的出錯原因的優先順序高於報警器，因此不能執行報警器的重定的相關處理。

(b) 發生了電池出錯時

如果在更換了電池後執行了 LEDR 指令，則對 CPU 模組前面的 “ BAT.ARM/BAT. ” LED 或者出錯顯示進行重定。
此時 SM51 也將變爲 OFF。

(2) 報警器 (F) 處於 ON 狀態時的動作

(a) 對於前面無 LED 顯示器的 CPU 模組

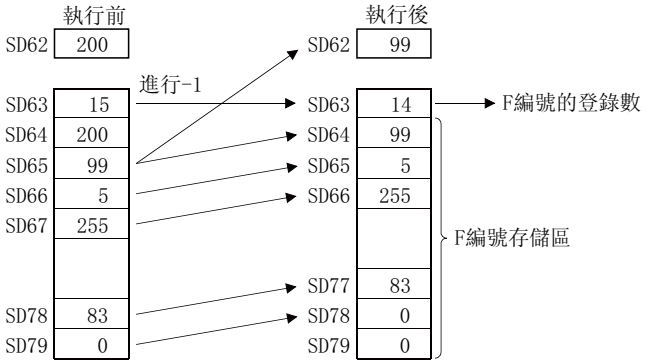
如果執行了 LEDR 指令，將執行以下動作：

1) “ USER ” LED 閃爍、熄燈。

2) 對 SD62、SD64 中存儲的報警器 (F) 進行重定後，SD65 ～ SD79 的 F 編號向前填充對齊。

3) 將 SD64 中新存儲的資料傳送至 SD62 中。

4) 對 SD63 的資料進行 -1。但是 SD63 爲 0 時保持爲 0 不變。



(b) 對於前面有 LED 顯示器的 CPU 模組

如果執行了 LEDR 指令，將執行以下動作：

1) 對 CPU 模組顯示的 F 編號進行重定。

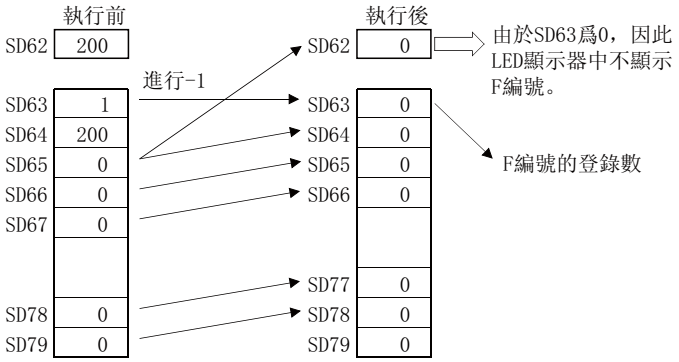
2) “ USER ” LED 閃爍、熄燈。

3) 對 SD62、SD64 中存儲的報警器 (F) 進行重定後，SD65 ～ SD79 的 F 編號向前填充對齊。

4) 將 SD64 中新存儲的資料傳送至 SD62 中。

5) 對 SD63 的資料進行 -1。但是 SD63 爲 0 時保持爲 0 不變。

6) 在 LED 顯示器上顯示 SD62 中存儲的 F 編號。但是，SD63 爲 0 時不顯示任何內容。



備 註

1. 特殊寄存器 SD207 ～ SD209 中設置的原因編號的內容及優先順序的預設值如下所示。

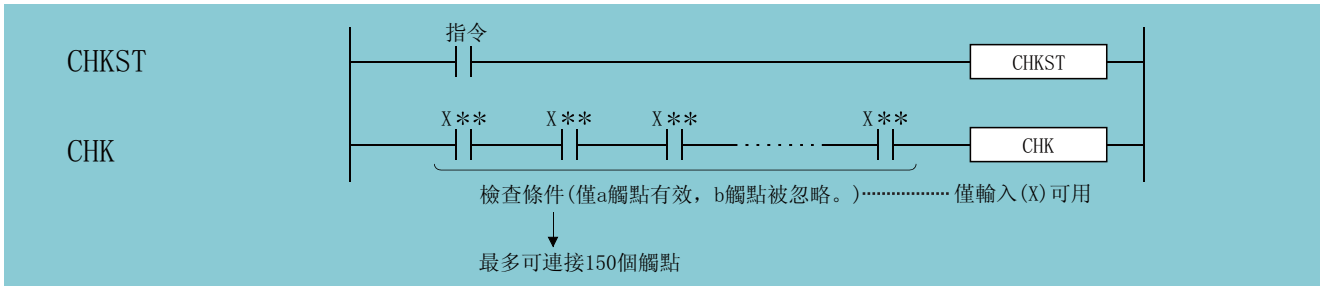
優先順序	原因編號 (16 進制數)	內容	備註
1	1	AC DOWN SINGLE PS. DOWN SINGLE PS. ERROR	電源斷開 冗餘基板電源電壓過低 (僅 QCPU) 冗餘電源模組異常 (僅 QCPU)
2	2	UNIT VERIFY ERR. FUSE BREAK OFF SP. UNIT ERROR SP.UNIT DOWN	I/O 模組校驗出錯 (僅 QCPU) 保險絲熔斷 (僅 QCPU) 特殊功能模組校驗出錯 (僅 QCPU) 智慧功能模組校驗出錯 智慧功能模組異常 (僅 LCPU)
3	3	OPERATION ERROR LINK PARA. ERROR SFCP OPE. ERROR SFCP EXE. ERROR REMOTE PASS.FAIL SNTP OPE.ERROR	運算出錯 鏈結參數出錯 (僅 QCPU) SFC 指令運算出錯 (僅 QCPU) SFC 程式執行出錯 (僅 QCPU) 遠端口令出錯 (僅 LCPU) SNTP 出錯 (僅 LCPU)
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 FLASH ROM ERROR	存儲卡操作出錯 文件訪問出錯 擴展指令出錯 (僅 QCPU) 運行狀態、開關不匹配 (僅 QCPU) 當前模式下功能無法執行 (僅 QCPU) 熱備資料通信出錯 (僅 QCPU) 熱備容量超出出錯 (僅 QCPU) 熱備電纜未連接．故障 (僅 QCPU) 快閃卡訪問次數溢出出錯 (僅 LCPU)
5	5	PRG.TIME OVER	恒定掃描設置時間超時 低速執行監視時間超時 (僅 QCPU)
6	6	CHK 指令	---
7	7	報警器	---
8	8	LED 指令	---
9	9	BATTERY ERR.	---
10	A	時鐘資料	---
11	B	CAN'T SWITCH STANDBY SYS. DOWN MEM. COPY EXE.	系統切換出錯 (僅 QCPU) 待機系統未啓動 / 停止出錯 (僅 QCPU) 實施記憶體複製功能 (僅 QCPU)
12	C	DISPLAY ERROR	顯示模組出錯 (僅 LCPU)

2. 如果將報警器設置爲最高優先順序，可通過 LEDR 指令對報警器進行優先重定。(對於基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU)

7.10 調試和故障診斷指令

7.10.1 CHKST、CHK

Basic | High performance | Process | Redundant | Universal | LCPU



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

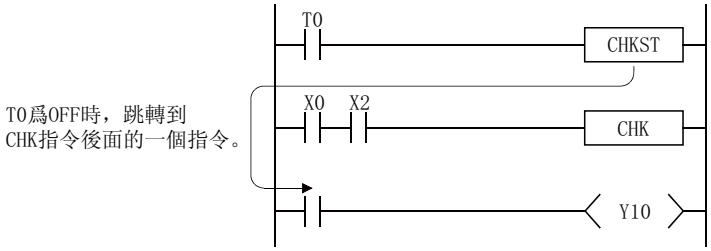
功 能

CHKST

CHKST 指令是啓動 CHK 指令的指令。

如果 CHKST 指令的執行指令爲 OFF，則跳轉到 CHK 指令後面的一個指令。

如果 CHKST 指令的執行指令爲 ON，則執行 CHK 指令。



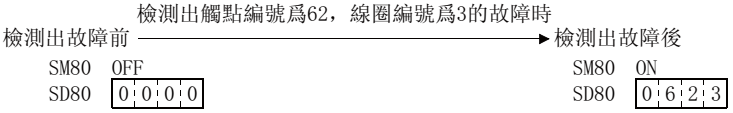
CHK

(1) CHK 指令是用於對下頁所示的進行往復運動的系統的故障內容進行確認的指令。

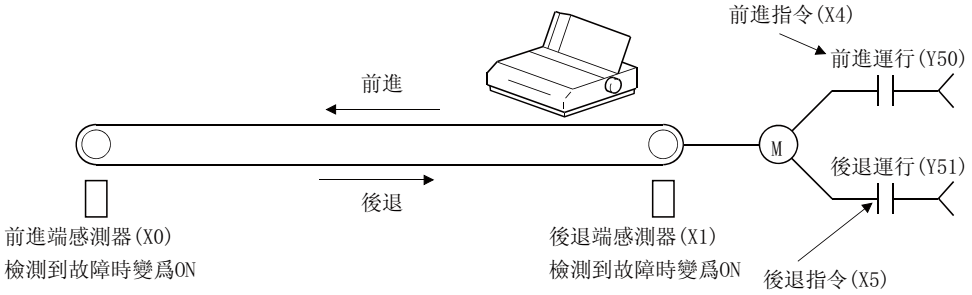
(a) 當執行 CHK 指令時，通過指定的檢查條件進行故障診斷檢查，如果檢測出故障，則 SM80 變爲 ON，且以 BCD 值將故障編號存儲到 SD80 中。

此外，檢測出故障時將發生出錯代碼爲 9010 的出錯。

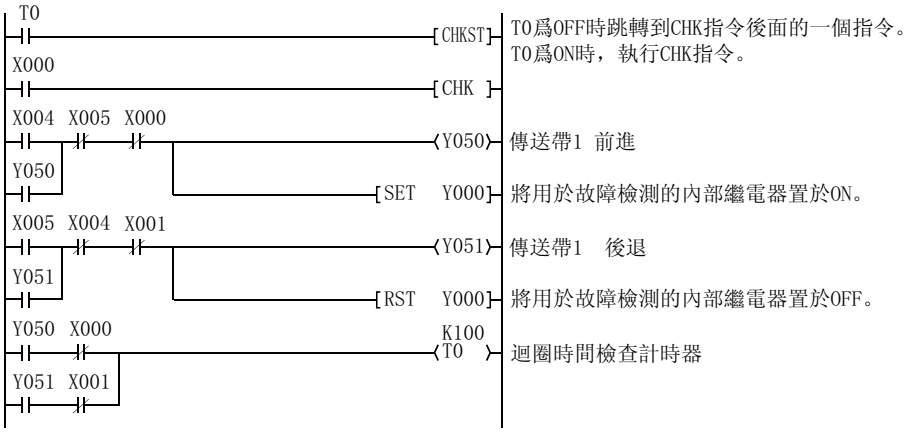
檢測出故障的觸點編號 (參閱 426 頁 7.10.1 項 (3)) 將被存儲到 SD80 的高 3 位元中，檢測出故障的線圈編號 (參閱 426 頁 7.10.1 項 (2)) 將被存儲到 SD80 的低 1 位中。



(b) 位於 CHK 指令之前的觸點指令不是用於對 CHK 指令的執行進行控制，而是用於進行檢查條件設置。



(c) 對上圖的系統進行迴圈時間超時檢查時，創建如下圖所示的梯形圖。



(d) 創建使用 CHK 指令的梯形圖時的注意事項如下所示：

1) 前進端感測器與後退端感測器的觸點編號 (X□) 必須爲連續編號。此外，前進端感測器的觸點編號 (X) 應該小於後退端感測器的觸點編號。

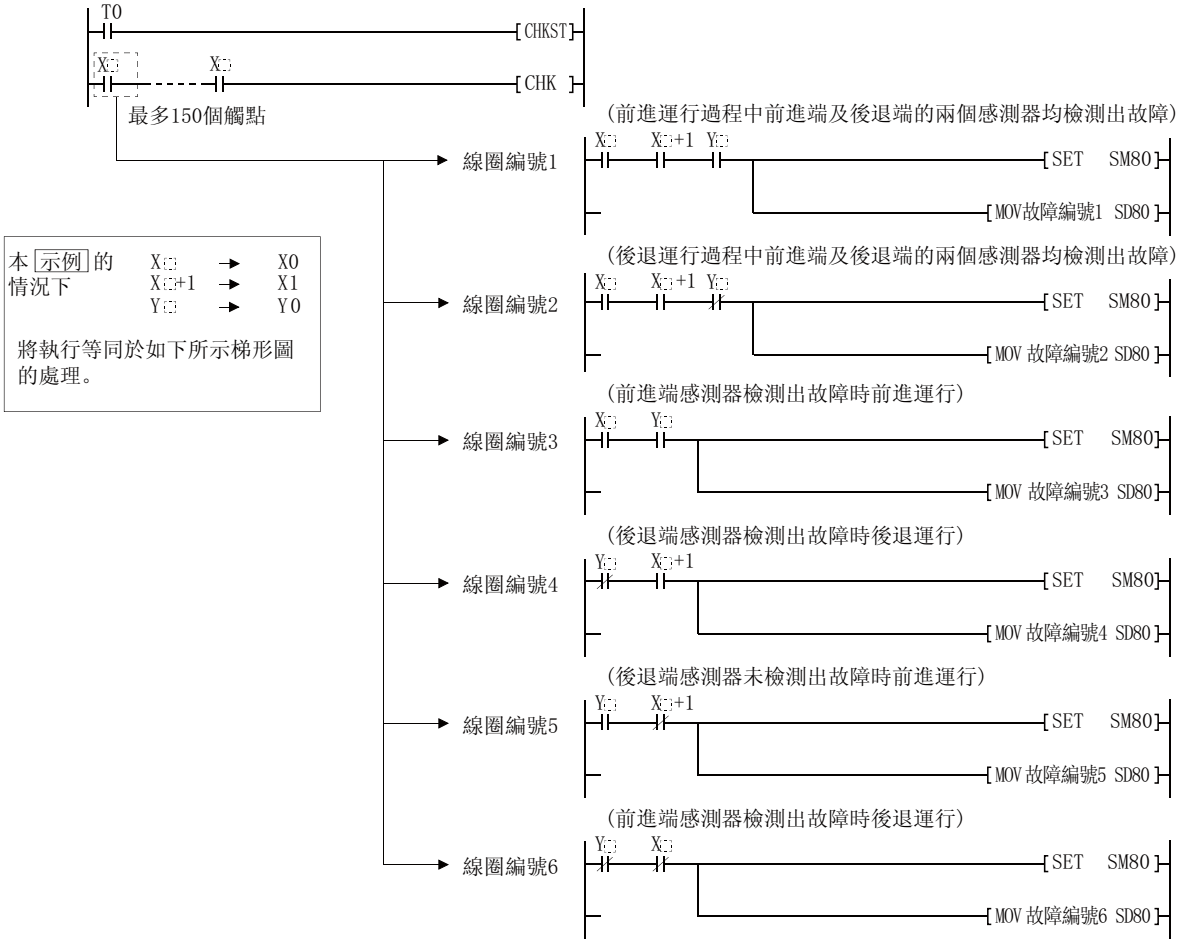
2) 對與前進端感測器觸點編號 (X□) 與相同編號的輸出 (Y□)*1 應進行如下所示的控制：

進行前進運行時 ..... 變爲 ON

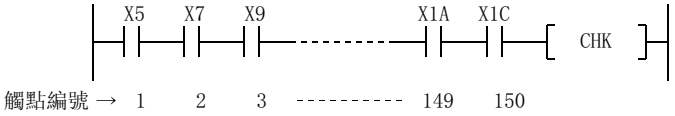
進行後退運行時 ..... 變爲 OFF

*1: 輸出 (Y□) 被作爲內部繼電器使用，不能對外部進行輸出。

(2) CHK 指令根據指定的觸點，執行等同於如下所示梯形圖的處理。



(3) 檢測出故障時的觸點編號從左側的縱母線開始按順序被分配爲 1 ～ 150。

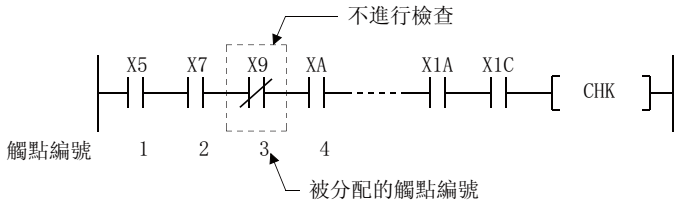


(4) 執行 CHK 指令時，應預先對 SM80、SD80 進行重定。
此外，執行 CHK 指令後，如果未對 SM80、SD80 進行重定，將不能再次執行 CHK 指令。
(SM80、SD80 的內容在用戶執行重定之前將被保持。)

(5) 在 CHK 指令的前面需要設置 CHKST 指令。
在 CHK 指令與 CHKST 指令之間如果存在有除 LD、LDI、AND、ANI 指令以外的指令，將會變爲出錯狀態。
(出錯代碼：4235)

(6) CHK 指令可以被寫入到程式的任意步中。
但是，使用 CHK 指令的數量有如下限制。
· 在所有正被執行的程式文件中最多可在兩個位置使用 CHK 指令。
· 在一個程式文件中只能在一個位置使用 CHK 指令。
如果 CHK 指令的使用超出上述限制，將會變爲出錯狀態。
(出錯代碼：4235)

(7) CHK 指令的前面應通過 LD、AND 指令設置檢查條件。
不能通過其他觸點指令設置檢查條件。
通過 LDI、ANI 指令設置了檢查條件時，不執行與檢查條件相關的處理。
但是，LDI、ANI 指令也將被分配檢測出故障時的觸點編號。



(8) 故障檢測方法根據 SM701 的 ON/OFF 狀態而有所不同。
(a) SM701 爲 OFF 時，按觸點順序進行線圈編號 1 ～ 6 的檢查。
執行 CHK 指令時，對觸點編號 1 的線圈編號 1 ～ 6 按編號順序進行檢查。觸點編號 1 的線圈編號 6 檢查完畢後，對觸點編號 2 的線圈編號 1 ～ 6 按編號順序進行檢查。
觸點編號 n 的線圈編號 6 檢查完畢後，結束 CHK 指令。
(b) SM701 爲 ON 時，按線圈順序進行觸點編號 1 ～ n 的檢查。
執行 CHK 指令時，對線圈編號 1 的梯形圖按觸點編號 1 ～ n 的編號順序進行檢查。線圈編號 1 的梯形圖的觸點編號 n 檢查完畢後，對線圈編號 2 的梯形圖按觸點編號 1 ～ n 的編號順序進行檢查。
線圈編號 6 的觸點編號 n 檢查完畢後，結束 CHK 指令。

(9) 檢測出多個故障時，最先檢測出的故障編號將被存儲。
後檢測出的故障編號將被忽略。

(10) CHK 指令不能用於低速執行型程式。
如果將包含有 CHK 指令的程式文件設置到低速執行型程式中，將變爲運算出錯狀態，CPU 模組運算將停止。

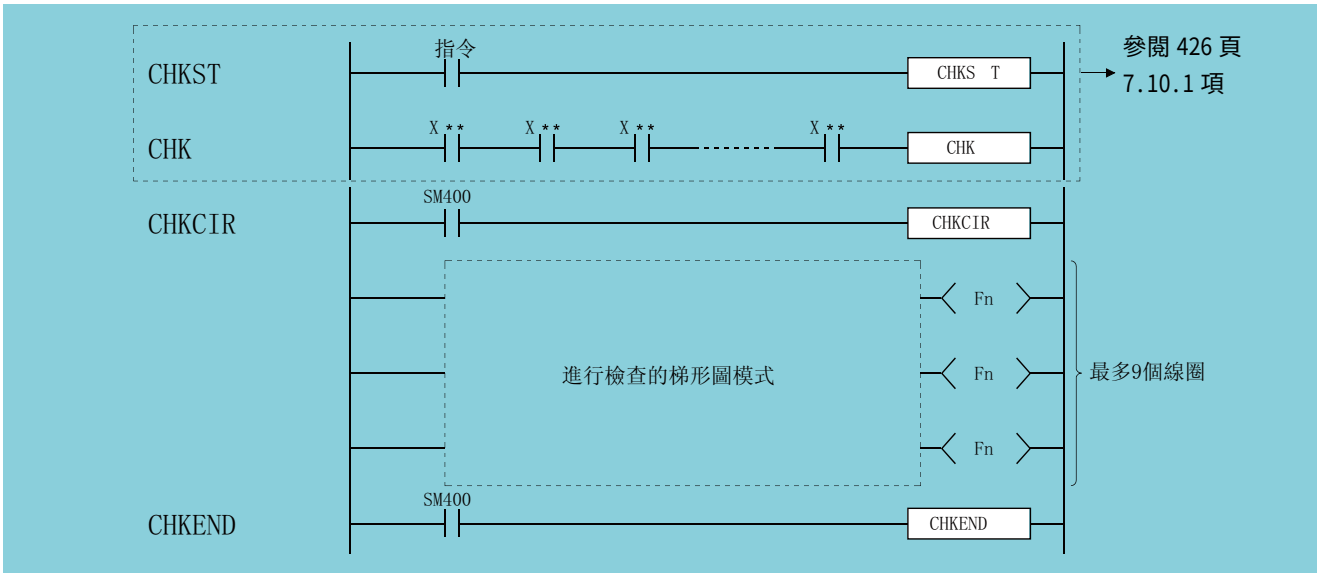
出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4235	存在有並行梯形圖時。 存在有 NOP 指令時。 觸點指令超過了 150 個時。 執行 CHKST 指令後，未執行 CHK 指令時。 在未執行 CHKST 指令的情況下執行了 CHK 指令時。 CHKST 指令、CHK 指令被用於低速執行型程式中時。 在 CHK 指令與 CHKST 指令之間存在有除 LD、LDI、AND、ANI 指令以外的指令時。 在處於執行狀態的所有程式文件中在 3 個或以上位置處使用了 CHK 指令時。 在 1 個程式文件中的 2 個或以上位置處使用了 CHK 指令時。	---	○	○	○	---	---

7.10.2 CHKCIR、CHKEND

Basic　High performance　Process　Redundant　Universal　LCPU



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功　能

CHKCIR,CHKEND

(1) 將 CHK 指令中使用的檢查梯形圖模式更改爲任意的格式。
　　實際的故障檢查是通過 CHKST、CHK 指令進行的。

(2) 根據 CHK 指令中指定的檢查條件及 CHKCIR ～ CHKEND 指令之間記述的梯形圖模式進行故障檢查。

備 註

關於 CHKCIR、CHKEND 指令，請參閱 426 頁 7.10.1 項。

要 點

使用 CHKCIR ～ CHKEND 指令對 CHK 指令的檢查格式進行變更時，應由用戶創建附加了變址修飾 (Z0) 的梯形圖。

(a) 檢查條件 (下圖的 X2、X8) 中所示的軟元件編號將成爲梯形圖模式中記述的各軟元件編號 (報警器 (F) 除外) 的變址修飾值。

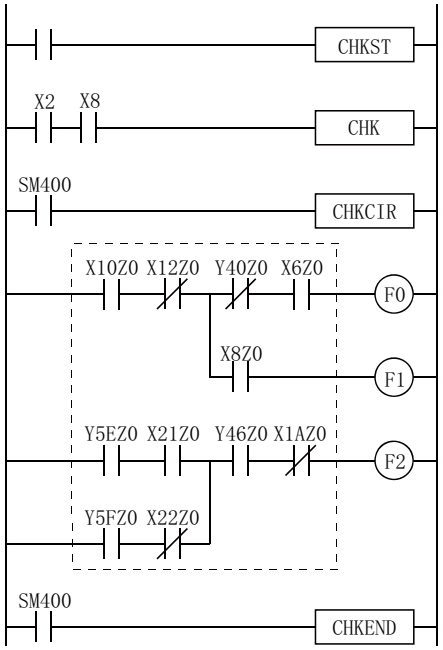
例 下圖中的 X10 的情況如下：

對應于檢查條件 X2 時 X12
對應于檢查條件 X8 時 X18
} 進行處理。

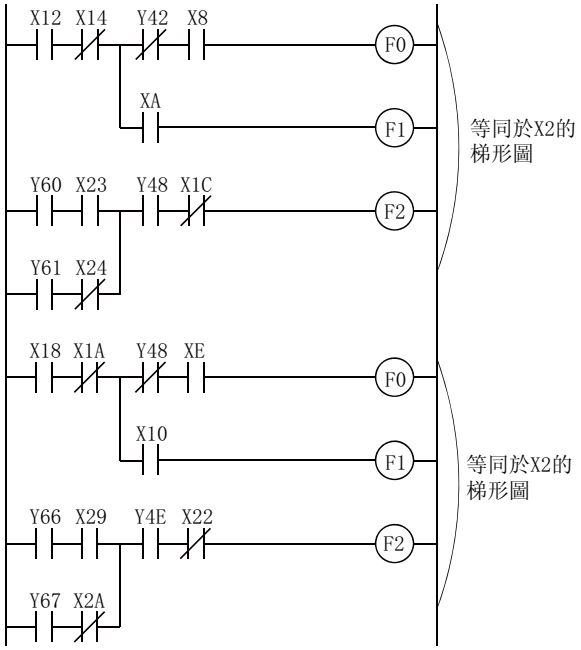
但是，根據 SM701 的 ON/OFF 狀態，故障檢查的順序有所不同。

1)SM701 爲 OFF 時，按觸點順序從線圈編號 1 開始進行檢查。

[CHKCIR ～ CHKEND 指定的梯形圖]

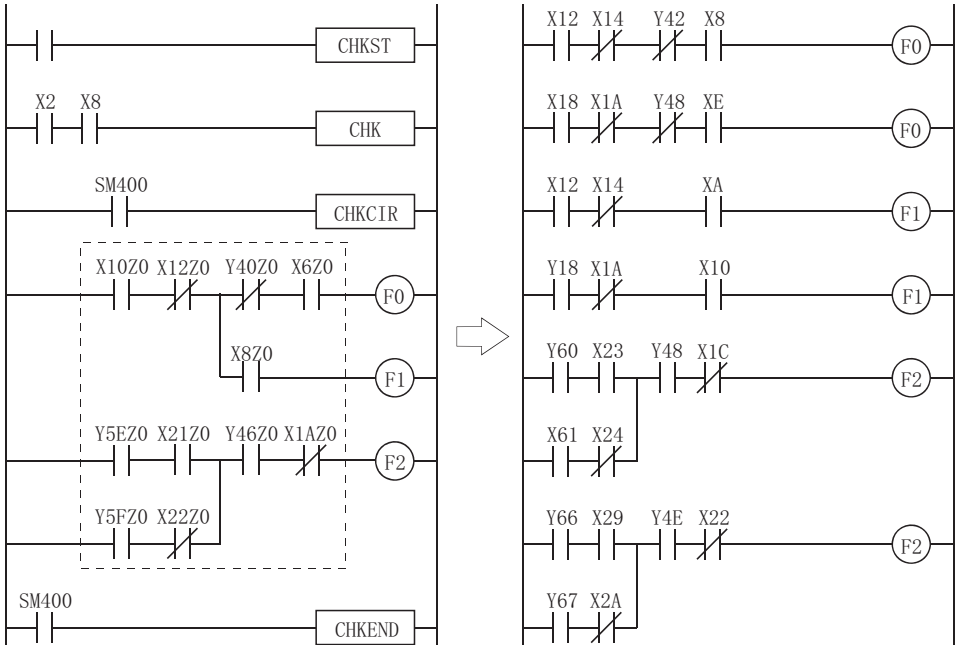


[CPU 模組的檢查順序]



2)SM701 爲 ON 時，按線圈順序從觸點編號 1 開始進行檢查。

[CHKCIR ～ CHKEND 指定的梯形圖]　　[CPU 模組的檢查順序]



(b) 進行故障檢查時，根據各檢查條件中的梯形圖模式，對 OUT F□的 ON/OFF 狀態進行檢查。
在所有的檢查條件中，只要梯形圖模式中有一個 OUT F□變爲 ON，則 SM80 將變爲 ON。
此外，將變爲 ON 狀態的 OUT F□所對應的出錯編號 (觸點編號及線圈編號) 按 BCD 順序存儲到 SD80 中。

(c) 梯形圖模式中可使用的指令如下所示：
觸點 .....LD、LDI、AND、ANI、OR、ORI、ANB、ORB、MPS、MPP、MRD 比較運算指令
線圈 .....OUT F□

(d) 梯形圖模式的觸點中可使用的軟元件如下所示：
輸入 (X)、輸出 (Y)

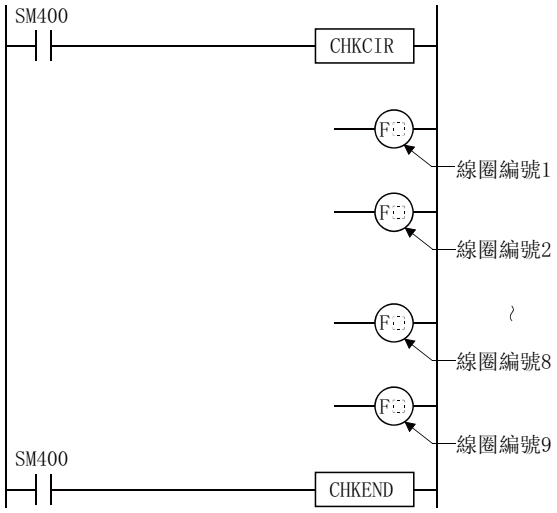
(e) 梯形圖模式的線圈中可使用的軟元件僅爲報警器 (F)。
但是，由於報警器 (F) 爲虛擬部件，因此可以設置任意的值。
此外，即使重復也沒有關係。

(f) 即使將與 CHK 指令中使用的報警器 (F) 具有相同編號的報警器 (F) 用於除 CHK 指令以外的其他位置，也可正常地進行 ON/OFF 控制。在 CHK 指令中及除 CHK 指令以外的其他位置將分別進行處理。

(g) 由於 CHK 指令中使用的報警器 (F) 不進行實際的 ON/OFF，因此即使通過週邊設備進行監控也不進行 ON/OFF。

(h) 可創建最多爲 256 步的梯形圖模式。
此外，OUT F□最多可使用 9 個線圈。

(3) CHKCIR ～ CHKEND 中指定的梯形圖的線圈編號按從上至下的順序被分配爲 1 ～ 9 號。



(4) CHKCIR、CHKEND 指令可被寫入到程式中的任意步中。
在所有處於執行狀態的程式文件中最多可在 2 個位置處使用。
但是，在一個程式文件中只能在一個位置處使用 CHK 指令。

(5) CHKCIR、CHKEND 指令不能用於低速執行型程式。
將記述了 CHKCIR、CHKEND 指令的程式文件設置到低速執行型程式中時，將變爲運算出錯狀態。此外，在高性能型 QCPU/過程 CPU/ 冗余 CPU 中將停止運算。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4230	執行了 CHKCIR 指令後，未執行 CHKEND 指令時。 在未執行 CHKCIR 指令的情況下，使用了 CHKEND 指令時。	---	○	○	○	---	---
4235	在所有程式文件中在 3 個或以上位置處使用了 CHKCIR ～ CHKEND 指令時。 在 1 個程式文件中的 2 個或以上位置處使用了 CHKCIR ～ CHKEND 指令時。 將 CHKCIR、CHKEND 指令用於低速執行型程式中時。 梯形圖模式中存在有 10 個或以上的 F 時。 梯形圖模式爲 257 步或以上時。 存在有梯形圖模式中不能使用的軟元件時。 對梯形圖模式的軟元件進行了變址修飾時。	---	○	○	○	---	---

7.11 字串處理指令

7.11.1 BINDA、BINDAP、DBINDA、DBINDAP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：進行 ASCII 碼轉換的 BIN 資料 (BIN16/32 位)。

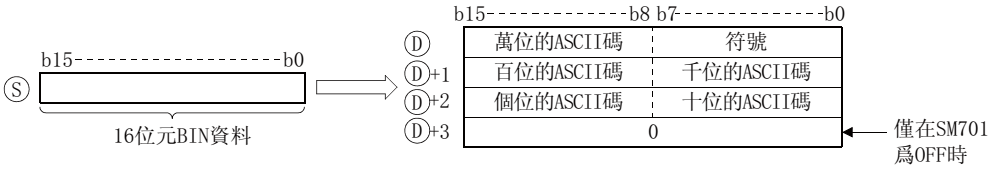
Ⓓ ：存儲轉換結果的軟元件的起始編號 (字串)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○		○					---
Ⓓ	---	○		---					---

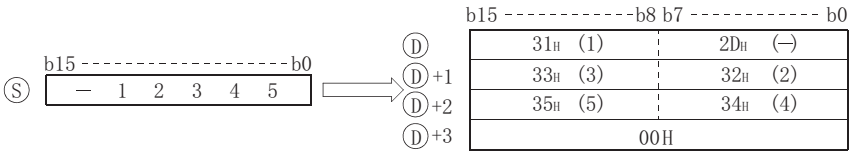
功　能

BINDA

(1) 將Ⓢ中指定的 BIN16 位元資料以 10 進制數表示時的各個位數值轉換爲 ASCII 碼後，存儲到Ⓓ中指定的軟元件編號的後面。



例如，在Ⓢ中指定了 -12345 時，按以下方式存儲到Ⓓ的後面。

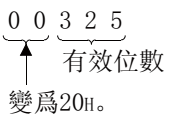


(2) 在Ⓢ中可指定的範圍爲 -32768 ～ 32767。

(3) Ⓓ中存儲的運算結果如下所示。

(a) BIN 資料爲正數時在 “ 符號 ” 中存儲 “ 20_H ”，爲負數時在 “ 符號 ” 中存儲 “ $2D_H$ ”。

(b) 在有效位元數的左側 “ 0 ” 中存儲 “ 20_H ”。(進行 0 刪除。)



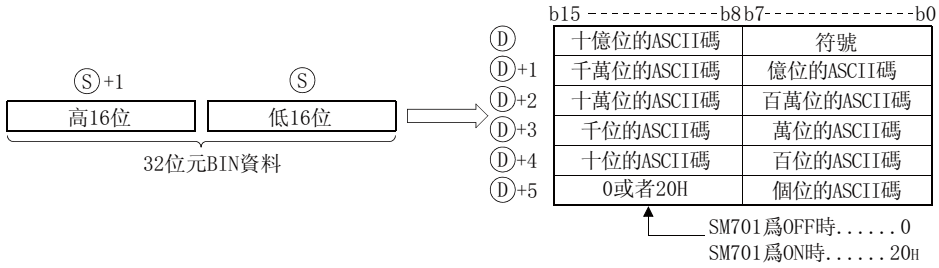
(c) 至Ⓓ+3 中指定的軟元件的資料存儲根據 SM701(輸出字元數切換信號) 的 ON/OFF 狀態而有所不同。

SM701 爲 OFF 時 ... 存儲 “ 0 ”。

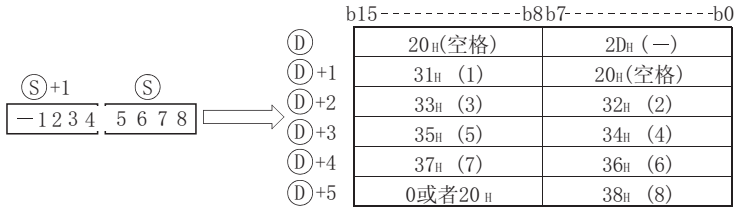
SM701 爲 ON 時 .... 無變化。

DBINDA

(1) 將Ⓢ中指定的 BIN32 位元資料以 10 進制數表示時的各個位數值轉換爲 ASCII 碼後，存儲到Ⓓ中指定的軟元件編號的後面。



例如，在Ⓢ中指定了 -12345678 時，按以下方式存儲到Ⓓ的後面。



(2) 在Ⓢ中可指定的 BIN 資料的範圍爲 -2147483648 ～ 2147483647。

(3) Ⓓ中存儲的結果如下所示。

(a) BIN 資料爲正數時在 “ 符號 ” 中存儲 “ 20_H ”，爲負數時在 “ 符號 ” 中存儲 “ $2D_H$ ”。

(b) 在有效位元數的左側 “ 0 ” 中存儲 “ 20_H ”。(進行 0 刪除。)

00 1 2 0 3 4 5 6 0

變爲 20H。　有效位數

(c) Ⓓ+5 中指定的軟元件的高 8 位元中存儲的資料根據 SM701(輸出字元數切換信號) 的 ON/OFF 狀態而有所不同。

SM701 爲 OFF 時... 存儲 “ 0 ”。

SM701 爲 ON 時.... 存儲 “ 20_H ”。

出　錯

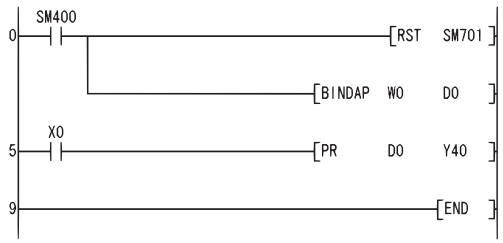
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 16 位元 BIN 資料的 W0 值通過 PR 指令轉換爲 10 進制數的 ASCII 碼後輸出到 Y40 ～ Y48 中的程式。

[梯形圖模式]



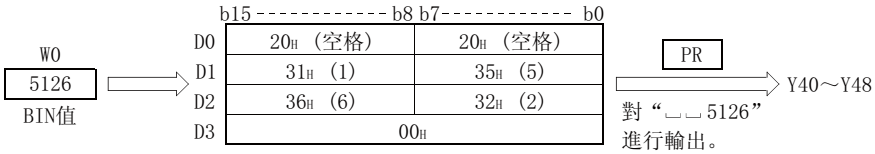
[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	RST	SM701	
2	BINDAP	W0	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

[動作]

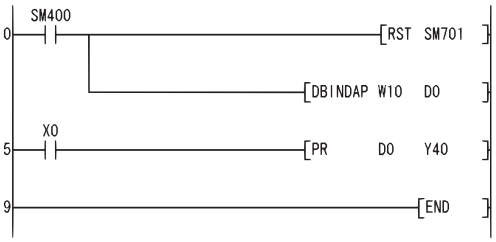
X0 變爲 ON 時，通過 PR 指令對 Y40 ～ Y48 進行 ASCII 碼輸出。

由於 SM701 處於 OFF 狀態，因此 PR 指令執行輸出直至 ASCII 碼 00H 爲止。



(2) 以下爲將 32 位元 BIN 資料 W10、W11 的值通過 PR 指令轉換爲 10 進制數的 ASCII 碼後輸出到 Y40 ～ Y48 中的程式。

[梯形圖模式]



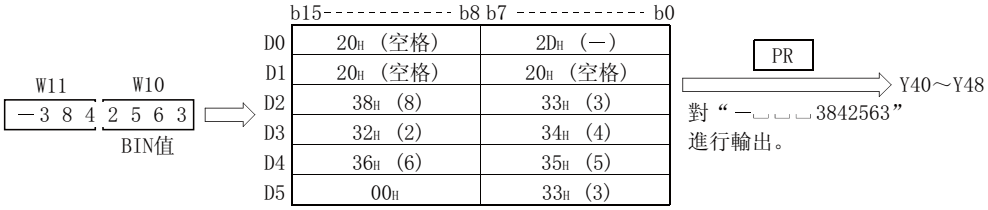
[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	RST	SM701	
2	DBINDAP	W10	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

[動作]

X0 變爲 ON 時，通過 PR 指令對 Y40 ～ Y48 進行 ASCII 碼輸出。

由於 SM701 處於 OFF 狀態，因此 PR 指令執行輸出直至 ASCII 碼 00_H 爲止。



7.11.2 BINHA、BINHAP、DBINHA、DBINHAP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：進行 ASCII 碼轉換的 BIN 資料 (BIN16/32 位)。

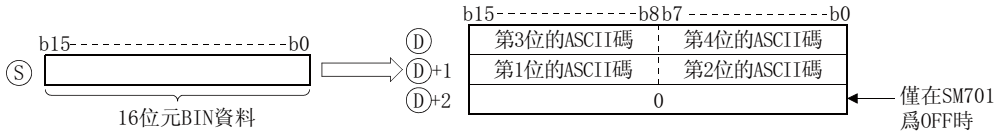
Ⓓ ：存儲轉換結果的軟元件的起始編號 (字串)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○		○					---
Ⓓ	---	○		---					---

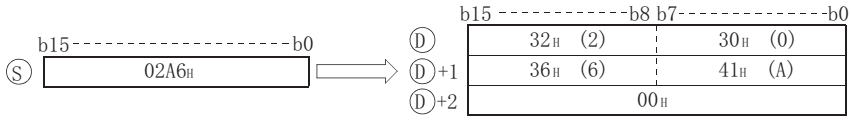
功　能

BINHA

(1) 將Ⓢ中指定的 BIN16 位元資料以 16 進制數表示時的各個位數值轉換爲 ASCII 碼後，存儲到Ⓓ中指定的軟元件編號的後面。



例如，在Ⓢ中指定了 $02A6_H$ 時，按以下方式存儲到Ⓓ的後面。



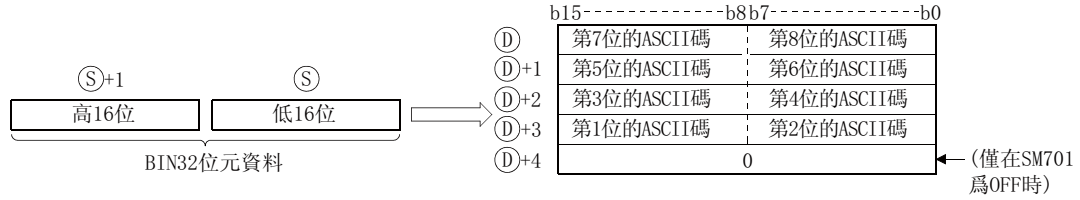
(2) 在Ⓢ中可指定的範圍爲 0_H ～ $FFFF_H$。

(3) Ⓓ中存儲的運算結果被處理爲 4 位數的 16 進制數。
因此，有效位數的左側的 “ 0 ” 將被處理爲 “ 0 ”。(不進行 0 刪除。)

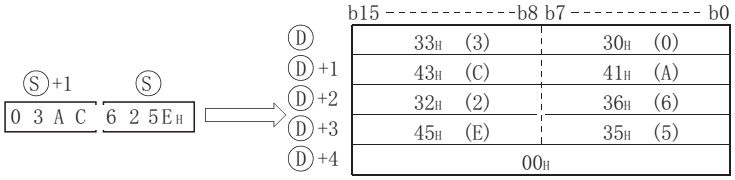
(4) 至Ⓓ+2 中指定的軟元件的資料存儲根據 SM701(輸出字元數切換信號) 的 ON/OFF 狀態而有所不同。
SM701 爲 OFF 時 ... 存儲 “ 0 ”。
SM701 爲 ON 時 .... 無變化。

DBINHA

(1) 將Ⓢ中指定的 BIN32 位元資料以 16 進制數表示時的各個位數值轉換爲 ASCII 碼後，存儲到Ⓓ中指定的軟元件編號的後面。



例如，在Ⓢ中指定了 $03AC625E_H$ 時，按以下方式存儲到Ⓓ的後面。



(2) 在Ⓢ中可指定的範圍爲 0_H ～ $FFFFFFFF_H$。

(3) Ⓓ中存儲的運算結果被處理爲 8 位數的 16 進制數。
因此，有效位數的左側的 “ 0 ” 將被處理爲 “ 0 ”。(不進行 0 刪除。)

(4) 至Ⓓ+2 中指定的軟元件的資料存儲根據 SM701(輸出字元數切換信號) 的 ON/OFF 狀態而有所不同。
SM701 爲 OFF 時 ... 存儲 “ 0 ”。
SM701 爲 ON 時 .... 無變化。

出　錯

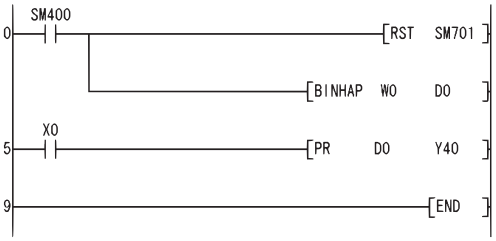
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 16 位元 BIN 資料的 W0 值通過 PR 指令轉換爲 16 進制數的 ASCII 碼後輸出到 Y40 ～ Y48 中的程式。

[梯形圖模式]



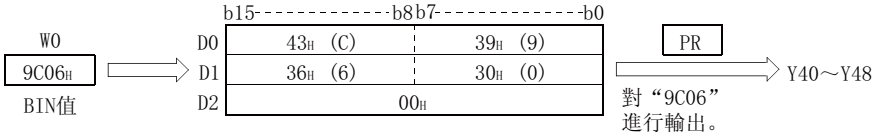
[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	RST	SM701	
2	BINHAP	W0	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

[動作]

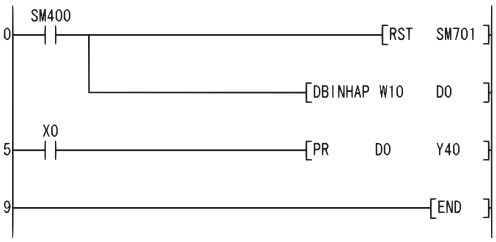
X0 變爲 ON 時，通過 PR 指令對 Y40 ～ Y48 進行 ASCII 碼輸出。

由於 SM701 處於 OFF 狀態，因此 PR 指令執行輸出直至 ASCII 碼 00_H 爲止。



(2) 以下爲將 32 位元 BIN 資料 W10、W11 的值通過 PR 指令轉換爲 16 進制數的 ASCII 碼後輸出到 Y40 ～ Y48 中的程式。

[梯形圖模式]



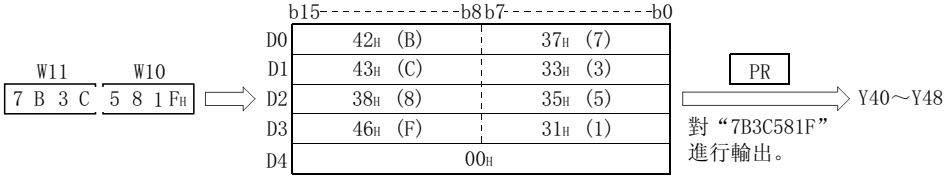
[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	RST	SM701	
2	DBINHAP	W10	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

[動作]

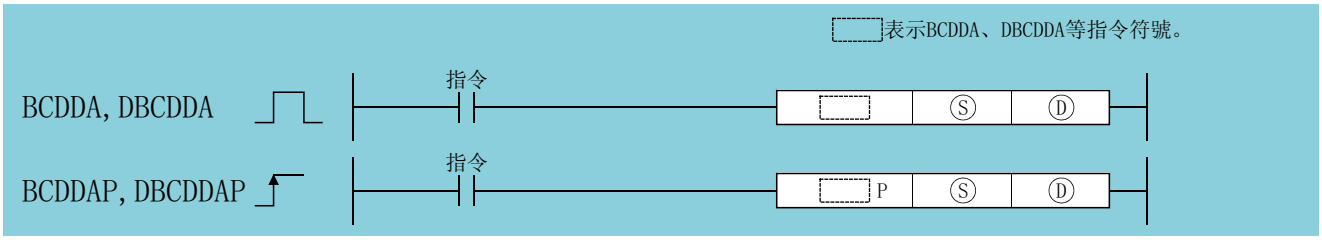
X0 變爲 ON 時，通過 PR 指令對 Y40 ～ Y48 進行 ASCII 碼輸出。

由於 SM701 處於 OFF 狀態，因此 PR 指令執行輸出直至 ASCII 碼 00_H 爲止。



7.11.3 BCDDA、BCDDAP、DBCDDA、DBCDDAP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：進行 ASCII 碼轉換的 BCD 資料 (BCD4 位元 /8 位)。

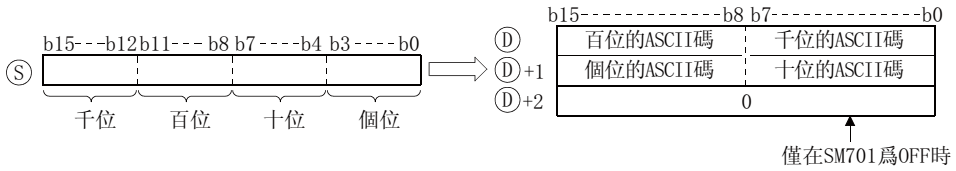
Ⓓ ：存儲轉換結果的軟元件的起始編號 (字串)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○		○					---
Ⓓ	---	○		---					---

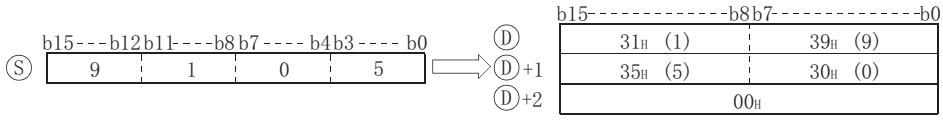
功　能

BCDDA

(1) 將Ⓢ中指定的 BCD4 位元資料的各個位元數值轉換爲 ASCII 碼後，存儲到Ⓓ中指定的軟元件編號的後面。

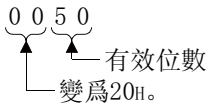


例如，在Ⓢ中指定了 9105 時，按以下方式存儲到Ⓓ的後面。



(2) 在Ⓢ中可指定的 BCD 資料範圍爲 0 ～ 9999。

(3) 在Ⓓ中存儲的運算結果中，有效位數的左側的 “ 0 ” 將被進行 0 刪除。



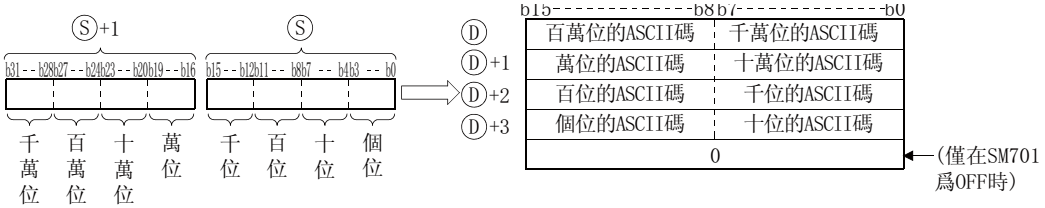
(4) 至Ⓓ+2 中指定的軟元件的資料存儲根據 SM701(輸出字元數切換信號) 的 ON/OFF 狀態而有所不同。

SM701 爲 OFF 時 ... 存儲 “ 0 ”。

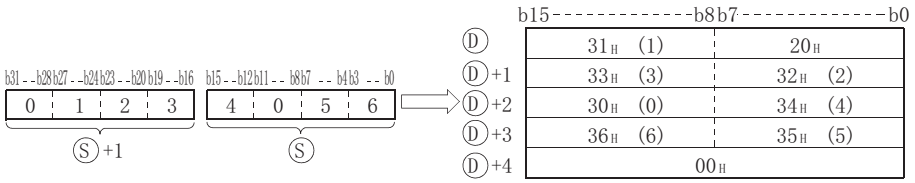
SM701 爲 ON 時 .... 無變化。

DBCDDA

(1) 將Ⓢ中指定的 BCD8 位元資料的各個位元數值轉換爲 ASCII 碼後，存儲到Ⓓ中指定的軟元件編號的後面。

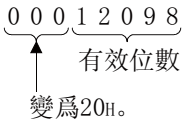


例如，在Ⓢ中指定了 01234056 時，按以下方式存儲到Ⓓ的後面。



(2) 在Ⓢ中可指定的 BCD 資料範圍爲 0 ～ 99999999。

(3) 在Ⓓ中存儲的運算結果中，有效位數的左側的 “ 0 ” 將被進行 0 刪除。



(4) 至Ⓓ+4 中指定的軟元件的資料存儲根據 SM701(輸出字元數切換信號) 的 ON/OFF 狀態而有所不同。
SM701 爲 OFF 時 ... 存儲 “ 0 ”。
SM701 爲 ON 時 .... 無變化。

出　錯

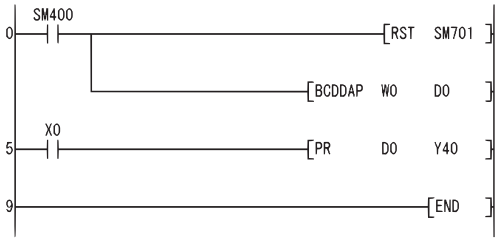
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	使用 BCDDA 指令時，Ⓢ的資料超出了 0 ～ 9999 的範圍時。 使用 DBCDDA 指令時，Ⓢ的資料超出了 0 ～ 99999999 的範圍時。	---	○	○	○	○	○
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 BCD4 位元資料 (W0 值) 通過 PR 指令轉換爲 10 進制數的 ASCII 碼後輸出到 Y40 ～ Y48 中的程式。

[梯形圖模式]



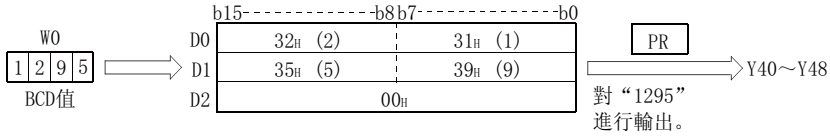
[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	RST	SM701	
2	BCDDAP	W0	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

[動作]

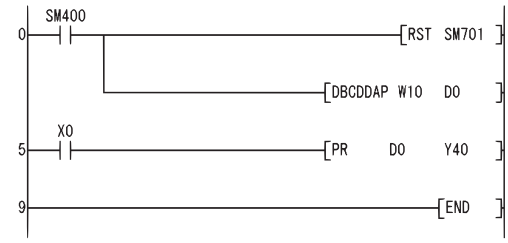
X0 變爲 ON 時，通過 PR 指令對 Y40 ～ Y48 進行 ASCII 碼輸出。

由於 SM701 處於 OFF 狀態，因此 PR 指令執行輸出直至 ASCII 碼 00_H 爲止。



(2) 以下爲將 BCD8 位元資料 (W10、W11 的值) 通過 PR 指令轉換爲 10 進制數的 ASCII 碼後輸出到 Y40 ～ Y48 中的程式。

[梯形圖模式]



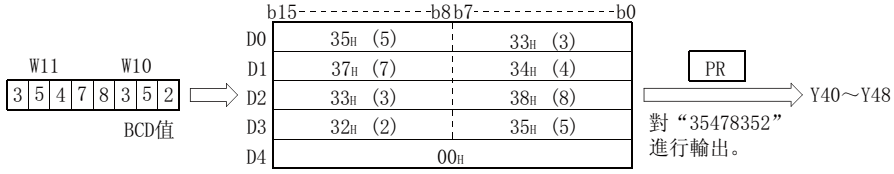
[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	RST	SM701	
2	DBCDDAP	W10	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

[動作]

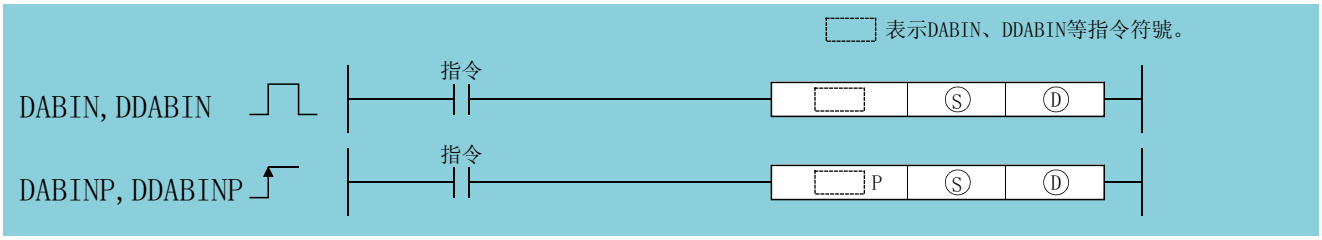
X0 變爲 ON 時，通過 PR 指令對 Y40 ～ Y48 進行 ASCII 碼輸出。

由於 SM701 處於 OFF 狀態，因此 PR 指令執行輸出直至 ASCII 碼 00_H 爲止。



7.11.4 DABIN、DABINP、DDABIN、DDABINP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：進行 BIN 值轉換的 ASCII 碼資料或者存儲 ASCII 碼資料的軟元件的起始編號 (字串)。

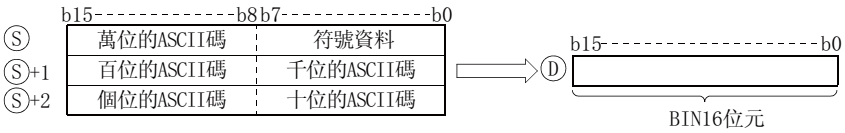
Ⓓ ：存儲轉換結果的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	○	○		○				---	---

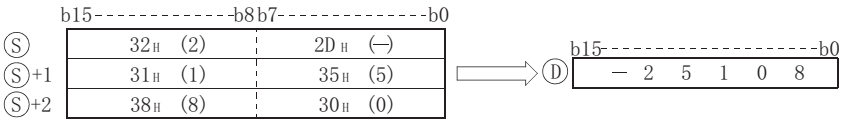
功 能

DABIN

(1) 將存儲在Ⓢ中指定的軟元件編號後面的 10 進制 ASCII 資料轉換爲 BIN16 位元資料後，存儲到Ⓓ中指定編號的軟元件中。



例如，Ⓢ後面指定了 -25108_H 的 ASCII 碼時，按以下方式存儲到Ⓓ中。



(2) 在Ⓢ～Ⓢ+2 中可指定的 ASCI 資料的範圍爲 -32768 ～ 32767。

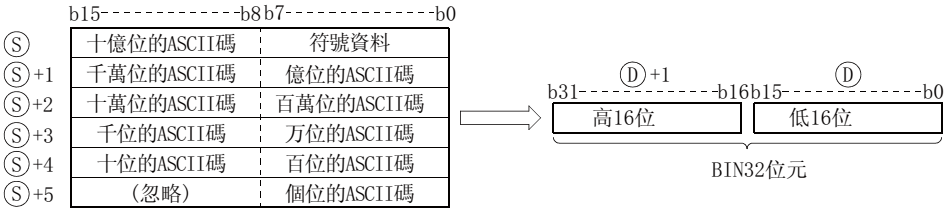
(3) 轉換的資料爲正數時在 “ 符號 ” 資料中設置 “ 20_H ”，爲負數時在 “ 符號 ” 資料中設置 “ $2D_H$ ”。
(設置了除 “ 20_H ”、 “ $2D_H$ ” 以外時，將作爲正的資料處理。)

(4) 各位數中可設置的 ASCII 碼的範圍爲 “ 30_H ” ～ “ 39_H ”。

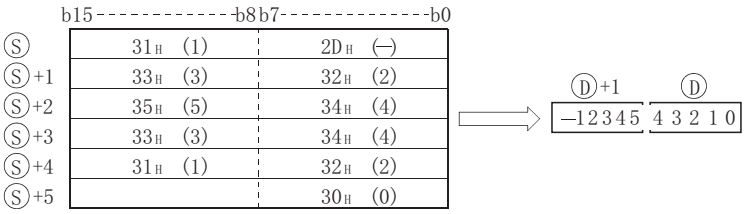
(5) 各位數中設置的 ASCII 碼爲 “ 20_H ”、 “ 00_H ” 時，將作爲 “ 30_H ” 處理。

DDABIN

(1) 將存儲在Ⓢ中指定的軟元件編號後面的 10 進制 ASCII 資料轉換爲 BIN32 位元資料後，存儲到Ⓓ中指定編號的軟元件中。



例如，Ⓢ後面指定了 -1234543210_H 的 ASCII 碼時，按以下方式存儲到Ⓓ+1、Ⓓ中。



(2) 在Ⓢ～Ⓢ+5 中可指定的 ASCI 資料的範圍爲 -2147483648 ～ 2147483647。
此外，Ⓢ+5 的高位位元組中存儲的資料將被忽略。

(3) 轉換的資料爲正數時在 “ 符號 ” 資料中設置 “ 20_H ”，爲負數時在 “ 符號 ” 資料中設置 “ $2D_H$ ”。
(設置了除 “ 2_H ”、 “ $2D_H$ ” 以外時，將作爲正的資料處理。)

(4) 各位中可設置的 ASCII 碼的範圍爲 “ 30_H ” ～ “ 39_H ”。

(5) 各位中設置的 ASCII 碼爲 “ 20_H ”、 “ 00_H ” 時，將作爲 “ 30_H ” 處理。

出　錯

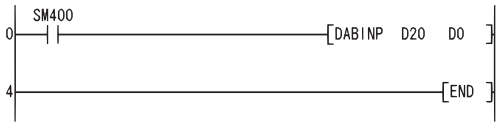
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	在Ⓢ～Ⓢ+5 中指定的 ASCI 碼爲 “30_H” ～ “39_H”、“20_H”、“00_H” 以外時。 在Ⓢ～Ⓢ+5 中指定的 ASCI 資料超出了以下範圍時。 使用 DABIN 指令時..... -32768 ～ 32767 使用 DDABIN 指令時..... -2147483648 ～ 2147483647	---	○	○	○	○	○
4101	Ⓢ 中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 D20 ～ D22 中設置的符號及 10 進制 5 位元數的 ASCII 資料轉換爲 BIN 值後，存儲到 D0 中的程式。

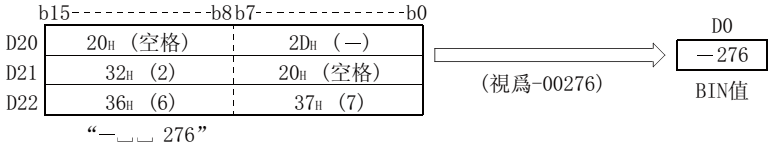
[梯形圖模式]



[列表模式]

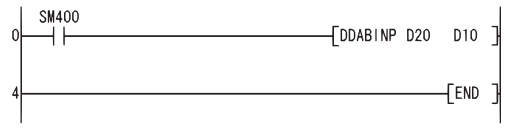
步	指令	軟元件	
0	LD	SM400	
1	DABINP	D20	D0
4	END		

[動作]



(2) 以下爲將 D20 ～ D25 中設置的符號及 10 進制 10 位元數的 ASCII 資料轉換爲 BIN 值後，存儲到 D10、D11 中的程式。

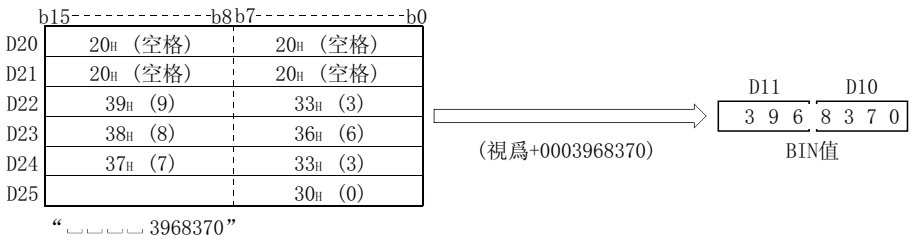
[梯形圖模式]



[列表模式]

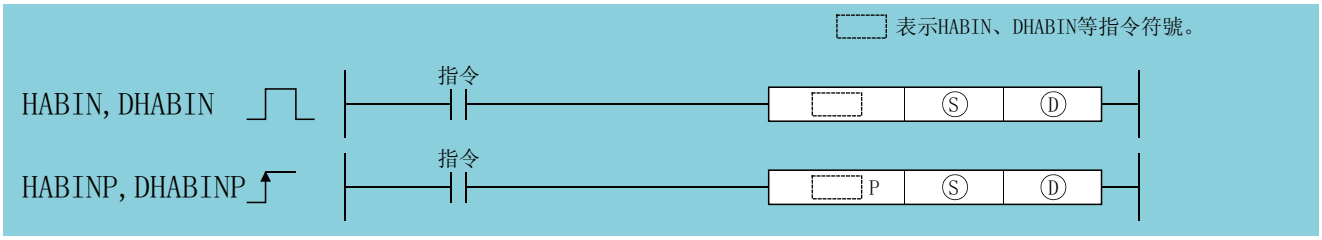
步	指令	軟元件	
0	LD	SM400	
1	DDABINP	D20	D10
4	END		

[動作]



7.11.5 HABIN、HABINP、DHABIN、DHABINP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：轉換爲 BIN 值的 ASCII 資料或者存儲 ASCII 資料的軟元件的起始編號 (字串)。

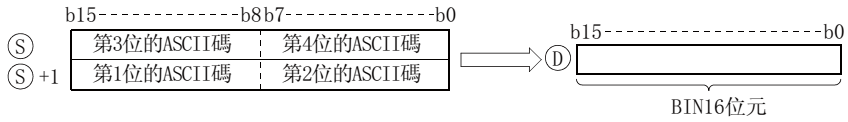
Ⓓ ：存儲轉換結果的軟元件的起始編號 (BIN16/32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	○	○		○				---	---

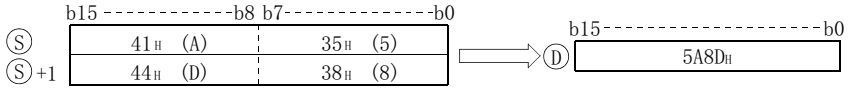
功　能

HABIN

(1) 將Ⓢ中指定的軟元件編號後面存儲 16 進制 ASCII 資料轉換爲 BIN16 位元資料後，存儲到Ⓓ中指定編號的軟元件中。



例如，Ⓢ在後面指定了 $5A8D_H$ 的 ASCII 碼時，按以下方式存儲到Ⓓ中。

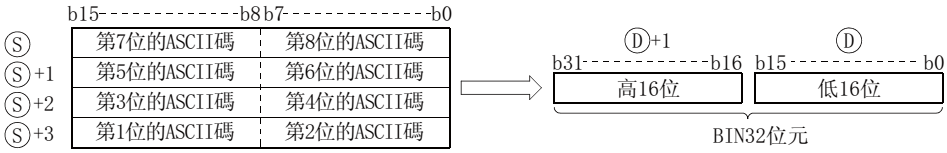


(2) 在Ⓢ～Ⓢ+1 中可指定的 ASCII 資料範圍爲 0000_H ～ $FFFF_H$。

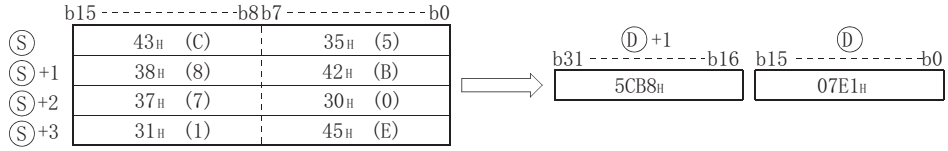
(3) 各位中可設置的 ASCII 碼的範圍爲 “ 30_H ” ～ “ 3_H ” 及 “ 41_H ” ～ “ 46_H ”。

DHABIN

(1) 將Ⓢ中指定的軟元件編號後面存儲 16 進制 ASCII 資料轉換爲 BIN32 位元資料後，存儲到Ⓓ中指定編號的軟元件中。



例如，Ⓢ在後面指定了 $5CB807E1_H$ 的 ASCII 碼時，按以下方式存儲到Ⓓ+1、Ⓓ中。



(2) 在Ⓢ～Ⓢ+3 中可指定的 ASCII 資料範圍爲 00000000_H ～ $FFFFFFFF_H$。

(3) 各位中可設置的 ASCII 碼的範圍爲 “ 30_H ” ～ “ 39_H ” 及 “ 41_H ” ～ “ 46_H ”。

出　錯

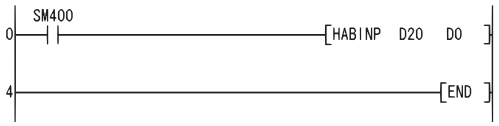
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	在Ⓢ～Ⓢ+3 中指定的各位的 ASCII 碼爲 “ 30_H ” ～ “ 39_H ”、“ 41_H ”、“ 46_H ” 以外時。	---	○	○	○	○	○
4101	Ⓢ 中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1)　以下爲將 D20、D21 中設置的 16 進制 4 位數的 ASCII 碼轉換爲 BIN 值後，輸出到 D0 中的程式。

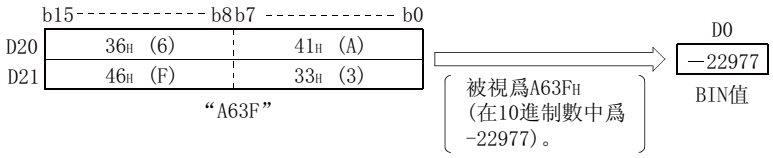
[梯形圖模式]



[列表模式]

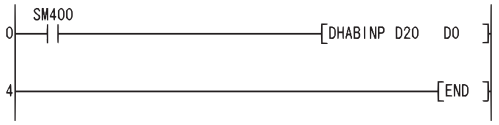
步	指令	軟元件	
0	LD	SM400	
1	HABINP	D20	D0
4	END		

[動作]



(2)　以下爲將 D20 ～ D23 中設置的 16 進制 8 位數的 ASCII 碼轉換爲 BIN 值後，輸出到 D10、D11 中的程式。

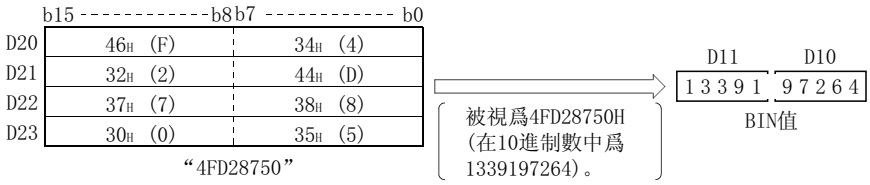
[梯形圖模式]



[列表模式]

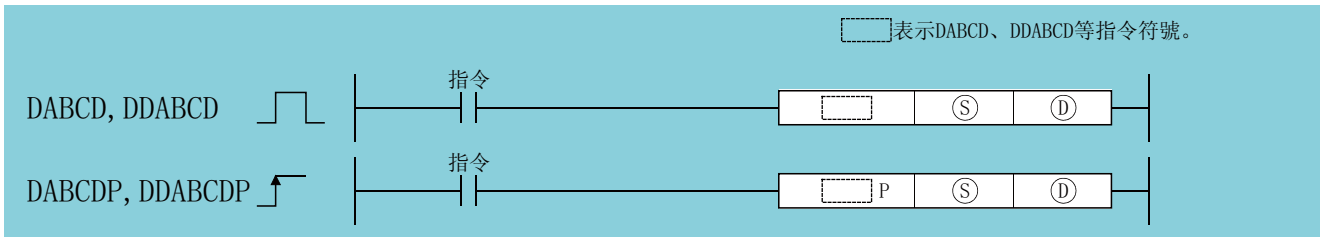
步	指令	軟元件	
0	LD	SM400	
1	DHABINP	D20	D0
4	END		

[動作]



7.11.6 DABCD、DABCDP、DDABCD、DDABCDP

Basic　High performance　Process　Redundant　Universal　LCPU



Ⓢ ： 轉換爲 BCD 值的 ASCII 資料或者存儲 ASCII 資料的軟元件的起始編號 (字串)。

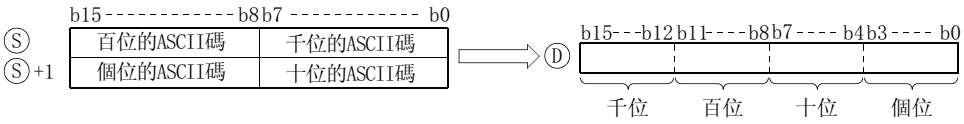
Ⓓ ： 存儲轉換結果的軟元件的起始編號 (BCD4 位 /8 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	○	○		○				---	---

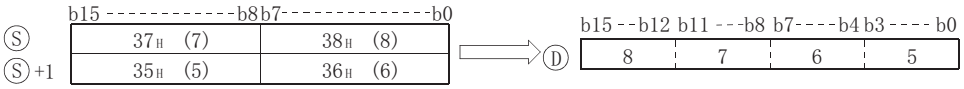
功　能

DABCD

(1) 將Ⓢ中指定的軟元件編號後面存儲的 10 進制 ASCII 資料轉換爲 BCD4 位元數資料後，存儲到Ⓓ中指定編號的軟元件中。



例如，Ⓢ在後面指定了 8765_H 的 ASCII 碼時，按以下方式存儲到Ⓓ中。



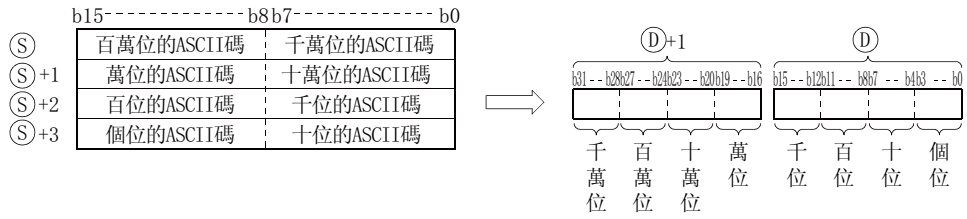
(2) 在Ⓢ～Ⓢ+1 中可指定的 ASCII 資料範圍爲 0 ～ 9999。

(3) 在各位數中可設置的 ASCII 碼的範圍爲 “ 30_H ” ～ “ 39_H ”。

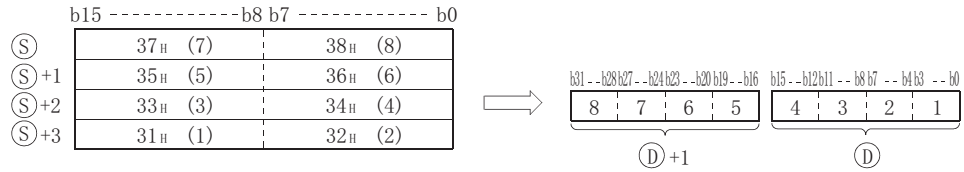
(4) 在各位數中設置的 ASCII 碼爲 “ 20_H ”、“ 00_H ” 時，將作爲 “ 30_H ” 處理。

DDABCD

(1) 將Ⓢ中指定的軟元件編號後面存儲的 10 進制 ASCII 資料轉換爲 BCD8 位元數資料後，存儲到Ⓓ中指定編號的軟元件後面。



例如，Ⓢ在後面指定了 87654321_H 的 ASCII 碼時，按以下方式存儲到Ⓓ+1、Ⓓ中。



(2) 在Ⓢ～Ⓢ+3 中可指定的 ASCII 資料範圍爲 0 ～ 99999999。

(3) 在各位數中可設置的 ASCII 碼的範圍爲 “ 30_H ” ～ “ 39_H ”。

(4) 在各位數中設置的 ASCII 碼爲 “ 20_H ” ～ “ 00_H ” 時，將作爲 “ 30_H ” 處理。

出　錯

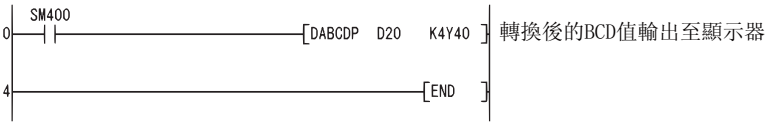
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ的資料中存在有除 “ 0 ” ～ “ 9 ” 以外的字元時。	---	○	○	○	○	○
4101	Ⓢ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 D20 ～ D22 中設置的 10 進制數的 ASCII 資料轉換爲 BCD4 位元數資料後，輸出到 Y40 ～ Y4F 中的程式。

[梯形圖模式]



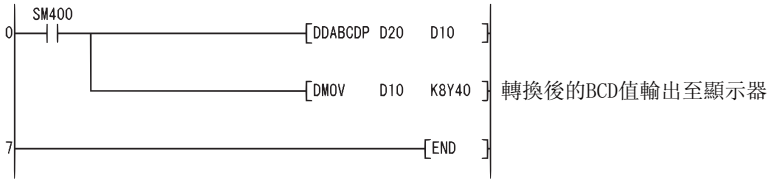
[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	DABCDP	D20	K4Y40
4	END		

[動作]

(2) 以下爲將 D20 ～ D23 中設置的 10 進制數的 ASCII 資料轉換爲 BCD8 位元數資料後，在存儲到 D10、D11 中的同時，輸出到 Y40 ～ Y5F 中的程式。

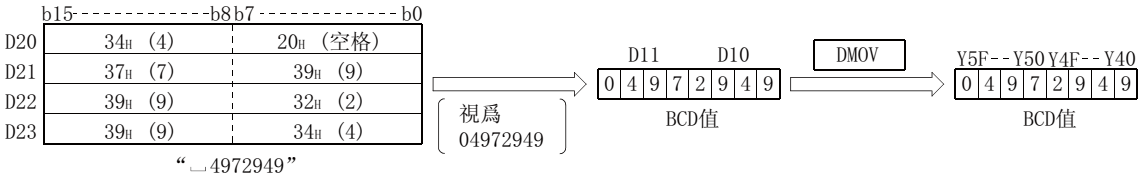
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	DDABCDP	D20	D10
4	DMOV	D10	K8Y40
7	END		

[動作]



7.11.7 COMRD、COMRDP

Basic | High performance | Process | Redundant | Universal | LCPU

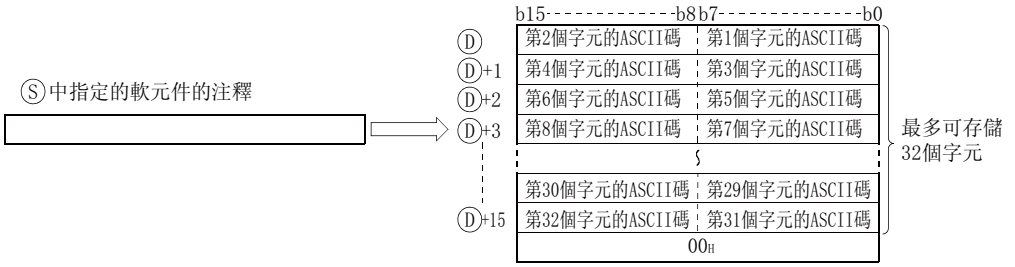
Ⓢ ：登錄了讀取的注釋的軟元件的起始編號（軟元件名）。

Ⓓ ：存儲讀取的注釋的軟元件的起始編號（字串）。

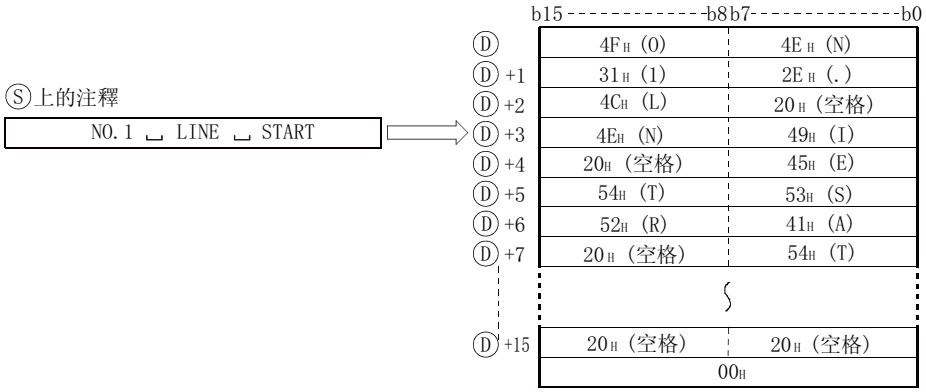
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他 BL\S、BL\TR、BL、P、I、J、U
	位	字		位	字				
Ⓢ	○	○		○			---		○
Ⓓ	---	○		---			---		---

功　能

(1) 對Ⓢ中指定的軟元件編號的注釋進行讀取後，以 ASCII 碼格式存儲到Ⓓ中指定編號的軟元件後面。



例如，在Ⓢ中中指定的軟元件注釋爲“No.1␣LINE␣TART␣␣”時，按以下方式存儲到Ⓓ以後。



(2) 在Ⓢ中指定的軟元件號未進行注釋範圍設置，注釋未被登錄的情況下，注釋的字元將全部被處理爲“20_H”（空格）。

(3) 存儲Ⓓ的最後字元的軟元件號 +1 根據 SM701（輸出字元數切換信號）的 ON/OFF 狀態而不同。

SM701 爲 OFF 時　：無變化。

SM701 爲 ON 時　：存儲“0”。

(4) 讀取注釋時，指令結束後 SM702 將 ON 一個掃描。

此外，指令執行過程中 SM721 處於 ON 狀態。

在 SM721 處於 ON 狀態時，不能執行 COMRD(P) 指令。執行的情況下將進行無處理。

要 點

1. COMRD(P) 指令中使用的軟元件注釋使用存儲卡及標準 ROM 中存儲的注釋文件。
   不能使用程式記憶體中存儲的注釋文件。
2. COMRD(P) 指令中使用的注釋文件是在可程式控制器參數的 “ 可程式控制器文件設置 ” 中設置。
   如果未在可程式控制器文件設置中對所使用的注釋文件進行設置，將不能通過 COMRD(P) 指令輸出軟元件注釋。
   在可程式控制器參數的 “ 可程式控制器文件設置 ” 中設置了注釋文件的情況下，在接通電源或重定時如果沒有設置的文件將發生 “ FILE SET ERROR ” (出錯代碼 : 2400)。
3. COMRD(P) 指令不能在中斷程式中執行。
   如果執行將變爲無處理。

出 錯

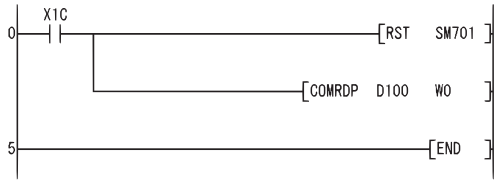
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ 中指定的軟元件編號的注釋未進行登錄時。	---	○	○	○	○	○
4101	Ⓓ 中指定編號的軟元件不是字軟元件時。	---	○	○	○	○	○
	Ⓓ 中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X1C 變爲 ON 時，將 D100 中設置的注釋以 ASCII 碼格式存儲到 W0 的後面的程式。

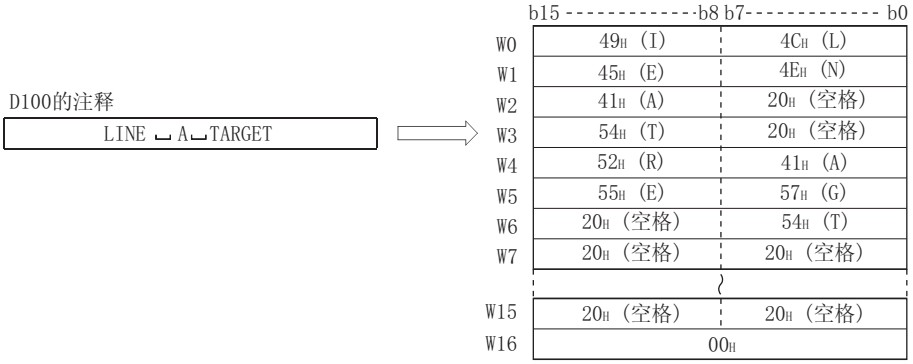
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X1C	
1	RST	SM701	
2	COMRDP	D100	W0
5	END		

[動作]



注意事項

(1) 在數次掃描後結束處理。

(2) 在指令結束前 (SM721 處於 ON 狀態) 即使將 COMRD(P)/PRC 指令的啓動信號 (執行指令) 置爲 ON，也不能執行 COMRD(P)/PRC 指令。應將程式設置爲在 SM721 處於 OFF 狀態時執行 COMRD(P)/PRC 指令。

(3) 不能同時訪問兩個或兩個以上的文件注釋。

(4) 以下指令由於共用 SM721，所以不能同時執行。

指令名	執行過程中 ON	結束後 ON 一個掃描	異常結束時 ON
SP.FREAD SP.FWRITE	SM721	通過指令指定	(通過指令指定的軟元件)+1
PRC COMRD		SM720	無

(5) 在 LCPU 中，使用 SD 存儲卡中存儲的注釋文件時，SM606(SD 存儲卡強制使用停止指示) 爲 ON 的過程中，不能執行本指令。如果執行將變爲無處理。

7.11.8 LEN、LENP

Basic / High performance / Process / Redundant / Universal / LCPU

Ⓢ ：字串或者存儲字串的軟元件起始編號 (字串)。

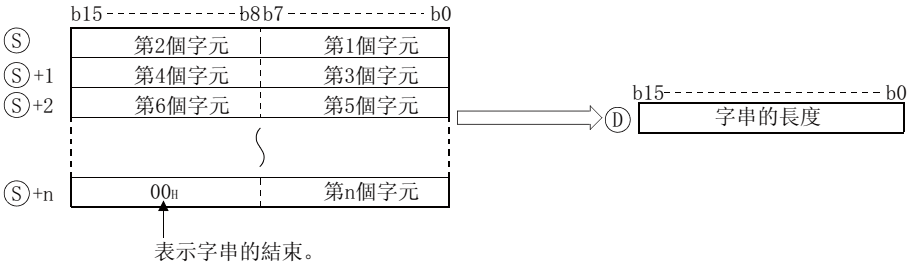
Ⓓ ：存儲檢測的字串長度的軟元件起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	○	○		○				---	---

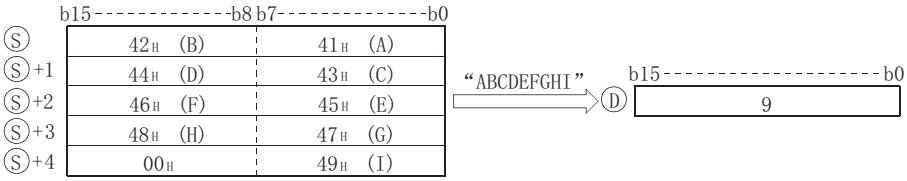
功　能

(1) 對Ⓢ中指定的字串的長度進行檢測後，存儲到Ⓓ中指定編號的軟元件後面。

將從Ⓢ中指定的軟元件編號開始，至存儲了 “ 00_H ” 的軟元件編號爲止的資料作爲字串處理。



例如，Ⓢ在後面存儲了 “ ABCDEFGHI ” 時，在Ⓓ中將存儲 9。



出　錯

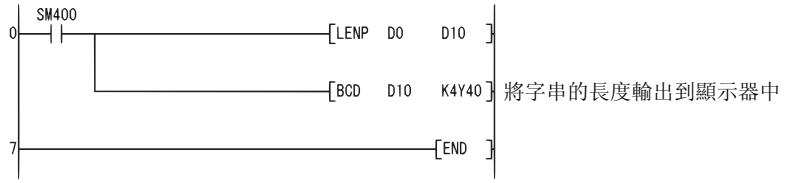
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓢ中指定的軟元件超出了相應軟元件的範圍時。	---	○	○	○	○	○

程式示例

(1) 以下爲將從 D0 開始的字串長度以 BCD4 位元數格式輸出到 Y40 ～ Y4F 中的程式。

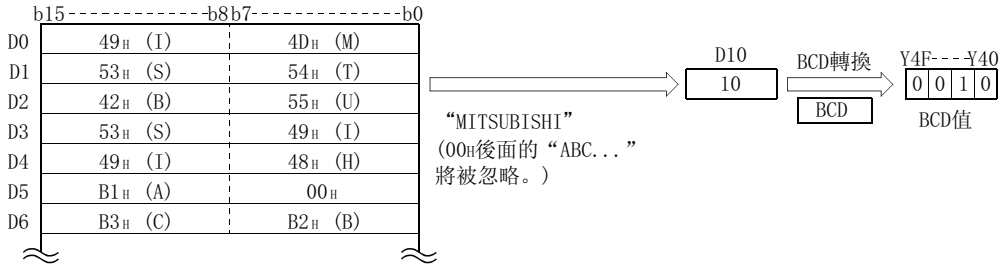
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	LENP	D0	D10
4	BCD	D10	K4Y40
7	END		

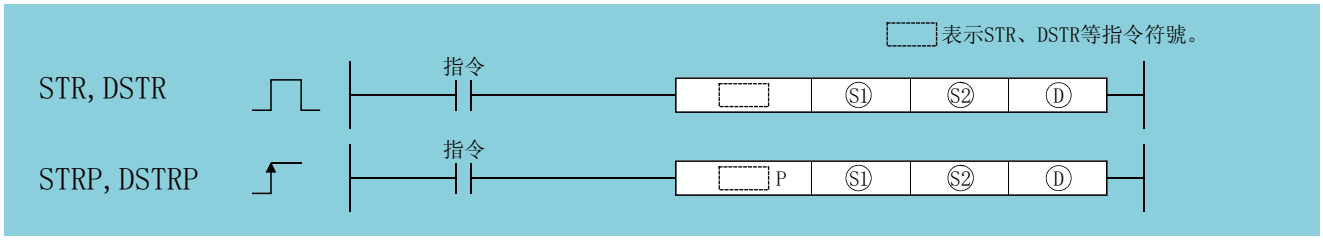
[動作]



7.11.9 STR、STRP、DSTR、DSTRP

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 " 04122 " 以後的基本型 QCPU 中可以使用。



(S1) : 轉換數值位數的軟元件的起始編號 (BIN16 位)。

(S2) : 進行轉換的 BIN 資料 (BIN16/32 位)。

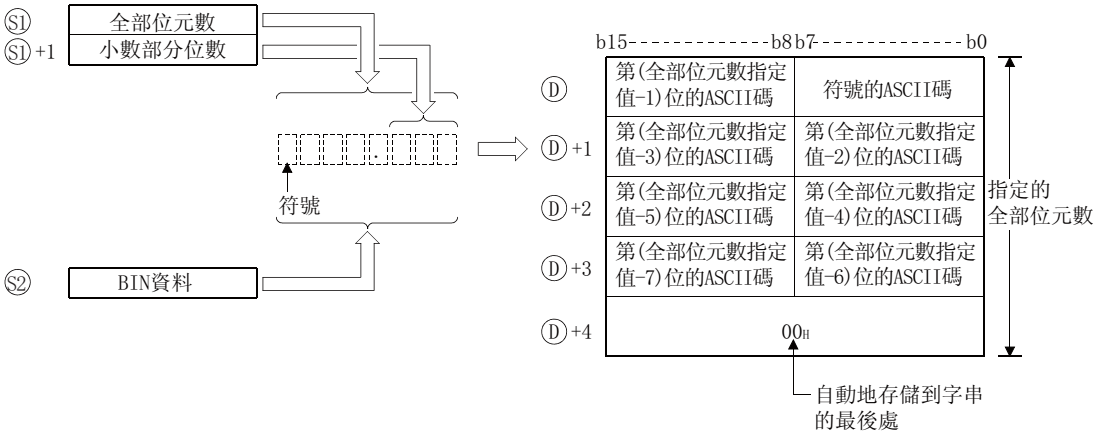
(D) : 存儲轉換後的字串的軟元件的起始編號 (字串)。

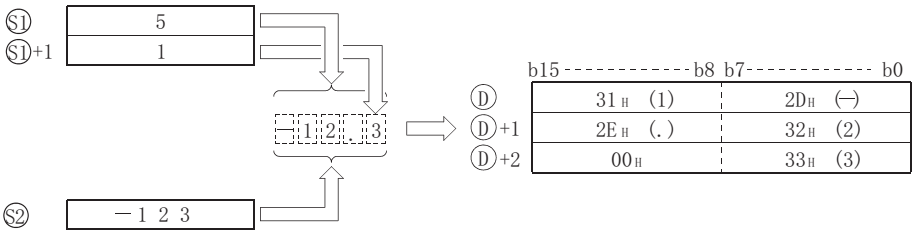
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
(S1)	○	○		○				---	---
(S2)	○	○		○				○	---
(D)	---	○		---				---	---

功 能

STR

(1) 將Ⓢ2中指定的 BIN16 位元資料轉換爲在Ⓢ1中指定位置附加了小數點的字串後，存儲到Ⓓ中指定編號的軟元件的後面。





(2) Ⓢ1中可指定的全部位元數爲 2 ～ 8 位。

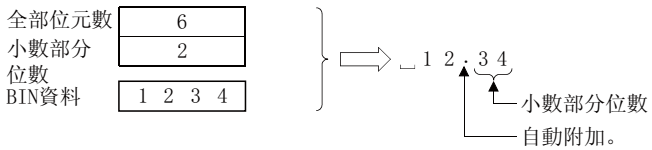
(3) Ⓢ1+1 中可指定的小數部分位元數爲 0 ～ 5 位。
但是，應設置爲小數部分位元數≤ (全部位元數 -3)。

(4) Ⓢ2中可指定的 BIN 資料範圍爲 -32768 ～ 32767。

(5) 轉換後的字串資料按以下方式存儲到Ⓓ的後面的軟元件編號中。

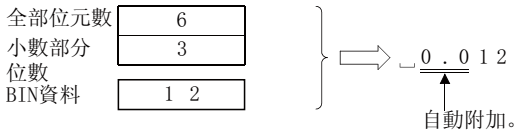
(a) BIN 資料爲正數時在 “ 符號 ” 中存儲 “ 20_H ” (空格)，爲負數時在 “ 符號 ” 中存儲 “ $2D_H$ ” (-)。

(b) 小數部分位元數被設置爲 “ 0 ” 以外時，第 (指定位數 +1) 位元中將自動存儲 “ $2E_H$ ” (.)。

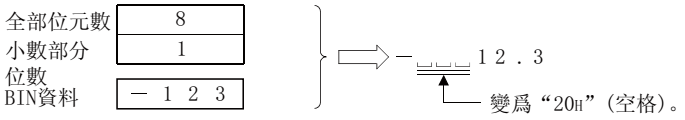


小數部分位元數爲 “ 0 ” 時，不存儲 “ $2E_H$ ” (.)。

(c) 在小數部分位元數的值大於 BIN 資料的位元數的情況下，在轉換時將自動地附加 0 後向右填充對齊，變爲 “ 0.□□□□□ ”。



(d) 如果全部位元數的值減去符號、小數點後的位元數仍然大於 BIN 資料的位元數時，將在符號與數值之間存儲 “ 20_H ” (空格)。

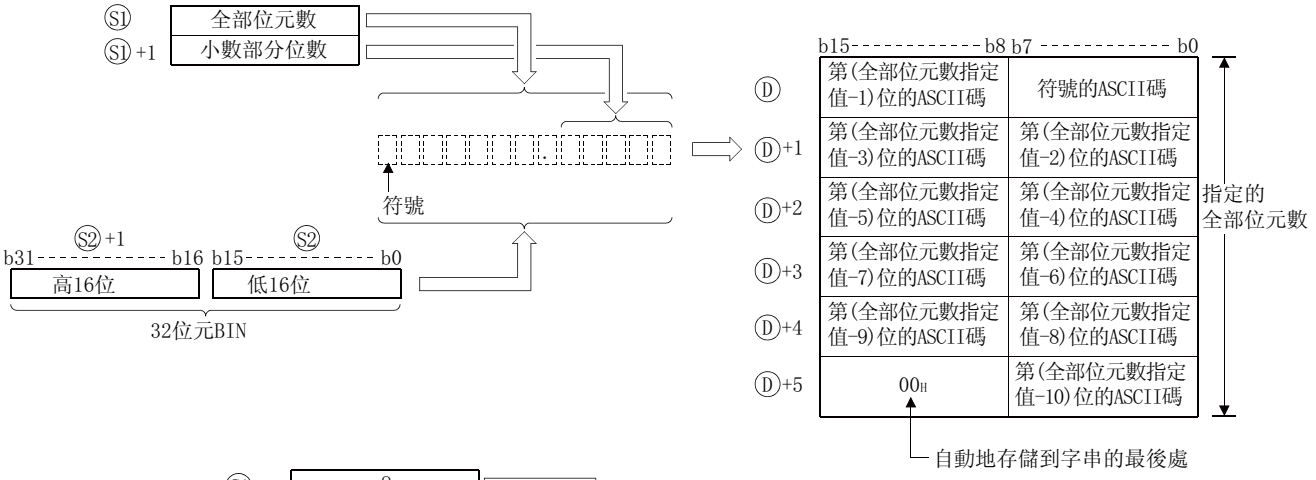


如果 BIN 資料的位元數大於全部位元數，將變爲出錯狀態。

(e) 在轉換後的字串的最末尾處將自動存儲 “ 00_H ”。

DSTR

(1) 將Ⓢ2中指定的 BIN32 位元資料轉換爲在Ⓢ1中指定位置附加了小數點的字串後，存儲到Ⓓ中指定編號的軟元件的後面。



(2) Ⓢ1中可指定的全部位元數爲 2 ～ 13 位。

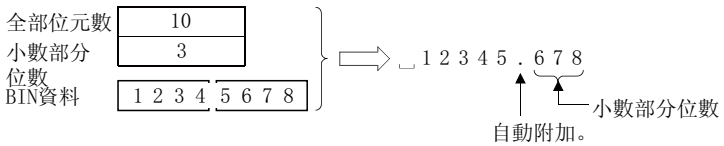
(3) Ⓢ1+1 中可指定的小數部分位元數爲 0 ～ 10 位。
但是，應設置爲小數部分位元數≤ (全部位元數 -3)。

(4) Ⓢ1、Ⓢ2+1 中可指定的 BIN 資料範圍爲 -2147483648 ～ 2147483647。

(5) 轉換後的字串資料按以下方式存儲到Ⓓ的後面的軟元件編號中。

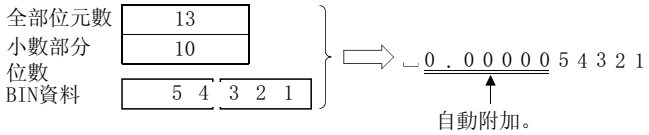
(a) BIN 資料爲正數時在 “ 符號 ” 中存儲 “ 20_H ” (空格)，爲負數時在 “ 符號 ” 中存儲 “ $2D_H$ ” (-)。

(b) 小數部分位元數被設置爲 “ 0 ” 以外時，第 (指定位數 +1) 位元中將自動存儲 “ $2E_H$ ” (.)。

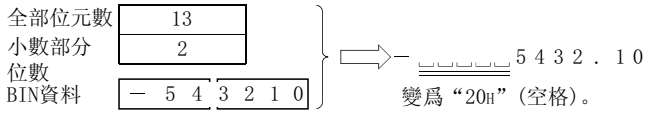


小數部分位元數爲 “ 0 ” 時，不存儲 “ $2E_H$ ” (.)。

(c) 在小數部分位元數的值大於 BIN 資料的位元數的情況下，在轉換時將自動地附加 0 後向右填充對齊，變爲 “ 0.□□□□□ ”。



(d) 如果全部位元數的值減去符號、小數點後的位元數仍然大於 BIN 資料的位元數時，將在符號與數值之間存儲“20_H”(空格)。



如果 BIN 資料的位元數大於全部位元數，將變爲出錯狀態。

(e) 在轉換後的字串的最末尾處將自動存儲“00_H”。

出　錯

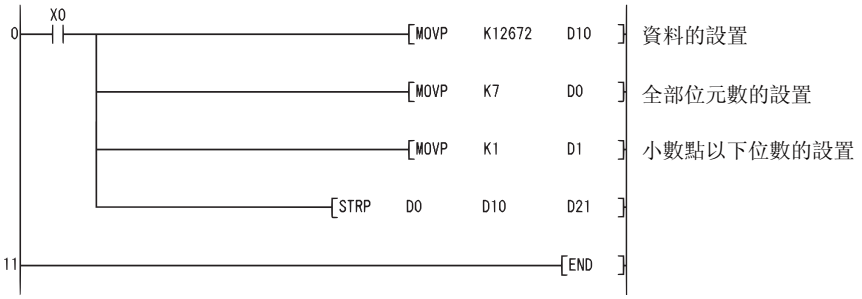
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	⑤1中指定的全部位元數指定超出了以下範圍時。 使用 STR 指令時 .....2 ～ 8 使用 DSTR 指令時 .....2 ～ 13 ⑤1+1 中指定的小數部分位元數指定超出了以下範圍時。 S 使用 STR 指令時 .....0 ～ 5 使用 DSTR 指令時 .....0 ～ 10 ⑤1中指定的全部位元數與⑤1+1 中指定的小數部分位元數指定值的關係不符合下述條件時。 全部位元數 -3 ≥小數部分位元數 ⑤1中指定的位元數小於⑤2中指定的 BIN 資料的位元數 +2 時。 (⑤1的位數 <⑤2的不含符號的 BIN 資料的位元數 + 符號 (+ 或 -) 的位數 + 小數點 (.) 的位數＝)	○	○	○	○	○	○
4101	存儲Ⓓ中指定的字串的軟元件範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D10 中存儲的 BIN16 位元資料按照 D0、D1 的位元數指定轉換爲字串後，存儲到 D20 ～ D23 中的程式。

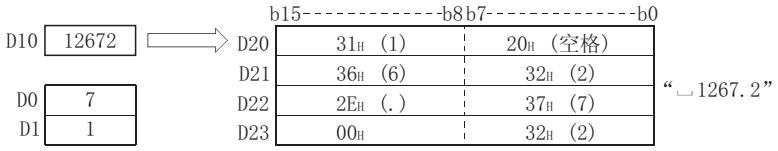
[梯形圖模式]



[列表模式]

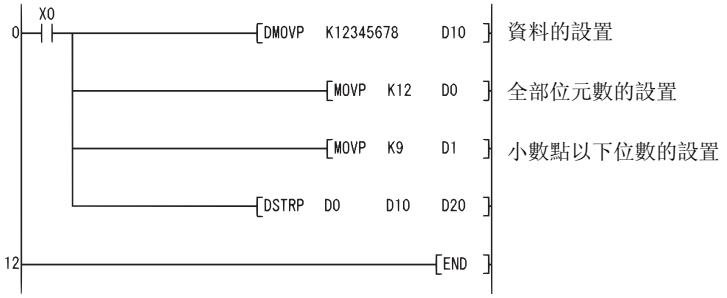
步	指令	軟元件		
0	LD	X0		
1	MOVP	K12672	D10	
3	MOVP	K7	D0	
5	MOVP	K1	D1	
7	STRP	D0	D10	D21
11	END			

[動作]



(2) 以下爲 X0 變爲 ON 時，將 D10、D11 中存儲的 BIN32 位元資料按照 D0、D1 的位元數指定轉換爲字串後，存儲到 D20 ～ D26 中的程式。

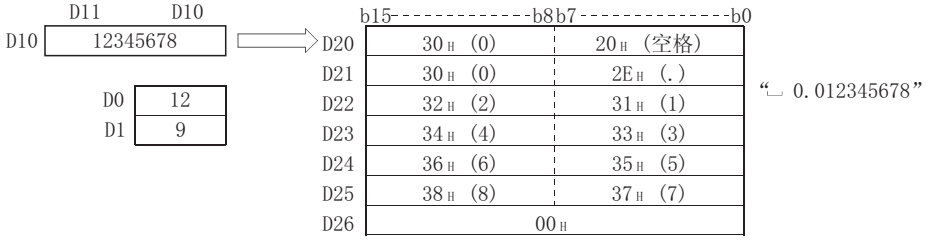
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X0		
1	DMOVP	K12345678		D10
4	MOVP	K12	D0	
6	MOVP	K9	D1	
8	DSTRP	D0	D10	D20
12	END			

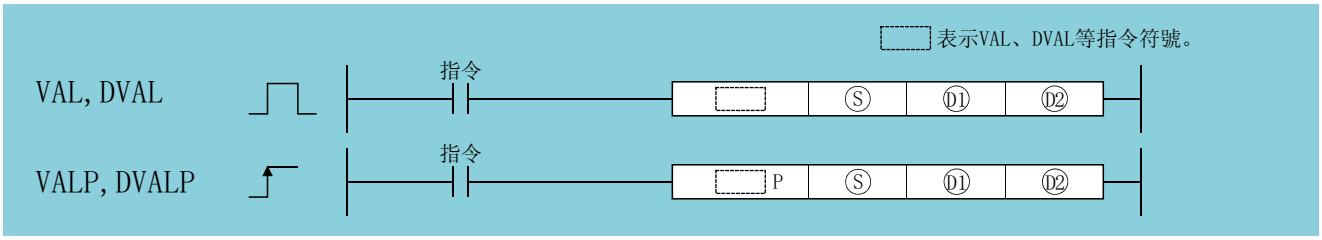
[動作]



Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。
(支持的 GX Developer: Version8.00A 以後)

7.11.10 VAL、VALP、DVAL、DVALP



Ⓢ : 轉換爲 BIN 資料的字串或者存儲字串的軟元件的起始編號 (字串)。

Ⓓ1 : 存儲轉換後的 BIN 資料的位元數的軟元件的起始編號 (BIN16 位)。

Ⓓ2 : 存儲轉換後的 BIN 資料的軟元件的起始編號 (BIN16/32 位)。

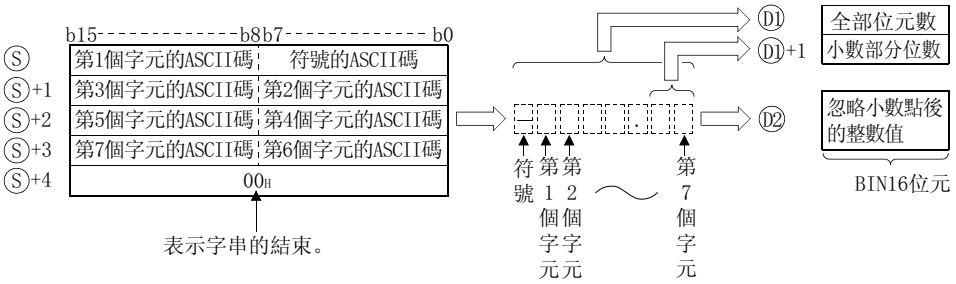
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ1	○	○		---				---	---
Ⓓ2	○	○		○				---	---

功 能

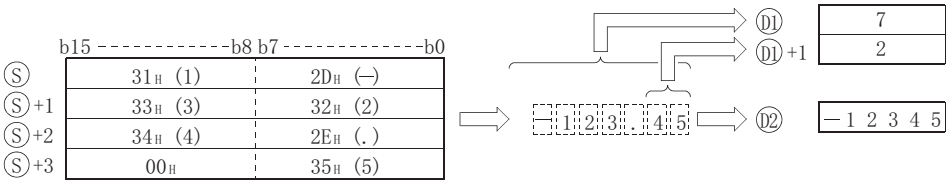
VAL

(1) 將Ⓢ中指定的軟元件編號後面存儲的字串轉換爲 BIN16 位元資料後，將位元數及 BIN 資料存儲到Ⓓ1、Ⓓ2中。

在字串→BIN 轉換中，將Ⓢ中指定的軟元件編號開始至存儲了“ 00_H ”軟元件編號爲止的資料作爲字串處理。



例如，Ⓢ在後面指定了“ -123.45 ”的字串時，將按以下方式存儲到Ⓓ1、Ⓓ2中。



(2) 在Ⓢ中可指定的字串的全部字元數爲 2 ～ 8 個字元。

(3) 在Ⓢ中指定的字串中的小數部分的字元數範圍爲 0 ～ 5 個字元。
但是，應在 (全部位元數 -3) 以下。

(4) 可轉換爲 BIN 值的數值字串的範圍 (小數點被忽略後的值) 爲 -32768 ～ 32767。
此外，符號及小數點除外的數值字串只能在“ 30_H ”～“ 39_H ”的範圍內進行指定。
小數點被忽略後的值如下所示。

例 “ -12345.6 ”→“ -123456 ”

(5) 表示正的數值時在“ 符號 ”中存儲“ 20_H ”，表示負的數值時在“ 符號 ”中存儲“ $2D_H$ ”。

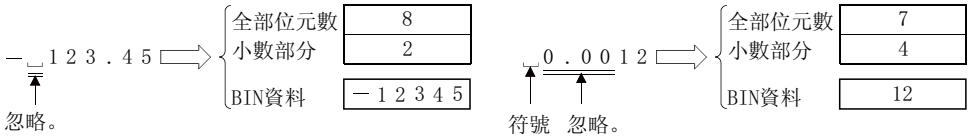
(6) 小數點中設置“ $2E_H$ ”。

(7) Ⓓ1中存儲的全部位元數是表示數值的字元 (包含符號、小數點) 的所有的字元數。

Ⓓ1+1 中存儲的小數部分位元數是表示“ $2E_H$ ”(.) 後面的小數部分的字元數。

Ⓓ2中存儲的 BIN 資料是由忽略了小數點後的字串所轉換成的 BIN 值。

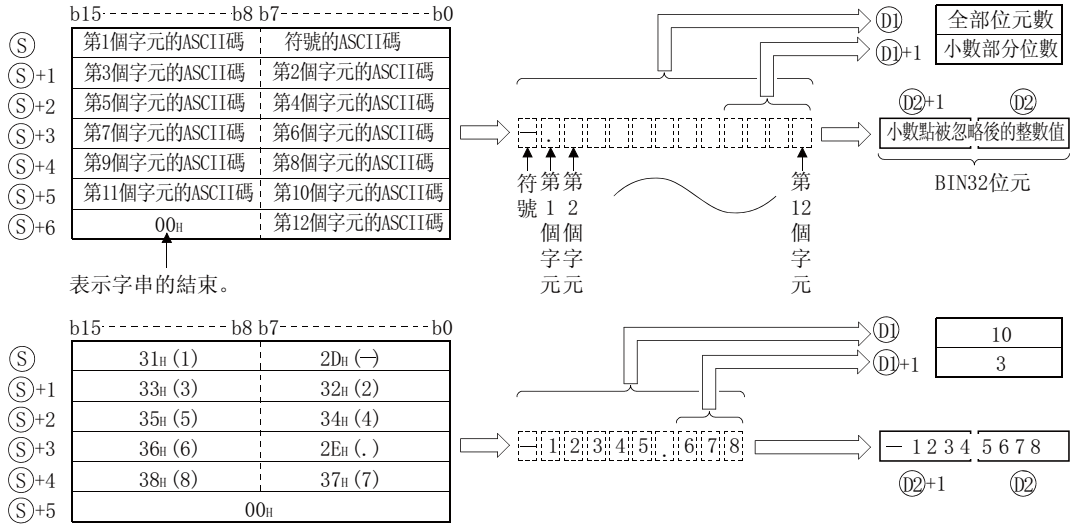
(8) 在Ⓢ中指定的字串中，在符號和第一個非零的數值之間存在有“ 20_H ”(空格) 或“ 30_H ”(0) 時，則在忽略“ 20_H ”、“ 30_H ”的狀況下將其轉換爲 BIN 值。



DVAL

(1) 將Ⓢ中指定的軟元件編號後面存儲的字串轉換爲 BIN32 位元資料後，將位元數及 BIN 資料存儲到Ⓓ1、Ⓓ2中。

在字串→ BIN 轉換中，將Ⓢ中指定的軟元件編號開始至存儲了 “ 00_H ” 軟元件編號爲止的資料作爲字串處理。



(2) 在Ⓢ中可指定的字串的全部字元數爲 2 ～ 13 個字元。

(3) 在Ⓢ中指定的字串中的小數部分的字元數範圍爲 0 ～ 10 個字元。
但是，應在 (全部位元數 -3) 以下。

(4) 可轉換爲 BIN 值的數值字串的範圍 (小數點被忽略後的值) 爲 -2147483648 ～ 2147483647。
此外，符號及小數點除外的數值字串只能在 “ 30_H ” ～ “ 39_H ” 的範圍內進行指定。

(5) 表示正的數值時在 “ 符號 ” 中存儲 “ 20_H ”，表示負的數值時在 “ 符號 ” 中存儲 “ $2D_H$ ”。

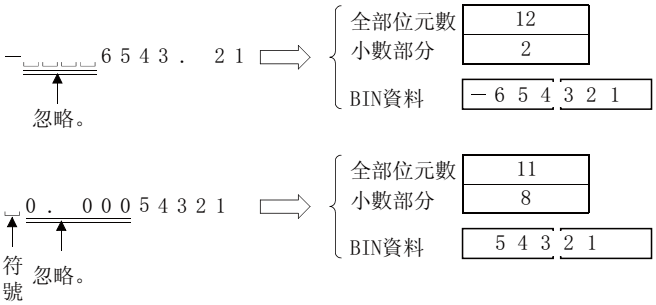
(6) 小數點中設置 “ $2E_H$ ”。

(7) 中存儲的全部位元數是表示數值的字元 (包含符號、小數點) 的所有的字元數。

Ⓓ1+1 中存儲的小數部分位元數是表示 “ $2E_H$ ” (.) 後面的小數部分的字元數。

Ⓓ2中存儲的 BIN 資料是由忽略了小數點後的字串所轉換成的 BIN 值。

(8) 在Ⓢ中指定的字串中，在符號與第一個非零的數值之間存在有 “ 20_H ” (空格) 或 “ 30_H ” (0) 時，則在忽略 “ 20_H ”、“ 30_H ” 的狀況下將其轉換爲 BIN 值。



出　錯

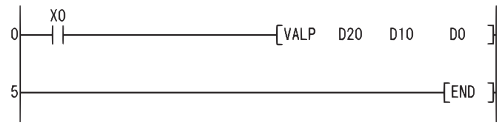
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	ⓢ1中指定的字串的字元數超出了下述範圍時。 使用 VAL 指令時 ..... 2 ～ 8 個字元 使用 DVAL 指令時 .... 2 ～ 13 個字元 ⓢ 中指定的小數部分字元數超出了下述範圍時。 使用 VAL 指令時 ..... 0 ～ 5 個字元 使用 DVAL 指令時 .... 0 ～ 10 個字元 ⓢ 中指定的全部字元數與小數部分字元數的關係超出了下述範圍時。 全部字元數 -3 ≥小數部分字元數 符號中設置了除 “ 20_H ”、“ $2D_H$ ” 以外的 ASCII 碼時。 各數位的位元數中設置了除 “ 30_H ” ～ “ 39_H ” 以及 “ $2E_H$ ” (小數點) 以外的 ASCII 碼時。 設置了多個小數點時。 轉換後的 BIN 值超出了下述範圍時。 使用 VAL 指令時 ..... 32768 ～ 32767 使用 DVAL 指令時 .... -2147483648 ～ 2147483647	○	○	○	○	○	○
4101	在從ⓢ 中指定的軟元件開始至相應軟元件的最終軟元件編號爲止之間未設置 “ 00_H ” 時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D20 ～ D22 中存儲的字串資料視爲整數值轉換爲 BIN 值後，存儲到 D0 中的程式。

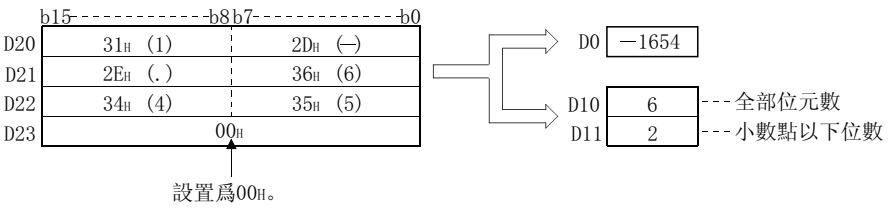
[梯形圖模式]



[列表模式]

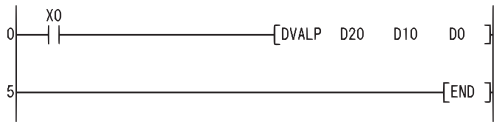
步	指令	軟元件		
0	LD	X0		
1	VALP	D20	D10	D0
5	END			

[動作]



(2) 以下爲 X0 變爲 ON 時，將 D20 ～ D24 中存儲的字串資料視爲整數值轉換爲 BIN 值後，存儲到 D0 中的程式。

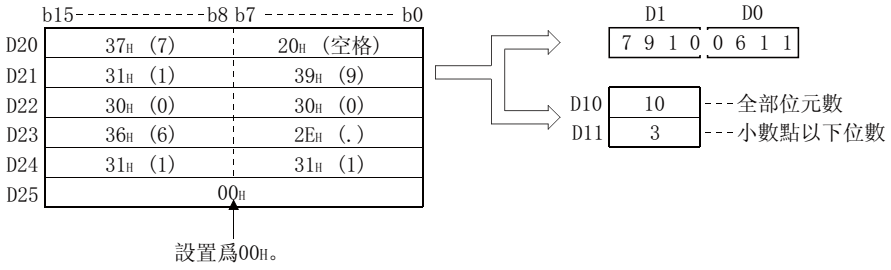
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X0		
1	DVALP	D20	D10	D0
5	END			

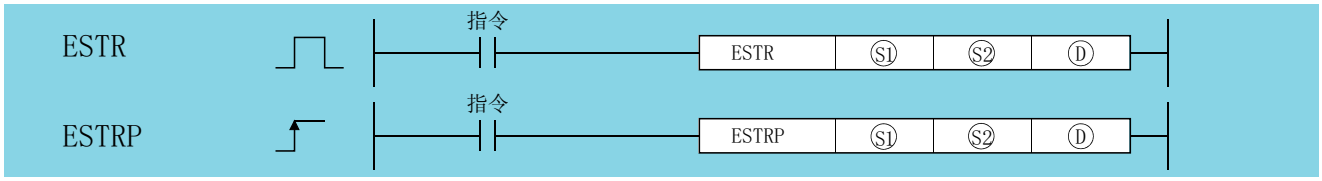
[動作]



7.11.11 ESTR、ESTRP

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。



(S1) ： 進行轉換的 32 位元浮點數據或者存儲資料的軟元件的起始編號 (實數)。

(S2) ： 存儲進行轉換的數值的顯示指定的軟元件的起始編號 (BIN16 位)。

(D) ： 存儲轉換後的字串的軟元件的起始編號 (字串)。

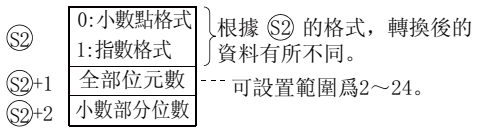
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
(S1)	---	○		---	○		○ *1	○	---
(S2)	---	○		---	---		---	---	---
(D)	---	○		---	---		---	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

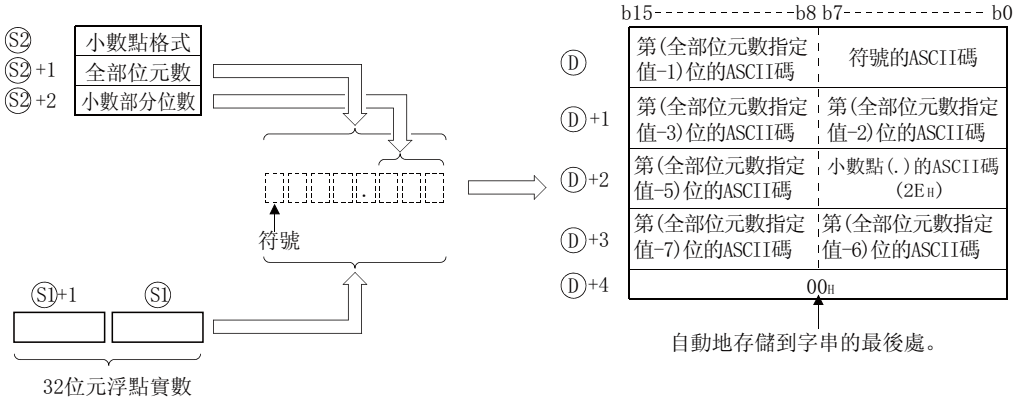
功 能

(1) 將(S1)中指定的 32 位元浮點實數資料按照(S2)中指定的顯示指定轉換爲字串後，存儲到(D)中指定編號的軟元件的後面。

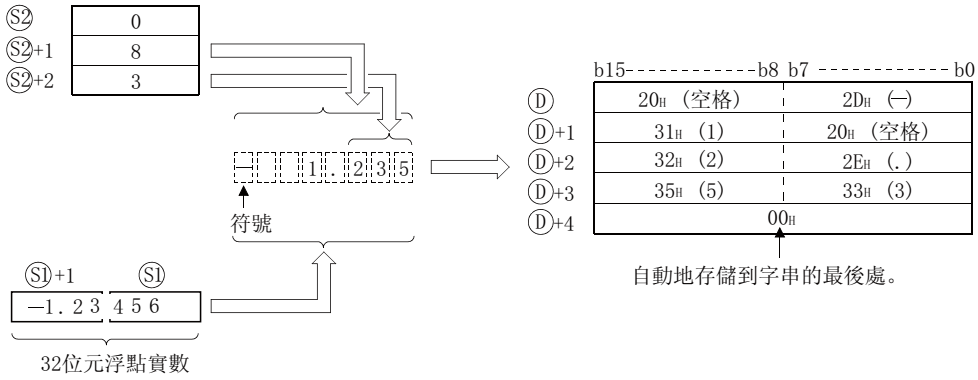
(2) 根據(S2)中指定的顯示指定，轉換後的資料有所不同。



小數點格式時



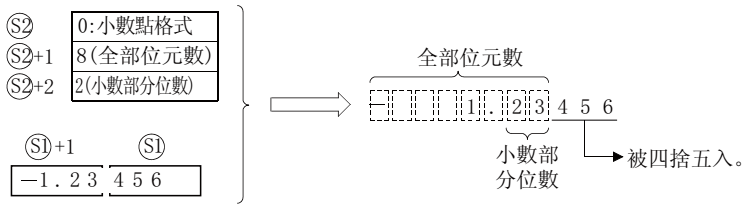
例如，全部位元數爲 8，小數部分位元數爲 3 時，指定了 -1.23456 的情況下，按以下方式儲存到Ⓓ的後面。



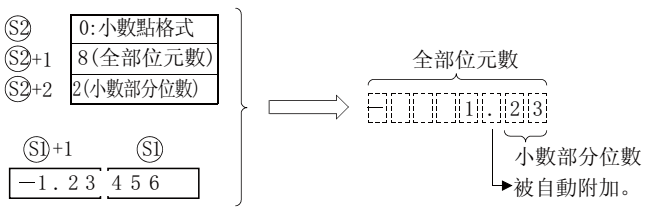
(a) Ⓢ2+1 中可指定的全部位元數如下所示：
小數部分位元數爲 “ 0 ” 時 ..... 位數 (最大： 24) ≥ 2
小數部分位元數爲 “ 0 ” 以外時 ..... 位數 (最大： 24) ≥ (小數部分位元數 +3)

(b) Ⓢ2+2 中可指定的小數部分位元數爲 0 ～ 7 位。
但是，設置時應滿足以下條件： 小數部分位元數≤ (全部位元數 -3)。

(c) 轉換後的字串資料按以下方式存儲到Ⓓ後面的軟元件編號中。
1) 32 位元浮點實數資料爲正數時在 “ 符號 ” 中存儲 “ 20_H ” (空格)，爲負數時在 “ 符號 ” 中存儲 “ $2D_H$ ” (-)。
2) 小數部分位元數的範圍中，不能容納 32 位元浮點實數資料的小數部分時，低位元小數部分將被四捨五入。

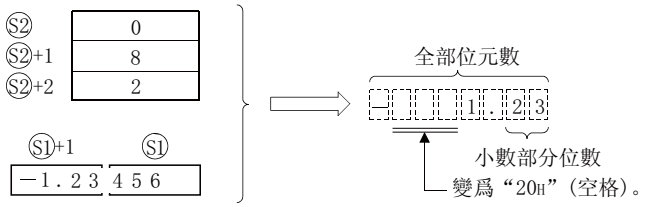


3) 將小數部分位元數設置爲 “ 0 ” 以外時，在指定的小數部分位元數 +1 位元中將自動存儲 “ $2E_H$ ” (.)。



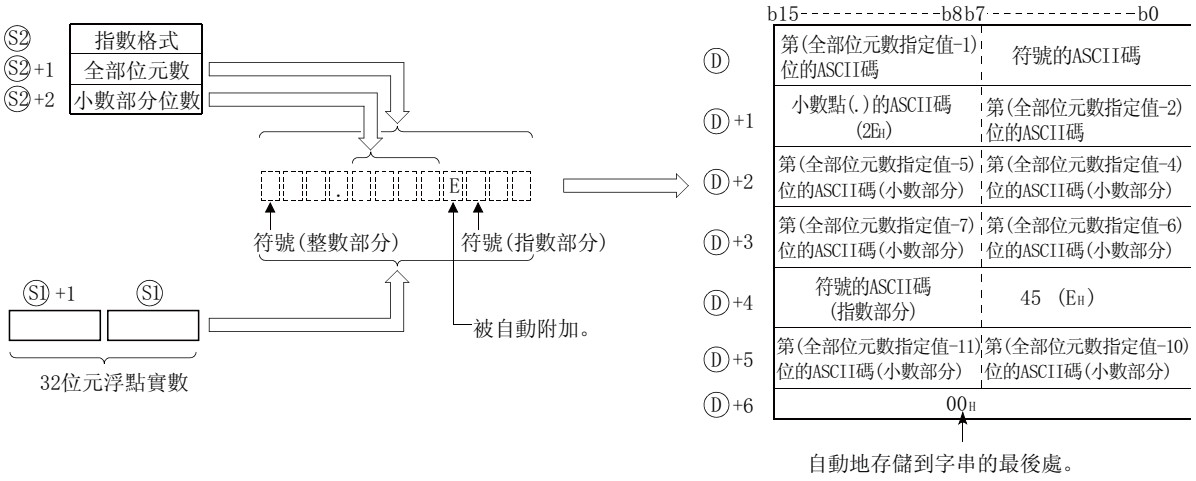
小數部分位元數爲 “ 0 ” 時，不存儲 “ $2E_H$ ” (.)。

4) 從全部位元數中減去符號、小數點、小數部分後的位元數大於 32 位元浮點實數資料的整數部分時，在符號與整數部分之間將存儲 “ 20_H ”(空格)。

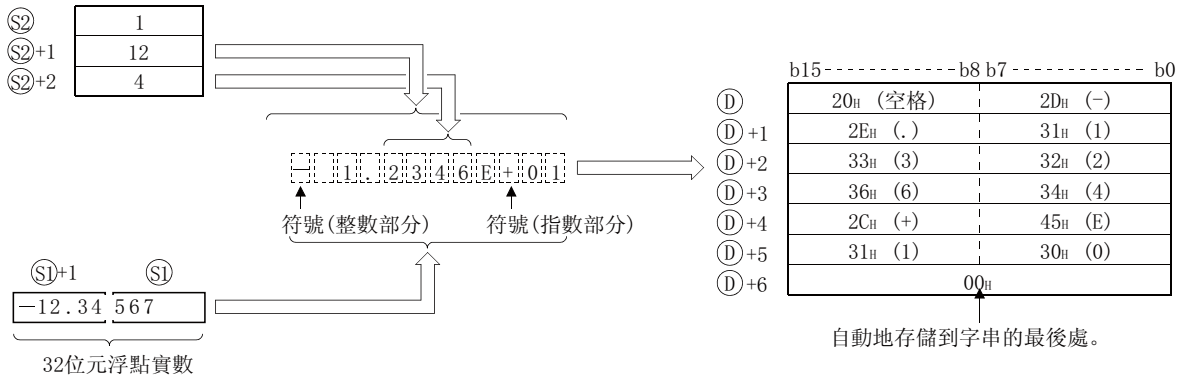


5) 轉換後的字串的末尾處將自動存儲 “ 00_H ”。

指數格式時



例如，全部位元數爲 12，小數部分位元數爲 4 時，指定了 -12.34567 的情況下，按以下方式儲存到、Ⓓ的後面。

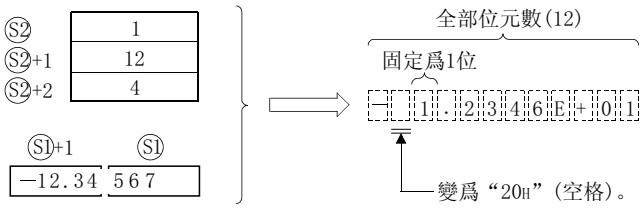


(a) Ⓢ2+1 中可指定的全部位元數如下所示：
小數部分位元數爲 “ 0 ” 時 ..... 位數 (最大： 24) ≥ 2
小數部分位元數爲 “ 0 ” 以外時 ..... 位數 (最大： 24) ≥ (小數部分位元數 +7)

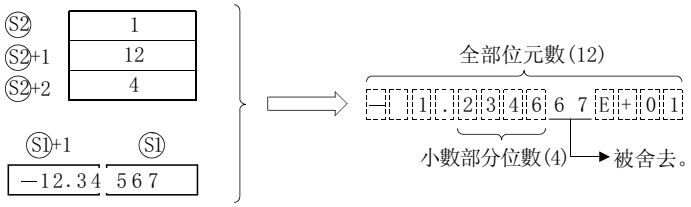
(b) Ⓢ2+2 中可指定的小數部分位元數爲 0 ～ 7 位。
但是，設置時應滿足以下條件： 小數部分位元數≤ (全部位元數 -7)。

(c) 轉換後的字串資料按以下方式存儲到Ⓓ後面的軟元件編號中。
1) 32 位元浮點實數資料爲正數時，在整數部分的 “ 符號 ” 中存儲 “ 20_H ”(空格)，爲負數時在整數部分的 “ 符號 ” 中存儲 “ $2D_H$ ”(-)。

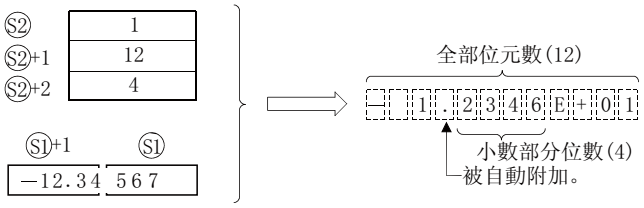
2) 整數部分固定爲 1 位元。
在整數部分與符號之間存儲 “ 20_H ”(空格)。



3) 小數部分位元數的範圍中，不能容納 32 位元浮點實數資料的小數部分時，低位元小數部分將被四捨五入。



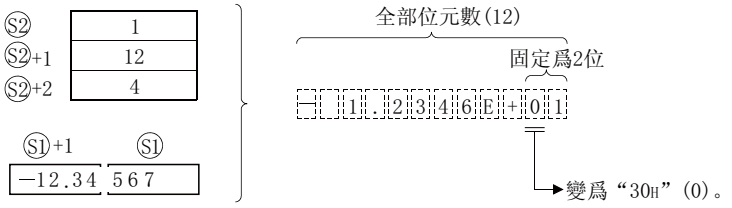
4) 將小數部分位元數設置爲 “ 0 ” 以外時，在指定的小數部分位元數 +1 位元中將自動存儲 “ $2E_H$ ” (.)。



小數部分位元數爲 “ 0 ” 時，不存儲 “ $2E_H$ ” (.)。

5) 指數爲正數時，在指數部分的 “ 符號 ” 中存儲 “ $2C_H$ ” (+)，爲負數時在 “ 符號 ” 中存儲 “ $2D_H$ ” (-)。

6) 指數部分固定爲 2 位元。
指數部分爲 1 位元時，在與指數部分的符號之間存儲 “ 30_H ” (0)。



7) 轉換後的字串的末尾處將自動存儲 “ 00_H ”。

出　錯

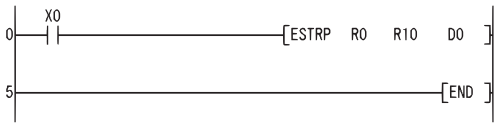
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	⑤超出了下述範圍時。 $0,2^{-126} \leqq \lvert ⑤ \rvert < 2^{128}$ ⑥中指定的格式指定爲 0、1 以外時。 ⑥+1 中指定的全部位元數指定超出了下述範圍時。 小數點格式時 小數部分位元數爲 “ 0 ” 時 .....全部位元數≥ 2 小數部分位元數爲 “ 0 ” 以外時..... 全部位元數≥ (小數部分位元數 +3) 指數格式時 小數部分位元數爲 “ 0 ” 時..... 全部位元數≥ 6 小數部分位元數爲 “ 0 ” 以外時..... 全部位元數≥ (小數部分位元數 +7) ⑥+2 中指定的小數部分位元數指定超出了下述範圍時。 小數點格式時 ..... 小數部分位元數≤(全部位元數-3) 指數格式時 ..... 小數部分位元數≤ (全部位元數 -7) 全部位元數中指定了超出 “ 24 ” 的值時。	○	○	○	○	○	○
4101	存儲⑦中指定的字串的軟元件範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○
	⑥中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○
4140	指定軟元件的內容爲 -0、非正規數、非數、±∞時。	○	○	○	○	○	○

程式示例

(1)　以下爲 X0 變爲 ON 時，將 R0、R1 中存儲的 32 位元浮點實數資料，按照 R10 ～ R12 中存儲的轉換指定進行轉換後，存儲到 D0 後面的程式。

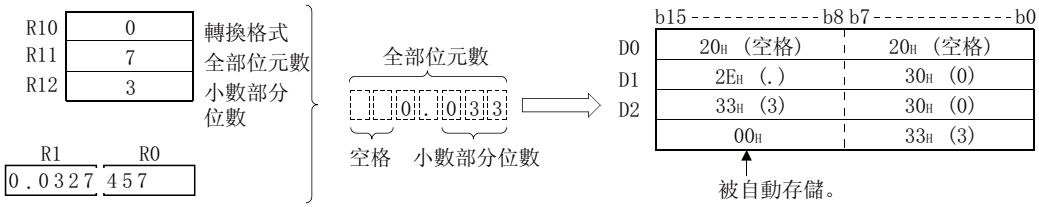
[梯形圖模式]



[列表模式]

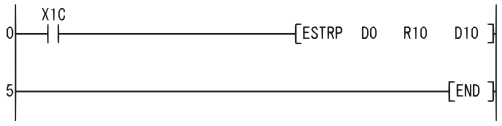
步	指令	軟元件		
0	LD	X0		
1	ESTRP	R0	R10	D0
5	END			

[動作]



(2)　以下爲 X1C 變爲 ON 時，將 D0、D1 中存儲的 32 位元浮點實數資料，按照 R10 ～ R12 中存儲的轉換指定進行轉換後，存儲到 D10 後面的程式。

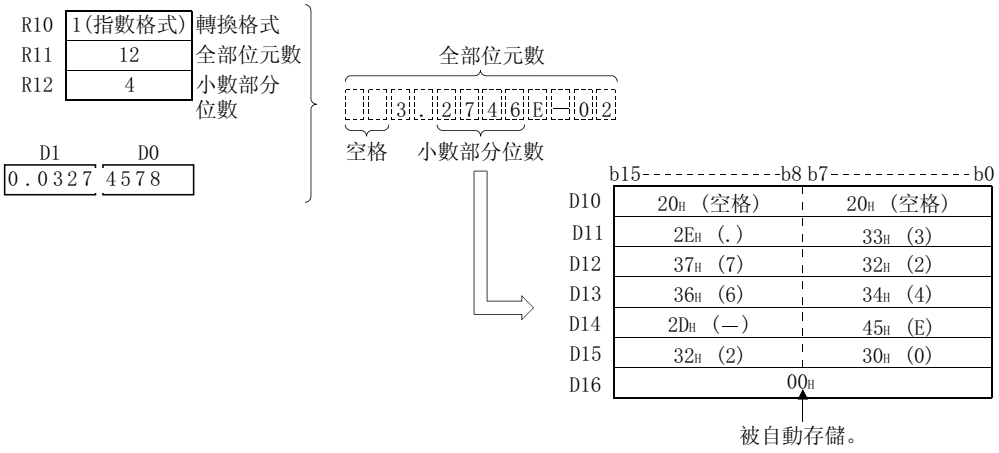
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X1C		
1	ESTRP	D0	R10	D10
5	END			

[動作]



7.11.12 EVAL、EVALP

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

Ⓢ ： 要轉換爲 32 位元浮點實數資料的字串資料或者存儲字串資料的軟元件的起始編號 (字串)。

Ⓓ ： 存儲轉換後的 32 位元浮點實數資料的軟元件的起始編號 (實數)。

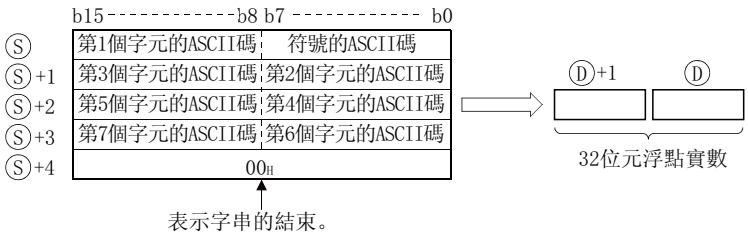
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---	---		---	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

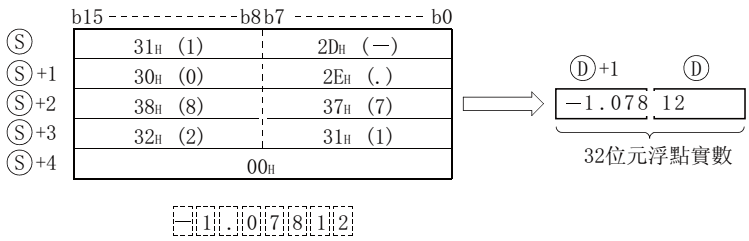
功 能

(1) 將Ⓢ中指定的軟元件編號後面存儲的字串轉換爲 32 位元浮點實數後，存儲到Ⓓ中指定的軟元件中。

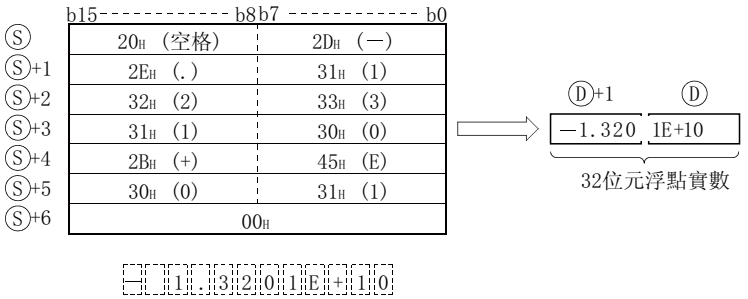
(2) 無論指定的字串是小數點格式還是指數格式，均可轉換爲 32 位元浮點實數資料。



(a) 小數點格式時

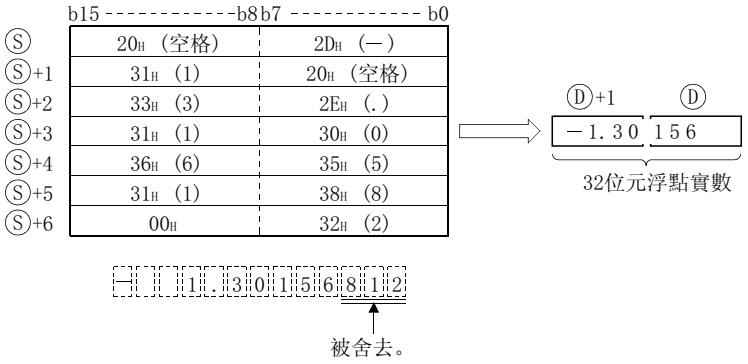


(b) 指數格式時

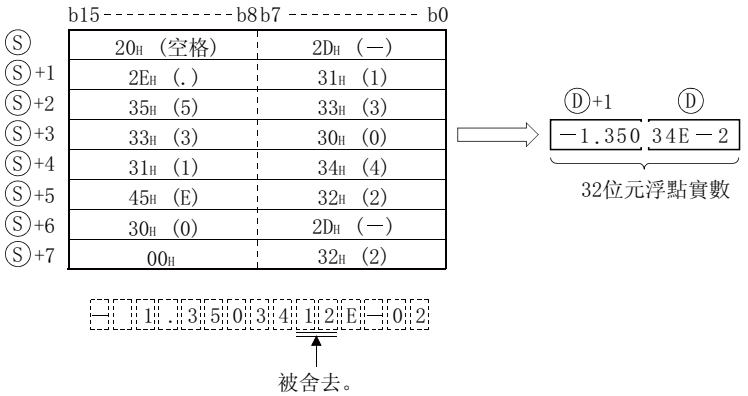


(3) 在Ⓢ中指定的字串中，要轉換爲 32 位元浮點實數的字串在除符號、小數點、指數部分外的 6 位數有效，在轉換時從第 7 位以後將被舍去。

(a) 小數點格式時



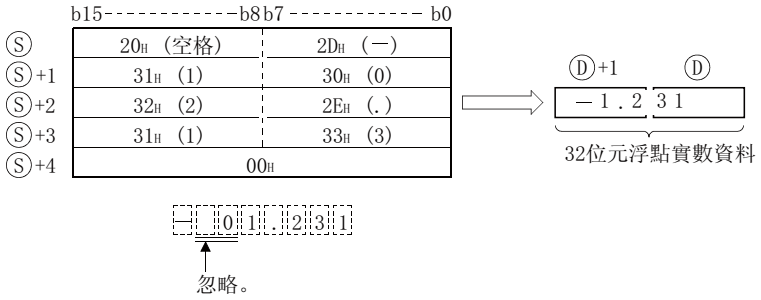
(b) 指數格式時



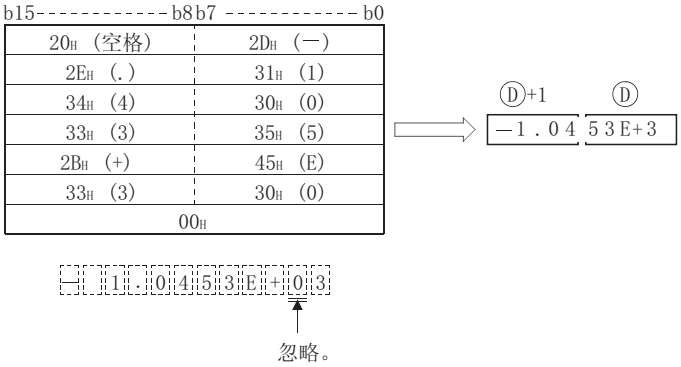
(4) 小數點格式中如果符號中指定了“ $2B_H$ ”(+)，或者符號被省略，則作爲正值進行轉換。
此外，如果符號中指定了“ $2D_H$ ”(-)，則作爲負值進行轉換。

(5) 指數格式中如果符號中指定了“ $2B_H$ ”(+)，或者符號被省略，則作爲正值進行轉換。
如果符號中指定了“ $2D_H$ ”(-)，則作爲負值進行轉換。

(6) 在Ⓢ中指定的字串中，在符號和第一個非零的數值之間存在有“ 20_H ”(空格) 或“ 30_H ”(0) 時，則在忽略“ 20_H ”、“ 30_H ”的狀況下進行轉換。



(7) 在指數格式的字串中，在“E”與數值之間存在有，“30_H”(0) 時，則在忽略“30H”的狀況下進行轉換。



(8) 在字串中包含有“20_H”(空格) 時，將在忽略“20_H”的狀況下進行轉換。

(9) 字串最多可設置 24 個字元。
字串中“20_H”(空格)、“30_H”(0) 也作爲 1 個字元計數。

出　錯

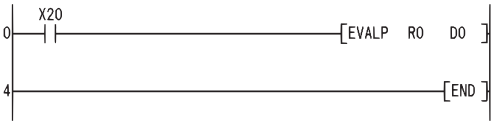
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	整數部分、小數部分中存在有除“30_H”(0) ～“39_H”(9) 以外的字元時。 Ⓓ中指定的字串中存在有 2 個或以上的“$2E_H$”(.) 時。 指數部分中存在有除“45_H”(E)、“$2B_H$”(+)、“45_H”(E)、“$2D_H$”(-) 以外的字元時，或者有多個指數部分時。 轉換後的資料不在下述範圍內時。 0、$2^{-126} \leq$ | 轉換後的資料 | $< 2^{128}$ Ⓢ後面的字元數爲 0 或者超過 24 個字元時。	○	○	○	○	○	○
4101	從Ⓢ開始的相應軟元件範圍內沒有“00_H”時。	○	○	○	○	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 R0 後面存儲的字串轉換爲 32 位元浮點實數後，存儲到 D0、D1 中的程式。

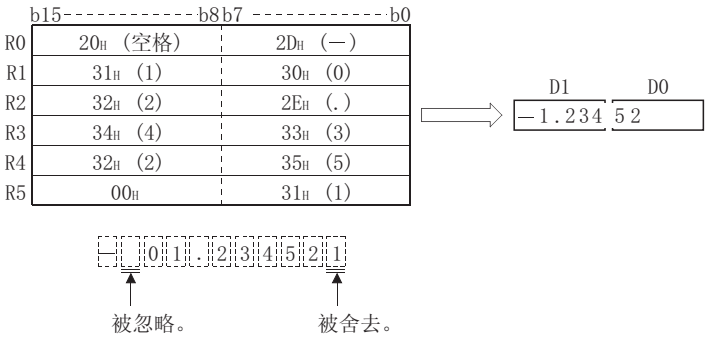
[梯形圖模式]



[列表模式]

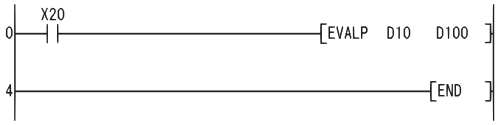
步	指令	軟元件	
0	LD	X20	
1	EVALP	R0	D0
4	END		

[動作]



(2) 以下爲 X20 變爲 ON 時，將 D10 後面存儲的字串轉換爲 32 位元浮點實數後，存儲到 D100、D101 中的程式。

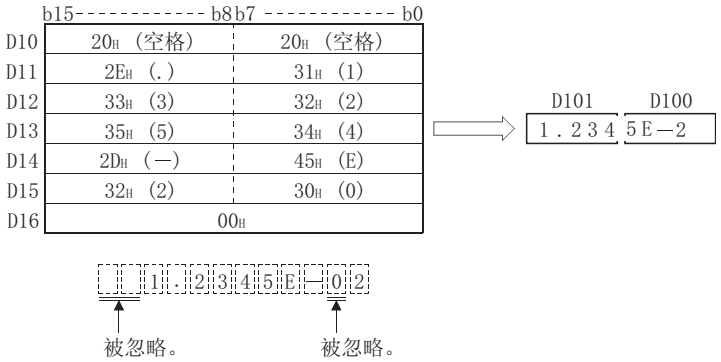
[梯形圖模式]



[列表模式]

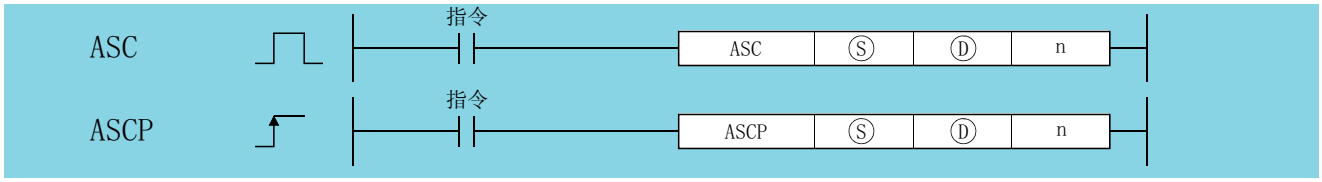
步	指令	軟元件	
0	LD	X20	
1	EVALP	D10	D100
4	END		

[動作]



7.11.13 ASC、ASCP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ： 存儲要轉換爲字串的 BIN 資料的軟元件的起始編號 (BIN16 位)。

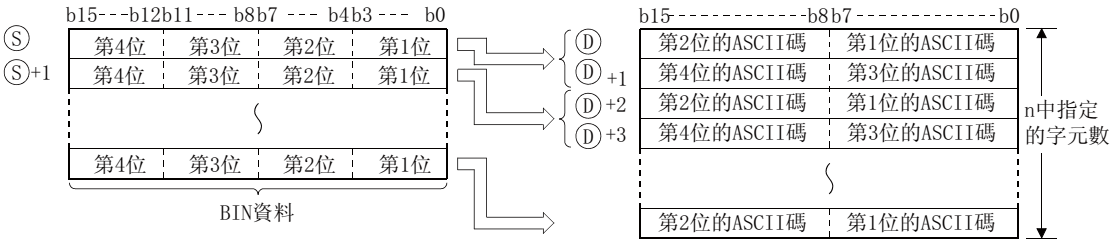
Ⓓ ： 存儲轉換後的字串的軟元件的起始編號 (字串)。

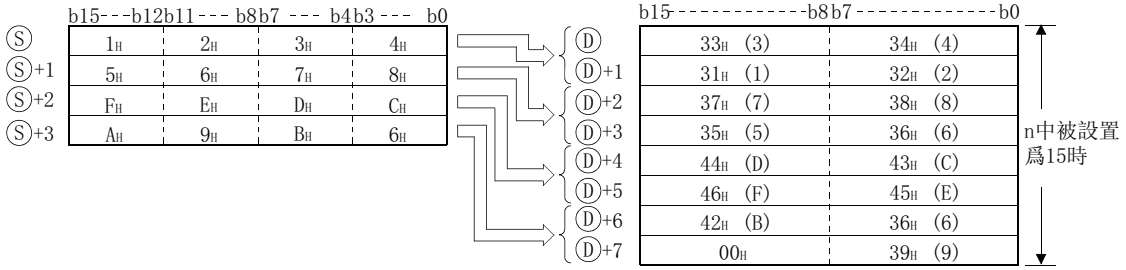
n ： 存儲的字元數 (BIN16 位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---					---
Ⓓ	---	○		---					---
n	○	○		○					---

功　能

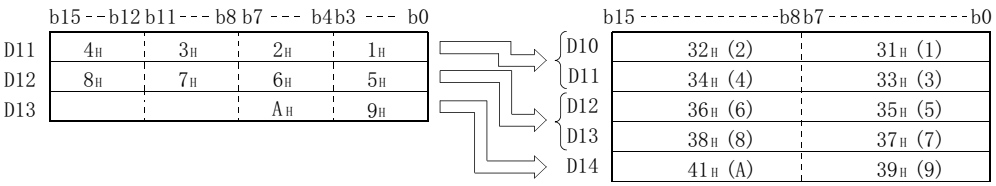
(1) 將Ⓢ中指定的軟元件編號後面存儲的 BIN16 位元資料，通過 16 進制數處理轉換爲 ASCII 後，以 n 中指定的字元數範圍存儲到Ⓓ中指定的軟元件編號後面。





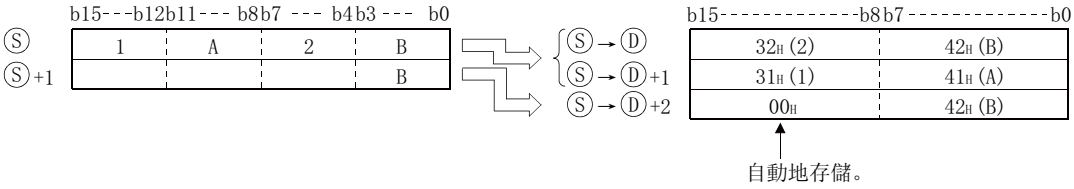
(2) 通過在 n 中設置字元數，Ⓢ中指定的 BIN 資料的範圍以及Ⓓ中指定的字串的存儲軟元件的範圍將被自動確定。

(3) 即使存儲要轉換的 BIN 資料的軟元件範圍與存儲轉換後的 ASCII 資料軟元件範圍重復時，也可正常進行處理。



(4) n 中指定的字元數爲奇數時，存儲字串的軟元件範圍的最終軟元件編號的高 8 位元中將自動存儲“ 00_H ”。

n 的字元數爲 5 時



(5) n 中指定的字元數爲“ 0 ”時，不執行轉換處理。

出　錯

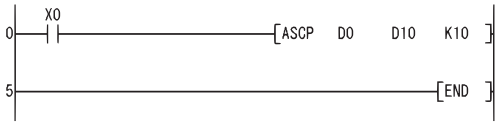
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓢ中指定的軟元件編號後面的 n 中指定的字元數範圍超出了相應軟元件的範圍時。 Ⓓ中指定的軟元件編號後面的 n 中指定的字元數範圍超出了相應軟元件的範圍時。	---	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D0 中存儲的 BIN 資料視爲 16 進制數轉換爲字串後，存儲到 D10 ～ D14 中的程式。

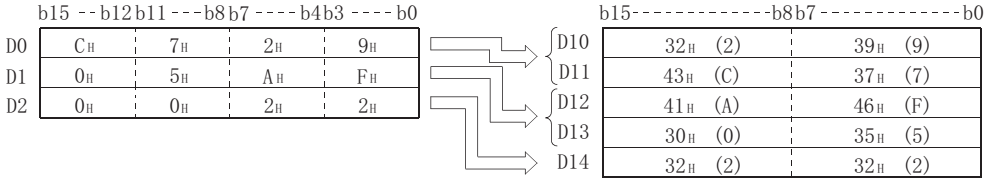
[梯形圖模式]



[列表模式]

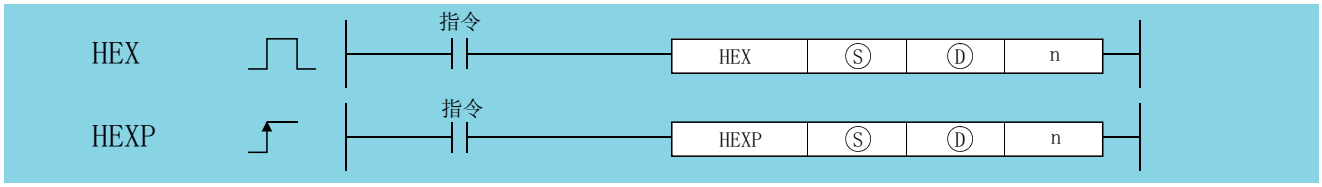
步	指令	軟元件		
0	LD	X0		
1	ASCP	D0	D10	K10
5	END			

[動作]



7.11.14 HEX、HEXP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：儲要轉換爲 BIN 資料的字串的軟元件的起始編號 (字串)。

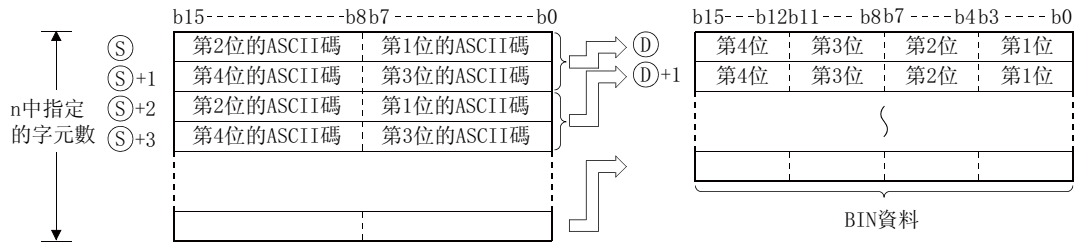
Ⓓ ：存儲轉換後的 BIN 資料的軟元件的起始編號 (BIN16 位)。

n ：存儲的字元數 (BIN16 位元)。

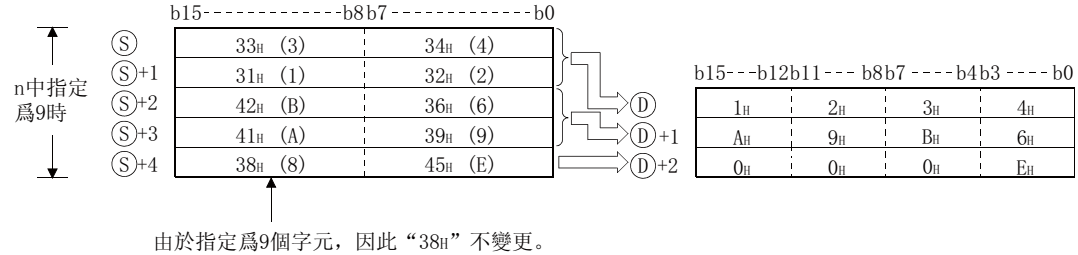
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---					---
Ⓓ	---	○		---					---
n	○	○		○					---

功　能

(1) 將存儲在Ⓢ中指定的軟元件編號後面的 n 中指定的字元數的 16 進制 ASCII 資料轉換爲 BIN 值後，存儲到Ⓓ中指定的軟元件編號的後面。

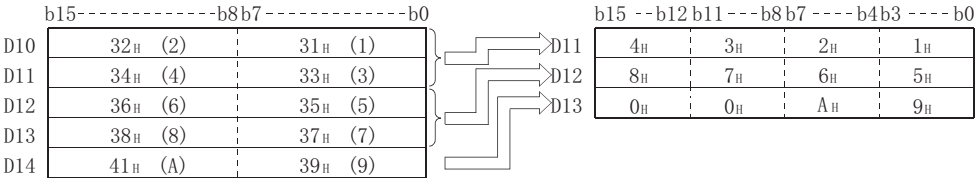


例如，n 中指定了 9 時，其情況如下所示。

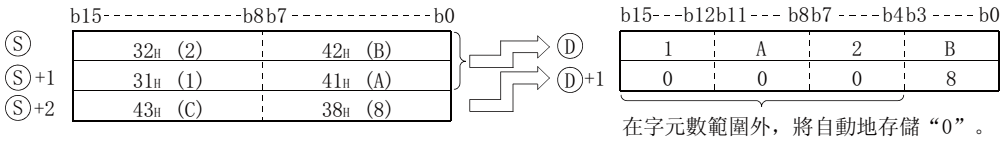


(2) 在 n 中設置了字元數時，存儲Ⓢ中指定的字串的範圍以及Ⓓ中指定的 BIN 資料的軟元件範圍將被自動確定。

(3) 即使存儲要轉換的 ASCII 資料的軟元件範圍與存儲轉換後的 BIN 資料軟元件範圍重復時，也可正常進行處理。



(4) n 中指定的字元數不是 4 的倍數時，在存儲轉換後的 BIN 值的軟元件編號中，最終軟元件編號的指定字元數後面的位元數中將自動地存儲“ 0 ”。



(5) n 中指定的字元數爲 “ 0 ” 時，不執行轉換處理。

(6) Ⓢ中可指定的 ASCII 碼在 “ 30_H ” ～ “ 39_H ” ，“ 41_H ” ～ “ 46_H ” 的範圍內。

出　錯

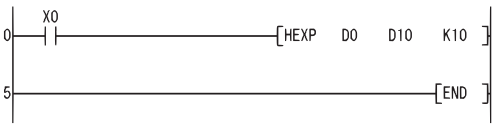
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中設置了除 16 進制數值的字串以外的字元 (“ 30_H ” ～ “ 39_H ” 、“ 41_H ” ～ “ 46_H ” 以外的 ASCII 碼) 時。	---	○	○	○	○	○
4101	在Ⓢ中指定的軟元件編號後面的 n 中指定字元數的範圍超出了相應軟元件的範圍時。 在Ⓓ中指定的軟元件編號後面的 n 中指定字元數的範圍超出了相應軟元件的範圍時。 n 中指定的字元數爲負數時。	---	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D0 ～ D4 中存儲的字串資料轉換爲 BIN 資料後，存儲到 D10 ～ D14 中的程式。

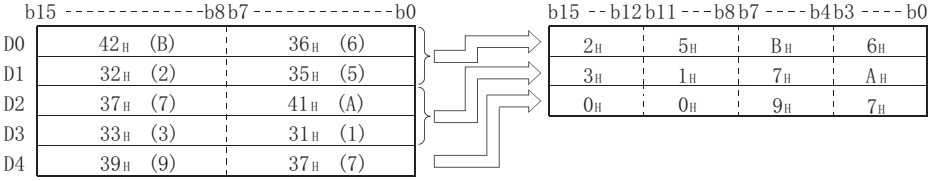
[梯形圖模式]



[列表模式]

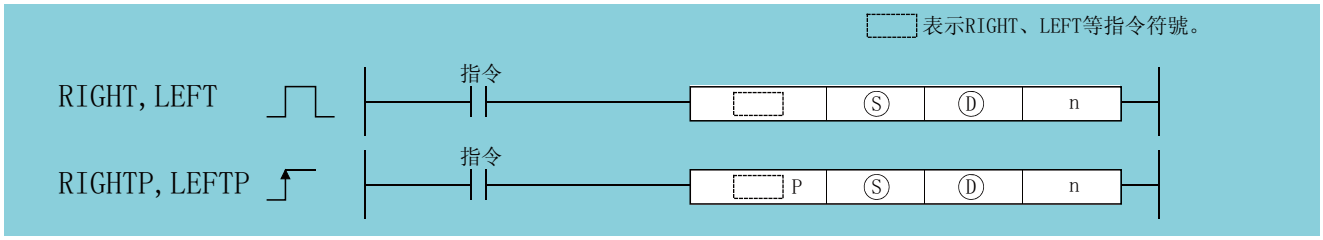
步	指令	軟元件		
0	LD	X0		
1	HEXP	D0	D10	K10
5	END			

[動作]



7.11.15 RIGHT、RIGHTP、LEFT、LEFTP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：字串或者存儲字串的軟元件的起始編號（字串）。

Ⓓ ：存儲從Ⓢ的右側或者左側開始的 n 個字元的字串的軟元件的起始編號（字串）。

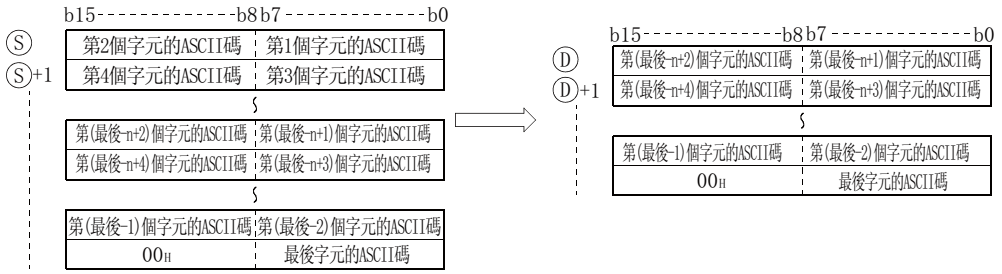
n ：提取的字元數 (BIN16 位元)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數		其他
	位	字		位	字			K、H	$	
Ⓢ	---	○		---				---	○	---
Ⓓ	---	○		---				---	---	---
n	○	○		○				○	---	---

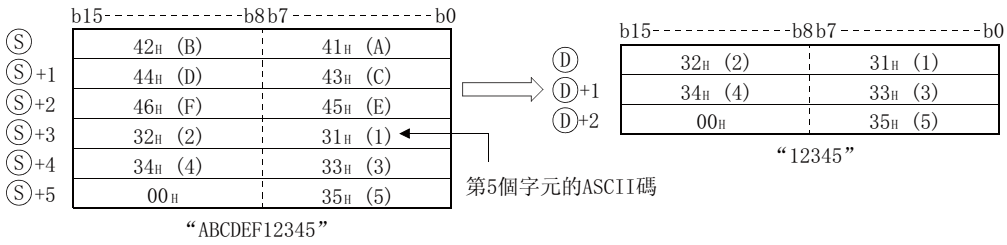
功　能

RIGHT

(1) 從Ⓢ中指定的軟元件編號後面存儲的字串資料的右（字串的最後）側開始，將 n 個字元的資料存儲到Ⓓ中指定的軟元件編號的後面。



n=5 時

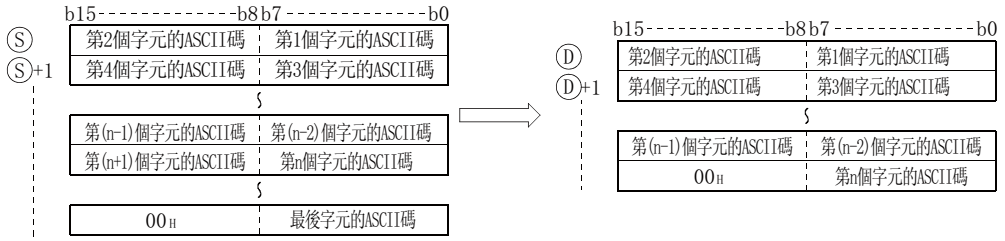


(2) 表示字串的結尾的 NULL 碼 (00_H) 將被自動附加到字串的結尾處。
關於字串資料的格式，請參閱 87 頁 3.2.5 項。

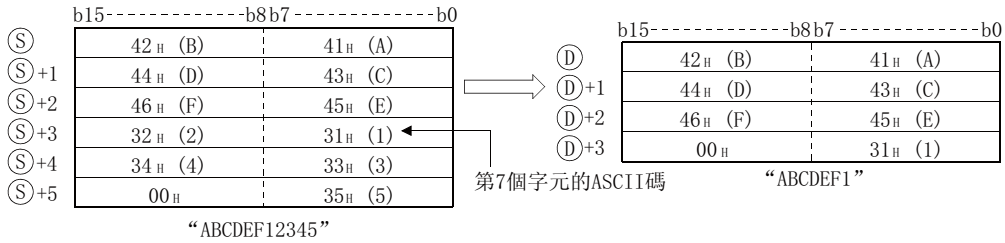
(3) n 中指定的字元數爲 “ 0 ” 時，則Ⓓ中將存儲 NULL 碼 (00_H)。

LEFT

(1) 從Ⓢ中指定的軟元件編號後面存儲的字串資料的左（字串的起始）側開始，將 n 個字元的資料存儲到Ⓓ中指定的軟元件編號的後面。



n=7 時



(2) 表示字串的結尾的 NULL 碼 (00_H) 將被自動附加到字串的結尾處。
關於字串資料的格式，請參閱 87 頁 3.2.5 項。

(3) n 中指定的字元數爲 " 0 " 時，則Ⓓ中將存儲 NULL 碼 (00_H)。

出　錯

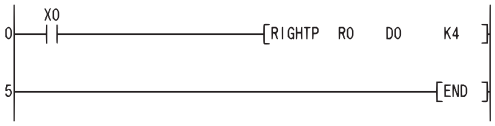
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	n 超過了Ⓢ中指定的字元數時。 從Ⓓ開始的 n 個字元的範圍超出了相應軟元件的範圍時。	---	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將存儲在 R0 後面的字串資料的右側開始的 4 個字元的資料存儲到 D0 的後面。

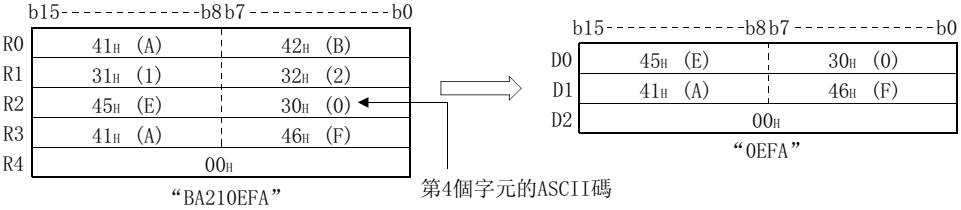
[梯形圖模式]



[列表模式]

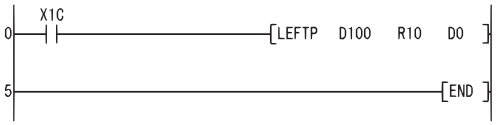
步	指令	軟元件		
0	LD	X0		
1	RIGHTP	R0	D0	K4
5	END			

[動作]



(2) 以下爲 X1C 變爲 ON 時，從存儲在 D100 後面的字串資料的左側開始，按 D0 中存儲的字元數將資料存儲到 R10 後面的程式。

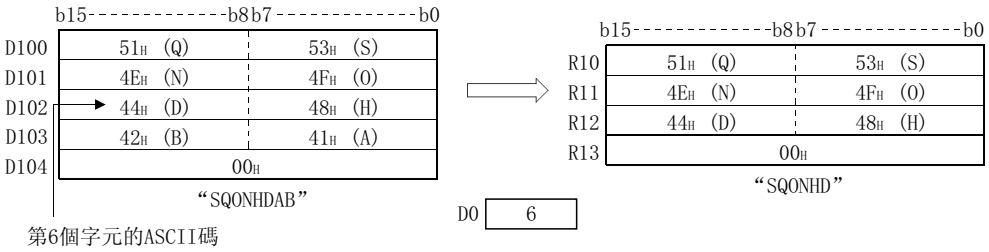
[梯形圖模式]



[列表模式]

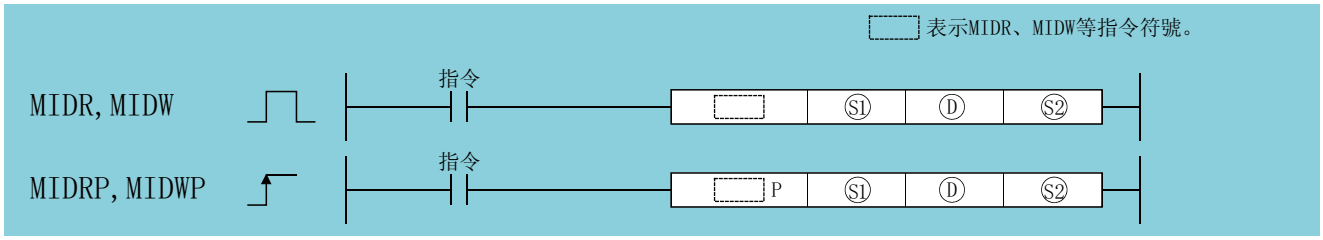
步	指令	軟元件		
0	LD	X1C		
1	LEFTP	D100	R10	D0
5	END			

[動作]



7.11.16 MIDR、MIDRP、MIDW、MIDWP

Basic | High performance | Process | Redundant | Universal | LCPU



(S1) ：字串或者存儲字串的軟元件的起始編號 (字串)。

(D) ：存儲運算結果的字串資料的軟元件的起始編號 (字串)。

(S2) ：存儲起始字元的位置以及字元數的軟元件的起始編號 (BIN16 位)。

·(S2)：起始字元的位置

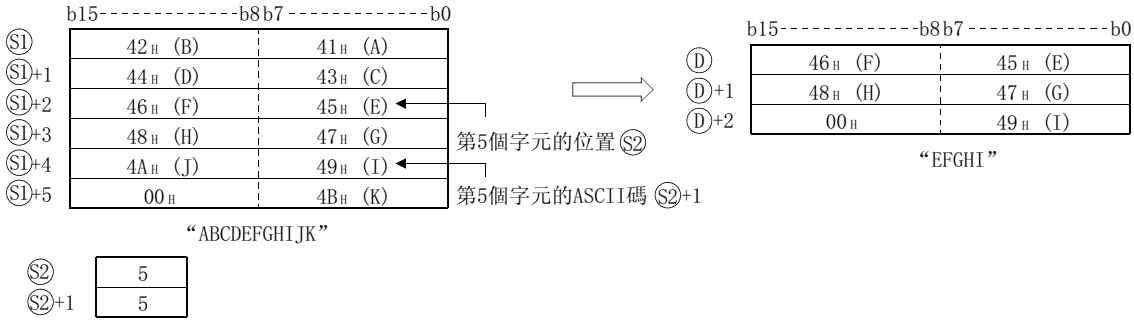
·(S2)+1：字元數

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
(S1)	---	○		---				○	---
(D)	---	○		---				---	---
(S2)	○	○		○				---	---

功　能

MIDR

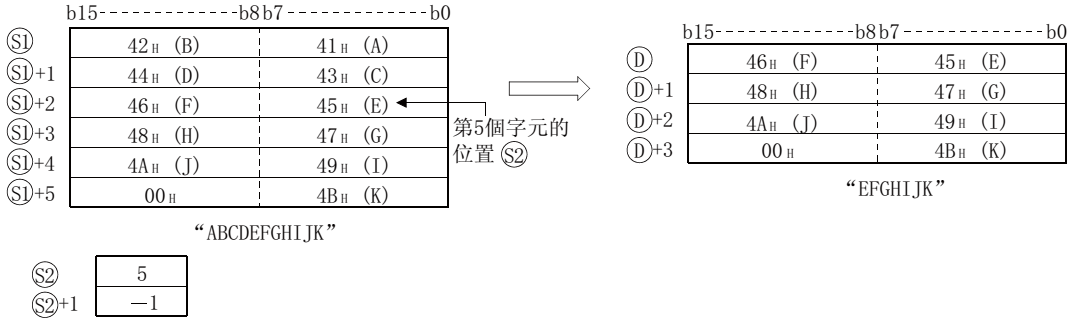
(1) 從(S1)中指定的字串資料的左側開始，從(S2)中指定的位置開始將(S2)+1 中中指定的字元數的資料存儲到(D)中中指定的軟元件編號的後面。



(2) 表示字串的結尾的 NULL 碼 (00_H) 將被自動附加到字串的結尾處。
關於字串資料的格式，請參閱 87 頁 3.2.5 項。

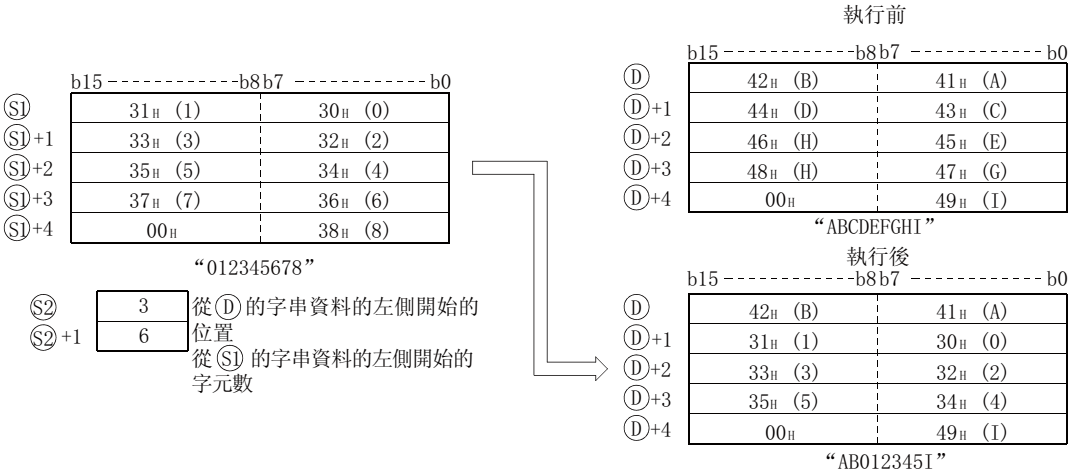
(3) ⓢ2+1 指定的字元數爲 “ 0 ” 時，不執行處理。

(4) ⓢ2+1 中指定的字元數爲 “ -1 ” 時，將至ⓢ中指定的最後字元資料爲止的資料存儲到ⓓ中指定的軟元件後面。



MIDW

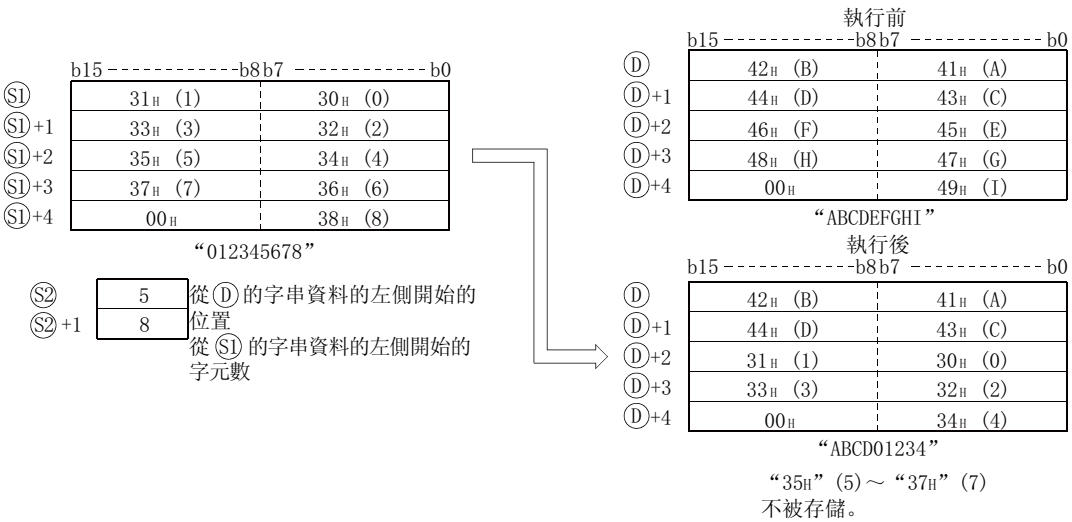
(1) 從ⓢ1中指定的字串資料的左側開始，將ⓢ2+1 中指定字元數的資料，存儲到從ⓓ中指定的字串資料左側開始的ⓢ2中中指定的位置後面。



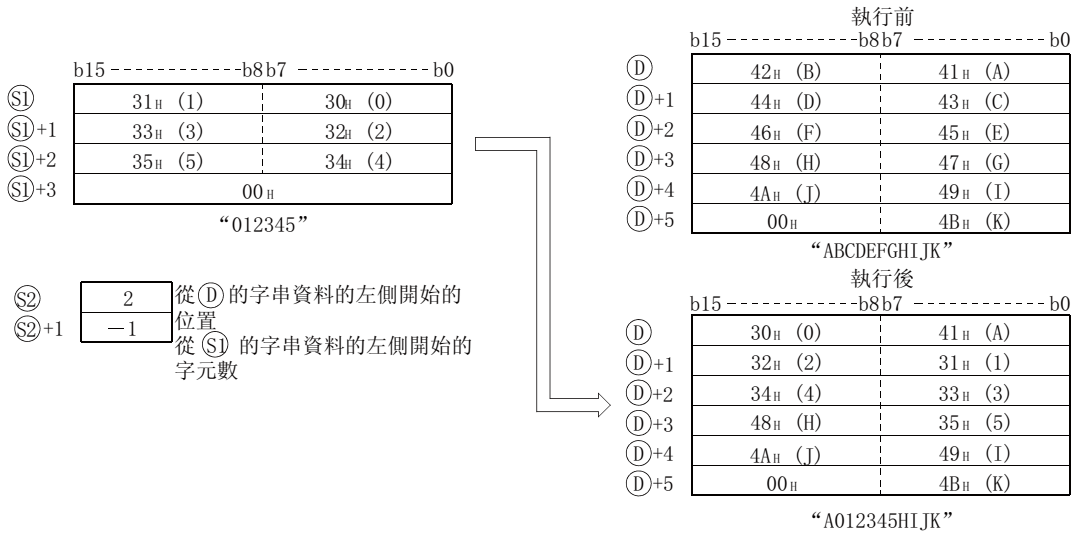
(2) 表示字串的結尾的 NULL 碼 (00_H) 將被自動附加到字串的結尾處。
關於字串資料的格式，請參閱 87 頁 3.2.5 項。

(3) ⓢ2+1 中指定的字元數爲 “ 0 ” 時，不執行處理。

(4) ⓢ2+1 中指定的字元數超過了ⓓ中指定的字串資料的最後字元時，存儲至最後字元爲止的資料。



(5) ⓢ2+1 中指定的字元數爲“ -1 ”時，將至ⓢ1中指定的最後字元資料爲止的資料存儲到ⓓ中指定的軟元件後面。



出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

MIDR 指令時

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	ⓢ2的值超過了ⓢ1的字元數時。 從ⓓ位置開始的ⓢ2+1 的字元數超出了ⓓ的軟元件的範圍時。 ⓢ2+0 的值爲 0 時。 ⓢ1中指定的軟元件編號後面相應軟元件的範圍中不存在“ 00_H ”時。	---	○	○	○	○	○

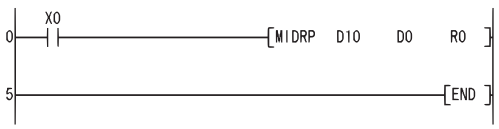
MIDW 指令時

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	ⓢ2的值超過了ⓓ的字元數時。 ⓢ2+1 的值超出了ⓢ1的字元數時。 ⓢ2+0 的值爲 0 時。 ⓢ1中指定的軟元件編號後面相應軟元件的範圍中不存在“ 00_H ”時。	---	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將存儲在 D10 後面的字串資料左側開始的第 3 個字元至第 6 個字元的資料，存儲到 D0 後面的程式。

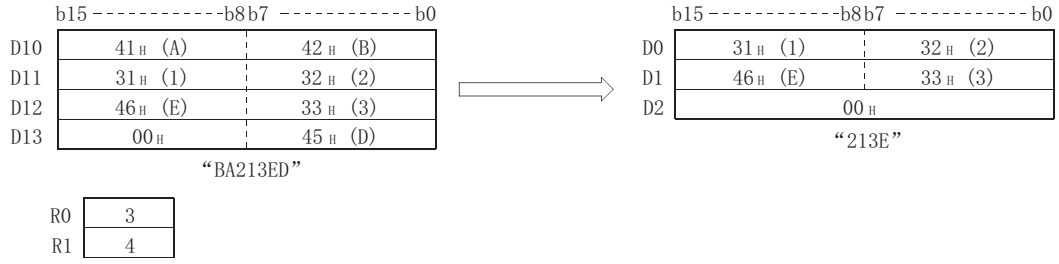
[梯形圖模式]



[列表模式]

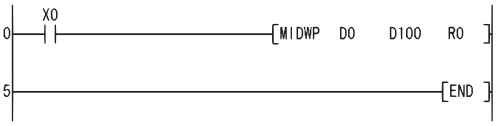
步	指令	軟元件		
0	LD	X0		
1	MIDRP	D10	D0	R0
5	END			

[動作]



(2) 以下爲 X0 變爲 ON 時，將存儲在 D0 後面的 4 個字元的字串資料，存儲到 D100 後面的字串資料左起第 3 個字元的後面的程式。

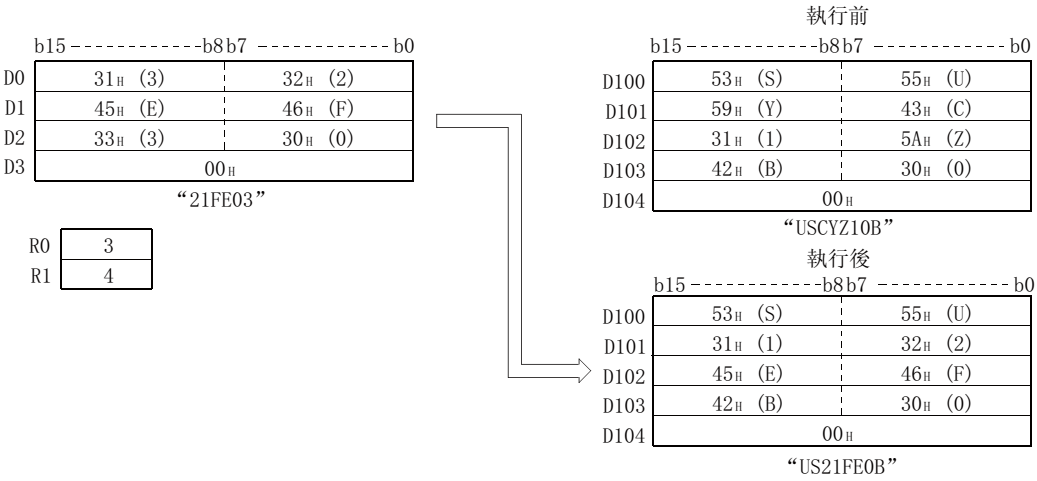
[梯形圖模式]



[列表模式]

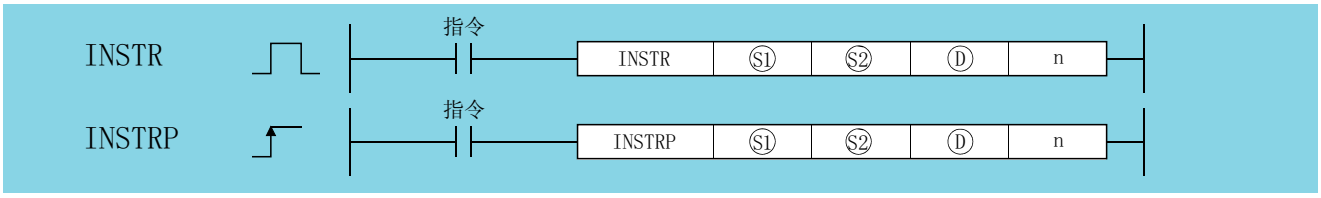
步	指令	軟元件		
0	LD	X0		
1	MIDWP	D0	D100	R0
5	END			

[動作]



7.11.17 INSTR、INSTRP

Basic High performance Process Redundant Universal LCPU



(S1) ：要搜索的字串或者存儲搜索的字串的軟元件的起始編號 (字串)。

(S2) ：被搜索的字串或者存儲被搜索的字串的軟元件的起始編號 (字串)。

(D) ：存儲搜索結果的軟元件的起始編號 (BIN16 位)。

n ：搜索開始位置 (BIN16 位)。

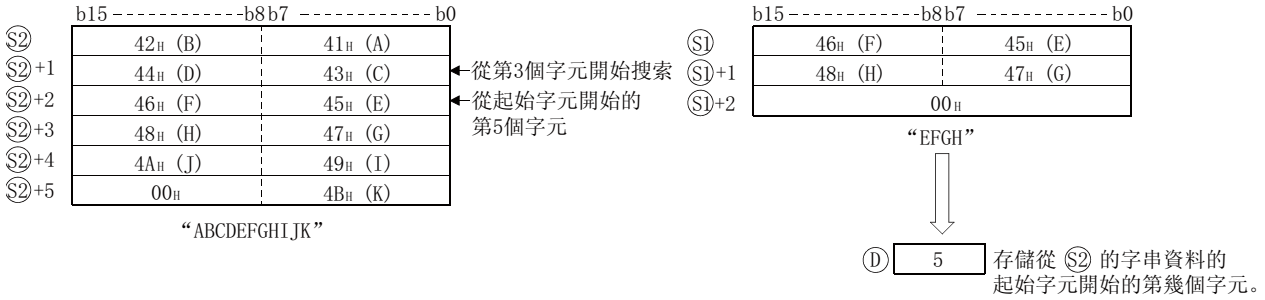
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數		其他
	位	字		位	字			K、H	$	
(S1)	---	○		---				---	○	---
(S2)	---	○		---				---	○	---
(D)	○	○		○				---	---	---
n	○	○		○				○	---	---

功　能

(1) 從Ⓢ2中指定的字串資料左起第 n 個字元開始，對Ⓢ1中指定的字串資料進行搜索，並將搜索結果存儲到Ⓓ中指定的軟元件中。

存儲從Ⓢ2的字串資料的起始字元算起的第幾個字元。

n=3 時



(2) 如果沒有一致的字串資料，將在Ⓓ中存儲 " 0 "。

出　錯

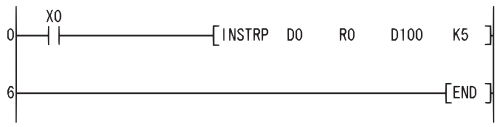
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 的值超過了Ⓢ2的字元數時。 Ⓢ1、Ⓢ2中指定的軟元件後面相應軟元件範圍內沒有 00_H(NULL) 時。 n 的值爲負數或者 " 0 " 時。	---	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D0 後面存儲的字串資料，從 R0 後面存儲的字串資料的左起第 5 個字元開始進行搜索，並將搜索結果存儲到 D100 中的程式。

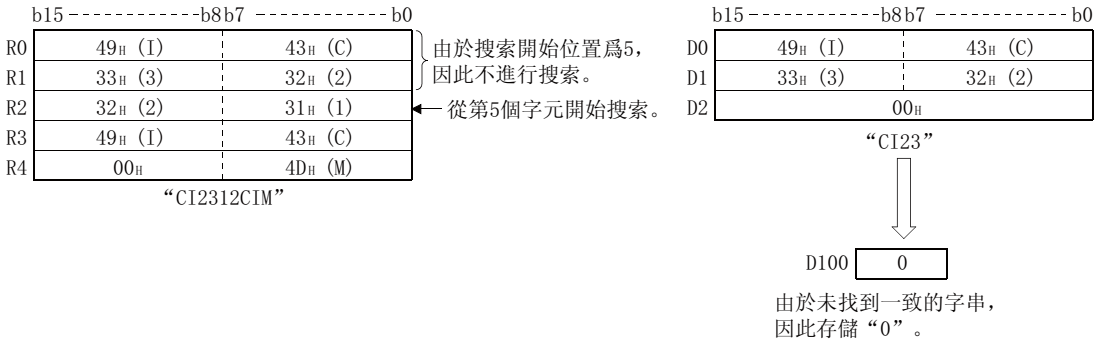
[梯形圖模式]



[列表模式]

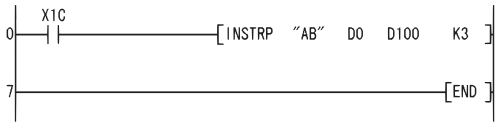
步	指令	軟元件
0	LD	X0
1	INSTRP	D0 R0 D100 K5
6	END	

[動作]



(2) 以下爲 X1C 變爲 ON 時，從存儲在 D0 後面的字串資料左起第 3 個字元開始對字串資料 “ AB ” 進行搜索，並將結果存儲到 D100 中的程式。

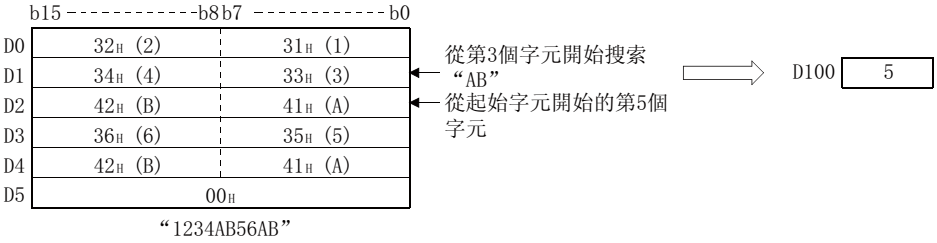
[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X1C			
1	INSTRP	"AB"	D0	D100	K3
7	END				

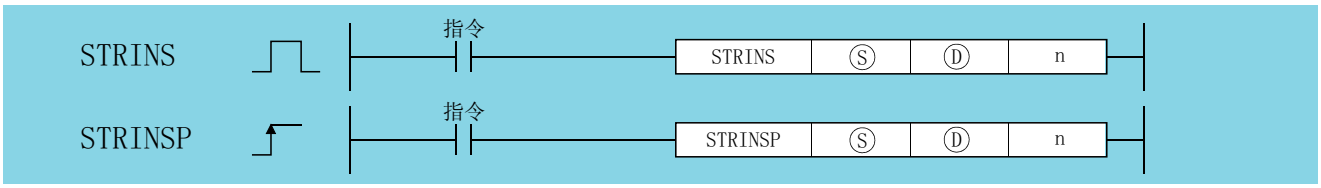
[動作]



Basic / High performance / Process / Redundant / Universal (Ver.) / LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。

7.11.18 STRINS、STRINSP



Ⓢ ： 插入的字串或者存儲插入字串的軟元件的起始編號 (字串)。

Ⓓ ： 存儲被插入字串的軟元件的起始編號 (字串)。

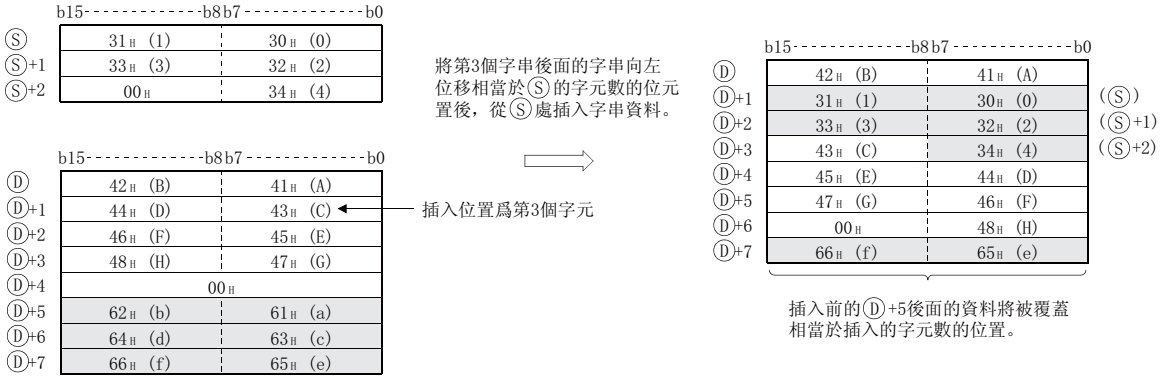
n ： 插入位置 (設置範圍 1 ≤ n ≤ 16383)(BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數		其他
	位	字		位	字			K、H	$	
Ⓢ	---	○		---				---	○	---
Ⓓ	---	○		---				---	---	---
n	---	○		○				○	---	---

功 能

(1) 將Ⓢ中指定的字串資料，插入到Ⓓ中指定的字元資料從起始字元算起的第 n 個字元 (插入位置) 處。

插入位置 n=3 時



(2) 插入後的字串 (Ⓢ+Ⓓ) 爲偶數時，字串的末尾的下一個軟元件 (1 字) 中將存儲 NULL 碼 (00_H)。

(3) 插入後的字串 (Ⓢ+Ⓓ) 爲奇數時，在字串的末尾軟元件 (高 8 位元) 中將存儲 NULL 碼 (00_H)。

(4) n 中指定了Ⓓ的字元數 +1 時，Ⓢ的字串將被合併到Ⓓ的字串的末尾處。

出　錯

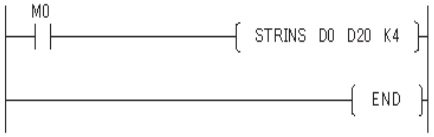
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	字串Ⓢ、字串Ⓓ、或者插入後的字串 (Ⓢ+Ⓓ) 的字元數超過了 16383 個字元時。 n 超出了範圍時。(1 ≤ n ≤ 16383) n 中指定的值超過了字串Ⓓ的字元數 +1 時。	---	---	---	---	○	○
4101	字串Ⓢ與字串Ⓓ的軟元件有部分重復時。 插入後的字串 (Ⓢ+Ⓓ) 超出了指定軟元件的範圍時。 Ⓢ、Ⓓ中指定的軟元件後面，指定軟元件範圍內沒有 NULL 碼 (00_H) 時。 插入後的字串 (Ⓢ+Ⓓ) 與存儲Ⓢ的字串的軟元件重復時。	---	---	---	---	○	○

程式示例

(1) 以下爲 M0 變爲 ON 時，將軟元件 D0 後面存儲的字串資料插入到軟元件 D20 後面的字串資料從起始算起的第 4 個字元處的程式。

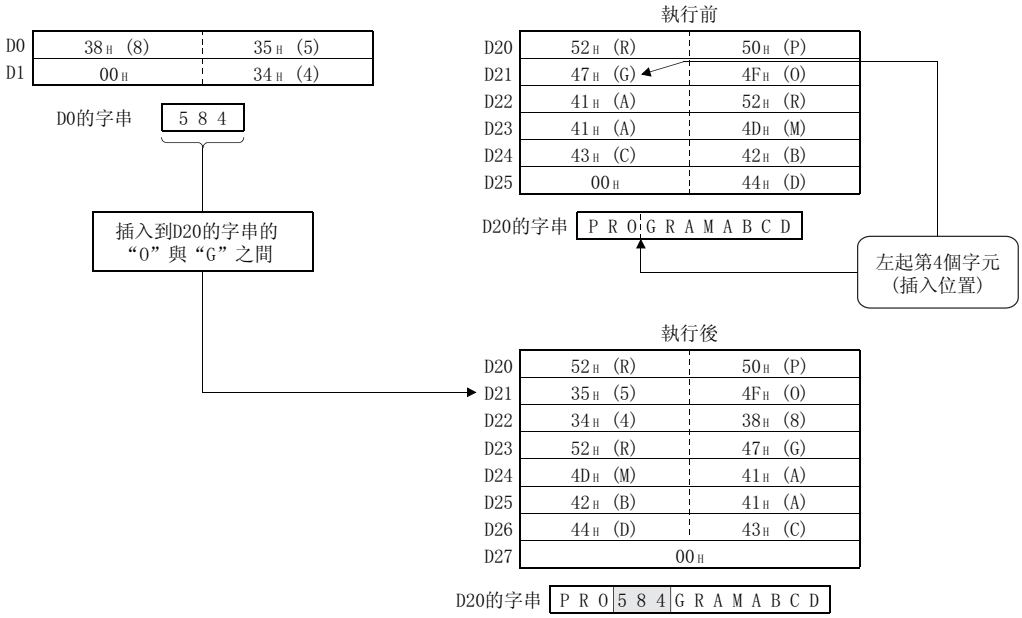
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	M0
1	STRINS	D0 D20 K4
5	END	

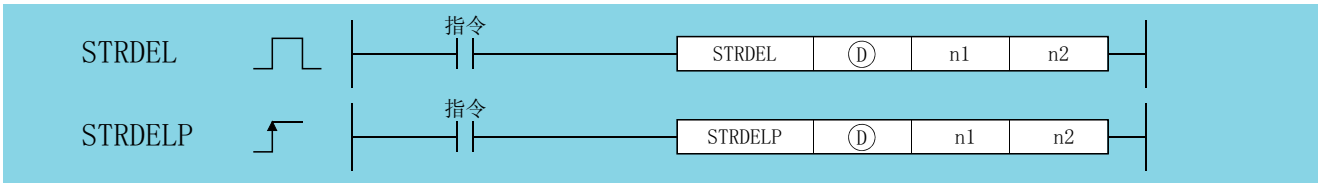
[動作]



7.11.19 STRDEL、STRDELP

Basic × High performance × Process × Redundant × Ver. Universal LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。

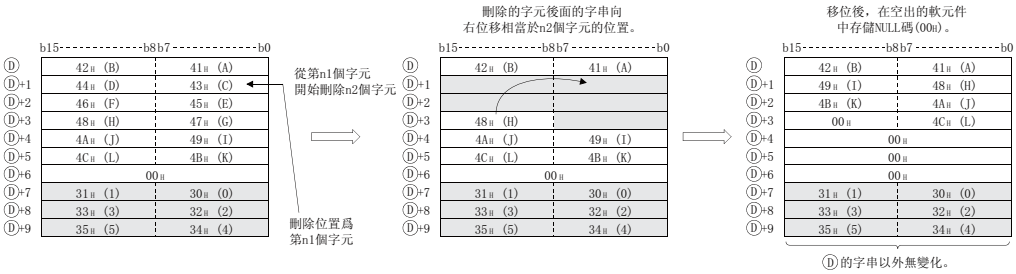


Ⓓ ： 存儲刪除字串的軟元件的起始編號 (字串)。
n1 ： 刪除開始位置 (設置範圍 1 ≤ n1 ≤ 16383)(BIN16 位)。
n2 ： 刪除字元數 (設置範圍 0 ≤ n2 ≤ 16384-n1)(BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓓ	---	○		---				---	---
n1	---	○		○				○	---
n2	---	○		○				○	---

功　能

(1) 從Ⓓ中指定的字串資料從起始算起的第 n1 個字元位置 (刪除開始位置) 開始，刪除 n2 個字元。
刪除位置： n1=3
刪除字元數： n2=5 時



(2) 刪除後，字串Ⓓ爲偶數時，在字串的末尾的下一個軟元件 (1 字) 中將存儲 NULL 碼 (00_H)。

(3) 刪除後，字串Ⓓ爲奇數時，在字串的末尾軟元件 (高 8 位元) 中將存儲 NULL 碼 (00_H)。

(4) 刪除後的字串後面的字串向右移位元 n2 個字元後，在空出的軟軟元件中存儲 NULL 碼 (00_H)。

出　錯

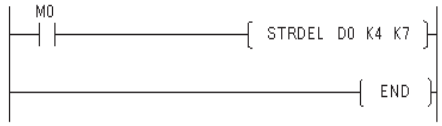
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	字串Ⓓ的字元數超過了 16383 個字元時。 n1 超出了範圍時。(1 ≤ n1 ≤ 16383) n1 中指定的值超過了字串Ⓓ的字元數時。 n2 中指定的值超過了從字串Ⓓ的 n1 開始至最後字元爲止的字元數時。 n2 中指定的值爲負數時。	---	---	---	---	○	○

程式示例

(1) 以下爲 M0 變爲 ON 時，從軟元件 D0 後面存儲的字串資料的第 4 個字元開始刪除 7 個字元資料的程式。

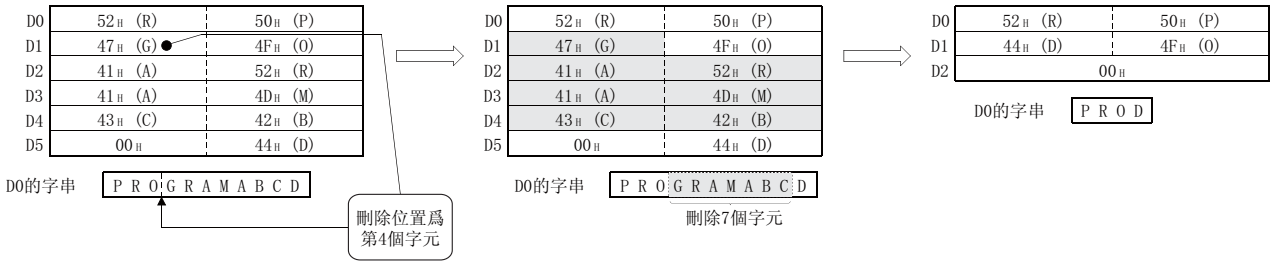
[梯形圖模式]



[列表模式]

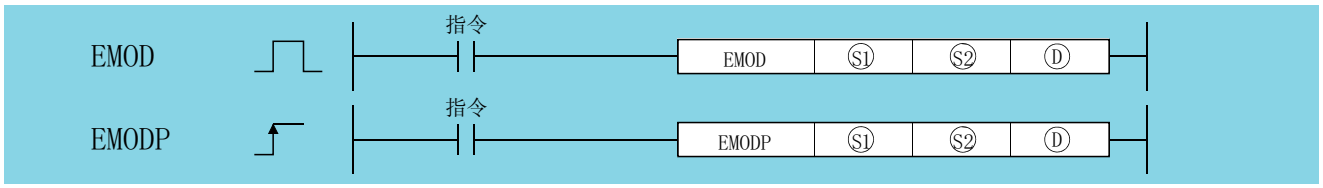
步	指令	軟元件
0	LD	M0
1	STRDEL	D0 K4 K7
5	END	

[動作]



7.11.20 EMOD、EMODP

Basic | High performance | Process | Redundant | Universal | LCPU



S1 ： 32 位元浮點實數資料或者存儲浮點實數資料的軟元件的起始編號 (實數)。

S2 ： 小數部分位元數資料 (BIN16 位元)。

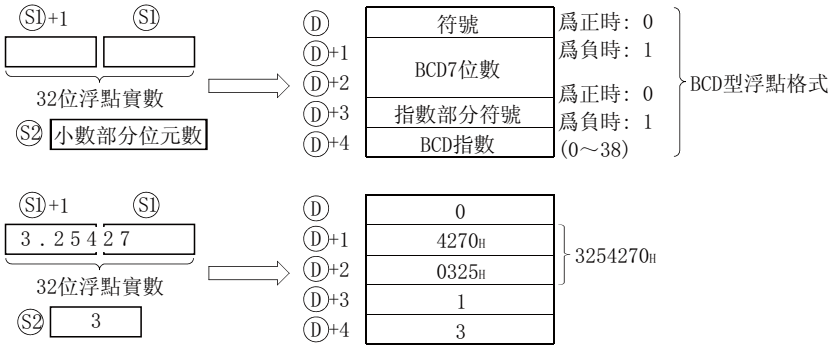
D ： 存儲進行了 BCD 分解的資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數		其他
	位	字		位	字			K、H	E	
S1	---	○		---	○		○ *1	---	○	---
S2	○	○		○	○		○	○	---	---
D	---	○		---	---		---	---	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功　能

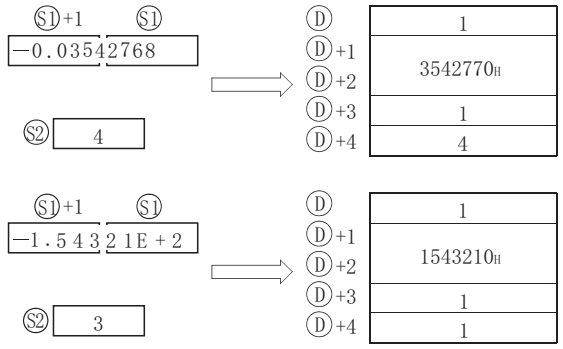
(1) 將(S1)中指定的 32 位元浮點實數資料，根據(S2)中指定的小數部分位元數分解爲 BCD 型浮點格式後，存儲到(D)中指定的軟元件編號後面。



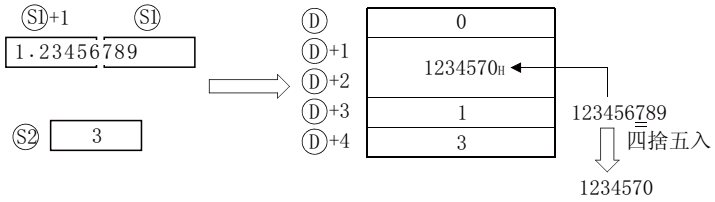
(S2)是用於指定(S1)的 32 位元浮點實數資料的小數部分位元數。在上圖示例中的情況如下所示。

3. 25427

(S2)=3



(2) (D)+1、(D)+2 中存儲的 BCD 的有效位元數爲將第 7 位進行四捨五入後的 6 位數。



出　錯

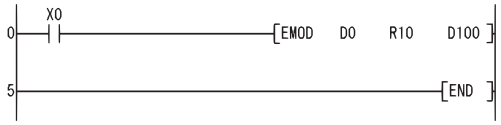
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	(S2)中指定的小數部分位元數超出了 0 ～ 7 的範圍時。 (S1)中指定的 32 位浮點實數不在下述範圍內時。 0、2^{-126} ｜軟元件｜ 2^{128}	---	○	○	○	○	○
4101	(D)中指定的軟元件範圍超出了相應軟元件的範圍時。	---	○	○	○	○	○
	(D)中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○
4140	指定的軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D0、D1 中存儲的 32 位元浮點實數資料，根據 R10 中存儲的值的小數部分位元數分解爲 BCD 後，存儲到 D100 後面的程式。

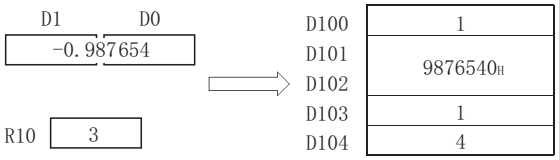
[梯形图模式]



[列表模式]

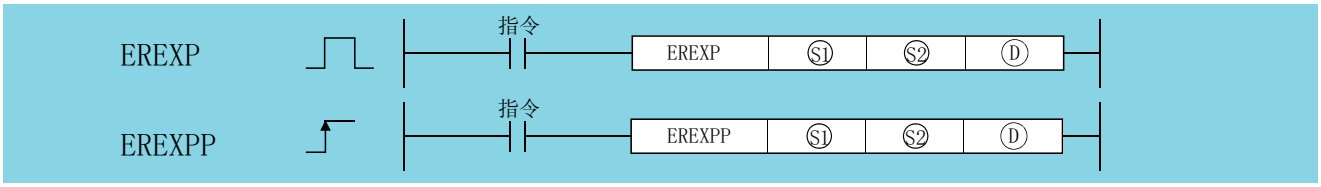
步	指令	軟元件		
0	LD	X0		
1	EMOD	D0	R10	D100
5	END			

[動作]



7.11.21 EREXP、EREXPP

Basic | High performance | Process | Redundant | Universal | LCPU



(S1) ：存儲 BCD 型浮點格式資料的軟元件的起始編號 (BIN16 位)。

(S2) ：小數部分位元數資料 (BIN16 位元)。

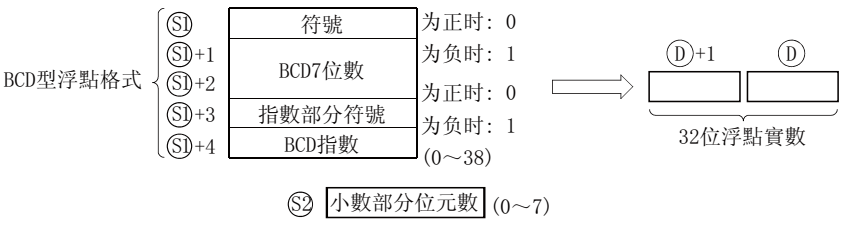
(D) ：存儲 32 位元浮點實數資料的軟元件 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
(S1)	---	○		---	---		---	---	---
(S2)	○	○		○	○		○	○	---
(D)	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功　能

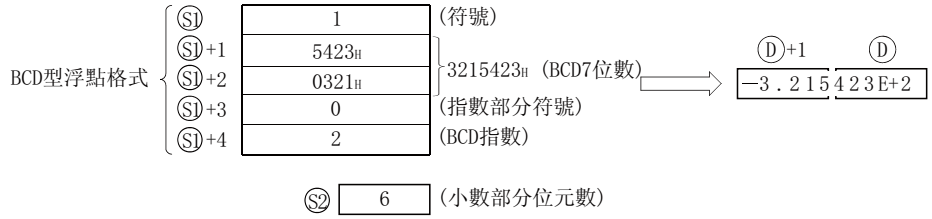
(1) 將(S1)中指定的 BCD 型浮點格式資料，根據(S2)中指定的小數部分位元數轉換爲 32 位元浮點實數資料後，存儲到(D)中指定的軟元件編號後面。



(2) 在(S1)的符號及(S1)+3 的指數部分符號中，爲正時設置爲 0，爲負時設置爲 1。

(3) (S1)+4 的 BCD 指數中，可設置範圍爲 0 ～ 38。

(4) 在(S2)的小數部分位元數中，可設置範圍爲 0 ～ 7。



出　錯

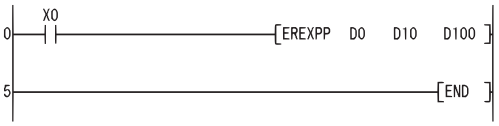
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	(S1)中指定的格式指定爲 0 及 1 以外時。 (S1)+1、(S1)+2 的各位數中存在有除 0 ～ 9 以外的值時。 (S1)+3 中指定的格式指定爲 0 及 1 以外時。 (S1)+4 中指定的指數資料超出了 0 ～ 38 的範圍時。 (S2)中指定的小數部分位元數超出了 0 ～ 7 的範圍時。	---	○	○	○	○	○
4101	(S1)中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D0 後面存儲的 BCD 型浮點格式資料，根據 D10 中存儲的小數部分位元數轉換爲 32 位元浮點實數資料後，存儲到 D100、D101 中的程式。

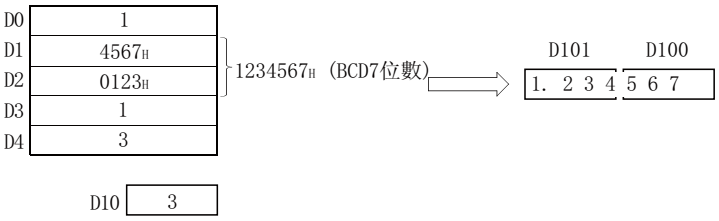
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X0		
1	EREXPP	D0	D10	D100
5	END			

[動作]

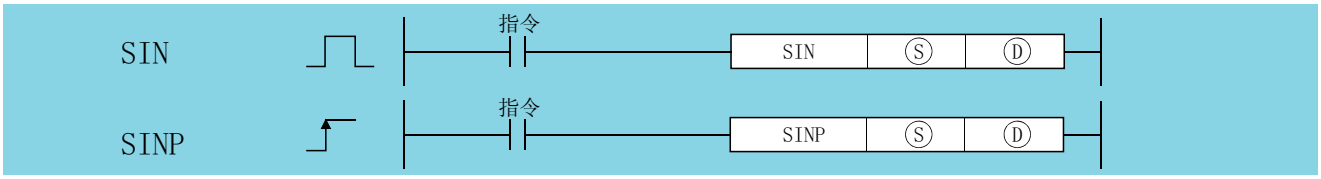


7.12 特殊函數指令

7.12.1 SIN、SINP

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。



Ⓢ ： 進行 SIN(正弦) 運算的角度資料或者存儲角度資料的軟元件的起始編號 (實數)。

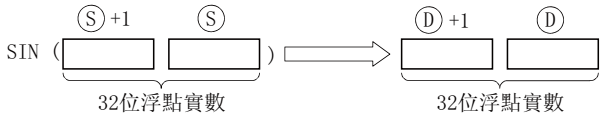
Ⓓ ： 存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在對於通用型 QCPU、LCPU 中可以使用。

功 能

(1) 對Ⓢ中指定的角度值進行 SIN(正弦) 運算後，將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ中指定的角度是以弧度單位 (角度×π ÷ 180) 設置的。
關於角度↔弧度的轉換，請參閱 RAD 指令、DEG 指令。

出 錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

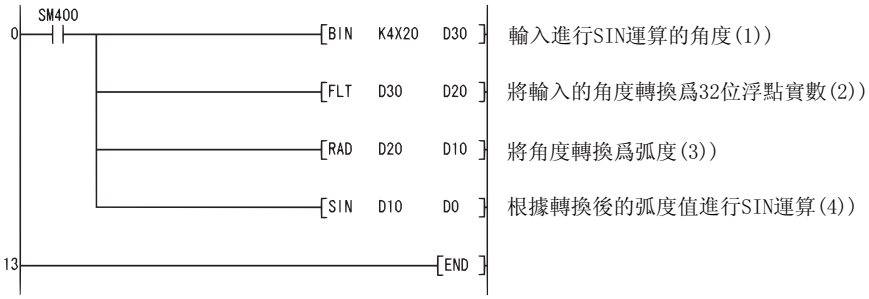
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定軟元件的內容爲 -0 時。*2	○	○	○	○	---	---
4140	指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定爲 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

(1) 以下爲對 X20 ～ X2F 中以 BCD4 位數設置的角度進行 SIN 運算後，以 32 位元浮點實數格式存儲到 D0、D1 中的程式。

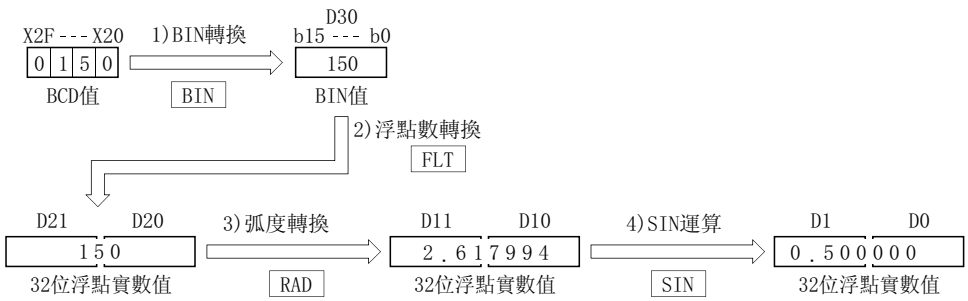
[梯形圖模式]



[列表模式]

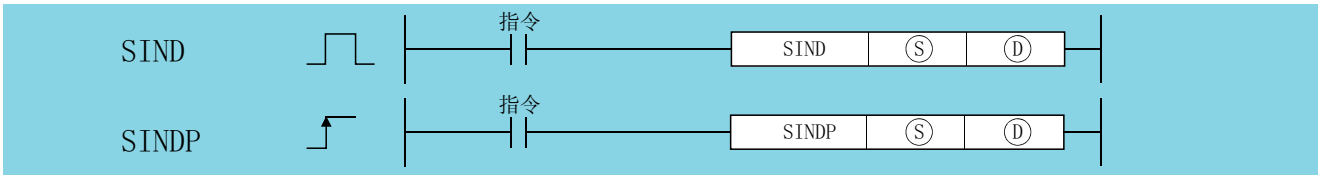
步	指令	軟元件	
0	LD	SM400	
1	BIN	K4X20	D30
4	FLT	D30	D20
7	RAD	D20	D10
10	SIN	D10	D0
13	END		

[在 X20 ～ X2F 中指定了 150 時的動作]



7.12.2 SIND、SINDP

Basic | High performance | Process | Redundant | Universal | LCPU



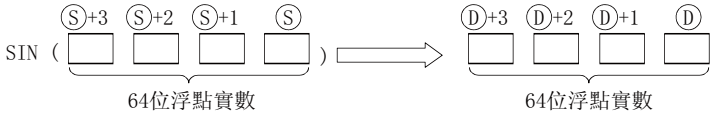
Ⓢ ：進行 SIN(正弦) 運算的角度資料或者存儲角度資料的軟元件的起始編號 (實數)。

Ⓓ ：存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 對Ⓢ中指定的角度值進行 SIN(正弦) 運算後，將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ中指定的角度是以弧度單位 (角度×π÷ 180) 設置的。
關於角度↔弧度的轉換，請參閱 RADD 指令、DEGD 指令。

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出　錯

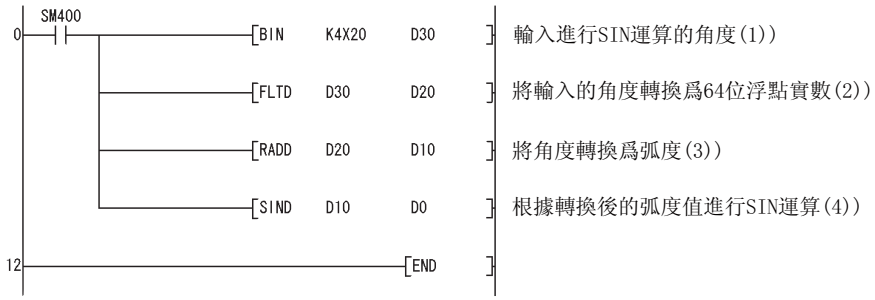
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲對 X20 ～ X2F 中以 BCD4 位數設置的角度進行 SIN 運算後，以 64 位元浮點實數格式存儲到 D0 ～ D3 中的程式。

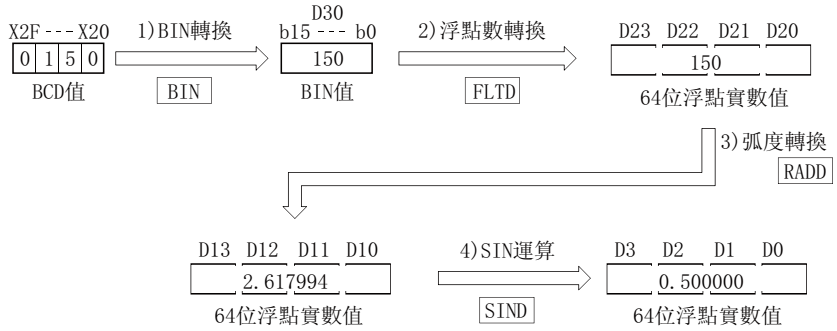
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	BIN	K4X20	D30
3	FLTD	D30	D20
6	RADD	D20	D10
9	SIND	D10	D0
12	END		

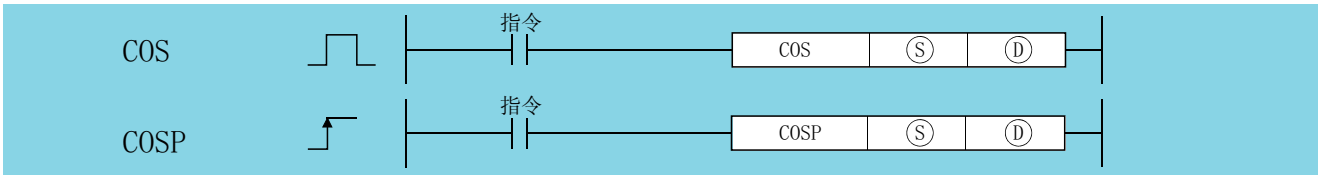
[在 X20 ～ X2F 中指定了 150 時的動作]



Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

7.12.3 COS、COSP



Ⓢ ： 進行 COS(余弦) 運算的角度資料或者存儲角度資料的軟元件的起始編號 (實數)。

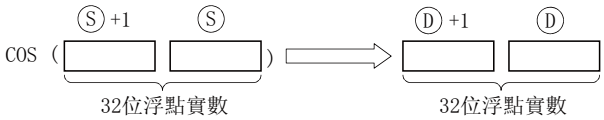
Ⓓ ： 存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

(1) 對Ⓢ中指定的角度值進行 COS(余弦) 運算後，將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ中指定的角度是以弧度單位 (角度×π÷180) 設置的。
關於角度↔弧度的轉換，請參閱 RAD 指令、DEG 指令。

出 錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

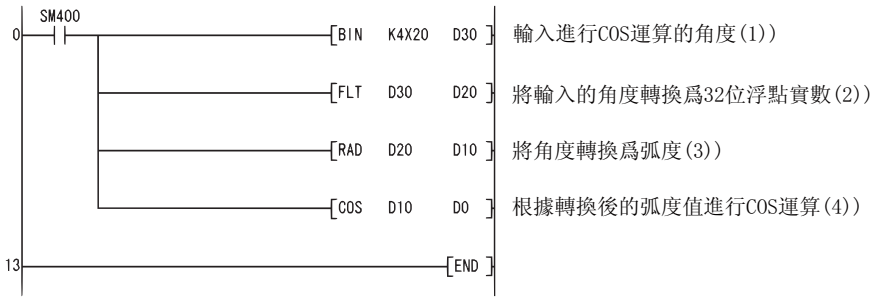
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定軟元件的內容爲 -0 時。*2	○	○	○	○	---	---
4140	指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。(發生了溢出時。) $2^{128} \leq \lvert$ 運算結果 $\rvert$	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定爲 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

(1) 以下爲對 X20 ～ X2F 中以 BCD4 位數設置的角度進行 COS 運算後，以 32 位元浮點實數格式存儲到 D0、D1 中的程式。

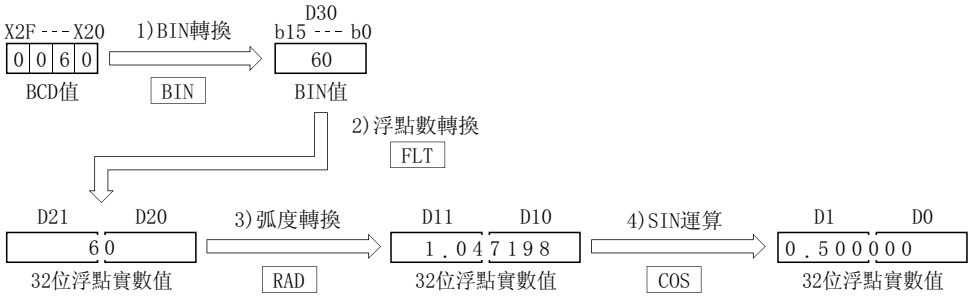
[梯形圖模式]



[列表模式]

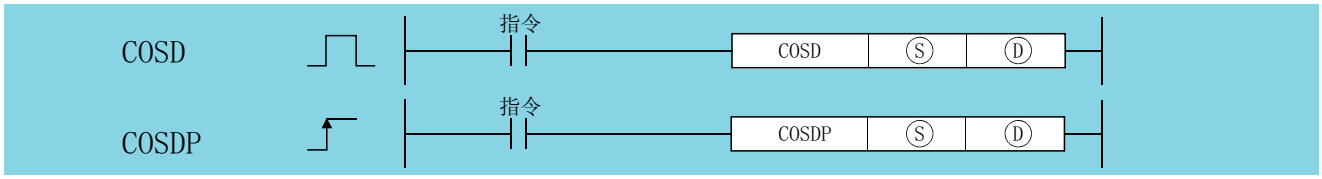
步	指令	軟元件	
0	LD	SM400	
1	BIN	K4X20	D30
4	FLT	D30	D20
7	RAD	D20	D10
10	COS	D10	D0
13	END		

[在 X20 ～ X2F 中指定了 60 時的動作]



7.12.4 COSD、COSDP

Basic High performance Process Redundant Universal LCPU



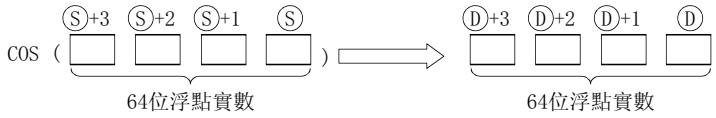
Ⓢ ：進行 COS(余弦) 運算的角度資料或者存儲角度資料的軟元件的起始編號 (實數)。

Ⓓ ：存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 對Ⓢ中指定的角度值進行 COS(余弦) 運算後，將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ中指定的角度是以弧度單位 (角度×π÷ 180) 設置的。
關於角度↔弧度的轉換，請參閱 RADD 指令、DEGD 指令。

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出　錯

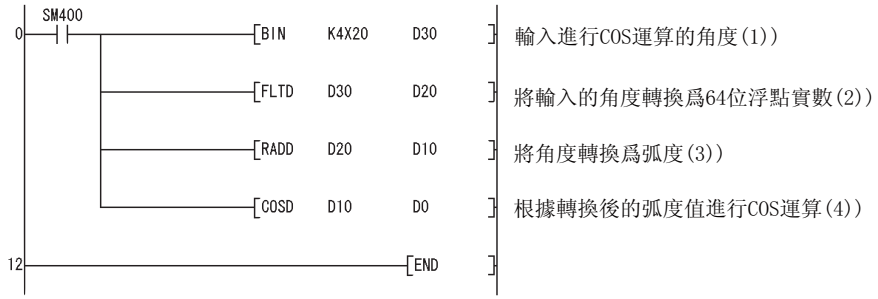
(1) 在以下情況下將變爲運算出錯狀態，出錯標誌 (SM0) 將變爲 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。(發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲對 X20 ～ X2F 中以 BCD4 位數設置的角度進行 COS 運算後，以 64 位元浮點實數格式存儲到 D0 ～ D3 中的程式。

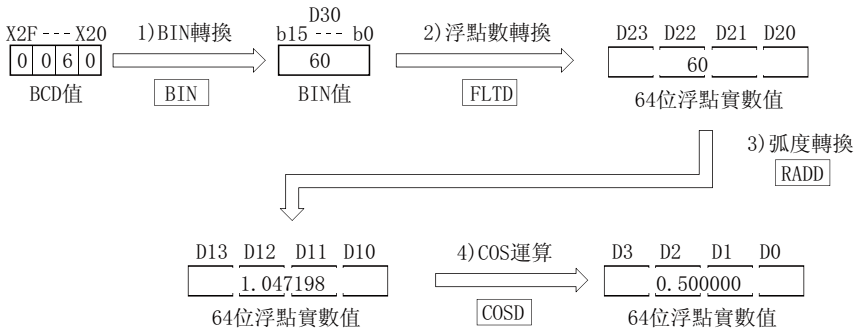
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	BIN	K4X20	D30
3	FLTD	D30	D20
6	RADD	D20	D10
9	COSD	D10	D0
12	END		

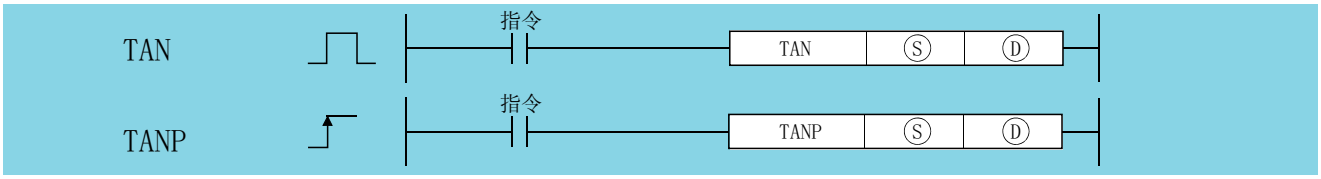
[在 X20 ～ X2F 中指定了 60 時的動作]



Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

7.12.5 TAN、TANP



Ⓢ ： 進行 TAN(正切) 運算的角度資料或者存儲角度資料的軟元件的起始編號 (實數)。

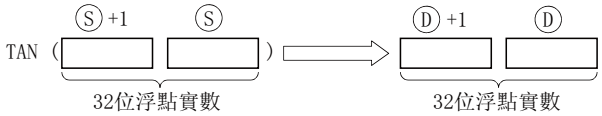
Ⓓ ： 存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

(1) 對Ⓢ中指定的角度值進行 TAN(正切) 運算後，將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ中指定的角度是以弧度單位 (角度×π ÷ 180) 設置的。
關於角度↔弧度的轉換，請參閱 RAD 指令、DEG 指令。

(3) Ⓢ中指定的角度爲π /2 弧度、(3/2) π弧度時，將會產生弧度值運算誤差，且不會變爲出錯狀態，應加以注意。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

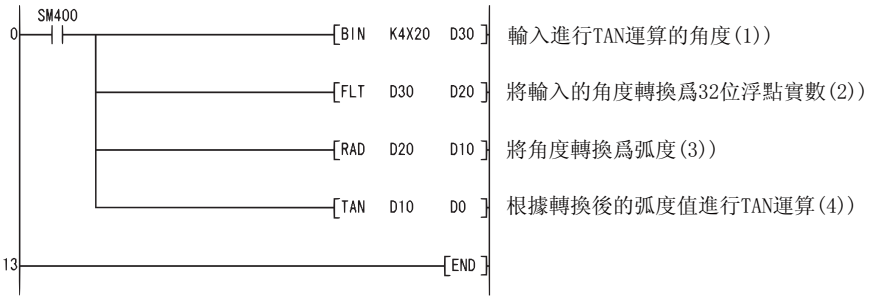
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定軟元件的內容不在下述範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 -0 時。*2	○	○	○	○	---	---
4140	指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定了 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

(1) 以下爲對 X20 ～ X2F 中以 BCD4 位數設置的角度進行 TAN 運算後，以 32 位元浮點實數格式存儲到 D0、D1 中的程式。

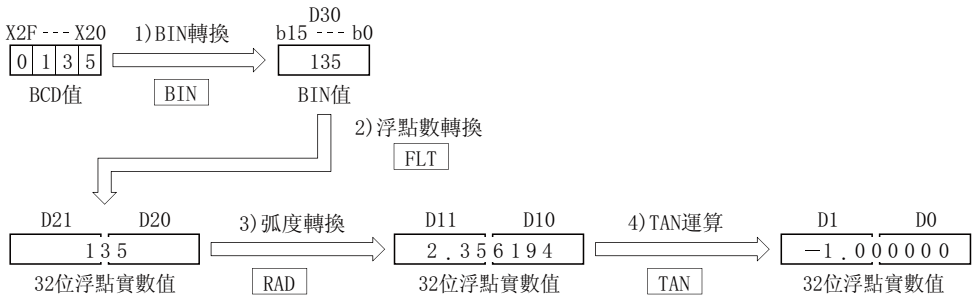
[梯形圖模式]



[列表模式]

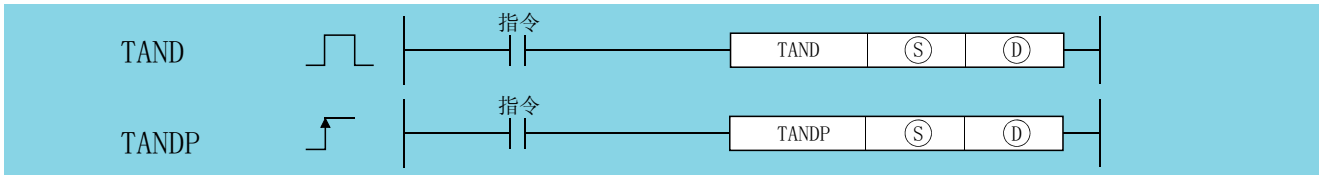
步	指令	軟元件	
0	LD	SM400	
1	BIN	K4X20	D30
4	FLT	D30	D20
7	RAD	D20	D10
10	TAN	D10	D0
13	END		

[在 X20 ～ X2F 中指定了 135 時的動作]



7.12.6 TAND、TANDP

Basic High performance Process Redundant Universal LCPU



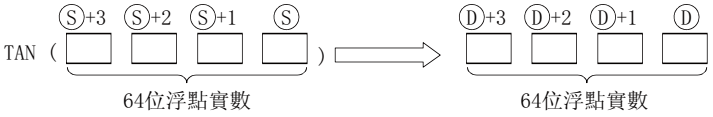
Ⓢ ：進行 TAN(正切) 運算的角度資料或者存儲角度資料的軟元件的起始編號 (實數)。

Ⓓ ：存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 對Ⓢ中指定的角度值進行 TAN(正切) 運算後，將運算結果存儲到Ⓓ中指定的軟元件中。



(2) Ⓢ中指定的角度是以弧度單位 (角度×π ÷ 180) 設置的。
關於角度↔弧度的轉換，請參閱 RADD 指令、DEGD 指令。

(3) Ⓢ中指定的角度爲π /2 弧度、(3/2) π弧度時，將會產生弧度值運算誤差，且不會變爲出錯狀態，應加以注意。

(4) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出　錯

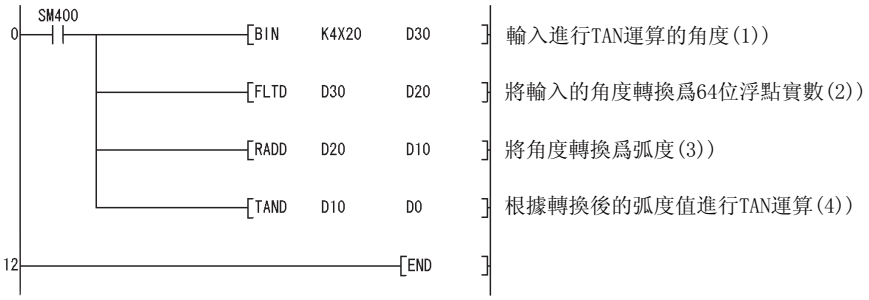
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲對 X20 ～ X2F 中以 BCD4 位數設置的角度進行 TAN 運算後，以 64 位元浮點實數格式存儲到 D0 ～ D3 中的程式。

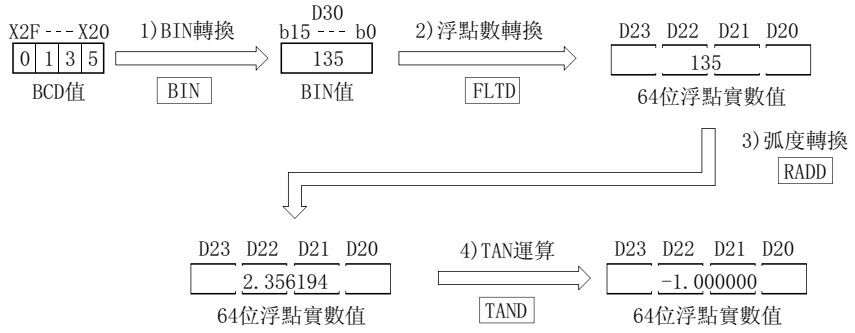
[梯形圖模式]



[列表模式]

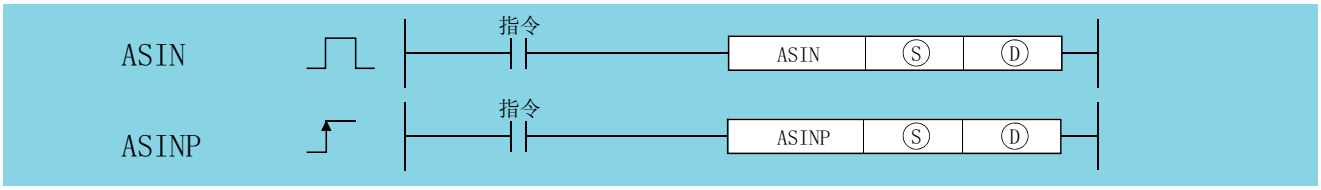
步	指令	軟元件	
0	LD	SM400	
1	BIN	K4X20	D30
3	FLTD	D30	D20
6	RADD	D20	D10
9	TAND	D10	D0
12	END		

[在 X20 ～ X2F 中指定了 135 時的動作]



7.12.7 ASIN、ASINP

Basic (×) | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：進行 SIN^{-1}(反正弦) 運算的 SIN 值或者存儲 SIN 值的軟元件的起始編號 (實數)。

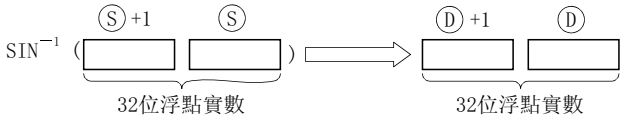
Ⓓ ：存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功　能

(1) 通過Ⓢ中指定的 SIN 值進行角度運算後，將運算結果存儲到Ⓓ中指定的字軟元件中。



(2) Ⓢ中指定的 SIN 值的可設置範圍爲 -1.0 ～ 1.0。

(3) Ⓓ中存儲的角度 (運算結果) 是以弧度單位存儲的。
關於角度↔弧度的轉換，請參閱 RAD 指令、DEG 指令。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的值超過了 -1.0 ～ 1.0 的範圍時。	---	○	○	○	○	○
4100	指定軟元件的內容爲 -0 時。*2	---	○	○	○	---	---
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定爲 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

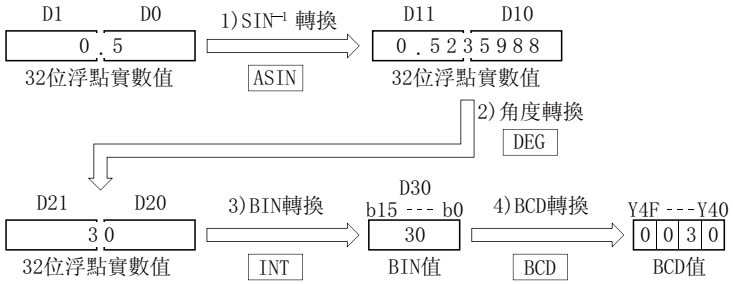
(1) 以下爲求出 D0、D1 的 32 位浮點實數的 SIN^{-1} 後，將該角度以 BCD4 位數格式輸出到 Y40 ～ Y4F 中的程式。

[梯形圖模式]

[列表模式]

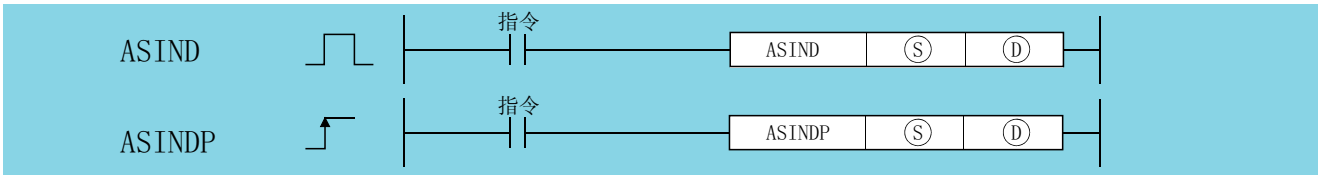
步	指令	軟元件	
0	LD	SM400	
1	ASIN	D0	D10
4	DEG	D10	D20
7	INT	D20	D30
10	BCD	D30	K4Y40
13	END		

[D0、D1 的值爲 0.5 時的動作]



7.12.8 ASIND、ASINDP

Basic High performance Process Redundant Universal LCPU



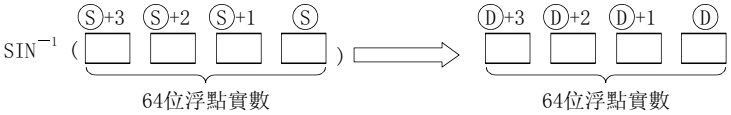
Ⓢ ：進行 SIN^{-1}(反正弦) 運算的 SIN 值或者存儲 SIN 值的軟元件的起始編號 (實數)。

Ⓓ ：存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 通過Ⓢ中指定的 SIN 值進行角度運算後，將運算結果存儲到Ⓓ中指定的字軟元件中。



(2) Ⓢ中指定的 SIN 值的可設置範圍爲 -1.0 ～ 1.0。

(3) Ⓓ中存儲的角度 (運算結果) 是以弧度單位存儲的。
關於角度↔弧度的轉換，請參閱 RADD 指令、DEGD 指令。

(4) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出　錯

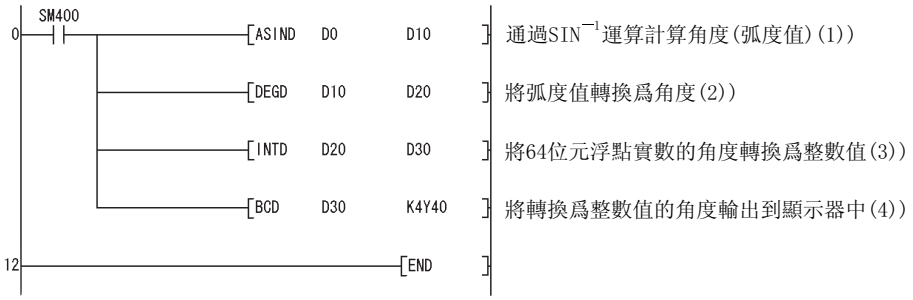
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的值在雙精度浮點數範圍內超過了 -1.0 ～ 1.0 的範圍時。	---	---	---	---	○	○
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲求出 D0 ～ D3 的 64 位浮點實數的 SIN^{-1} 後，將該角度以 BCD4 位數格式輸出到 Y40 ～ Y4F 中的程式。

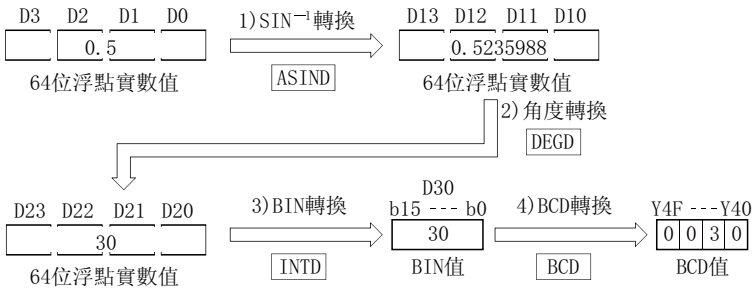
[梯形圖模式]



[列表模式]

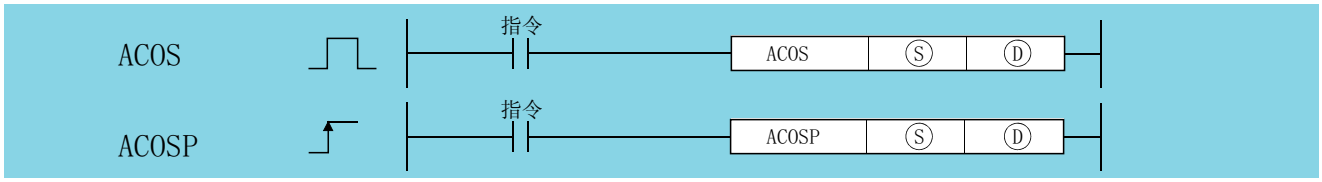
步	指令	軟元件	
0	LD	SM400	
1	ASIND	D0	D10
4	DEGD	D10	D20
7	INTD	D20	D30
10	BCD	D30	K4Y40
12	END		

[D0 ～ D3 的值爲 0.5 時的動作]



7.12.9 ACOS、ACOSP

Basic High performance Process Redundant Universal LCPU



Ⓢ ： 進行 COS^{-1}(反余弦) 運算的 COS 值或者存儲 COS 值的軟元件的起始編號 (實數)。

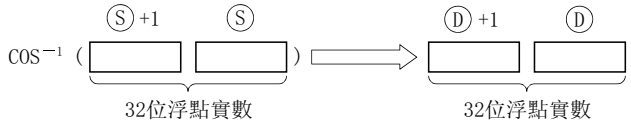
Ⓓ ： 存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功　能

(1) 通過Ⓢ中指定的 COS 值進行角度運算後，將運算結果存儲到Ⓓ中指定的字軟元件中。



(2) Ⓢ中指定的 COS 值的可設置範圍爲 -1.0 ～ 1.0。

(3) Ⓓ中存儲的角度（運算結果）是以弧度單位存儲的。
關於角度↔弧度的轉換，請參閱 RAD 指令、DEG 指令。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

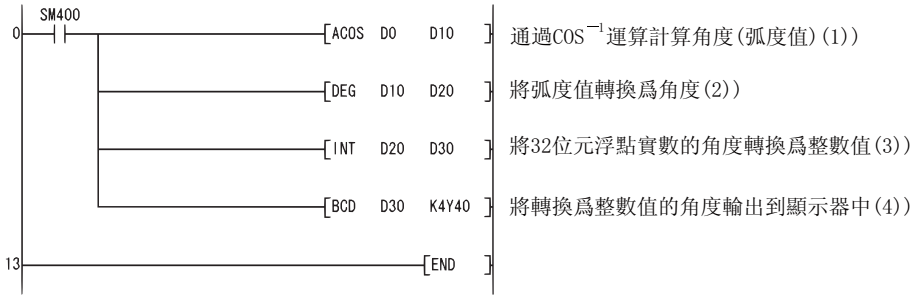
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的值超過了 -1.0 ～ 1.0 的範圍時。	---	○	○	○	○	○
	指定軟元件的內容爲 -0 時。*2	---	○	○	○	---	---
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 （發生了溢出時。） $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定爲 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

(1) 以下爲求出 D0、D1 的 32 位浮點實數的 COS^{-1} 後，將該角度以 BCD4 位數格式輸出到 Y40 ～ Y4F 中的程式。

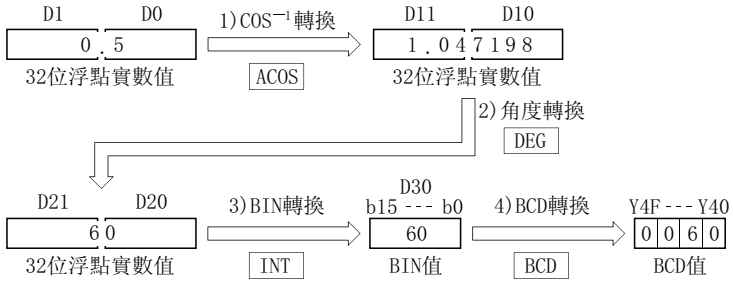
[梯形圖模式]



[列表模式]

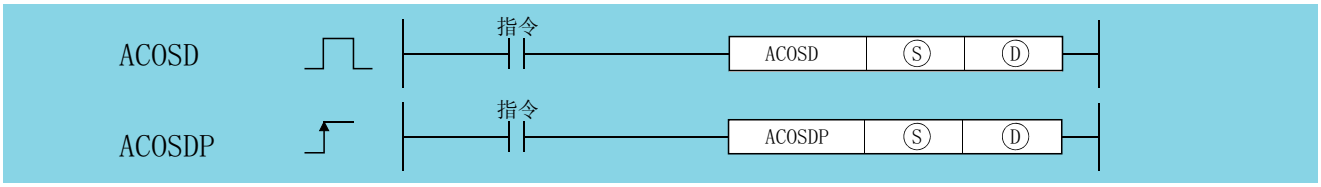
步	指令	軟元件	
0	LD	SM400	
1	ACOS	D0	D10
4	DEG	D10	D20
7	INT	D20	D30
10	BCD	D30	K4Y40
13	END		

[D0、D1 的值爲 0.5 時的動作]



7.12.10 ACOSD、ACOSDP

Basic | High performance | Process | Redundant | Universal | LCPU



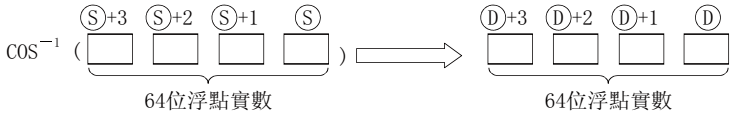
Ⓢ ：進行 COS^{-1}(反余弦) 運算的 COS 值或者存儲 COS 值的軟元件的起始編號 (實數)。

Ⓓ ：存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 通過Ⓢ中指定的 COS 值進行角度運算後，將運算結果存儲到Ⓓ中指定的字軟元件中。



(2) Ⓢ中指定的 COS 值的可設置範圍爲 -1.0 ～ 1.0。

(3) Ⓓ中存儲的角度 (運算結果) 是以弧度單位存儲的。
關於角度↔弧度的轉換，請參閱 RADD 指令、DEGD 指令。

(4) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出　錯

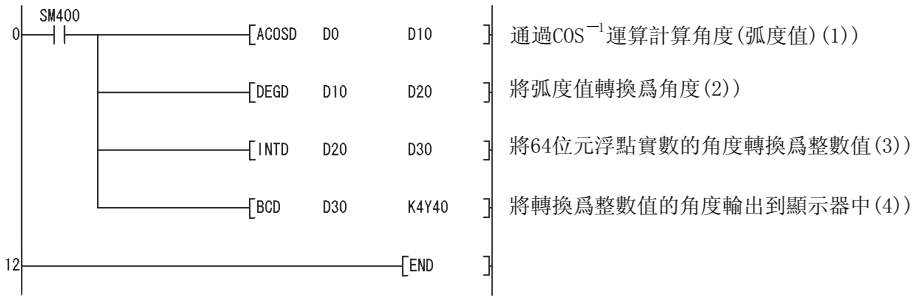
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的值在雙精度浮點數範圍內超過了 -1.0 ～ 1.0 的範圍時。	---	---	---	---	○	○
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲求出 D0 ～ D3 的 64 位浮點實數的 COS^{-1} 後，將該角度以 BCD4 位數格式輸出到 Y40 ～ Y4F 中的程式。

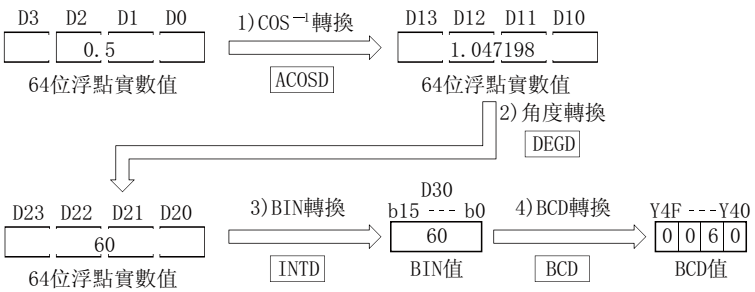
[梯形圖模式]



[列表模式]

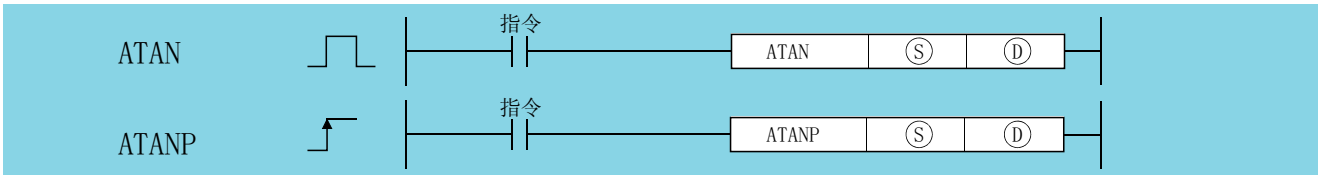
步	指令	軟元件	
0	LD	SM400	
1	ACOSD	D0	D10
4	DEGD	D10	D20
7	INTD	D20	D30
10	BCD	D30	K4Y40
12	END		

[D0 ～ D3 的值爲 0.5 時的動作]



7.12.11 ATAN、ATANP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：進行 TAN^{-1}(反正切) 運算的 TAN 值或者存儲 TAN 值的軟元件的起始編號 (實數)。

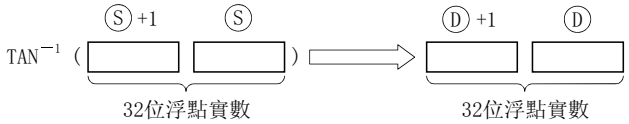
Ⓓ ：儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功　能

(1) 通過Ⓢ中指定的 TAN 值進行角度運算後，將運算結果存儲到Ⓓ中指定的字軟元件中。



(2) Ⓓ中存儲的角度（運算結果）是以弧度單位存儲的。
關於角度↔弧度的轉換，請參閱 RAD 指令、DEG 指令。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

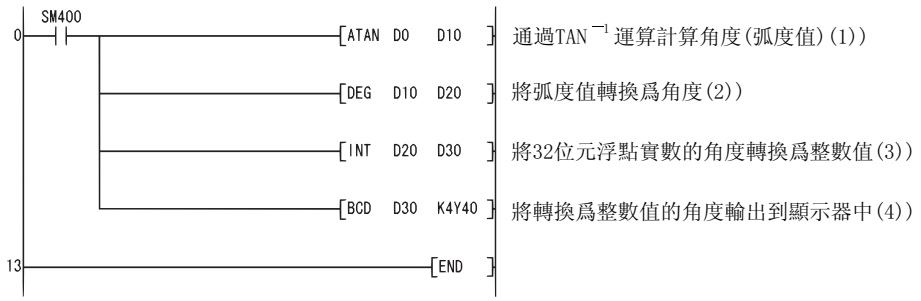
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定軟元件的內容爲 -0 時。*2	---	○	○	○	---	---
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 （發生了溢出時。） $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定爲 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

(1) 以下爲求出 D0、D1 的 32 位浮點實數的 TAN^{-1} 後，將該角度以 BCD4 位數格式輸出到 Y40 ～ Y4F 中的程式。

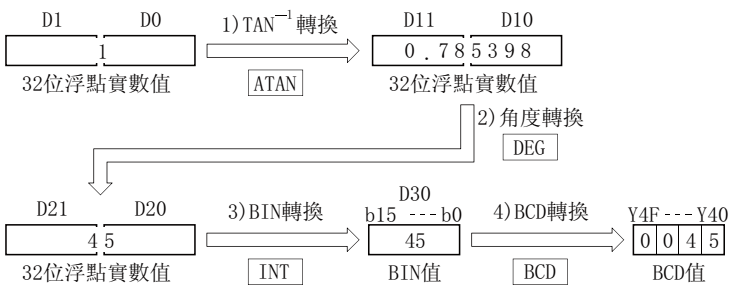
[梯形圖模式]



[列表模式]

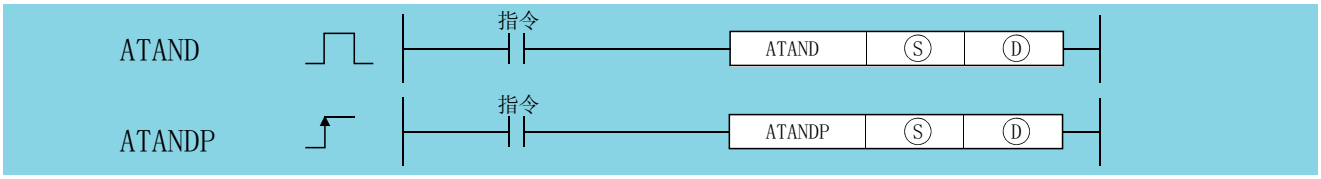
步	指令	軟元件	
0	LD	SM400	
1	ATAN	D0	D10
4	DEG	D10	D20
7	INT	D20	D30
10	BCD	D30	K4Y40
13	END		

[D0、D1 的值爲 1 時的動作]



7.12.12 ATAND、ATANDP

Basic × | High performance × | Process × | Redundant × | Universal | LCPU



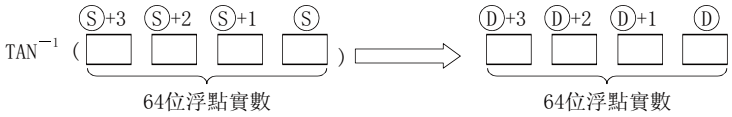
Ⓢ ：進行 TAN^{-1}（反正切）運算的 TAN 值或者存儲 TAN 值的軟元件的起始編號（實數）。

Ⓓ ：存儲運算結果的軟元件的起始編號（實數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 通過Ⓢ中指定的 TAN 值進行角度運算後，將運算結果存儲到Ⓓ中指定的字軟元件中。



(2) Ⓓ中存儲的角度（運算結果）是以弧度單位存儲的。
關於角度↔弧度的轉換，請參閱 RADD 指令、DEGD 指令。

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

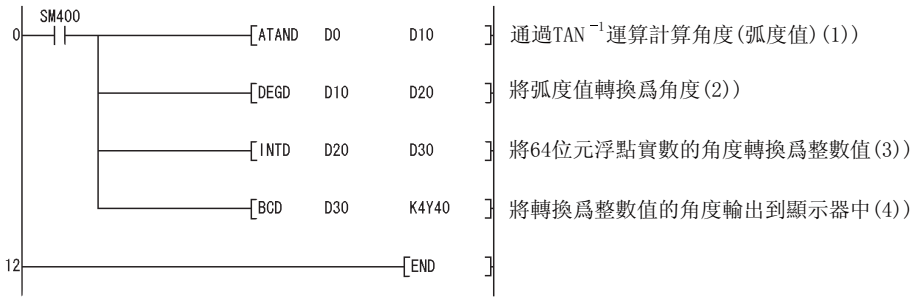
出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 （發生了溢出時。） $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

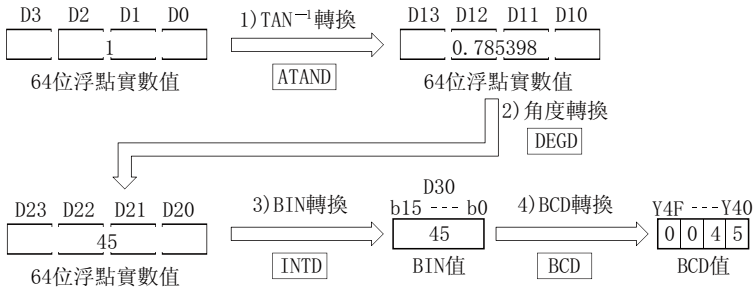
(1) 以下爲求出 D0 ～ D3 的 64 位浮點實數的 TAN^{-1} 後，將該角度以 BCD4 位數格式輸出到 Y40 ～ Y4F 中的程式。
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	ATAND	D0	D10
4	DEGD	D10	D20
7	INTD	D20	D30
10	BCD	D30	K4Y40
12	END		

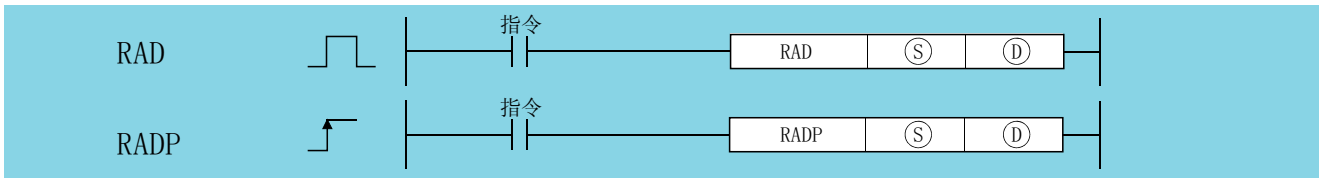
[D0 ～ D3 的值爲 1 時的動作]



Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

7.12.13 RAD、RADP



Ⓢ ： 要轉換爲弧度單位的角度或者存儲角度的軟元件的起始編號 (實數)。

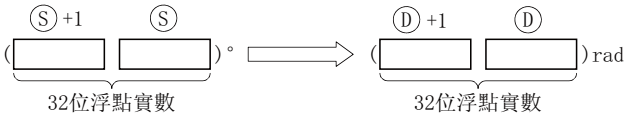
Ⓓ ： 存儲轉換爲弧度單位的值的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

(1) 將角度的大小單位從Ⓢ中指定的度單位轉換爲弧度單位後，將運算結果存儲到Ⓓ中指定編號的軟元件中。



(2) 度單位→弧度單位的轉換公式如下所示。

$$\text{弧度單位} = \text{度單位} \times \frac{\pi}{180}$$

出　錯

(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

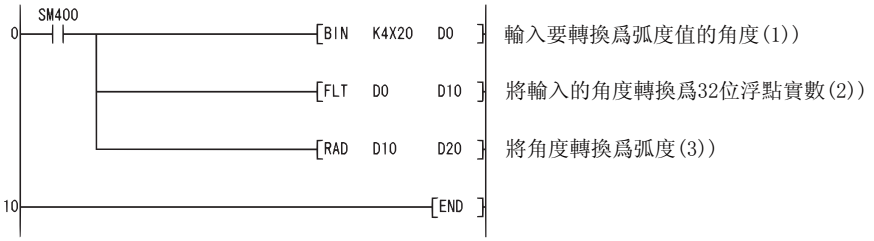
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定軟元件的內容爲 -0 時。*2	○	○	○	○	---	---
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○

*2:　有的 CPU 模組即使指定爲 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

(1)　以下爲將 X20 ～ X2F 中以 BCD4 位數設置的角度轉換爲弧度後，以 32 位元浮點實數存儲到 D20、D21 中的程式。

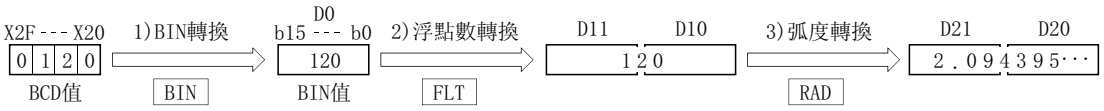
[梯形圖模式]



[列表模式]

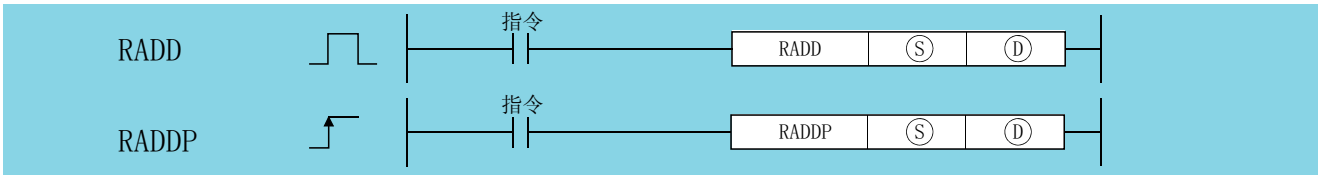
步	指令	軟元件	
0	LD	SM400	
1	BIN	K4X20	D0
4	FLT	D0	D10
7	RAD	D10	D20
10	END		

[在 X20 ～ X2F 中指定了 120 時的動作]



7.12.14 RADD、RADDP

Basic　High performance　Process　Redundant　Universal　LCPU



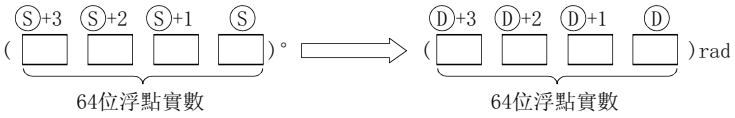
Ⓢ ：要轉換爲弧度單位的角度或者存儲角度的軟元件的起始編號 (實數)。

Ⓓ ：存儲轉換爲弧度單位的值的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字			E	
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 將角度的大小單位從Ⓢ中指定的度單位轉換爲弧度單位後，將運算結果存儲到Ⓓ中指定編號的軟元件中。



(2) 度單位→弧度單位的轉換公式如下所示。

$$\text{弧度單位} = \text{度單位} \times \frac{\pi}{180}$$

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出　錯

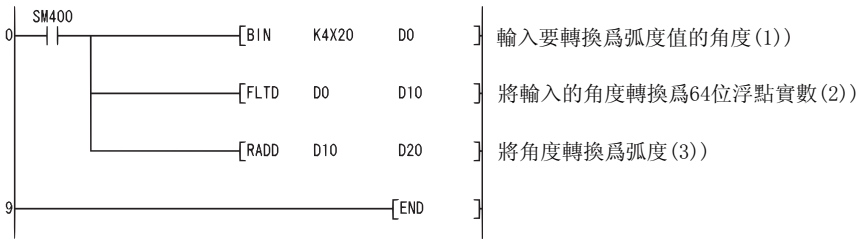
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲將 X20 ～ X2F 中以 BCD4 位數設置的角度轉換爲弧度後，以 64 位元浮點實數存儲到 D20 ～ D23 中的程式。

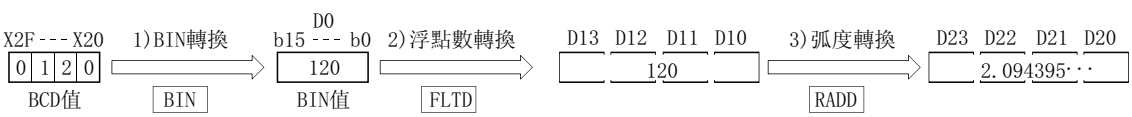
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	BIN	K4X20	D0
3	FLTD	D0	D10
6	RADD	D10	D20
9	END		

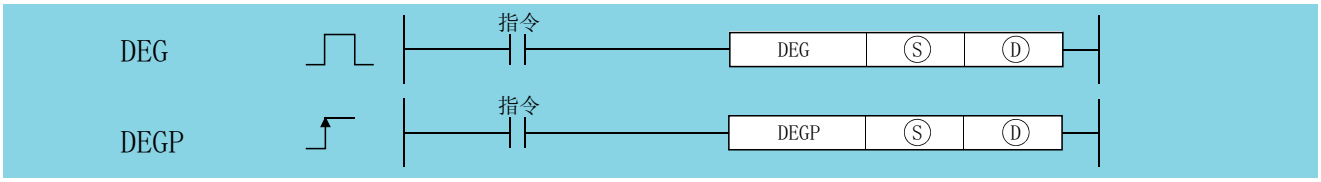
[在 X20 ～ X2F 中指定了 120 時的動作]



Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

7.12.15 DEG、DEGP



Ⓢ ： 要轉換爲度單位的弧度角度或者存儲弧度角度的軟元件的起始編號 (實數)。

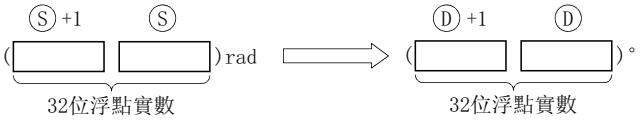
Ⓓ ： 存儲轉換爲度單位的值的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

(1) 將角度的大小單位從Ⓢ中指定的弧度單位轉換爲度單位後，將運算結果存儲到Ⓓ中指定編號的軟元件中。



(2) 弧度單位→度單位的轉換公式如下所示。

$$\text{度單位} = \text{弧度單位} \times \frac{180}{\pi}$$

出 錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

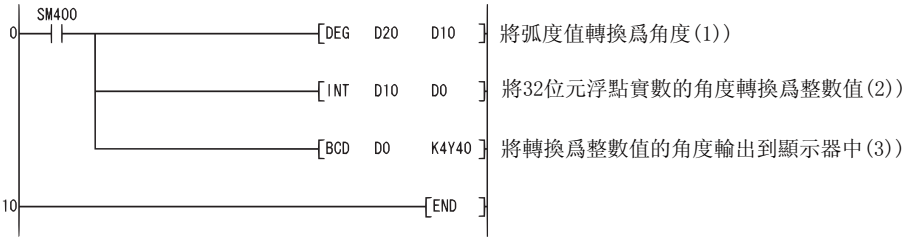
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定軟元件的內容爲 -0 時。 *2	○	○	○	○	---	---
4140	指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定爲 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

(1) 以下爲將 D20、D21 中以 32 位浮點實數設置的弧度值轉換爲角度後，以 BCD 值格式輸出到 Y40 ～ Y4F 中的程式。

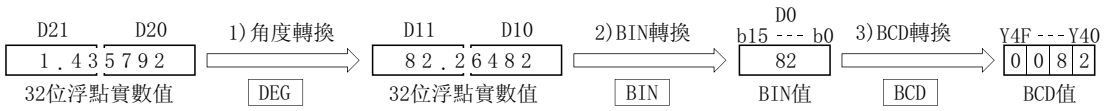
[梯形圖模式]



[列表模式]

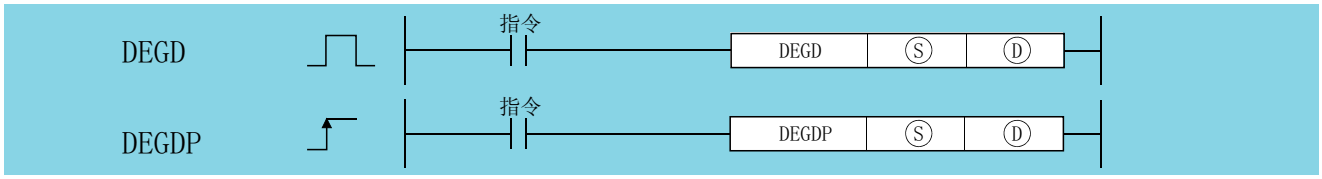
步	指令	軟元件	
0	LD	SM400	
1	DEG	D20	D10
4	INT	D10	D0
7	BCD	D0	K4Y40
10	END		

[D20、D21 的值爲 1.435792 時的動作]



7.12.16 DEGD、DEGDP

Basic　High performance　Process　Redundant　Universal　LCPU



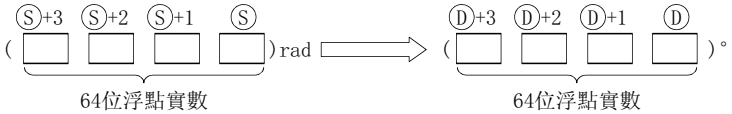
Ⓢ ： 要轉換爲度單位的弧度角度或者存儲弧度角度的軟元件的起始編號 (實數)。

Ⓓ ： 存儲轉換爲度單位的值的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 將角度的大小單位從Ⓢ中指定的弧度單位轉換爲度單位後，將運算結果存儲到Ⓓ中指定編號的軟元件中。



(2) 弧度單位→度單位的轉換公式如下所示。

$$\text{度單位} = \text{弧度單位} \times \frac{180}{\pi}$$

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 處理。

出　錯

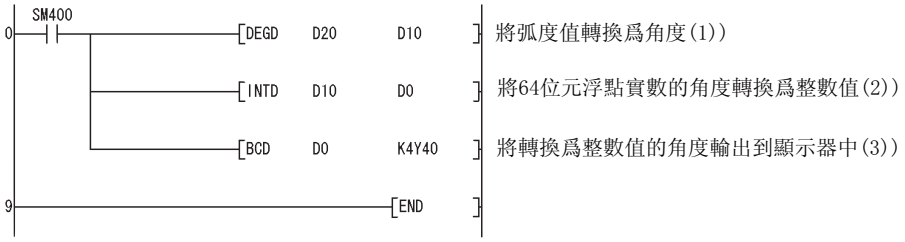
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲將 D20、D21 中以 64 位浮點實數設置的弧度值轉換爲角度後，以 BCD 值格式輸出到 Y40 ～ Y4F 中的程式。

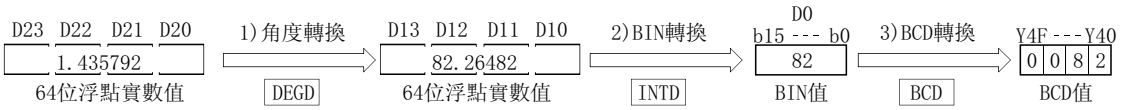
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	DEGD	D20	D10
4	INTD	D10	D0
7	BCD	D0	K4Y40
9	END		

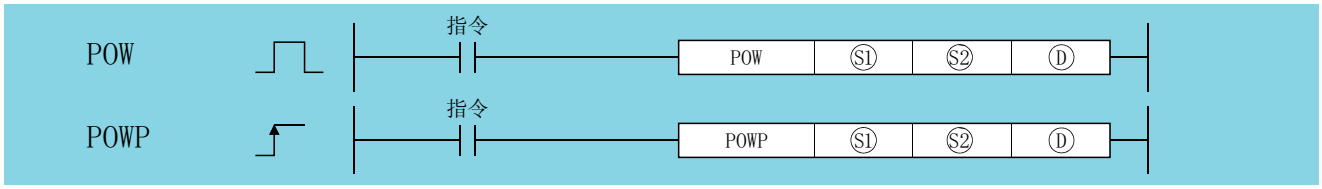
[D20 ～ D23 的值爲 1.435792 時的動作]



Basic × High performance × Process × Redundant × Ver. Universal LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。

7.12.17 POW、POWP



S1 ： 底數資料或者存儲底數資料的軟元件的起始編號 (實數)。

S2 ： 冪次方資料或者存儲冪次方資料的軟元件的起始編號 (實數)。

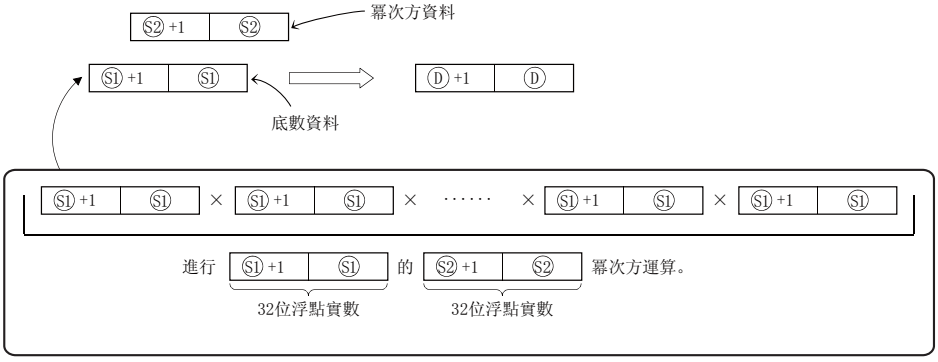
D ： 儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字			E	
S1	---	○		---	○		○	△ *1	---
S2	---	○		---	○		○	△ *1	---
D	---	○		---	○		○	---	---

*1: 僅實數可以使用。

功 能

(1) 以S1中指定的 32 位浮點實數作爲底數，以S2中指定的 32 位浮點實數作爲冪次方進行冪運算，並將運算結果存儲到D中指定的位元軟元件中。



(2) S1、S2中可指定的值及可存儲的值如下所示。

0、$2^{-126} \leq$ | 設置值 (存儲值) | $< 2^{128}$

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出 錯

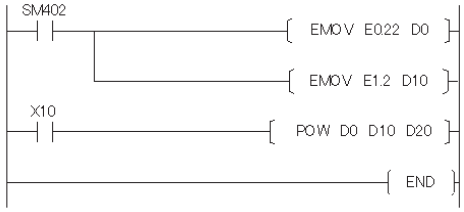
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	S1或者S2中指定的值不在下述範圍內時。 0、$2^{-126} \leq$	設置值 (存儲值)	$< 2^{128}$ S1或者S2的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。(發生了溢出時。) $2^{128} \leq$	運算結果		---	---	---	---	○	○

程式示例

(1) 以下爲 X10 變爲 ON 時，將 D0、D1 的 32 位浮點實數作爲底數，以 D10、D11 的 32 位浮點實數作爲冪進行冪運算，並將運算結果存儲到 D20、D21 中的程式。

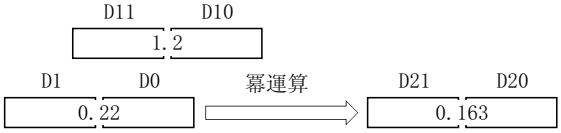
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	SM402
1	EMOV	E0.22 D0
4	EMOV	E1.2 D10
7	LD	X10
8	POW	D0 D10 D20
12	END	

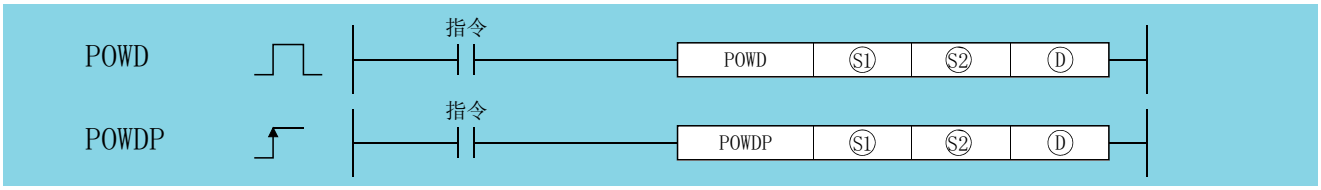
[動作]



Basic　High performance　Process　Redundant　Universal　LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。

7.12.18 POWD、POWDP



Ⓢ1 ： 底數資料或者存儲底數資料的軟元件的起始編號 (實數)。

Ⓢ2 ： 冪次方資料或者存儲冪次方資料的軟元件的起始編號 (實數)。

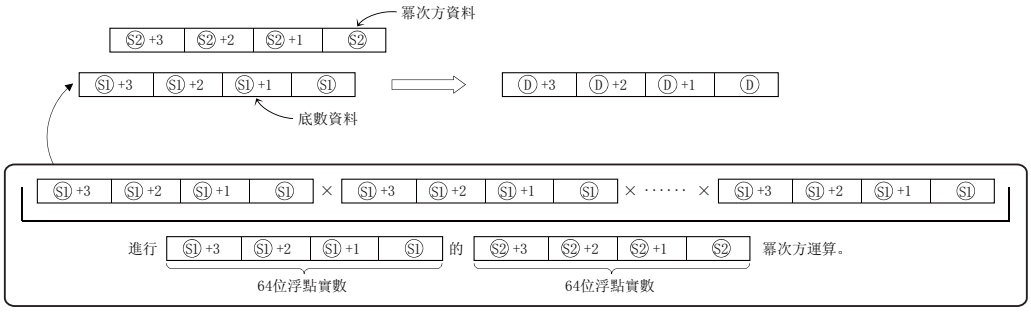
Ⓓ ： 存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 E	其他
Ⓢ1	---	○		---	○		---	△ *1	---
Ⓢ2	---	○		---	○		---	△ *1	---
Ⓓ	---	○		---	○		---	---	---

*1: 僅實數可以使用。

功　能

(1) 以Ⓢ1中指定的 64 位浮點實數作爲底數，以Ⓢ2中指定的 64 位浮點實數作爲冪次方進行冪運算，並將運算結果存儲到Ⓓ中指定的位元軟元件中。



(2) ⓢ1、ⓢ2中可指定的值及可存儲的值如下所示。

0、$2^{-1022} \leq$ | 設置值 (存儲值) | $< 2^{1024}$

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出　錯

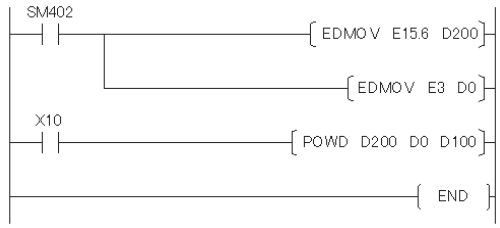
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	ⓢ1或者ⓢ2中指定的值不在下述範圍內時。 0、$2^{-1022} \leq$ | 設置值 (存儲值) | $< 2^{1024}$ ⓢ1或者ⓢ2的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。(發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲 X10 變爲 ON 時，將 D200 ～ D203 的 64 位浮點實數作爲底數，以 D0 ～ D3 的 64 位浮點實數作爲冪進行冪運算，並將運算結果存儲到 D100 ～ D103 中的程式。

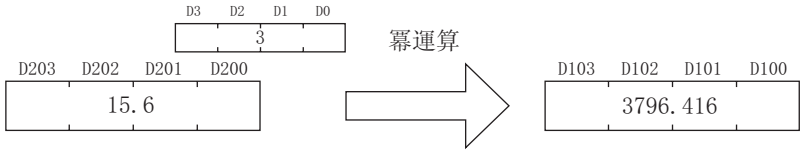
[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	SM402
1	EDMOV	E15.6 D200
4	EDMOV	E3 D0
7	LD	X10
8	POWD	D200 D0 D100
12	END	

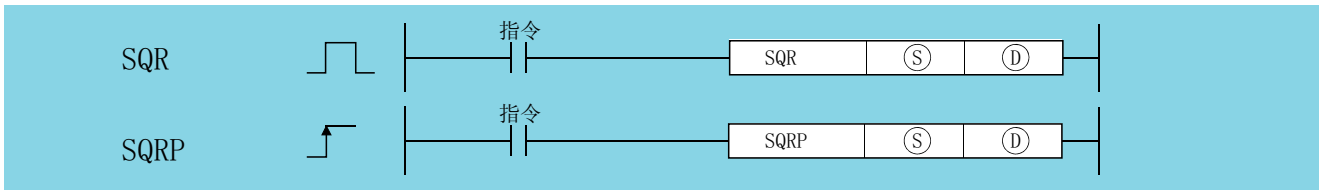
[動作]



Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

7.12.19 SQR、SQRP



ⓢ : 進行平方根運算的資料或者存儲資料的軟元件的起始編號 (實數)。

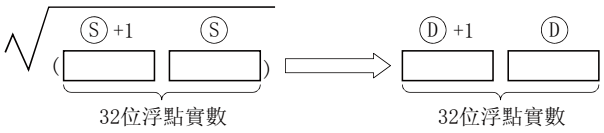
ⓓ : 存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 E	其他
ⓢ	---	○		---	○		○ *1	○	---
ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功　能

(1) 對Ⓢ中指定的值進行平方根運算後，將運算結果存儲到Ⓓ中指定編號的軟元件中。



(2) Ⓢ中指定的值只能設置爲正數。(爲負數時將無法運算。)

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的值爲負數時。	○	○	○	○	○	---
4100	指定軟元件的內容爲 -0 時。*2	○	○	○	○	---	---
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定爲 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

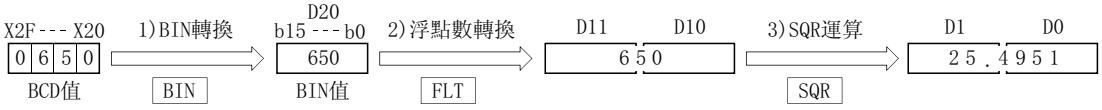
(1) 以下爲對 X20 ～ X2F 中以 BCD4 位數格式設置的值進行平方根運算後，將其結果以 32 位元浮點實數存儲到 D0、D1 中的程式。

[梯形圖模式]

[列表模式]

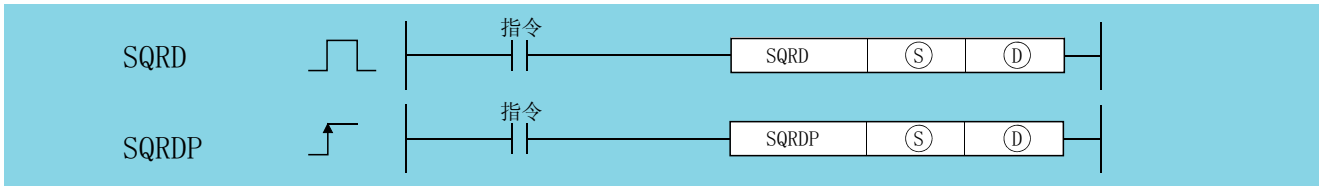
步	指令	軟元件	
0	LD	SM400	
1	BIN	K4X20	D20
4	FLT	D20	D10
7	SQRP	D10	D0
10	END		

[X20 ～ X2F 中指定爲 650 時的動作]



7.12.20 SQRD、SQRDP

Basic ✕ | High performance ✕ | Process ✕ | Redundant ✕ | Universal | LCPU



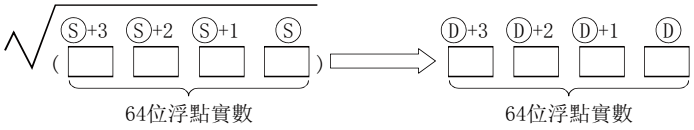
Ⓢ ：進行平方根運算的資料或者存儲資料的軟元件的起始編號（實數）。

Ⓓ ：存儲運算結果的軟元件的起始編號（實數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 對Ⓢ中指定的值進行平方根運算後，將運算結果存儲到Ⓓ中指定編號的軟元件中。



(2) Ⓢ中指定的值只能設置爲正數。（爲負數時將無法運算。）

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出　錯

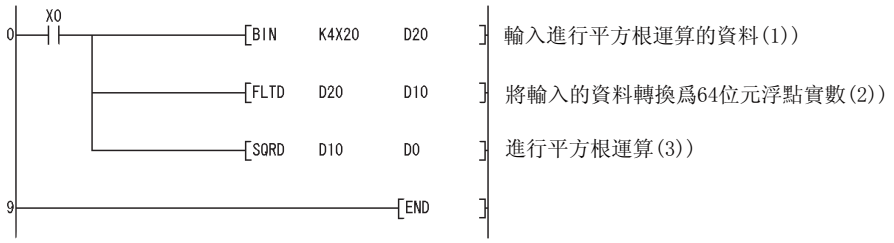
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ 中指定的值爲負數時。	---	---	---	---	○	○
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 （發生了溢出時。） $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲對 X20 ～ X2F 中以 BCD4 位數格式設置的值進行平方根運算後，將其結果以 64 位元浮點實數存儲到 D0 ～ D3 中的程式。

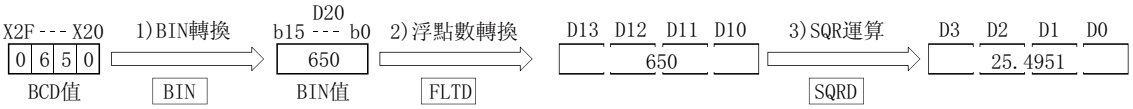
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X0	
1	BIN	K4X20	D20
3	FLTD	D20	D10
6	SQRD	D10	D0
9	END		

[X20 ～ X2F 中指定爲 650 時的動作]

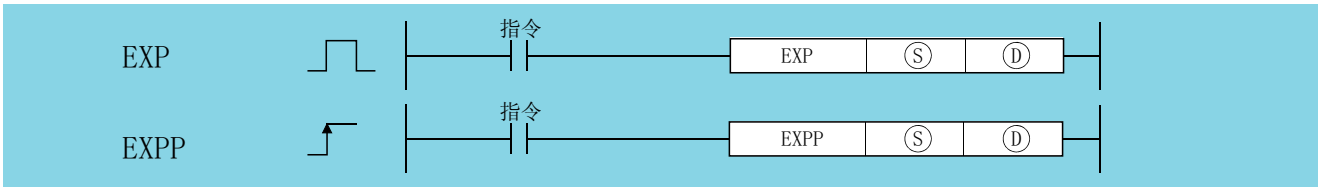


Ver.

Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

7.12.21 EXP、EXPP



Ⓢ ： 進行指數運算的資料或者存儲資料的軟元件的起始編號 (實數)。

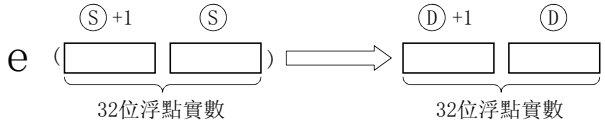
Ⓓ ： 存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

(1) 對Ⓢ中指定的值進行指數運算，並將運算結果存儲到Ⓓ中指定編號的位軟元件中。



(2) 在指數運算中，將底 (e) 作爲 “ 2.71828 ” 進行運算。

出 錯

(1) 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

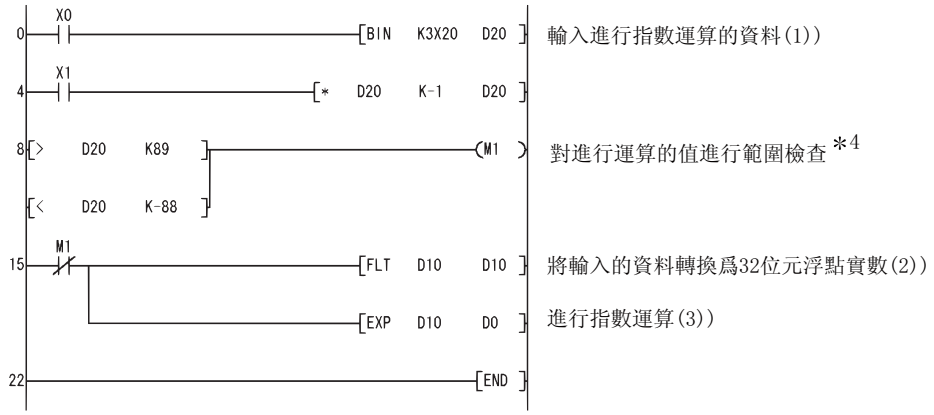
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	運算結果不在下述範圍內時。 $2^{-126} \leq \lvert$ 運算結果 $\rvert \leq 2^{128}$	---	○	---	---	---	---
	運算結果不在下述範圍內時。 $2^{-126} \leq \lvert$ 運算結果 $\rvert < 2^{128}$	○	---	○	○	---	---
	指定軟元件的內容爲 -0 時。*2	○	○	○	○	---	---
4140	指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{128} \leq \lvert$ 運算結果 $\rvert$	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定爲 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

(1) 以下爲以 X20 ～ X27 中以 BCD2 位數格式設置的值進行指數運算後，將其結果以 32 位元浮點實數存儲到 D0、D1 中的程式。

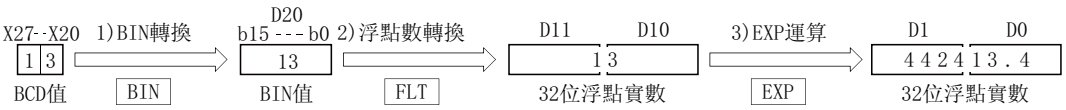
[梯形圖模式]



[列表模式]

步	指令	軟元件		
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			

[X20 ～ X27 中指定爲 13 時的動作]



*4: 由於 loge 2^{129}=89.4，爲了讓運算結果低於 2^{129}，X20 ～ X27 的 BCD 值應爲 89 以下。
由於設置值超過 90 以上時將會變爲運算出錯狀態，因此應將程式設置爲，如果設置了 90 以上的值，則 M1 將變爲 ON，不執行運算。

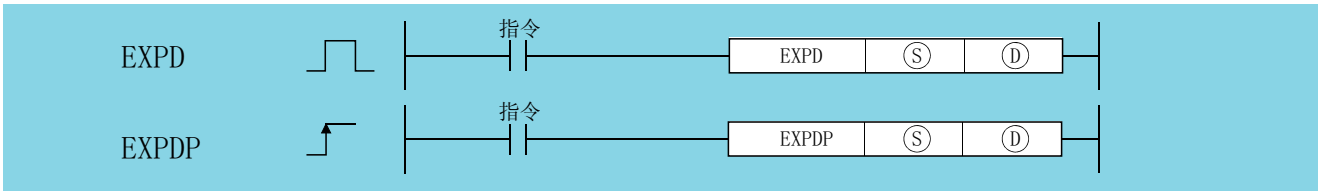
要 點

自然對數至常用對數的轉換
在 CPU 中是使用自然對數進行運算的。
若要對常用對數值進行計算，應將常用對數值用 0.43429 相除後的值指定到Ⓢ中。

$$10^{x}=e^{\frac{x}{0.43429}}$$

7.12.22 EXPD、EXPDP

Basic High performance Process Redundant Universal LCPU



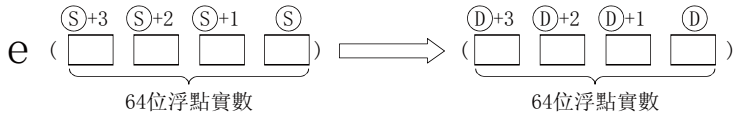
Ⓢ ：進行指數運算的資料或者存儲資料的軟元件的起始編號（實數）。

Ⓓ ：存儲運算結果的軟元件的起始編號（實數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 以Ⓢ中指定的值進行指數運算，並將運算結果存儲到Ⓓ中指定編號的位軟元件中。



(2) 在指數運算中，將底 (e) 作爲 “ 2.71828 ” 進行運算。

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出　錯

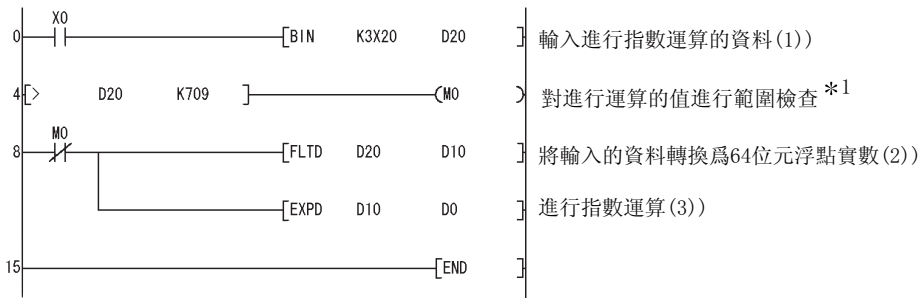
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 （發生了溢出時。） $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲以 X20 ～ X31 中以 BCD2 位數格式設置的值進行指數運算後，將其結果以 64 位元浮點實數存儲到 D0 ～ D3 中的程式。

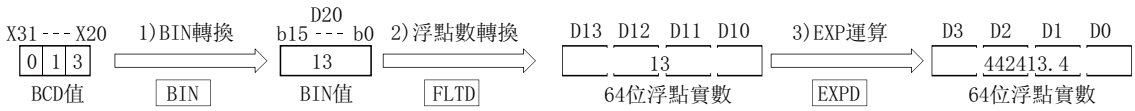
[梯形圖模式]



[列表模式]

步	指令	軟元件	
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		

[X20 ～ X31 中指定爲 13 時的動作]



*1: 由於 loge 2^{1024}=709.7832，爲了讓運算結果低於 2^{1024}，X20 ～ X31 的 BCD 值應爲 709 以下。由於設置值超過 710 以上時將會變爲運算出錯狀態，因此應將程式設置爲，如果設置了 710 以上的值，則 M0 將變爲 ON，不執行運算。

要 點

自然對數至常用對數的轉換

在 CPU 中是使用自然對數進行運算的。

若要對常用對數值進行計算，應將常用對數值用 0.43429 相除後的值指定到Ⓢ中。

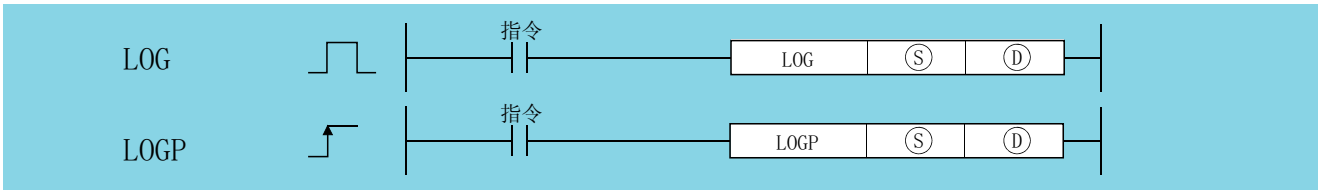
$$10^{x}=e^{\frac{x}{0.43429}}$$

Ver.

Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

7.12.23 LOG、LOGP



Ⓢ : 進行自然對數運算的資料或者存儲資料的軟元件的起始編號 (實數)。

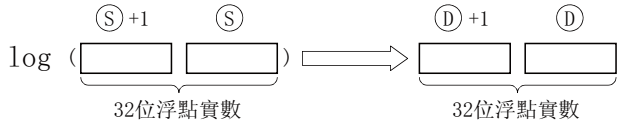
Ⓓ : 存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		○ *1	○	---
Ⓓ	---	○		---	○		○ *1	---	---

*1: 在通用型 QCPU、LCPU 中可以使用。

功 能

(1) 對Ⓢ中指定值的以自然對數 (e) 爲底時的對數進行運算，並將運算結果存儲到Ⓓ中指定編號的位軟元件中。



(2) Ⓢ中指定的值只能設置爲正數。(設置爲負數時不能執行運算。)

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

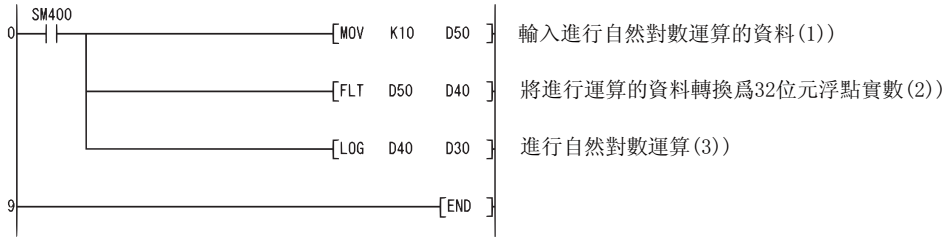
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的值爲負數時。 Ⓢ中指定的值爲 “ 0 ” 時。	○	○	○	○	○	---
	指定軟元件的內容爲 -0 時。*2	○	○	○	○	---	---
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ 指定軟元件的內容爲 -0、非正規數、非數、±∞時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。 (發生了溢出時。) $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○

*2: 有的 CPU 模組即使指定爲 -0 也不會變爲運算出錯狀態。
有關詳細內容請參閱 85 頁 3.2.4 項。

程式示例

(1) 以下爲求出 D50 中設置的 “ 10 ” 的自然對數後，存儲到 D30、D31 中的程式。

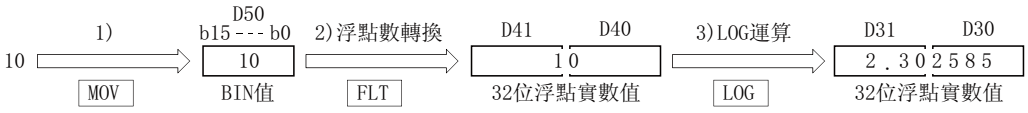
[梯形圖模式]



[列表模式]

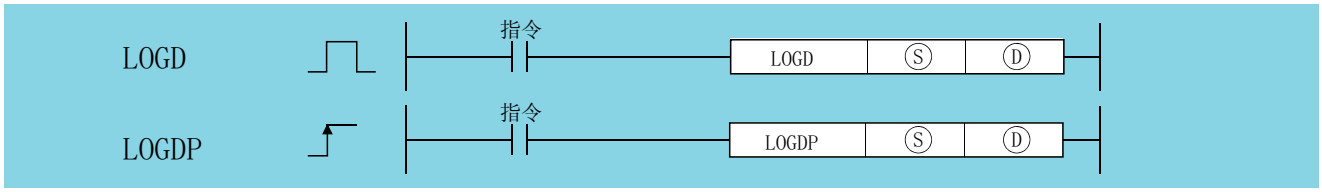
步	指令	軟元件	
0	LD	SM400	
1	MOV	K10	D50
3	FLT	D50	D40
6	LOG	D40	D30
9	END		

[動作]



7.12.24 LOGD、LOGDP

Basic High performance Process Redundant Universal LCPU



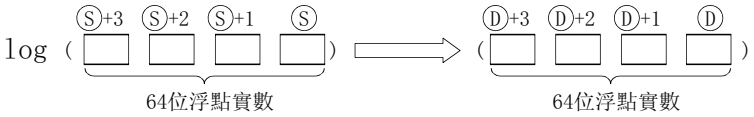
Ⓢ ：進行自然對數運算的資料或者存儲資料的軟元件的起始編號（實數）。

Ⓓ ：存儲運算結果的軟元件的起始編號（實數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 對Ⓢ中指定值的以自然對數 (e) 爲底時的對數進行運算，並將運算結果存儲到Ⓓ中指定編號的位軟元件中。



(2) Ⓢ中指定的值只能設置爲正數。（設置爲負數時不能執行運算。）

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 進行處理。

出　錯

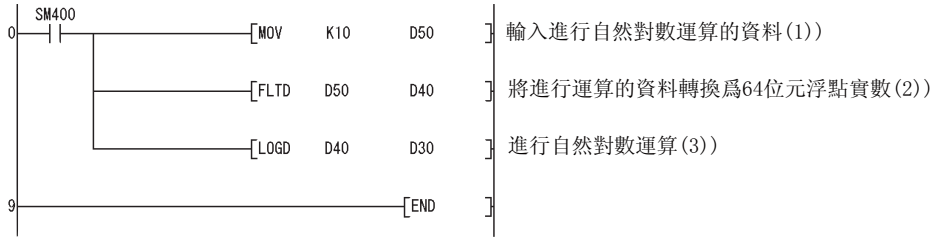
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的值爲負數時。 Ⓢ中指定的值爲“0”時。	---	---	---	---	○	○
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ 指定軟元件的內容爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出了下述範圍時。（發生了溢出時。） $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲求出 D50 中設置的“10”的自然對數後，存儲到 D30 ～ D33 中的程式。

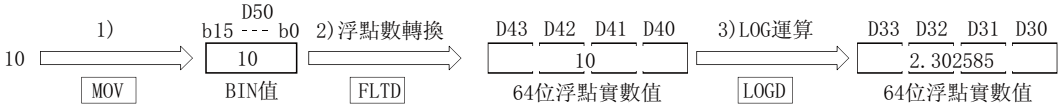
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	SM400	
1	MOV	K10	D50
3	FLTD	D50	D40
6	LOGD	D40	D30
9	END		

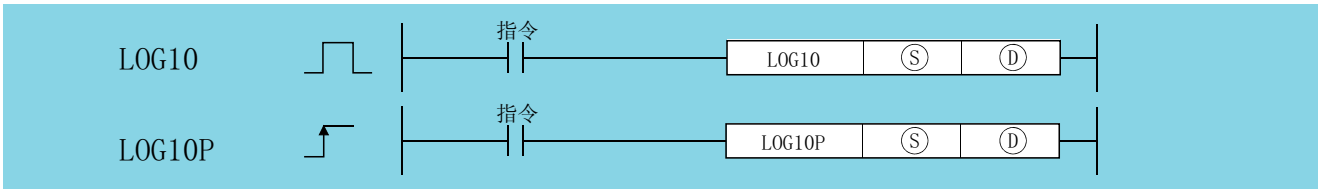
[動作]



Basic High performance Process Redundant Ver. Universal LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。

7.12.25 LOG10、LOG10P



Ⓢ ： 進行常用對數運算的資料或者存儲資料的軟元件的起始編號 (實數)。

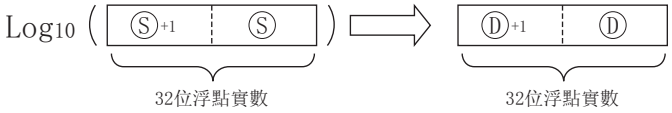
Ⓓ ： 存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 E	其他
	位	字		位	字				
Ⓢ	---	○		---	○		---	△ *1	---
Ⓓ	---	○		---	○		---	---	---

*1: 僅實數可以使用。

功 能

(1) 對Ⓢ中指定的值進行常用對數 (以 10 爲底的對數) 運算，並將運算結果存儲到Ⓓ中指定編號的位軟元件中。



(2) Ⓢ中指定的值只能設置爲正數。(設置爲負數時不能執行運算。)

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 處理。

出 錯

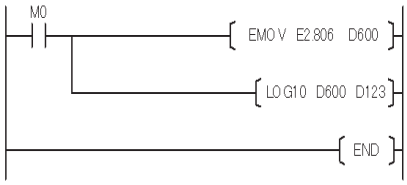
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的值爲負數時。 Ⓢ中指定的值爲 “ 0 ” 時。	---	---	---	---	○	○
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-126} \leq$ | 指定軟元件的內容 | $< 2^{128}$ Ⓢ中指定的值爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出以下範圍時。(發生了溢出時。) $2^{128} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲 M0 變爲 ON 時，對 D600、D601 中存儲的 32 位元浮點實數的常用對數進行計算後，將運算結果存儲到 D123、D124 中的程式。

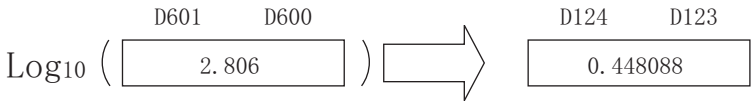
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	M0	
1	EMOV	E2.806	D600
4	LOG10	D600	D123
7	END		

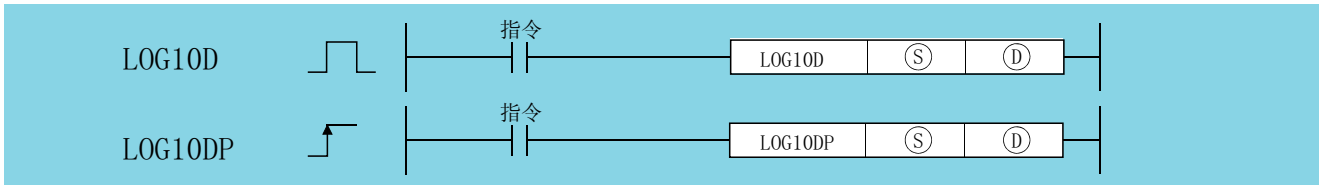
[動作]



7.12.26 LOG10D、LOG10DP

Basic ✕ | High performance ✕ | Process ✕ | Redundant ✕ | Ver. Universal | LCPU

· QnU(D)(H)CPU：序列號的前 5 位數爲 “ 10102 ” 以後



Ⓢ ：進行常用對數運算的資料或者存儲資料的軟元件的起始編號 (實數)。

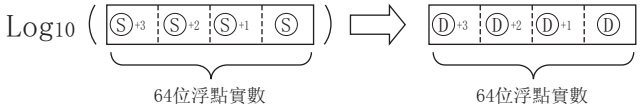
Ⓓ ：存儲運算結果的軟元件的起始編號 (實數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---				△ *1	---
Ⓓ	---	○		---				---	---

*1: 僅實數可以使用。

功　能

(1) 對Ⓢ中指定的值進行常用對數 (以 10 爲底的對數) 運算，並將運算結果存儲到Ⓓ中指定編號的位軟元件中。



(2) Ⓢ中指定的值只能設置爲正數。(設置爲負數時不能執行運算。)

(3) 運算結果爲 -0 或者發生了下溢時，將運算結果作爲 0 處理。

出　錯

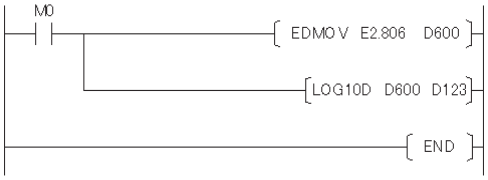
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ 中指定的值爲負數時。 Ⓢ 中指定的值爲 “ 0 ” 時。	---	---	---	---	○	○
4140	指定軟元件的內容不在下述範圍內時。 0、$2^{-1022} \leq$ | 指定軟元件的內容 | $< 2^{1024}$ Ⓢ 中指定的值爲 -0 時。	---	---	---	---	○	○
4141	運算結果超出以下範圍時。(發生了溢出時。) $2^{1024} \leq$ | 運算結果 |	---	---	---	---	○	○

程式示例

(1) 以下爲 M0 變爲 ON 時，對 D600 ～ D603 中存儲的 64 位元浮點實數的常用對數進行計算後，將運算結果存儲到 D123 ～ D126 中的程式。

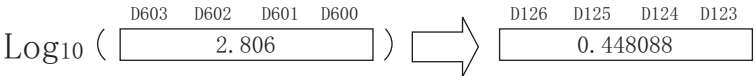
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	M0	
1	EDMOV	E2.806	D600
4	LOG10D	D600	D123
7	END		

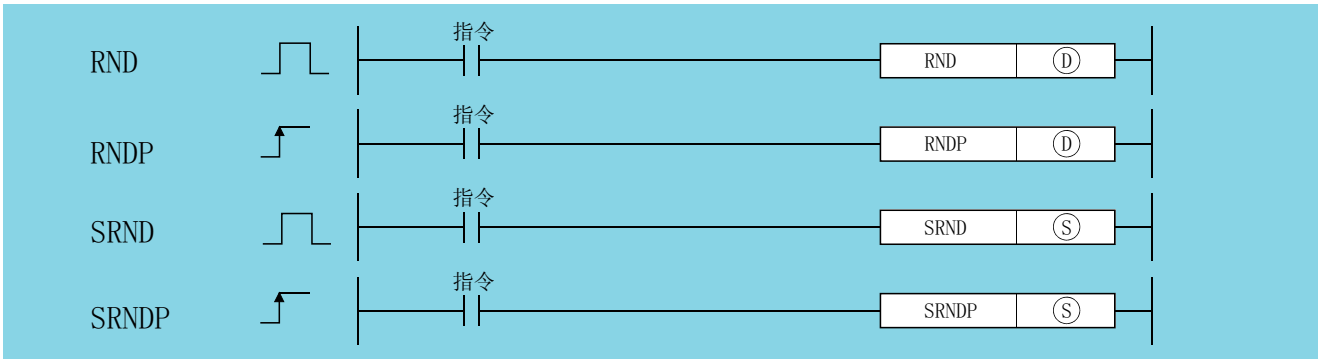
[動作]



Ver.
Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

7.12.27 RND、RNDP、SRND、SRNDP



Ⓓ ： 存儲亂數的軟元件的起始編號 (BIN16 位)。

Ⓢ ： 亂數系列資料或者存儲隨機系列資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓓ	○							---	---
Ⓢ	○							○	---

功　能

亂數產生指令用於按照某個計算公式生成亂數。在計算公式中，將上次的計算結果作爲係數使用。

通過系列變更指令可以更改亂數生成模式。

RND

生成 0 ～ 32767 的亂數後，存儲到ⓓ中指定的軟元件中。

SRND

按照Ⓢ中指定的軟元件中存儲的 16 位元 BIN 資料的內容對亂數系列進行變更。

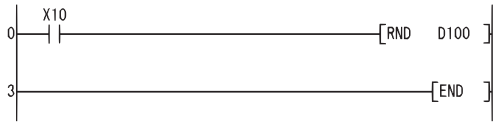
出　錯

(1)　在 RND(P)、SRND(P) 指令中無運算出錯。

程式示例

(1)　以下爲 X10 變爲 ON 時，將亂數存儲到 D100 中的程式。

[梯形圖模式]

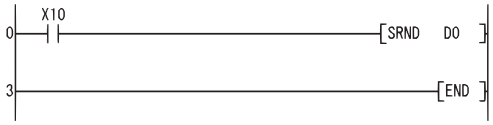


[列表模式]

步	指令	軟元件
0	LD	X10
1	RND	D100
3	END	

(2)　以下爲 X10 變爲 ON 時，按照 D0 中的內容對亂數系列進行變更的程式。

[梯形圖模式]

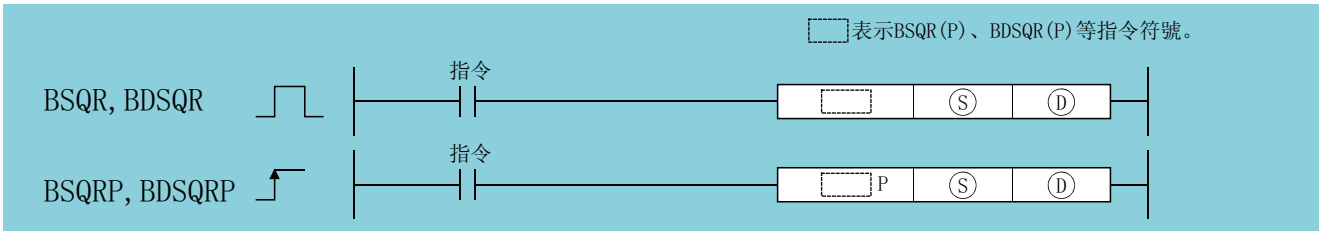


[列表模式]

步	指令	軟元件
0	LD	X10
1	SRND	D0
3	END	

7.12.28 BSQR、BSQRP、BDSQR、BDSQRP

Basic　High performance　Process　Redundant　Universal　LCPU



Ⓢ ：進行平方根運算的資料或者存儲資料的軟元件編號 BSQR(P)：BCD4 位；BDSQR(P)：BCD8 位。

ⓓ ：存儲運算結果的軟元件的起始編號 (BCD4 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○							○	---
ⓓ	○							---	---

功　能

BSQR

(1) 進行Ⓢ中指定值的平方根運算後，將運算結果存儲到Ⓓ中指定的位元軟元件中。

$$\sqrt{Ⓢ} = \boxed{Ⓓ\text{整數部分}}.\boxed{Ⓓ+1\text{小數部分}}$$

(2) Ⓢ中指定值的範圍爲最多 4 位數的 BCD 值 (0 ～ 9999)。

(3) Ⓓ、Ⓓ+1 的運算結果分別以 0 ～ 9999 的 BCD 值進行存儲。

(4) 運算結果爲小數部分第 5 位元數被四捨五入後的值。
因此，小數部分第 4 位元數將產生 ± 1 的誤差。

BDSQR

(1) 進行Ⓢ、Ⓢ+1 中指定值的平方根運算後，將運算結果存儲到Ⓓ中指定的位元軟元件中。

$$\sqrt{(\underbrace{\boxed{Ⓢ+1}\ \boxed{Ⓢ}}_{\text{2字資料}})} = \boxed{Ⓓ\text{整數部分}}.\boxed{Ⓓ+1\text{小數部分}}$$

(2) Ⓢ、Ⓢ+1 中指定值的範圍爲最多 8 位數的 BCD 值 (0 ～ 99999999)。

(3) Ⓓ、Ⓓ+1 的運算結果分別以 0 ～ 9999 的 BCD 值進行存儲。

(4) 運算結果爲小數部分第 5 位元數被四捨五入後的值。
因此，小數部分第 4 位元數將產生 ± 1 的誤差。

出　錯

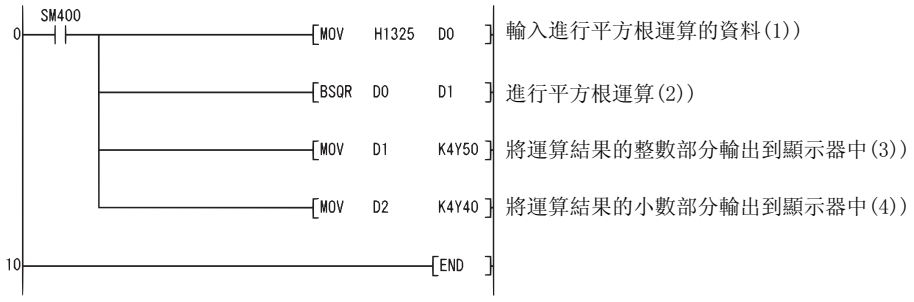
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的資料不是 BCD 值時。	---	○	○	○	○	○

程式示例

(1) 以下爲對 BCD 值 1325 進行平方根運算後，將整數部分以 BCD4 位元數輸出到 Y50 ～ Y5F 中，將小數部分以 BCD4 位元數輸出到 Y40 ～ Y4F 中的程式。

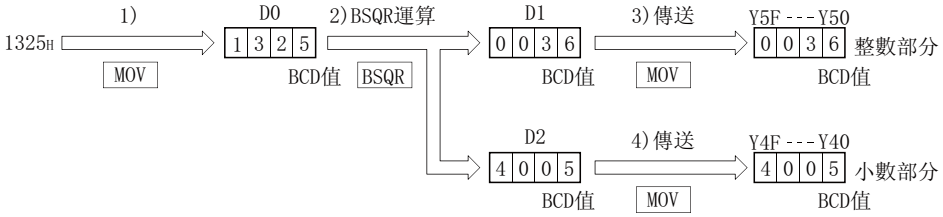
[梯形圖模式]



[列表模式]

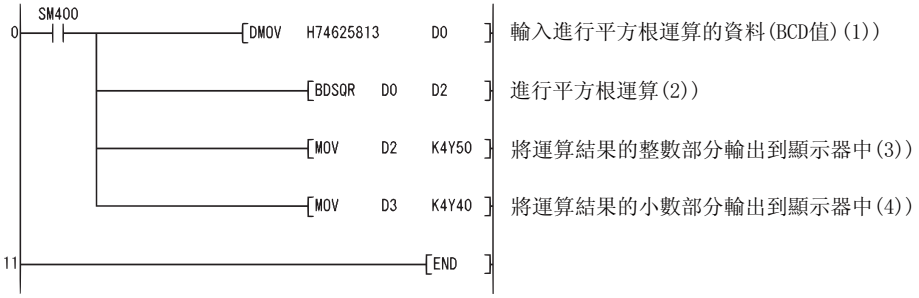
步	指令	軟元件	
0	LD	SM400	
1	MOV	H1325	D0
3	BSQR	D0	D1
6	MOV	D1	K4Y50
8	MOV	D2	K4Y40
10	END		

[動作]



(2) 以下爲對 BCD 值 74625813 進行平方根運算後，將整數部分以 BCD4 位元數輸出到 Y50 ～ Y5F 中，將小數部分以 BCD4 位元數輸出到 Y40 ～ Y4F 中的程式。

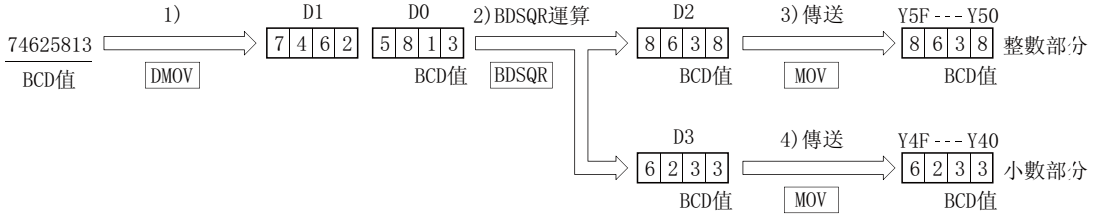
[梯形圖模式]



[列表模式]

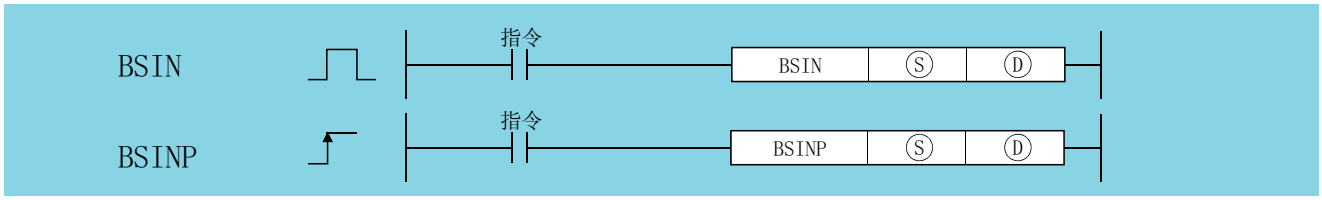
步	指令	軟元件	
0	LD	SM400	
1	DMOV	H74625813	D0
4	BDSQR	D0	D2
7	MOV	D2	K4Y50
9	MOV	D3	K4Y40
11	END		

[動作]



7.12.29 BSIN、BSINP

Basic (×) | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：進行 SIN(正弦) 運算的資料或者存儲資料的軟元件的起始編號 (BCD4 位數)。

Ⓓ ：存儲運算結果的軟元件的起始編號 (BCD4 位數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○				○			---
Ⓓ	---	○				---			---

功　能

(1) 對Ⓢ中指定的值（角度）進行 SIN（正弦）運算，並將運算結果的符號存儲到Ⓓ中指定的軟元件中，將運算結果存儲到Ⓓ+1、Ⓓ+2 中指定的軟元件中。

SIN Ⓢ ＝ [Ⓓ 符號] [Ⓓ+1 整數部分] . [Ⓓ+2 小數部分]

(2) Ⓢ中指定的值是以 0 ～ 360 °(DEG. 單位）的 BCD 值進行設置的。

(3) 對於Ⓓ中存儲的運算結果的符號，在運算結果爲正時存儲“ 0 ”，爲負時存儲“ 1 ”。

(4) 存儲到Ⓓ+1、Ⓓ+2 中的運算結果的範圍爲 -1.000 ～ 1.000 的 BCD 值。

(5) 運算結果爲小數部分第 5 位元數被四捨五入後的值。

出　錯

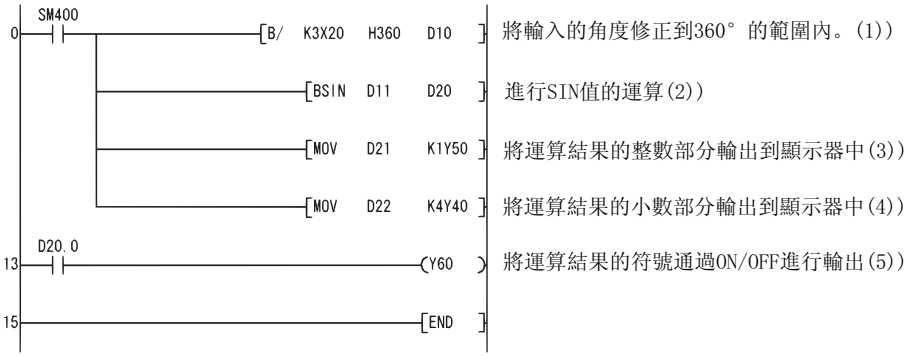
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的資料不是 BCD 值時。 Ⓢ中指定的資料超出了 0 ～ 360 的範圍時。	---	○	○	○	○	○
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲對 X20 ～ X2B 中以 BCD3 位元數指定的資料進行 SIN 運算後，將整數部分以 BCD1 位元數輸出到 Y50 ～ Y53 中，將小數部分以 BCD4 位元數輸出到 Y40 ～ Y4F 中的程式。
運算結果爲負數時將 Y60 置爲 ON。（在 X20 ～ X2F 中設置了 360 以上的值時，將所設置的值修正到 0 ～ 360 的範圍內。）

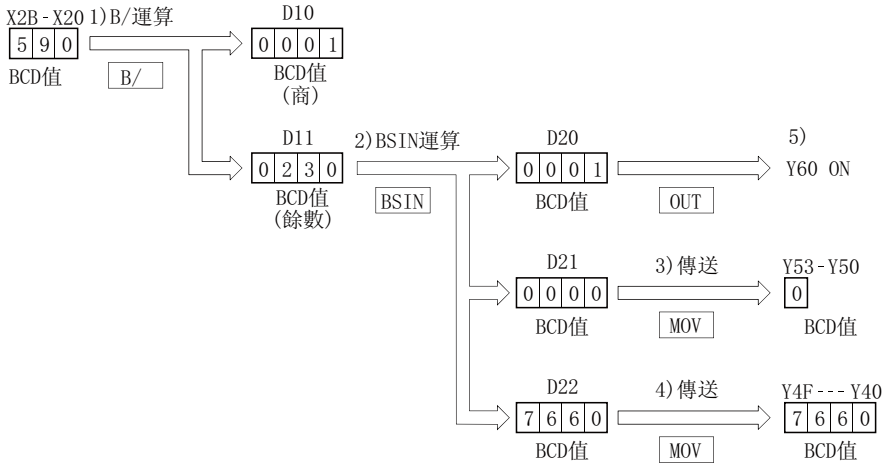
[梯形圖模式]



[列表模式]

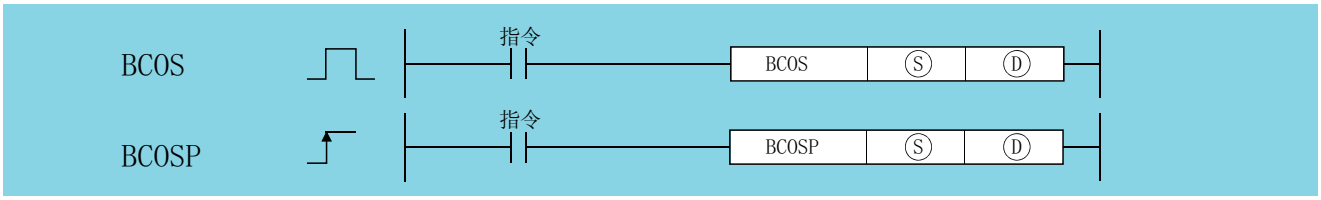
步	指令	軟元件		
0	LD	SM400		
1	B/	K3X20	H360	D10
5	BSIN	D11	D20	
8	MOV	D21	K1Y50	
11	MOV	D22	K4Y40	
13	LD	D20.0		
14	OUT	Y60		
15	END			

[在 X20 ～ X2B 中指定了 590 時的動作]



7.12.30 BCOS、BCOSP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：進行 COS(余弦) 運算的資料或者存儲資料的軟元件的起始編號 (BCD4 位數)。

Ⓓ ：存儲運算結果的軟元件的起始編號 (BCD4 位數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○		○					---
Ⓓ	---	○		---					---

功　能

(1) 對Ⓢ中指定的值 (角度) 進行 COS(余弦) 運算，並將運算結果的符號存儲到Ⓓ中指定的字軟元件中，將運算結果存儲到Ⓓ+1、Ⓓ+2 中指定的字軟元件中。

COS Ⓢ ＝ Ⓓ 符號　Ⓓ+1 整數部分 . Ⓓ+2 小數部分

(2) Ⓢ中指定的值是以 0 ～ 360 ° (DEG. 單位) 的 BCD 值進行設置的。

(3) 對於Ⓓ中存儲的運算結果的符號，在運算結果爲正時存儲 “ 0 ”，爲負時存儲 “ 1 ”。

(4) 存儲到Ⓓ+1、Ⓓ+2 中的運算結果的範圍爲 -1.000 ～ 1.000 的 BCD 值。

(5) 運算結果爲小數部分第 5 位元數被四捨五入後的值。

出　錯

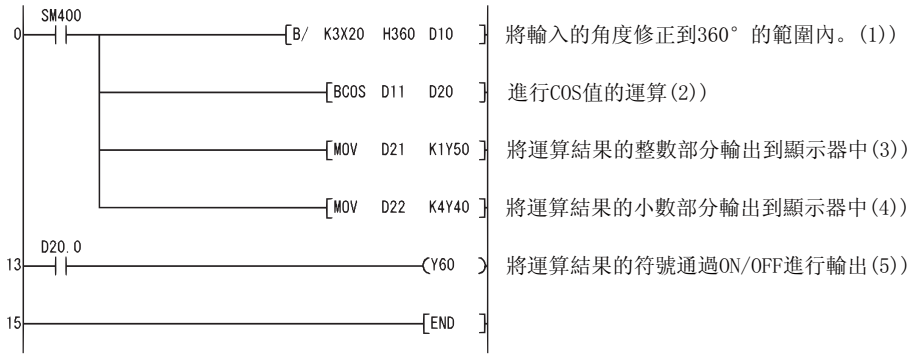
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的資料不是 BCD 值時。 Ⓢ中指定的資料超出了 0 ～ 360 的範圍時。	---	○	○	○	○	○
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲對 X20 ～ X2B 中以 BCD3 位元數指定的資料進行 COS 運算後，將整數部分以 BCD1 位元數輸出到 Y50 ～ Y53 中，將小數部分以 BCD4 位元數輸出到 Y40 ～ Y4F 中的程式。
運算結果爲負數時將 Y60 置爲 ON。

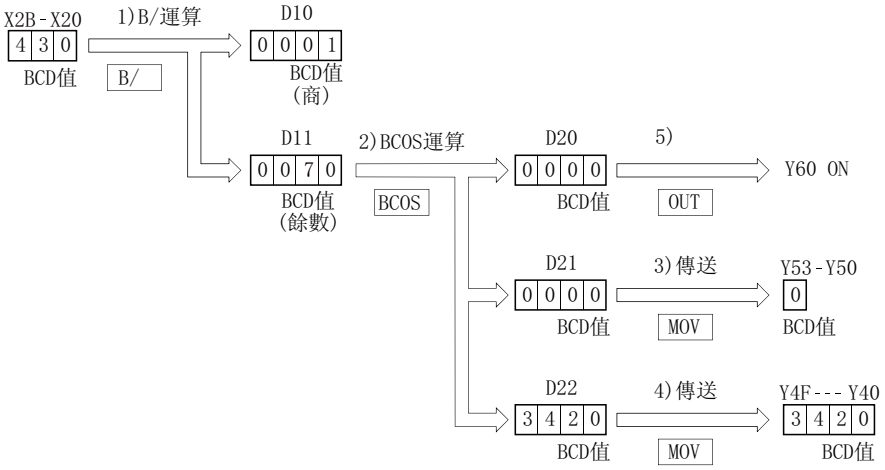
[梯形圖模式]



[列表模式]

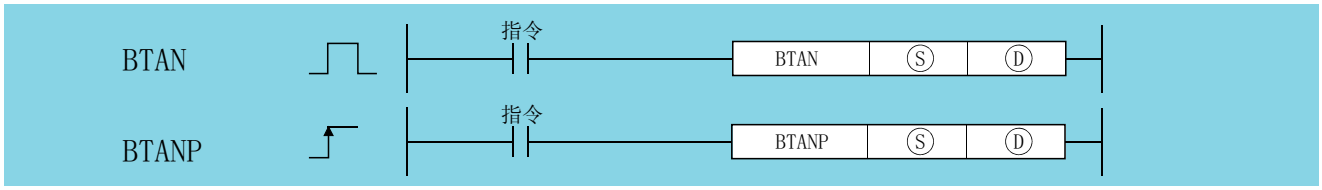
步	指令	軟元件		
0	LD	SM400		
1	B/	K3X20	H360	D10
5	BCOS	D11	D20	
8	MOV	D21	K1Y50	
11	MOV	D22	K4Y40	
13	LD	D20.0		
14	OUT	Y60		
15	END			

[在 X20 ～ X2B 中指定了 430 時的動作]



7.12.31 BTAN、BTANP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：進行 TAN(正切) 運算的資料或者存儲資料的軟元件的起始編號 (BCD4 位數)。

Ⓓ ：存儲運算結果的軟元件的起始編號 (BCD4 位數)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○		○					---
Ⓓ	---	○		---					---

功　能

(1) 對Ⓢ中指定的值 (角度) 進行 TAN(正切) 運算，並將運算結果的符號存儲到Ⓓ中指定的軟元件中，將運算結果存儲到Ⓓ+1、Ⓓ+2 中指定的軟元件中。

TAN Ⓢ ＝ [Ⓓ 符號] [Ⓓ+1 整數部分] . [Ⓓ+2 小數部分]

(2) Ⓢ中指定的值是以 0 ～ 360 ° (DEG. 單位) 的 BCD 值進行設置的。

(3) 對於Ⓓ中存儲的運算結果的符號，在運算結果爲正時存儲 “ 0 ”，爲負時存儲 “ 1 ”。

(4) 存儲到Ⓓ+1、Ⓓ+2 中的運算結果的範圍爲 -57.2901 ～ 57.2902 的 BCD 值。

(5) 運算結果爲小數部分第 5 位元數被四捨五入後的值。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

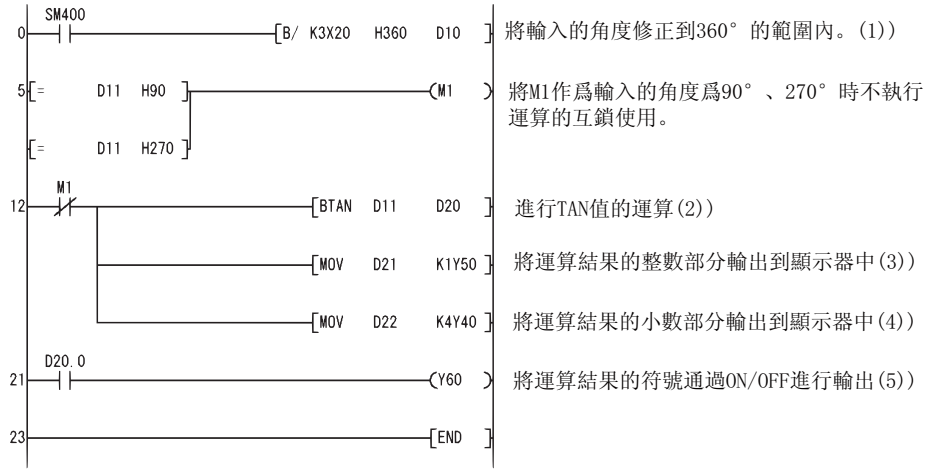
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的資料不是 BCD 值時。 Ⓢ中指定的資料超出了 0 ～ 360 的範圍時。 Ⓢ中指定的資料爲 90 °、270 ° 時。	---	○	○	○	○	○
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲對 X20 ～ X2B 中以 BCD3 位元數指定的資料進行 TAN 運算後，將整數部分以 BCD4 位元數輸出到 Y50 ～ Y53 中，將小數部分以 BCD4 位元數輸出到 Y40 ～ Y4F 中的程式。

運算結果爲負數時將 Y60 置爲 ON。

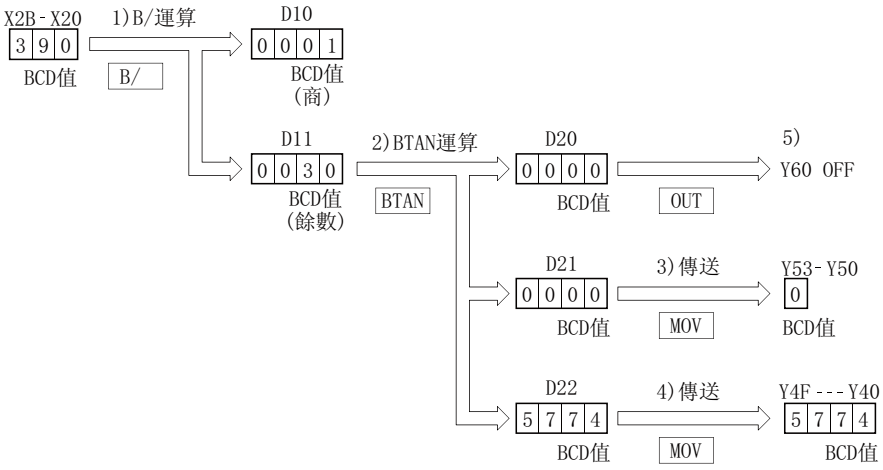
[梯形圖模式]



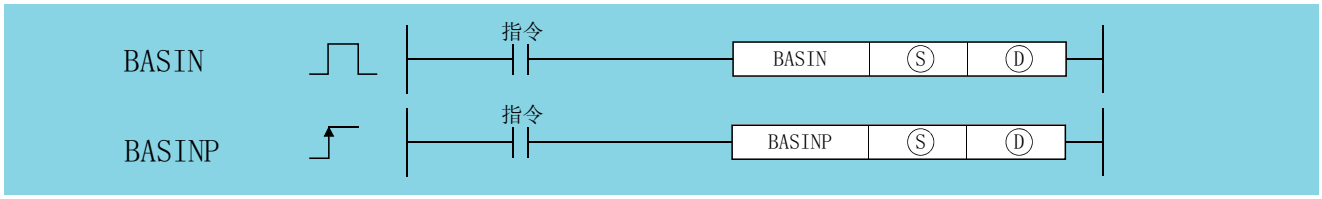
[列表模式]

步	指令	軟元件		
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			

[在 X20 ～ X2B 中指定了 390 時的動作]



7.12.32 BASIN、BASINP



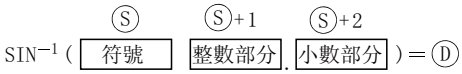
Ⓢ ：進行 SIN^{-1}（反正弦）運算的資料或者存儲資料的軟元件的起始編號（BCD4 位數）。

Ⓓ ：存儲運算結果的軟元件的起始編號（BCD4 位數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ	---	○		---				---	
Ⓓ	○	○		○				---	

功　能

(1) 對Ⓢ中指定的值（角度）進行 SIN^{-1}（反正弦）運算，並將運算結果（角度）存儲到Ⓓ中指定的軟元件中。



(2) 在Ⓢ中設置進行運算的資料的符號。
運算資料爲正時存儲“0”，爲負時存儲“1”。

(3) 在Ⓢ+1、Ⓢ+2 中，分別以 BCD 值存儲運算資料的整數部分及小數部分。
（可設置範圍爲 0 ～ 1.0000。）

(4) 存儲到Ⓓ中的運算結果爲 0 ～ 90 °、270 ～ 360 °（DEG. 單位）範圍內的 BCD 值。

(5) 運算結果爲小數部分被四捨五入後的值。

出　錯

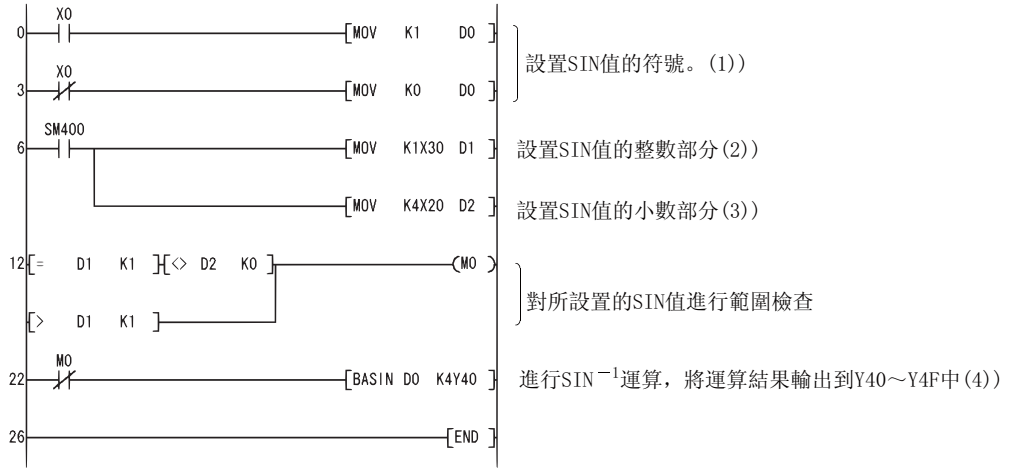
(1) 在以下情況下將變爲出錯狀態，出錯標誌（SM0）將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的資料不是 BCD 值時。 Ⓢ中指定的資料超出了 -1.0000 ～ 1.0000 的範圍時。	---	○	○	○	○	○
4101	Ⓢ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲對 X0 中設置了正負符號 (OFF 時爲正，ON 時爲負)，X30 ～ X33 中以 BCD1 位元數設置了整數部分，X20 ～ X2F 中以 BCD4 位元數設置了小數部分的值進行 SIN^{-1} 運算後，將求出的角度以 BCD4 位數輸出到 Y40 ～ Y4F 中的程式。

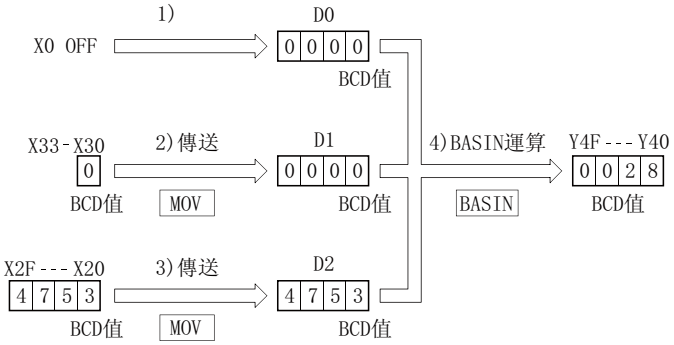
[梯形圖模式]



[列表模式]

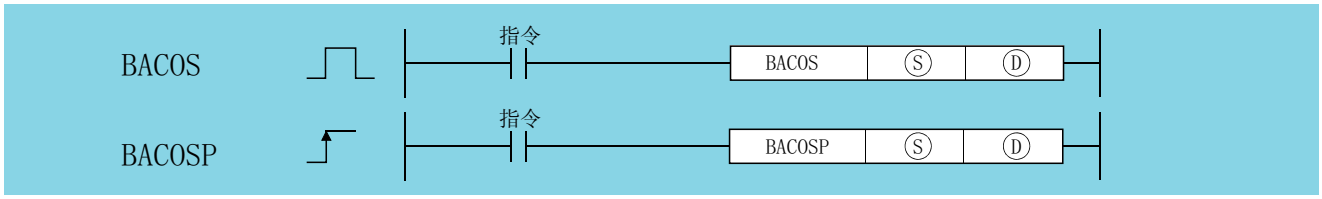
步	指令	軟元件	
0	LD	X0	
1	MOV	K1	D0
3	LDI	X0	
4	MOV	K0	D0
6	LD	SM400	
7	MOV	K1X30	D1
10	MOV	K4X20	D2
12	LD=	D1	K1
15	AND<>	D2	K0
18	OR>	D1	K1
21	OUT	M0	
22	LDI	M0	
23	BASIN	D0	K4Y40
26	END		

[在 X20 ～ X33 中指定了 0.4753 時的動作]



7.12.33 BACOS、BACOSP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：進行 COS^{-1}（反余弦）運算的資料或者存儲資料的軟元件的起始編號（BCD4 位數）。

Ⓓ ：存儲運算結果的軟元件的起始編號（BCD4 位數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ	---	○		---				---	
Ⓓ	○	○		○				---	

功　能

(1) 對Ⓢ中指定的值（角度）進行 COS^{-1}（反余弦）運算，並將運算結果（角度）存儲到Ⓓ中指定的軟元件中。

$$COS^{-1}(\boxed{\text{Ⓢ 符號}}\ \boxed{\text{Ⓢ+1 整數部分}}.\boxed{\text{Ⓢ+2 整數部分}}) = \text{Ⓓ}$$

(2) 在Ⓢ中設置進行運算的資料的符號。
運算資料爲正時存儲“0”，爲負時存儲“1”。

(3) 在Ⓢ+1、Ⓢ+2 中，分別以 BCD 值存儲運算資料的整數部分及小數部分。
（可設置範圍爲 0 ～ 1.0000。）

(4) 存儲到Ⓓ中的運算結果爲 0 ～ 180 °（DEG. 單位）範圍內的 BCD 值。

(5) 運算結果爲小數部分被四捨五入後的值。

出　錯

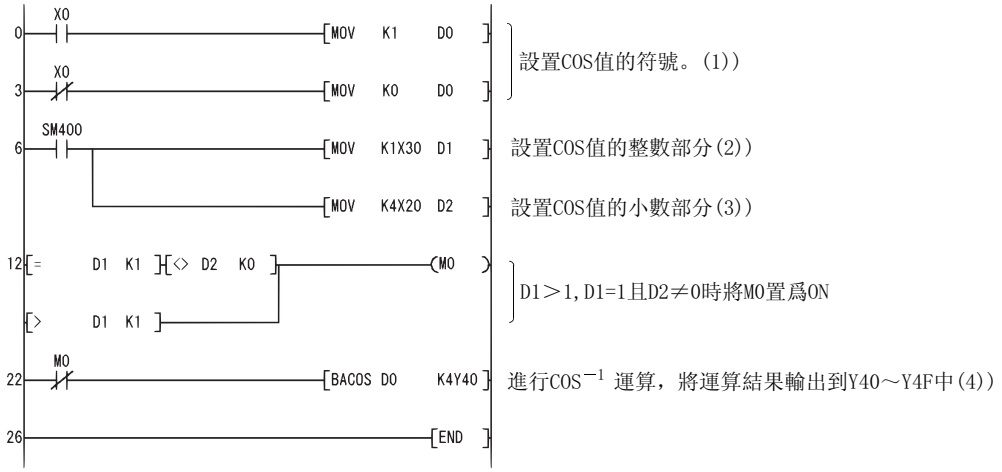
(1) 在以下情況下將變爲出錯狀態，出錯標誌（SM0）將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的資料不是 BCD 值時。 Ⓢ中指定的資料超出了 -1.0000 ～ 1.0000 的範圍時。	---	○	○	○	○	○
4101	Ⓢ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲對 X0 中設置了正負符號 (OFF 時爲正，ON 時爲負)，X30 ～ X33 中以 BCD1 位元數設置了整數部分，X20 ～ X2F 中以 BCD4 位元數設置了小數部分的值進行 COS^{-1} 運算後，將求出的角度以 BCD4 位數輸出到 Y40 ～ Y4F 中的程式。

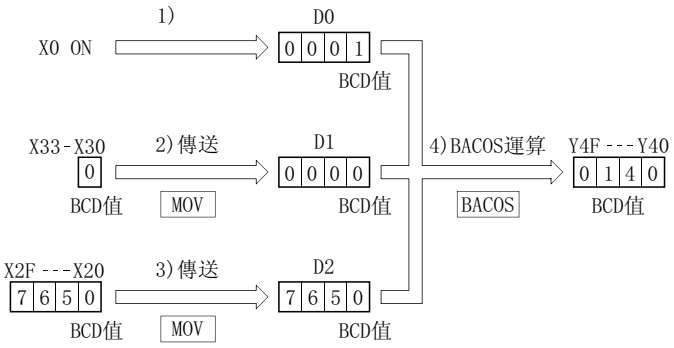
[梯形圖模式]



[列表模式]

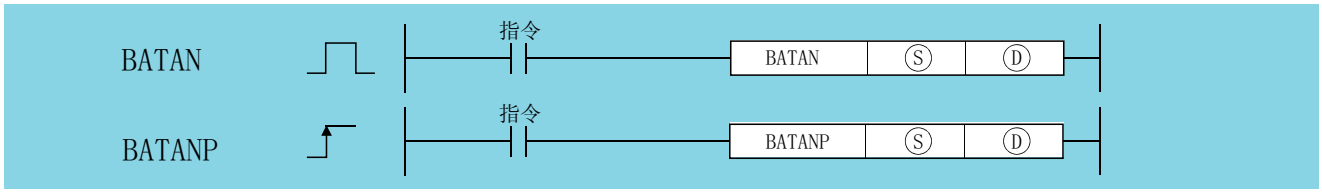
步	指令	軟元件	
0	LD	X0	
1	MOV	K1	D0
3	LDI	X0	
4	MOV	K0	D0
6	LD	SM400	
7	MOV	K1X30	D1
10	MOV	K4X20	D2
12	LD=	D1	K1
15	AND<>	D2	K0
18	OR>	D1	K1
21	OUT	M0	
22	LDI	M0	
23	BACOS	D0	K4Y40
26	END		

[在 X0、X20 ～ X33 中指定了 -0.7650 時的動作]



7.12.34 BATAN、BATANP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：進行 TAN^{-1}（反正切）運算的資料或者存儲資料的軟元件的起始編號（BCD4 位數）。

Ⓓ ：存儲運算結果的軟元件的起始編號（BCD4 位數）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ	---	○		---				---	
Ⓓ	○	○		○				---	

功　能

(1) 對Ⓢ中指定的值進行 TAN^{-1}（反正切）運算，並將運算結果（角度）存儲到Ⓓ中指定的軟元件中。

TAN^{-1}（ Ⓢ 符號 | Ⓢ+1 整數部分 . Ⓢ+2 小數部分 ）＝ Ⓓ

(2) 在Ⓢ中設置進行運算的資料的符號。
運算資料爲正時存儲“0”，爲負時存儲“1”。

(3) 在Ⓢ+1、Ⓢ+2 中，分別以 BCD 值存儲運算資料的整數部分及小數部分。
（可設置範圍爲 0 ～ 9999.9999。）

(4) 存儲到Ⓓ中的運算結果爲 0 ～ 90°、270 ～ 360°（DEG. 單位）範圍內的 BCD 值。

(5) 運算結果爲小數部分被四捨五入後的值。

出　錯

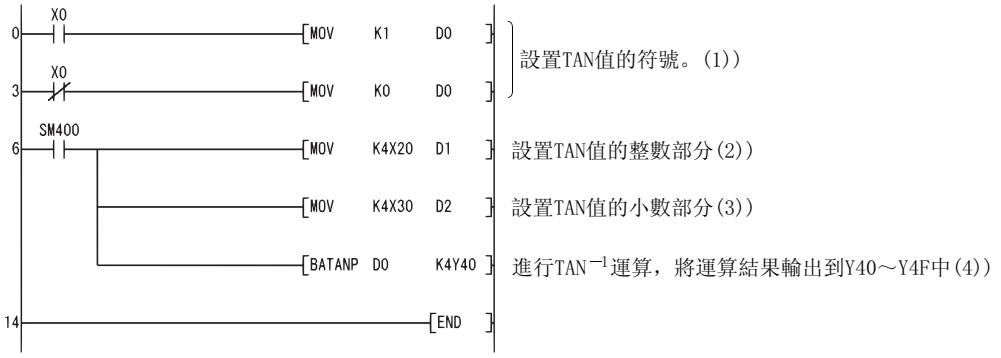
(1) 在以下情況下將變爲出錯狀態，出錯標誌（SM0）將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ中指定的資料不是 BCD 值時。	---	○	○	○	○	○
4101	Ⓢ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲對 X0 中設置了正負符號（OFF 時爲正，ON 時爲負），X20 ～ X2F 中以 BCD4 位元數設置了整數部分，X30 ～ X3F 中以 BCD4 位元數設置了小數部分的值進行 TAN^{-1} 運算後，將求出的角度以 BCD4 位數輸出到 Y40 ～ Y4F 中的程式。

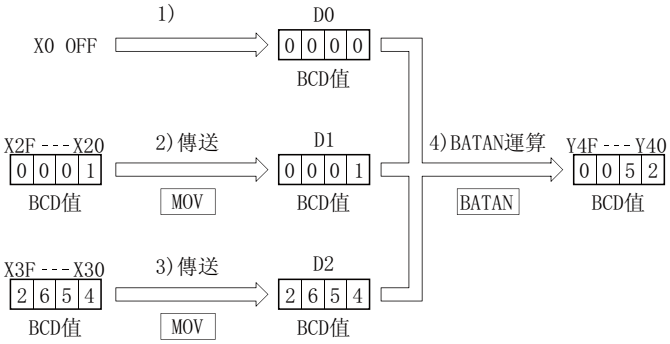
[梯形圖模式]



[列表模式]

步	指令	軟元件	
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		

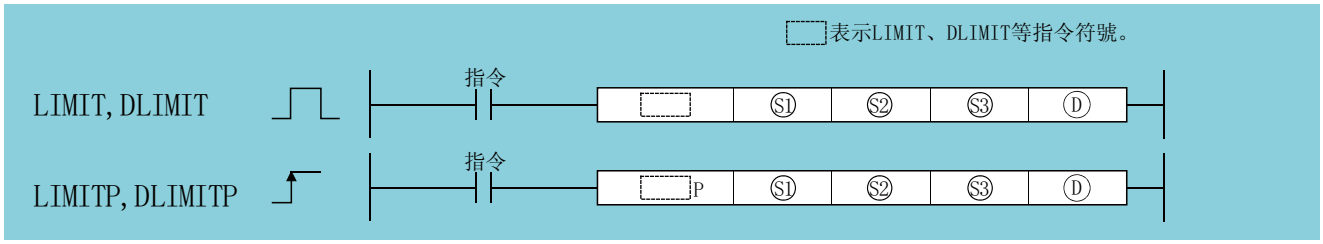
[在 X0、X20 ～ X2F 中指定了 1.2654 時的動作]



7.13 資料控制指令

7.13.1 LIMIT、LIMITP、DLIMIT、DLIMITP

Basic | High performance | Process | Redundant | Universal | LCPU



S1 ：下限值（最小輸出界限值）(BIN16/BIN32 位)。

S2 ：上限值（最大輸出界限值）(BIN16/BIN32 位)。

S3 ：通過上下限控制進行控制的輸入值 (BIN16/BIN32 位)。

D ：存儲通過上下限控制進行控制的輸入值的軟元件的起始編號 (BIN16/BIN32 位)。

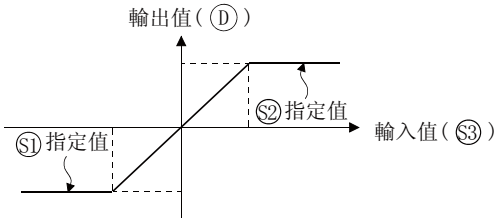
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
S1	○							○	---
S2	○							○	---
S3	○							○	---
D	○							---	---

功　能

LIMIT

(1) 根據S3中指定的輸入值 (BIN16 位值) 是否在S1、S2中指定的上下限值的範圍內，對存儲到D中指定的軟元件中的輸出值進行控制。

- S1 [下限值] > S3 [輸入值] 時 ..... S1 [下限值] → D [輸出值]
- S2 [上限值] < S3 [輸入值] 時 ..... S2 [上限值] → D [輸出值]
- S1 [下限值] ≤ S3 [輸入值] ≤ S2 [下限值] 時 ... S3 [輸入值] → D [輸出值]



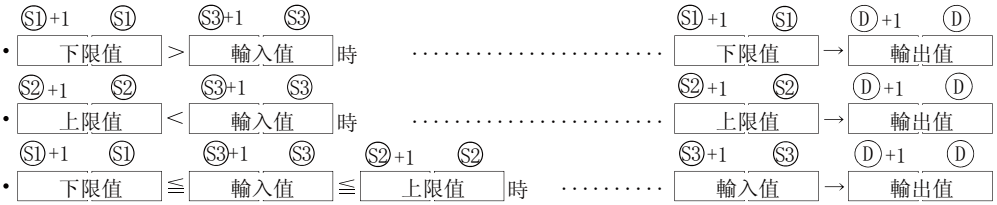
(2) S1、S2、S3中可指定的值的範圍爲 -32768 ～ 32767。

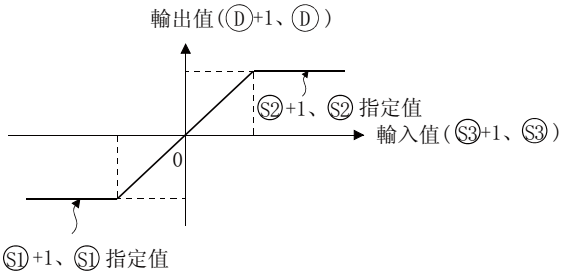
(3) 僅根據上限值進行控制時，在S1中指定的下限值中設置“ -32768 ”。

(4) 僅根據下限值進行控制時，在S2中指定的上限值中設置“ 32767 ”。

DLIMIT

(1) 根據 (Ⓢ3、Ⓢ3+1) 中指定的輸入值 (BIN32 位值) 是否在 (Ⓢ1、Ⓢ1+1)、(Ⓢ2、Ⓢ2+1) 中指定的上下限值的範圍內，對存儲到 (Ⓓ、Ⓓ+1) 中指定的軟元件中的輸出值進行控制。





(2) (Ⓢ1、Ⓢ1+1)、(Ⓢ2、Ⓢ2+1)、(Ⓢ3、Ⓢ3+1) 中可指定的值的範圍爲 -2147483648 ～ 2147483647。

(3) 僅根據上限值進行控制時，在 (Ⓢ1、Ⓢ1+1) 中指定的下限值中設置 “ -2147483648 ”。

(4) 僅根據下限值進行控制時，在 (Ⓢ2、Ⓢ2+1) 中指定的上限值中設置 “ 2147483647 ”。

出　錯

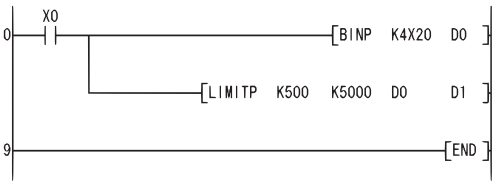
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ1中指定的下限值大於Ⓢ2中指定的上限值時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，對 X20 ～ X2F 中以 BCD 值設置的資料進行 500 ～ 5000 的限制控制，並將結果存儲到 D1 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X0			
1	BINP	K4X20	D0		
4	LIMITP	K500	K5000	D0	D1
9	END				

[動作]

· D0 < 500 時，D1 變爲 500。

例 D0 = 400 → D1 = 500

· 500 ≤ D0 ≤ 5000 時，D1 變爲 D0 的值。

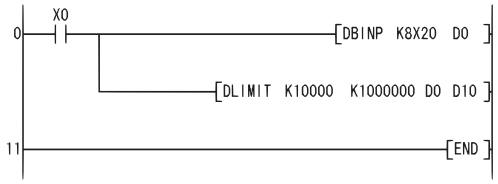
例 D0 = 1300 → D1 = 1300

· 5000 < D0 時，D1 變爲 5000。

例 D0 = 9600 → D1 = 5000

(2) 以下爲 X0 變爲 ON 時，對 X20 ～ X3F 中以 BCD 值設置的資料進行 10000 ～ 1000000 的限制控制，並將結果存儲到 D10、D11 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X0		
1	DBINP	K8X20	D0	
4	DLIMIT	K10000	K1000000 D0	D10
11	END			

[動作]

· (D1,D0) < 10000 時，(D11、D10) 變爲 10000。

例 (D1,D0) = 400 → (D11,D10) = 10000

· 10000 ≤ (D1,D0) ≤ 1000000 時，(D11、D10) 變爲 (D1、D0) 的值。

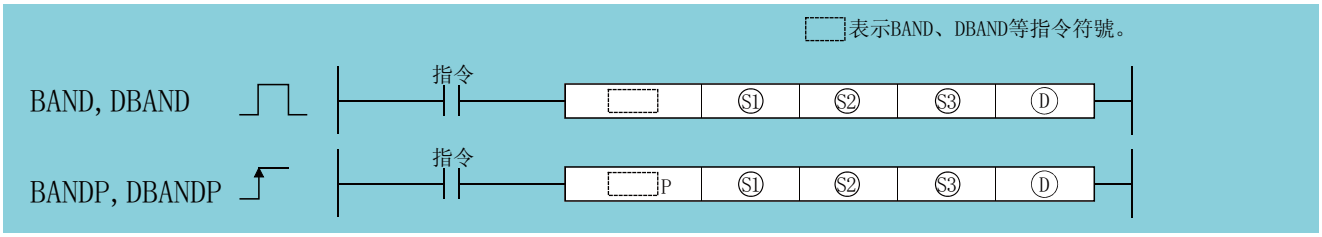
例 (D1,D0) = 345678 → (D11,D10) = 345678

· 1000000 < (D1,D0) 時，(D11、D10) 變爲 1000000。

例 (D1,D0) = 9876543 → (D11,D10) = 1000000

7.13.2 BAND、BANDP、DBAND、DBANDP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ1 ： 死區 (無輸出區域) 的下限值 (BIN16/BIN32 位)。

Ⓢ2 ： 死區 (無輸出區域) 的上限值 (BIN16/BIN32 位)。

Ⓢ3 ： 通過死區控制進行控制的輸入值 (BIN16/BIN32 位)。

Ⓓ ： 存儲通過死區控制進行控制的輸出值的軟元件的起始編號 (BIN16/BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1				○				○	---
Ⓢ2				○				○	---
Ⓢ3				○				○	---
Ⓓ				○				---	---

功 能

BAND

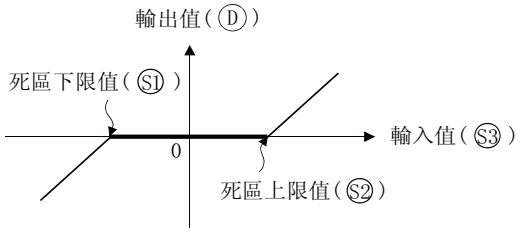
(1) 根據Ⓢ3中指定的輸入值 (BIN16 位值) 是否在Ⓢ1、Ⓢ2中指定的死區的上下限範圍內，對存儲到Ⓓ中指定的軟元件中的輸出值進行控制。。

輸出值的控制如下所示。

·Ⓢ1 [下限值] > Ⓢ3 [輸入值] 時 .....Ⓢ3 [輸入值] - Ⓢ1 [下限值] → Ⓓ [輸出值]

·Ⓢ2 [上限值] < Ⓢ3 [輸入值] 時 .....Ⓢ3 [輸入值] - Ⓢ2 [上限值] → Ⓓ [輸出值]

·Ⓢ1 [下限值] ≤ Ⓢ3 [輸入值] ≤ Ⓢ2 [上限值] 時 .....0 → Ⓓ [輸出值]



(2) Ⓢ1、Ⓢ2、Ⓢ3中可指定的值的範圍爲 -32768 ～ 32767。

(3) Ⓓ中存儲的輸出值爲帶符號的 16 位元 BIN 值。因此運算結果超出了 -32768 ～ 32767 的範圍時的情況如下所示。

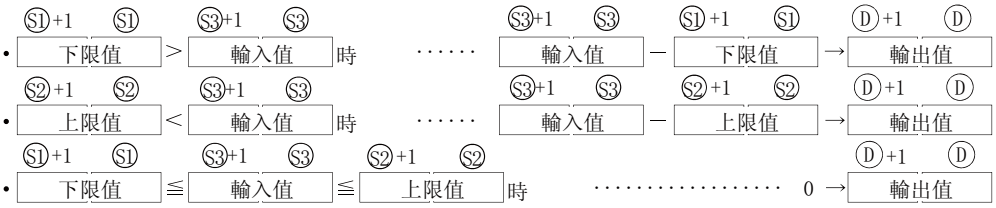
死區下限值 (Ⓢ1) 10
輸入值 (Ⓢ3) ..... -32768
} 時

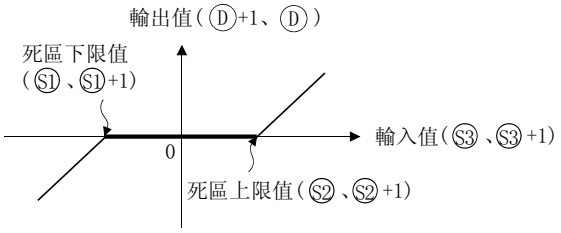
輸出值 =-32768-10=8000_H-A_H=$7FF6_H$=32758

DBAND

(1) 根據 (Ⓢ3、Ⓢ3+1) 中指定的輸入值 (BIN32 位值) 是否在 (Ⓢ1、Ⓢ1+1)、(Ⓢ2、Ⓢ2+1) 中指定的死區上下限值的範圍內，對存儲到Ⓓ中指定的軟元件中的輸出值進行控制。

輸出值的控制如下所示。





(2) (Ⓢ1、Ⓢ1+1)、(Ⓢ2、Ⓢ2+1)、(Ⓢ3、Ⓢ3+1) 中可指定的值的範圍爲 -2147483648 ～ 2147483647。

(3) Ⓓ、Ⓓ+1 中存儲的輸出值爲帶符號的 32 位元 BIN 值。因此運算結果超出了 -2147483648 ～ 2147483647 的範圍時的情況如下所示。

死區下限值 (Ⓢ1、Ⓢ1+1) 1000
輸入值 (Ⓢ3、Ⓢ3+1) ..... -2147483648
} 時

輸出值 =-2147483648-1000=80000000_H-$000003E8_H$
=$7FFFFC18_H$=2147482648。

出　錯

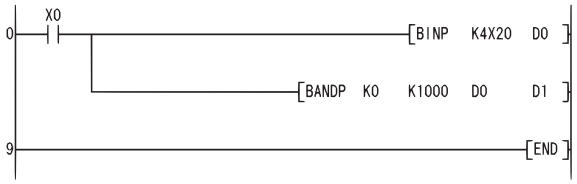
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	⑤1中指定的下限值大於⑤2中指定的上限值時。	---	---	---	---	○	○

程式示例

(1)　以下爲 X0 變爲 ON 時，對 X20 ～ X2F 中以 BCD 值設置的資料進行 0 ～ 1000 的死區控制，並將結果存儲到 D1 中的程序。

[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X0			
1	BINP	K4X20	D0		
4	BANDP	K0	K1000	D0	D1
9	END				

[動作]

· 0 ≤ D0 ≤ 1000 時，D1 中存儲 0。

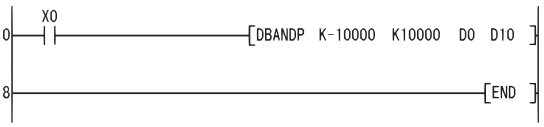
例　D0 = 500 → D1 = 0

· 1000 < D0 時，D1 中存儲 (D0)-1000 的值。

例　D0 = 7000 → D1 = 6000

(2)　以下爲 X0 變爲 ON 時，對 D0、D1 設置的資料進行 -10000 ～ 10000 的死區控制，並將結果存儲到 D10、D11 中的程序。

[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X0			
1	DBANDP	K-10000	K10000	D0	D10
8	END				

[動作]

· (D1、D0) < (-10000) 時，(D11、D10) 中存儲 (D1、D0)-(-10000) 的值。

例　(D1,D0) = -12345 → (D11,D10) = -2345

· -10000 ≤ (D1,D0) ≤ 10000 時，(D11、D10) 中存儲 0。

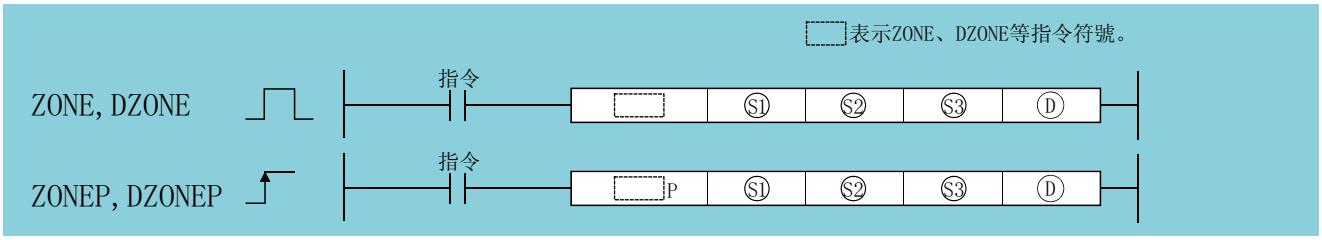
例　(D1,D0) = 6789 → (D11,D10) = 0

· 10000 < (D1,D0) 時，(D11、D10) 中存儲 (D1、D0)-10000 的值。

例　(D1,D0) = 50000 → (D11,D10) = 40000

7.13.3 ZONE、ZONEP、DZONE、DZONEP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ1 ： 對輸入值進行加法運算的負偏置值 (BIN16/BIN32 位)。

Ⓢ2 ： 對輸入值進行加法運算的正偏置值 (BIN16/BIN32 位)。

Ⓢ3 ： 用於進行區域控制的輸入值 (BIN16/BIN32 位)。

Ⓓ ： 存儲通過區域控制進行控制的輸入值的軟元件的起始編號 (BIN16/BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1				○				○	---
Ⓢ2				○				○	---
Ⓢ3				○				○	---
Ⓓ				○				---	---

功　能

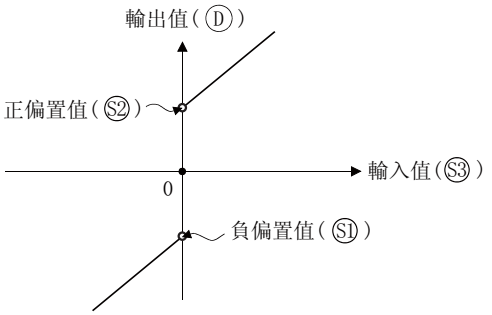
ZONE

(1) 在Ⓢ3中指定的輸入值上附加Ⓢ1或者Ⓢ2中指定的偏置值後，存儲到Ⓓ中指定編號的軟元件中。
偏置值的附加情況如下所示。

·Ⓢ3 [輸入值] < 0 時 ..... Ⓢ3 [輸入值] + Ⓢ1 [負偏置值] → Ⓓ [輸出值]

·Ⓢ3 [輸入值] = 0 時 ..... 0 → Ⓓ [輸出值]

·Ⓢ3 [輸入值] > 0 時 ..... Ⓢ3 [輸入值] + Ⓢ2 [正偏置值] → Ⓓ [輸出值]



(2) Ⓢ1、Ⓢ2、Ⓢ3中可指定的值的範圍爲 -32768 ～ 32767。

(3) Ⓓ中存儲的輸出值爲帶符號的 16 位元 BIN 值。因此運算結果超出了 -32768 ～ 32767 的範圍時的情況如下所示。

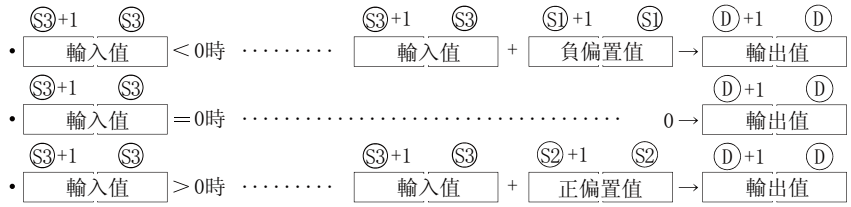
負偏置值 (Ⓢ1) ..... -100
輸入值 (Ⓢ3) ..... -32768
} 時

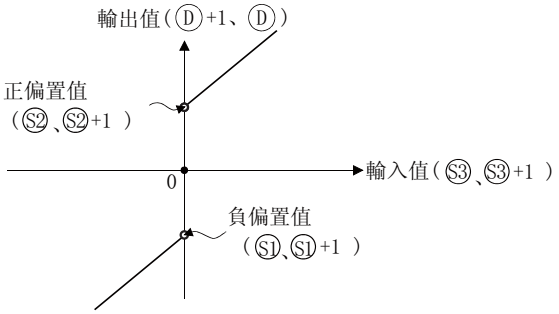
輸出值 = -32768+(-100)=8000_H+FF9C=7F9C$_H$=32668。

DZONE

(1) 在 (Ⓢ3、Ⓢ3+1) 中指定的輸入值中，附加了 (Ⓢ1、Ⓢ1+1) 或者 (Ⓢ2、Ⓢ2+1) 中指定的偏置值後，存儲到 (Ⓓ、Ⓓ+1) 中指定編號的軟元件中。

偏置值的附加情況如下所示。





(2) (Ⓢ1、Ⓢ1+1)、(Ⓢ2、Ⓢ2+1)、(Ⓢ3、Ⓢ3+1) 中可指定的值的範圍爲 -2147483648 ～ 2147483647。

(3) (Ⓓ、Ⓓ+1) 中存儲的值爲帶符號的 32 位元 BIN 值。

因此運算結果超出了 -2147483648 ～ 2147483647 的範圍時的情況如下所示。

負偏置值 (Ⓢ1、Ⓢ1+1) .....-1000
輸入值 (Ⓢ3、Ⓢ3+1) .....-2147483648
} 時

輸出值 = -2147483648+(-1000)=80000000_H+$FFFFFC18_H$

= 7FFFFC18=2147482648。

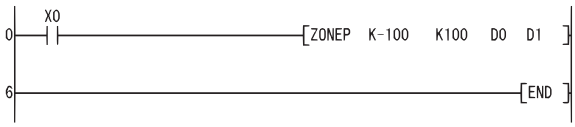
出　錯

(1) ZONE(P)、DZONE(P) 指令中無運算出錯。

程式示例

(1) 以下爲 X0 變爲 ON 時，對 D0 中設置的資料進行 -100 ～ 100 的區域控制，並將結果存儲到 D1 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X0			
1	ZONEP	K-100	K100	D0	D1
6	END				

[動作]

· D0 < 0 時，在 D1 中存儲 (D0)+(-100) 的值。

例 D0 = -200 → D1 = -300

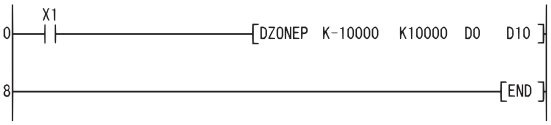
· D0 = 0 時，D1 中存儲 0。

· 0 < D0 時，D1 中存儲 (D0)+100 的值。

例 D0 = 700 → D1 = 800

(2) 以下爲 X0 變爲 ON 時，對 D0、D1 設置的資料進行 -10000 ～ 10000 的區域控制，並將結果存儲到 D10、D11 中的程序。

[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X1			
1	DZONEP	K-10000	K10000	D0	D10
8	END				

[動作]

· (D1,D0) < 0 時，(D11、D10) 中存儲 (D1、D0)+(-10000) 的值。

例 (D1,D0) = -12345 → (D11,D10) = -22345

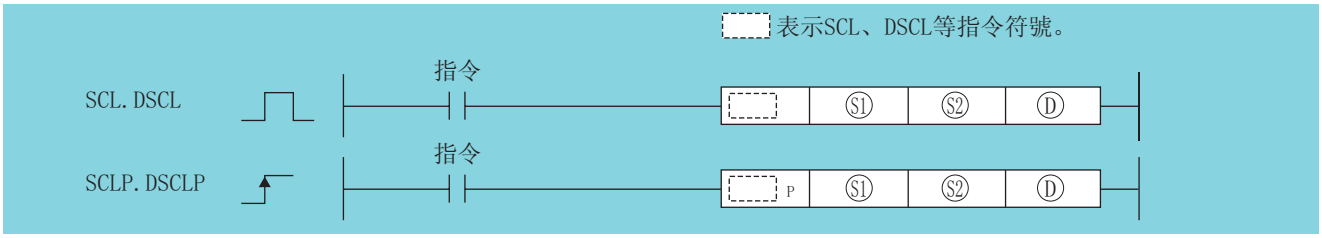
· (D1,D0) = 0 時，(D11、D10) 中存儲 0。

· 0 < (D1,D0) 時，(D11、D10) 中存儲 (D11、D10)+10000 的值。

例 (D1,D0) = 50000 → (D11,D10) = 60000

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。

7.13.4 SCL、SCLP、DSCL、DSCLP



Ⓢ1 ： 進行標度的輸入值或者存儲輸入值的軟元件的起始編號 (BIN16/32 位)。

Ⓢ2 ： 存儲標度用轉換資料的軟元件的起始編號 (BIN16/32 位)。

Ⓓ ： 存儲通過標度進行控制的輸出值的軟元件的起始編號 (BIN16/BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	---	○	○	○				○	---
Ⓢ2	---	○	○	---				---	---
Ⓓ	---	○	○	○				---	---

功 能

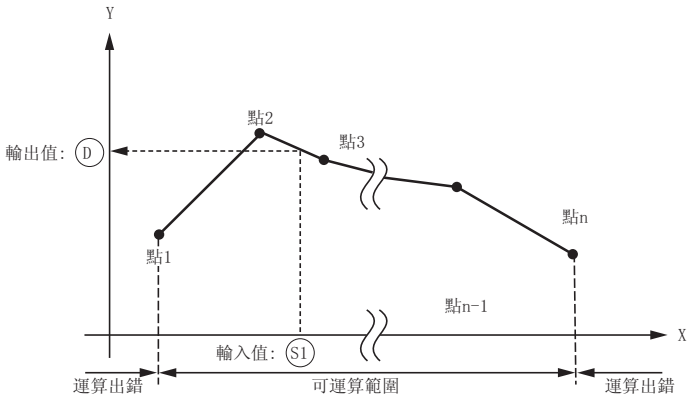
SCL(P)

(1) 對Ⓢ2中指定的標度用轉換資料 (16 位元資料單位)，通過Ⓢ1中指定的輸入值進行標度，並將運算結果存儲到Ⓓ中指定編號的軟元件中。

標度轉換是基於Ⓢ2中指定的軟元件後面存儲的標度用轉換資料進行的。

[標度用轉換資料的構成]n 表示Ⓢ2中指定的座標點數。

設置專案		軟元件分配
座標點數		Ⓢ2
點 1	X 座標	Ⓢ2+1
	Y 座標	Ⓢ2+2
點 2	X 座標	Ⓢ2+3
	Y 座標	Ⓢ2+4
⋮		
點 n	X 座標	Ⓢ2+2n-1
	Y 座標	Ⓢ2+2n



(2) 運算結果不是整數值時，對小數點以下第 1 位進行四捨五入。

(3) 標度用轉換資料的 X 座標資料應按昇冪進行設置。

(4) Ⓢ1應在標度用轉換資料範圍內 (Ⓢ2的軟元件值) 進行設置。

(5) 將多個點指定爲相同的 X 座標時，將對最大點號的點的 Y 座標值進行輸出。

(6) 標度用轉換資料的座標點數範圍爲 1 ～ 32767。

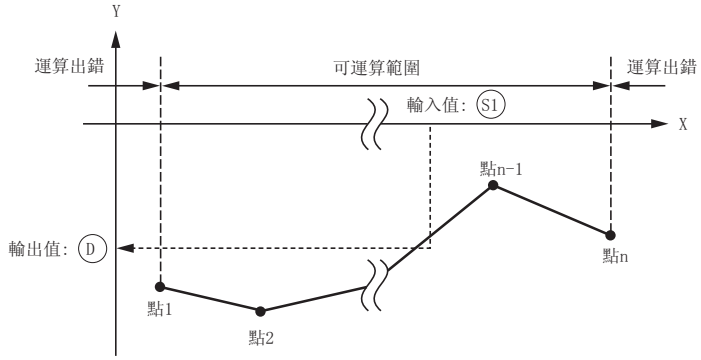
DSCL(P)

(1) 對Ⓢ2中指定的標度用轉換資料 (32 位元資料單位)，通過Ⓢ1中指定的輸入值進行標度，並將運算結果存儲到Ⓓ中指定編號的軟元件中。

標度轉換是基於Ⓢ2中指定的軟元件後面存儲的標度用轉換資料進行的。

[標度用轉換資料的構成]n 表示Ⓢ2中指定的座標點數。

設置專案		軟元件分配
座標點數		Ⓢ2+1、Ⓢ2
點 1	X 座標	Ⓢ2+3、Ⓢ2+2
	Y 座標	Ⓢ2+5、Ⓢ2+4
點 2	X 座標	Ⓢ2+7、Ⓢ2+6
	Y 座標	Ⓢ2+9、Ⓢ2+8
⋮		
點 n	X 座標	Ⓢ2+4n-1、Ⓢ2+4n-2
	Y 座標	Ⓢ2+4n+1、Ⓢ2+4n



(2) 運算結果不是整數值時，對小數點以下第 1 位進行四捨五入。

(3) 標度用轉換資料的 X 座標資料應按昇冪進行設置。

(4) Ⓢ1應在標度用轉換資料範圍內 (Ⓢ2、Ⓢ2+1 的軟元件值) 進行設置。

(5) 將多個點指定爲相同的 X 座標時，將對最大點號的點的 Y 座標值進行輸出。

(6) 標度用轉換資料的座標點數範圍爲 1 ～ 32767。

要 點

(1) 根據 SM750 的 ON/OFF 狀態其探索方法有所不同。

SM750	探索方法	探索次數範圍
OFF	逐次探索	1 ≤次數≤ 32767
ON	二分探索	1 ≤次數≤ 15

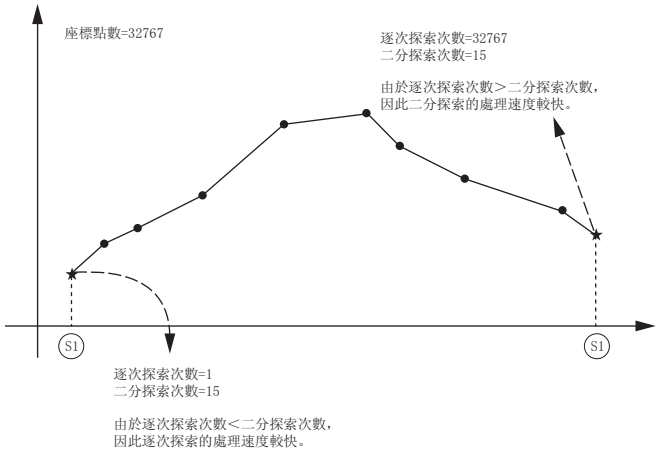
(2) 標度用轉換資料按昇冪排序時，根據 SM750 的狀態其探索方法有所不同，因此處理速度也不相同。
處理速度取決於探索次數，探索次數越少其處理速度越快。

(a) 逐次探索的處理速度較快的情況

(S1)的最大座標點數爲 1 ～ 15 之間時，由於逐次探索次數≤ 15，因此逐次探索的處理速度將較快。

(b) 二分探索的處理速度較快的情況

由於最大探索次數爲 15 次，在(S1)的座標點爲 16 點以上時，二分探索次數≤逐次探索次數，因此二分探索的處理速度較快。



出 錯

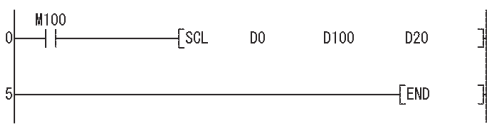
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	位於標度用轉換資料的(S1)前面的點的 X 座標資料未按昇冪設置時。(但是，SM750 爲 ON 時，不進行此出錯檢查。) (S1)中指定的輸入值超出了所設置的標度用轉換資料的範圍時。 從(S2)的軟元件開始的座標點數超出了 1 ～ 32767 的範圍時。	---	---	---	---	○	○
4101	從(S2)的軟元件開始的座標點數超出了指定軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 M100 變爲 ON 時，通過 D0 中指定的輸入值對 D100 後面設置的標度用轉換資料進行標度後，輸出到 D20 中的程式。

[梯形圖模式]



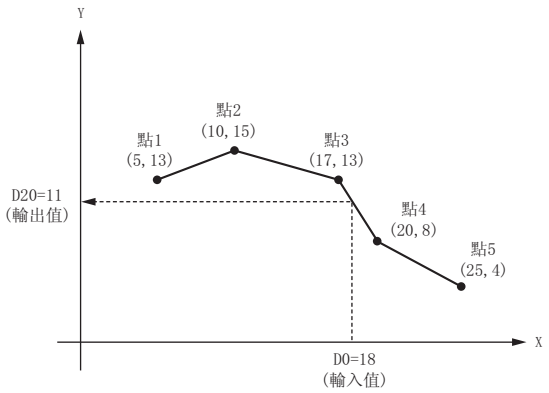
[列表模式]

步	指令	軟元件		
0	LD	M100		
1	SCL	D0	D100	D20
5	END			

[動作]

標度用轉換資料的構成

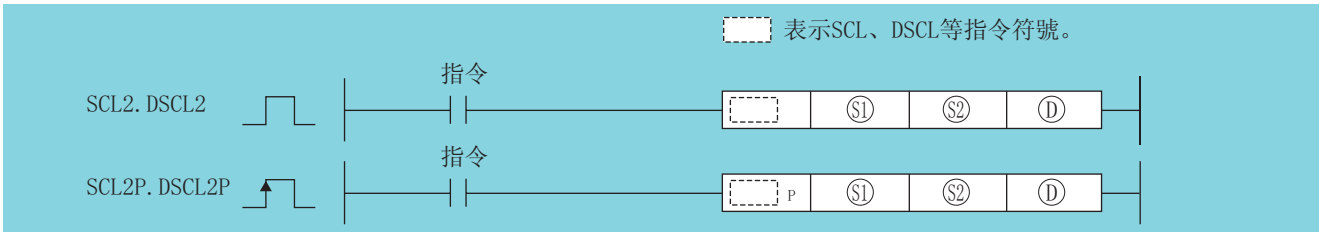
設置專案		軟元件	設置內容
座標點數		D100	K5
點1	X座標	D101	K5
	Y座標	D102	K13
點2	X座標	D103	K10
	Y座標	D104	K15
點3	X座標	D105	K17
	Y座標	D106	K13
點4	X座標	D107	K20
	Y座標	D108	K8
點5	X座標	D109	K25
	Y座標	D110	K22



7.13.5 SCL2、SCL2P、DSCL2、DSCL2P

Basic High performance Process Redundant Ver. Universal LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。



Ⓢ1 ： 進行標度的輸入值或者存儲輸入值的軟元件的起始編號 (BIN16/32 位)。

Ⓢ2 ： 存儲標度用轉換資料的軟元件的起始編號 (BIN16/32 位)。

Ⓓ ： 存儲通過標度進行控制的輸出值的軟元件的起始編號 (BIN16/BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	---	○	○	○				○	---
Ⓢ2	---	○	○	---				---	---
Ⓓ	---	○	○	○				---	---

功 能

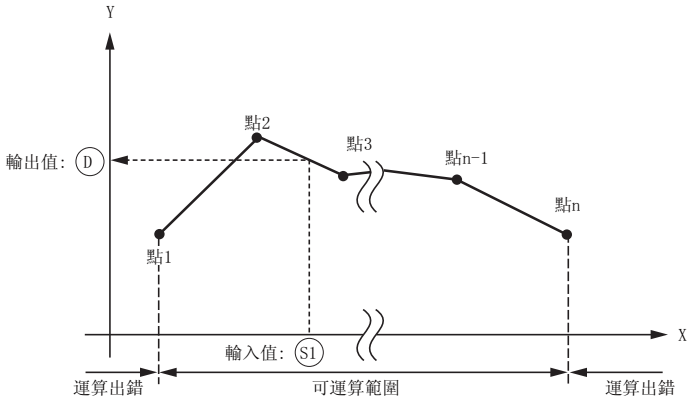
SCL2(P)

(1) 對Ⓢ2中指定的標度用轉換資料 (16 位元資料單位)，通過Ⓢ1中指定的輸入值進行標度，並將運算結果存儲到Ⓓ中指定編號的軟元件中。

標度轉換是基於Ⓢ2中指定的軟元件後面存儲的標度用轉換資料進行的。

[標度用轉換資料的構成]n 表示Ⓢ2中指定的座標點數。

設置專案		軟元件分配
座標點數		Ⓢ2
X 座標	點 1	Ⓢ2+1
	點 2	Ⓢ2+2
	⋮	⋮
	點 n	Ⓢ2+n
Y 座標	點 1	Ⓢ2+n+1
	點 2	Ⓢ2+n+2
	⋮	⋮
	點 n	Ⓢ2+2n



(2) 算結果不是整數值時，對小數點以下第 1 位進行四捨五入。

(3) 標度用轉換資料的 X 座標資料應按昇冪進行設置。

(4) Ⓢ1應在標度用轉換資料範圍內 (Ⓢ2的軟元件值) 進行設置。

(5) 將多個點指定爲相同的 X 座標時，將對最大點號的點的 Y 座標值進行輸出。

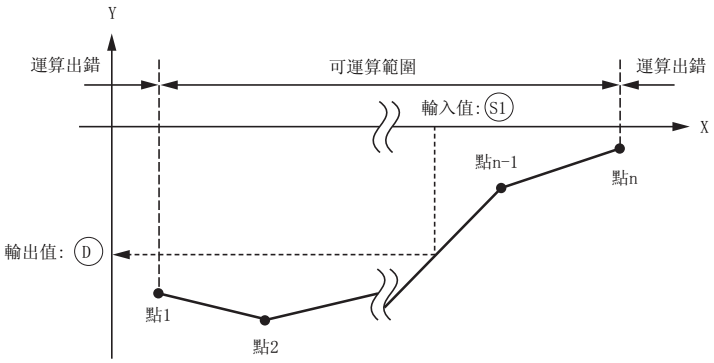
DSCL2(P)

(1) 對Ⓢ2中指定的標度用轉換資料 (32 位元資料單位)，通過Ⓢ1中指定的輸入值進行標度，並將運算結果存儲到Ⓓ中指定編號的軟元件中。

標度轉換是基於Ⓢ2中指定的軟元件後面存儲的標度用轉換資料進行的。

[標度用轉換資料的構成]n 表示Ⓢ2中指定的座標點數。

設置專案		軟元件分配
座標點數		Ⓢ2+1、Ⓢ2
X 座標	點 1	Ⓢ2+3、Ⓢ2+2
	點 2	Ⓢ2+5、Ⓢ2+4
	⋮	⋮
	點 n	Ⓢ2+2n+1、Ⓢ2+2n
Y 座標	點 1	Ⓢ2+2n+3、Ⓢ2+2n+2
	點 2	Ⓢ2+2n+5、Ⓢ2+2n+4
	⋮	⋮
	點 n	Ⓢ2+4n+1、Ⓢ2+4n



(2) 運算結果不是整數值時，對小數點以下第 1 位進行四捨五入。

(3) 標度用轉換資料的 X 座標資料應按昇冪進行設置。

(4) Ⓢ1應在標度用轉換資料範圍內 (Ⓢ2、Ⓢ2+1 的軟元件值) 進行設置。

(5) 將多個點指定爲相同的 X 座標時，將對最大點號的點的 Y 座標值進行輸出。

(6) 標度用轉換資料的座標點數範圍爲 1 ～ 32767。

要 點

標度用轉換資料按昇冪排序時，根據 SM750 的狀態其搜索方法有所不同，因此處理速度也不相同。
詳細內容請參閱 543 頁 7.13.4 項。

出　錯

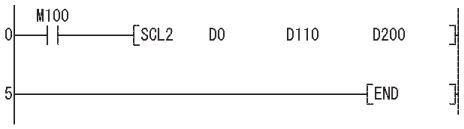
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	X 座標資料未按昇冪設置時。 (S1)中指定的輸入值超出了所設置的標度用轉換資料的範圍時。 (S2)中指定的座標點數超出了 1 ～ 32767 的範圍時。	---	---	---	---	○	○
4101	從(S2)的軟元件開始的座標點數超出了指定軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1)　以下爲 M0 變爲 ON 時，將 D0 中指定的輸入值通過 D110 後面設置的標度用轉換資料進行標度後，輸出到 D200 中的程式。

[梯形圖模式]



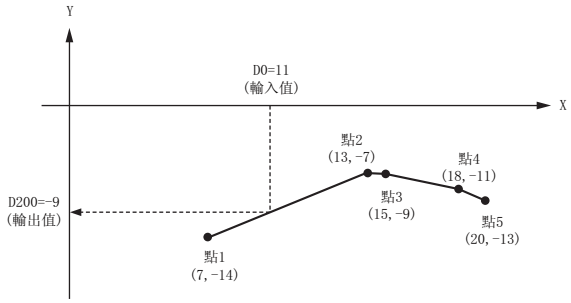
[列表模式]

步	指令	軟元件		
0	LD	M100		
1	SCL2	D0	D110	D200
5	END			

[動作]

標度用轉換資料的構成

設置專案		軟元件	設置內容
座標點數		D110	K5
X座標	點1	D111	K7
	點2	D112	K13
	點3	D113	K15
	點4	D114	K18
	點5	D115	K20
Y座標	點1	D116	K-14
	點2	D117	K-7
	點3	D118	K-15
	點4	D119	K-11
	點5	D120	K-18



7.14 文件寄存器切換指令

7.14.1 RSET、RSETP

Basic | High performance | Process | Redundant | Ver. Universal | LCPU

· 在 Q00UJCPU 以外的通用型 QCPU 中可以使用。

Ⓢ ：進行切換的塊號數據或者存儲塊號數據的軟元件的起始編號 (BIN16 位)。

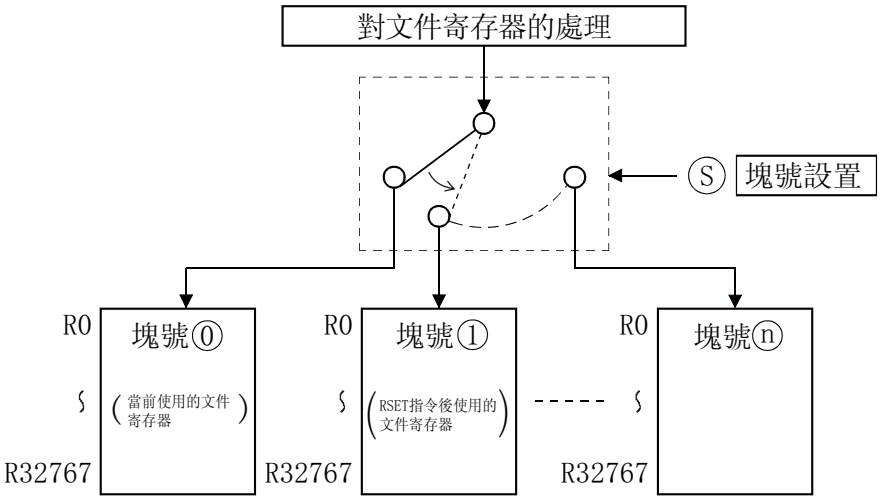
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○								---

功　能

將程式中使用的文件寄存器的塊號變更爲Ⓢ中指定的軟元件中存儲的塊號。

進行塊號變更後，順控程式中使用的所有文件寄存器均按變更後的塊號的文件寄存器進行處理。

例 從塊號 0 切換爲塊號 1 時



要 點

將文件寄存器 (R) 指定爲刷新軟元件，並通過 RSET 指令切換文件寄存器 (R) 的塊號時，應加以注意。
關於文件寄存器的限制事項，請參閱 115 頁 3.10 節。

出　錯

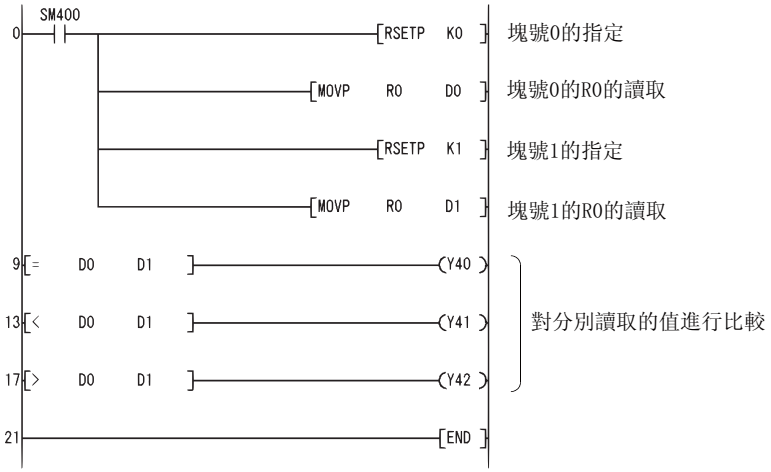
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ 中指定的塊號不存在時。	---	---	---	---	○	○
4101	文件寄存器不存在時。	---	---	---	---	○	○

程式示例

(1) 以下爲將塊號 0 與塊號 1 的 R0 進行比較的程式。

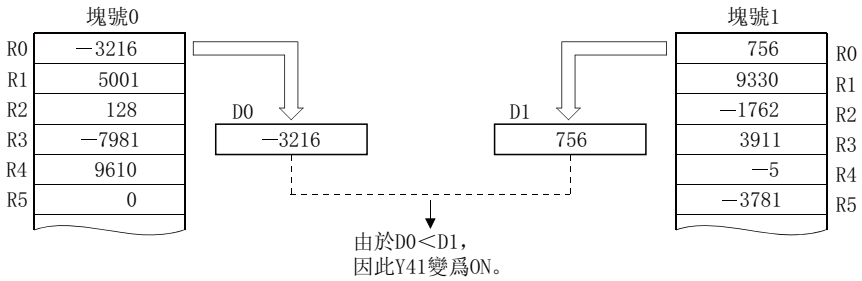
[梯形圖模式]



[列表模式]

步	指令	軟元件	
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		

[動作]



7.14.2 QDRSET、QDRSETP

Basic | High performance | Process | Redundant | Universal (Ver.) | LCPU

・在 Q00UJCPU 以外的通用型 QCPU 中可以使用。

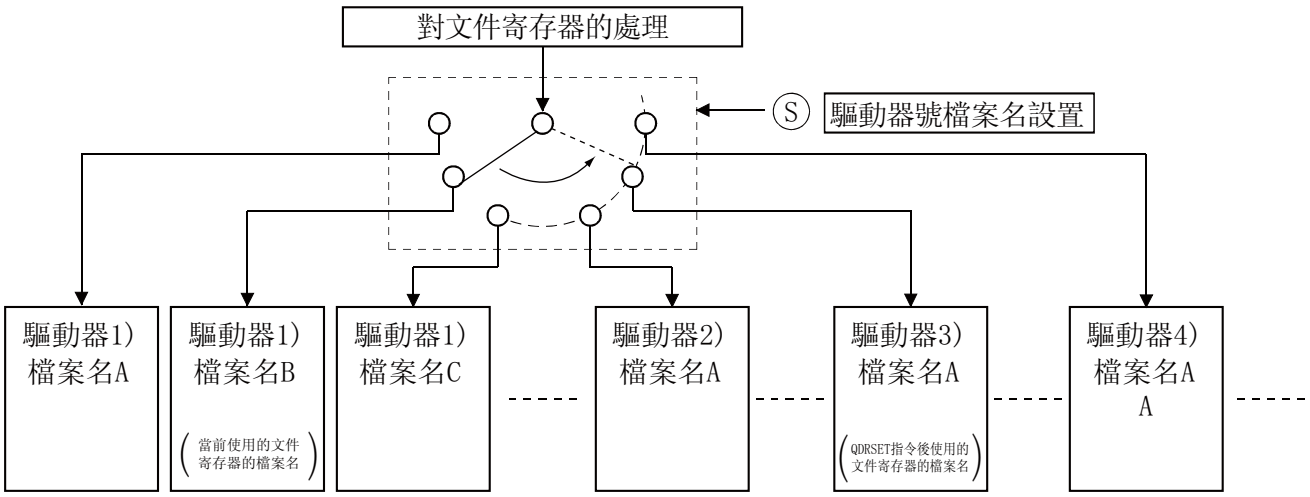
Ⓢ ：進行設置的文件寄存器的驅動器號及檔案名的字串資料或者存儲字串資料的軟元件的起始編號（字串）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---

功　能

(1) 將程式中使用的文件寄存器的檔案名變更爲Ⓢ中指定的軟元件中存儲的檔案名。
進行了檔案名變更後，順控程式中使用的所有文件寄存器均按變更後檔案名的文件寄存器進行處理。
變更後的文件寄存器的塊號將變爲 0。
塊號的切換是通過 RSET 指令進行。

例 從驅動器號 1 的檔案名 B 切換爲驅動器號 3 的檔案名 A 時



(2) 驅動器號可在 1 ～ 4 的範圍內指定。
(驅動器號不能指定爲驅動器 0(程式記憶體 / 內置記憶體)。)
但是，根據 CPU 模組的不同可指定的驅動器也有所不同。
請通過所使用的 CPU 模組的手冊對可指定的驅動器進行確認。

(3) 檔案名中無需指定副檔名 (.QDR)。

(4) 通過在檔案名中指定 NULL 字元 (00_H)，可以對檔案名的設置進行解除。

(5) 即使在參數中對驅動器號、檔案名進行了指定，本指令中指定的檔案名也將優先。

要 點

1. 即使通過 QDRSET 指令對檔案名進行了變更，如果進行了 CPU 模組的 STOP → RUN 操作，將恢復爲參數中設置的檔案名。
如果希望在進行了 CPU 模組的 STOP → RUN 操作時，仍保持爲通過 QDRSET 指令變更後的檔案名，應使用 STOP → RUN 時 1 個掃描 ON 的特殊繼電器後，執行 QDRSET 指令。
2. 將文件寄存器指定爲刷新軟元件時，不要通過 QDRSET 指令對文件寄存器的檔案名進行變更。
關於文件寄存器的限制事項，請參閱 115 頁 3.10 節。

出　錯

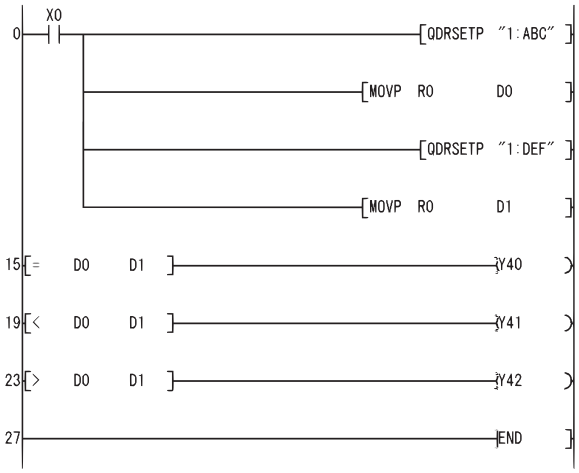
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	Ⓢ 中指定的驅動器號的檔案名不存在時。	---	○	○	○	○	---

程式示例

(1) 以下爲將驅動器號 1 的 ABC 與驅動器號 1 的 DEF 的 R0 進行比較的程式。

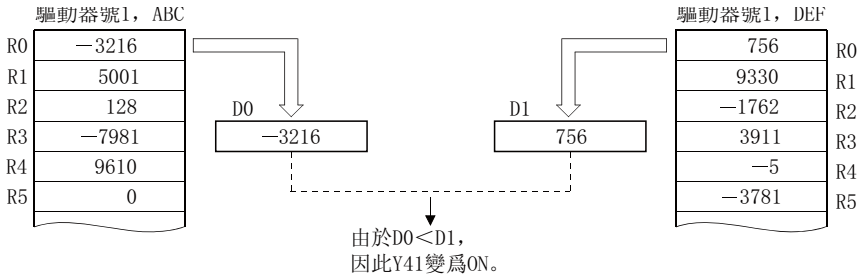
[梯形圖模式]



[列表模式]

步	指令	軟元件	
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		

[動作]



7.14.3 QCDSET、QCDSETP

Basic | High performance | Process | Redundant | Universal | LCPU

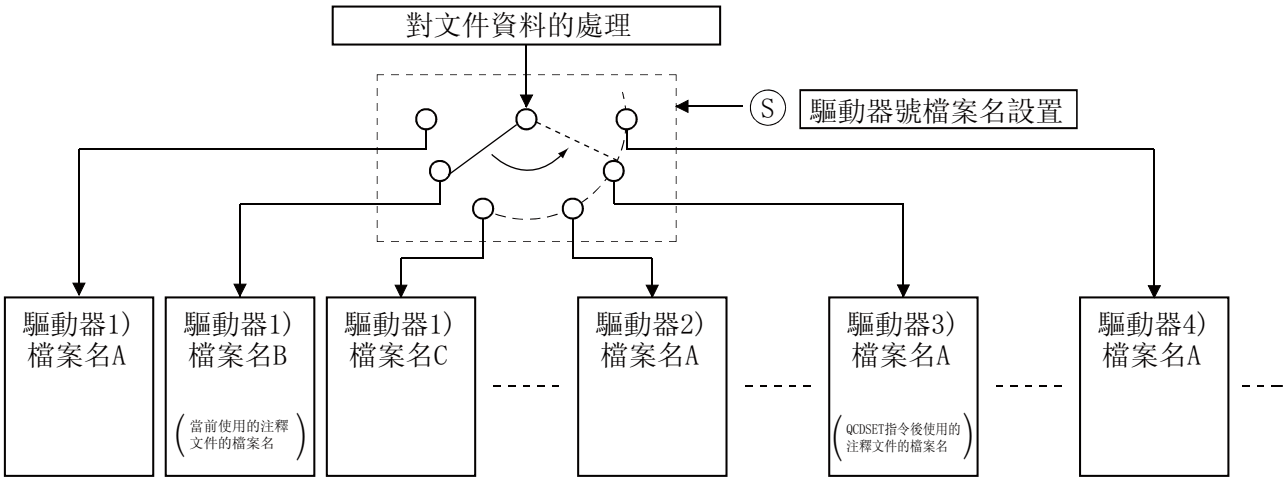
Ⓢ ： 進行設置的注釋文件的驅動器號及檔案名的字串資料或者存儲字串資料的軟元件的起始編號 (字串)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---

功　能

(1) 將程式中使用的注釋文件的檔案名變更爲Ⓢ中指定的軟元件中存儲的檔案名。
進行了檔案名變更後，順控程式中使用的所有注釋資料均按變更後檔案名的注釋資料進行處理。

例 從驅動器號 1 的文件 B 切換爲驅動器號 4 的文件 B 時



(2) 驅動器號可在 1 ～ 4 的範圍內指定。
(驅動器號不能指定爲驅動器 0(程式記憶體 / 內置記憶體)。)
但是，根據 CPU 模組的不同可指定的驅動器也有所不同。
請通過所使用的 CPU 模組的手冊對可指定的驅動器進行確認。

(3) 檔案名中無需指定副檔名 (.QCD)。

(4) 通過在檔案名中指定 NULL 字元 (00_H)，可以對檔案名的設置進行解除。

(5) 即使在參數中對驅動器號、檔案名進行了指定，本指令中指定的檔案名也將優先。

要 點

即使通過 QCDSET 指令對檔案名進行了變更，如果進行了 CPU 模組的 STOP → RUN 操作，將恢復爲參數中設置的檔案名。
如果希望在進行了 CPU 模組的 STOP → RUN 操作時，仍保持爲通過 QCDSET 指令變更後的檔案名，應使用 STOP → RUN 時 1 個掃描 ON 的特殊繼電器 SM402 後，執行 QCDSET 指令。

出　錯

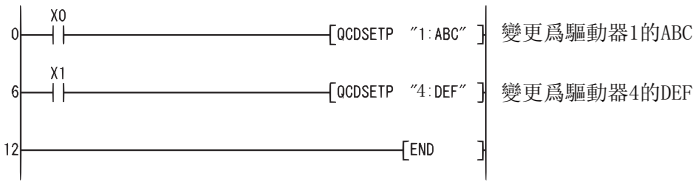
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	Ⓢ中指定的驅動器號的檔案名不存在時。	---	○	○	○	○	---

程式示例

(1) 以下爲 X0 變爲 ON 時，將注釋物件文件切換爲驅動器 1 的 ABC.QCD，X1 變爲 ON 時，將注釋物件文件切換爲驅動器 4 的 DEF.QCD 的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0
1	QCDSETP	"1:ABC"
6	LD	X1
7	QCDSETP	"4:DEF"
12	END	

注意事項

(1) 在通用型 QCPU、LCPU 中，SM721(文件訪問中) 爲 ON 狀態時，即使本指令的執行指令變爲 ON，也不能執行本指令。應將程式設置爲在 SM721 處於 OFF 狀態時執行本指令。

(2) 在 LCPU 中，驅動器號中指定了驅動器 2(SD 存儲卡) 的情況下，SM606(SD 存儲卡強制使用停止指示) 爲 ON 時，不執行本指令。執行的情況下將變爲無處理。

7.15 時鐘指令

7.15.1 DATERD、DATERDP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：存儲讀取的時鐘資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	---	○		---					

功　能

(1) 根據 CPU 模組的時鐘因數讀取 “ 年、月、日、時、分、秒、星期 ” 後，以 BIN 值存儲到Ⓓ中指定的軟元件的後面。

時鐘因數 ⇨

Ⓓ	年(西曆)	(1980～2079)
Ⓓ+1	月	(1～12)
Ⓓ+2	日	(1～31)
Ⓓ+3	時(24小時時鐘)	(0～23)
Ⓓ+4	分	(0～59)
Ⓓ+	秒	(0～59)
Ⓓ+6	星期	(0～6)

(2) Ⓓ的 “ 年 ” 以 西曆 4 位元數存儲。

(3) Ⓓ+6 的 “ 星期 ” 中以 “ 0 ～ 6 ” 存儲 “ 星期日～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
存儲資料	0	1	2	3	4	5	6

(4) 閏年時將進行自動修正。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

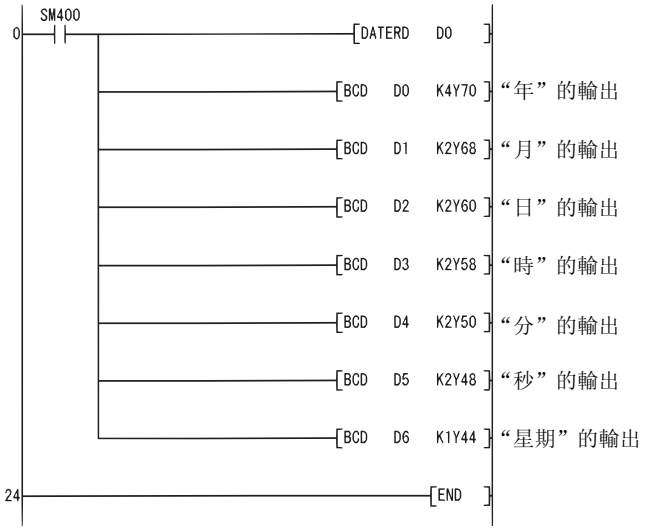
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將時鐘資料以 BCD 值進行輸出的程式。

年 ..... Y70 ～ Y7F
月 ..... Y68 ～ Y6F
日 ..... Y60 ～ Y67
時 ..... Y58 ～ Y5F
分 ..... Y50 ～ Y57
秒 ..... Y48 ～ Y4F
星期 ..... Y44 ～ Y47

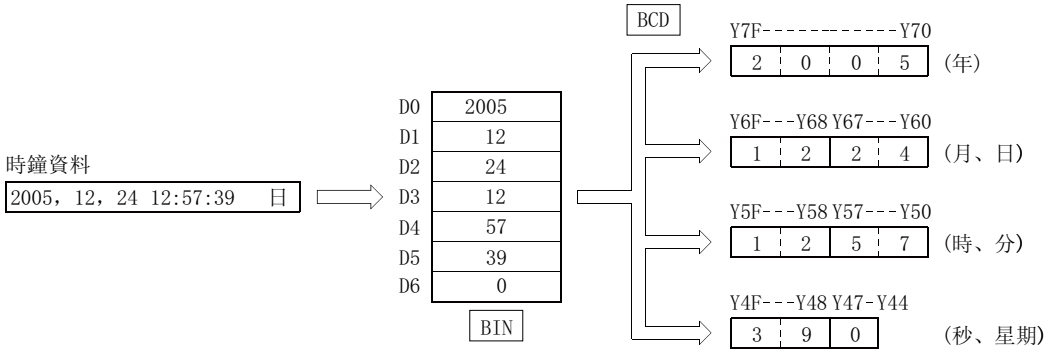
[梯形圖模式]



[列表模式]

步	指令	軟元件	
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		

[動作]



7.15.2 DATEWR、DATEWRP

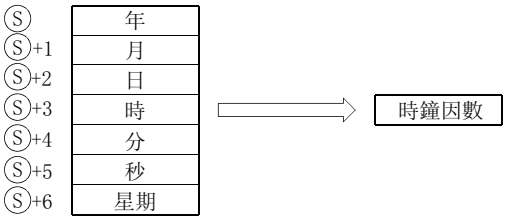
Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ： 存儲寫入到時鐘因數中的時鐘資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ	---	○		---					

功　能

(1) 將Ⓢ中指定的軟元件編號後面存儲的時鐘資料，寫入到 CPU 模組的時鐘因數中。



(2) 各項目的設置是以 BIN 值進行的。

(3) Ⓢ的“ 年 ”是以西曆 4 位數在 1980 ～ 2079 的範圍內進行設置。

(4) Ⓢ+1 的“ 月 ”是在 1 ～ 12(1 月～ 12 月) 的範圍內進行設置。

(5) Ⓢ+2 的“ 日 ”是在 1 ～ 31(1 日～ 31 日) 的範圍內進行設置。

(6) Ⓢ+3 的“ 時 ”是在 0 ～ 23(0 時～ 23 時) 的範圍內進行設置。
(設置爲 24 小時時鐘。)

(7) Ⓢ+4 的“ 分 ”是在 0 ～ 59(0 分～ 59 分) 的範圍內進行設置。

(8) Ⓢ+5 的“ 秒 ”是在 0 ～ 59(0 秒～ 59 秒) 的範圍內進行設置。

(9) Ⓢ+6 的“ 星期 ”是以“ 0 ～ 6 ”對“ 星期日～星期六 ”進行設置。

星期	星期日	星期一	星期二	星期三	星期四	星期五	星期六
存儲資料	0	1	2	3	4	5	6

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

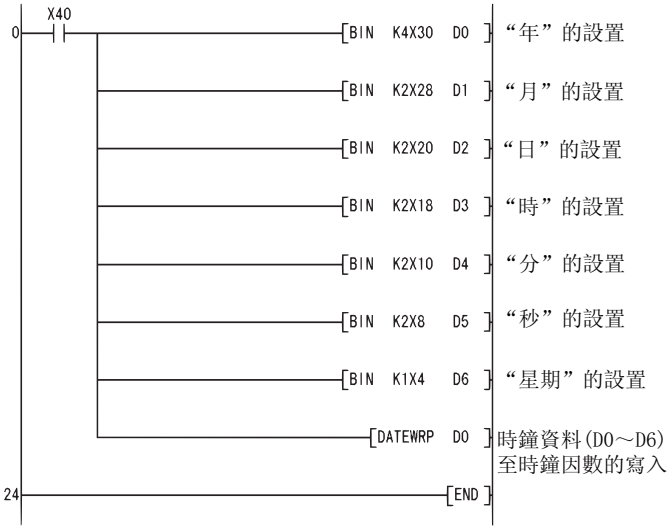
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	各專案中設置了超出設置範圍的資料時。	---	---	---	---	○	○
4101	Ⓢ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X40 變爲 ON 時，將以 BCD 值輸入的下述時鐘資料寫入到時鐘因數中的程式。

年　..... X30 ～ X3F　　時　..... X18 ～ X1F
月　..... X28 ～ X2F　　分　..... X10 ～ X17
日　..... X20 ～ X27　　秒　..... X8 ～ XF
星期 ..... X4 ～ X7

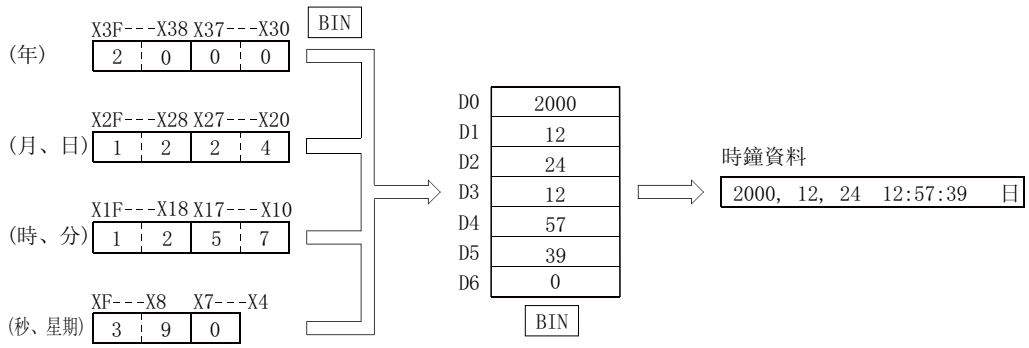
[梯形圖模式]



[列表模式]

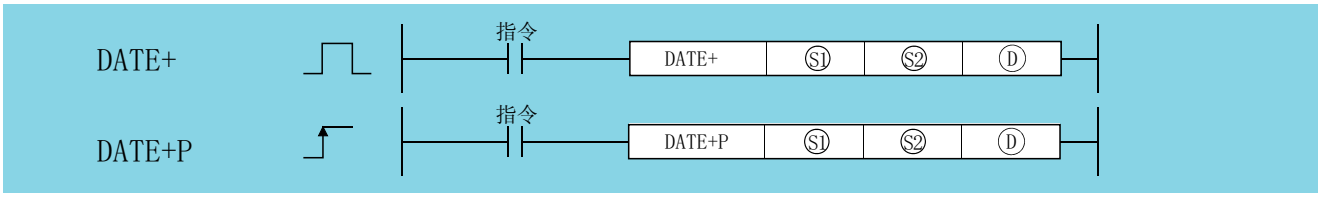
步	指令	軟元件	
0	LD	X40	
1	BIN	K4X30	D0
4	BIN	K2X28	D1
7	BIN	K2X20	D2
10	BIN	K2X18	D3
13	BIN	K2X10	D4
16	BIN	K2X8	D5
19	BIN	K1X4	D6
22	DATEWRP	D0	
24	END		

[動作]



7.15.3 DATE+、DATE+P

Basic | High performance | Process | Redundant | Universal | LCPU



ⓢ1 ： 存儲被進行加法運算的時間資料的軟元件的起始編號 (BIN16 位)。

ⓢ2 ： 存儲進行加法運算的時間資料的軟元件的起始編號 (BIN16 位)。

Ⓓ ： 存儲加法運算結果時間資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
ⓢ1	---	○		---					
ⓢ2	---	○		---					
Ⓓ	---	○		---					

功　能

(1) 將Ⓢ1中指定的時間資料與Ⓢ2中指定的時間資料進行加法運算，並將加法運算結果存儲到Ⓓ中指定的軟元件編號的後面。

		資料範圍			資料範圍			資料範圍
Ⓢ1	時	(0～23)	Ⓢ2	時	(0～23)	Ⓓ	時	(0～23)
Ⓢ1+1	分	(0～59) +	Ⓢ2+1	分	(0～59) ⇨	Ⓓ+1	分	(0～59)
Ⓢ1+2	秒	(0～59)	Ⓢ2+2	秒	(0～59)	Ⓓ+2	秒	(0～59)

例如，將 6 時 32 分 40 秒與 7 時 48 分 10 秒進行加法運算時的情況如下所示。

Ⓢ1	6時		Ⓢ2	7時		Ⓓ	14時
Ⓢ1+1	32分	+	Ⓢ2+1	48分	⇨	Ⓓ+1	20分
Ⓢ1+2	40秒		Ⓢ2+2	10秒		Ⓓ+2	50秒

(2) 運算結果的時間超過了 24 小時時，將該值減去 24 小時後的值作爲運算結果。

例如，將 14 時 20 分 30 秒與 20 時 20 分 20 秒進行加法運算時，其結果不是 34 時 40 分 50 秒，而是 10 時 40 分 50 秒。

Ⓢ1	14時		Ⓢ2	20時		Ⓓ	10時
Ⓢ1+1	20分	+	Ⓢ2+1	20分	⇨	Ⓓ+1	40分
Ⓢ1+2	30秒		Ⓢ2+2	20秒		Ⓓ+2	50秒

備 註

關於可設置爲時、分、秒的資料，請參閱 556 頁 7.15.2 項。

出　錯

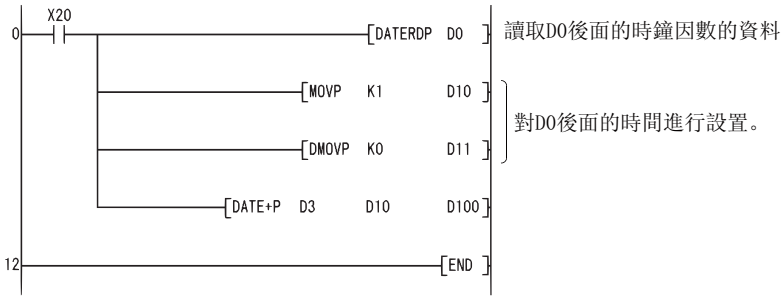
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ1、Ⓢ2的資料超出了範圍時。	---	---	---	---	○	○
4101	Ⓢ1、Ⓢ2、Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，對從時鐘因數中讀取的時間資料進行 1 小時的加法運算後，將運算結果存儲到 D100 後面的程式。

[梯形圖模式]

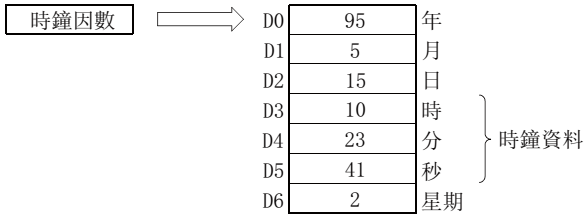


[列表模式]

步	指令	軟元件		
0	LD	X20		
1	DATERDP	D0		
3	MOVP	K1	D10	
5	DMOVP	K0	D11	
8	DATE+P	D3	D10	D100
12	END			

[動作]

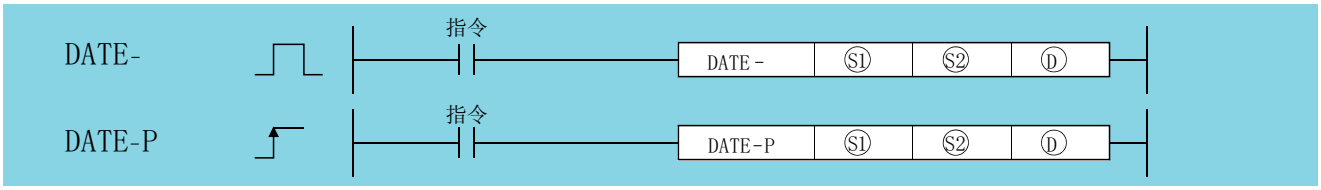
· 通過 DATERDP 指令讀取時間資料



· 通過 DATE+P 指令進行加法運算

7.15.4 DATE-、DATE-P

Basic | High performance | Process | Redundant | Universal | LCPU



ⓢ1 ： 存儲被進行減法運算的時間資料的軟元件的起始編號 (BIN16 位)。

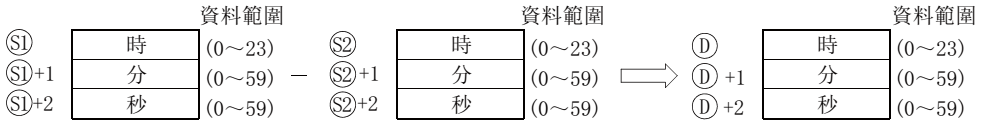
ⓢ2 ： 存儲進行減法運算的時間資料的軟元件的起始編號 (BIN16 位)。

Ⓓ ： 存儲減法運算結果時間資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
ⓢ1	---	○		---					
ⓢ2	---	○		---					
Ⓓ	---	○		---					

功 能

(1) 將ⓢ1中指定的時間資料與ⓢ2中指定的時間資料進行減法運算，並將減法運算結果存儲到Ⓓ中指定的軟元件編號的後面。



例如，將 10 時 40 分 20 秒與 3 時 50 分 10 秒進行減法運算時的情況如下所示。

(2) 運算結果的時間爲負數時，將該值加上 24 小時後的值作爲運算結果。

例如，將 4 時 50 分 32 秒與 10 時 42 分 12 秒進行減法運算時，其結果不是 -6 時 8 分 20 秒，而是 18 時 8 分 20 秒。

備 註

關於時、分、秒的可設置資料，請參閱 556 頁 7.15.2 項。

出　錯

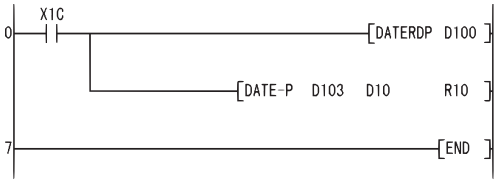
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	ⓢ1、ⓢ2的資料超出了範圍時。	---	---	---	---	○	○
4101	ⓢ1、ⓢ2、ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X1C 變爲 ON 時，將從時鐘因數中讀取的時間資料與 D10 後面存儲的時間資料進行減法運算後，將運算結果存儲到 R10 後面的程式。

[梯形圖模式]

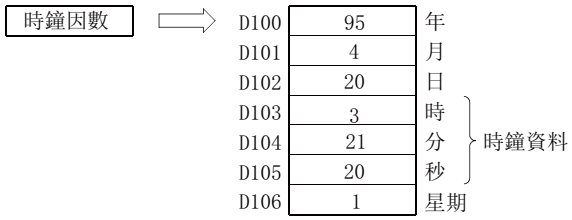


[列表模式]

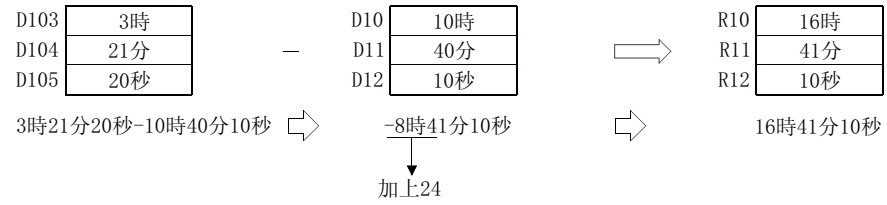
步	指令	軟元件		
0	LD	X1C		
1	DATERDP	D100		
3	DATE-P	D103	D10	R10
7	END			

[動作]

・通過 DATERDP 指令讀取時間資料

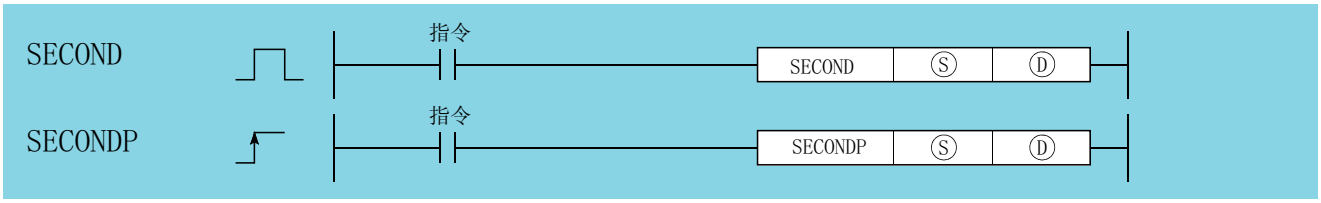


・通過 DATE-P 指令進行減法運算 (D10 ～ D12 中指定了 10 時 40 分 10 秒時)



7.15.5 SECOND、SECONDP

Basic　High performance　Process　Redundant　Universal　LCPU



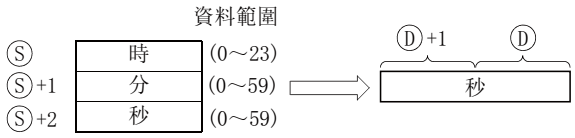
Ⓢ ：存儲轉換前的時鐘資料的軟元件的起始編號 (BIN16 位)。

Ⓓ ：存儲轉換後的時鐘資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ	---	○		---				---	
Ⓓ	○	○		○				---	

功　能

(1) 將Ⓢ中指定的軟元件編號後面存儲的時間資料換算爲秒後，將換算結果存儲到Ⓓ中指定的軟元件中。



例如，指定爲 4 時 29 分 31 秒時的情況如下所示。

出　錯

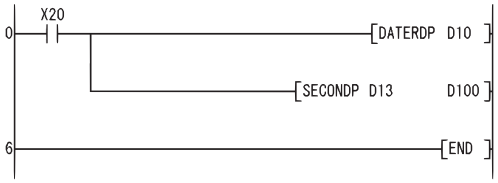
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ的資料超出了範圍時。	---	---	---	---	○	○
4101	Ⓢ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將從時鐘因數中讀取的時間資料換算爲秒後，存儲到 D100、D101 中的程式。

[梯形圖模式]

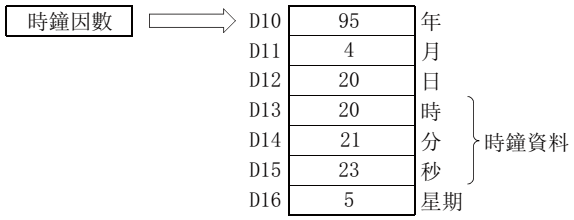


[列表模式]

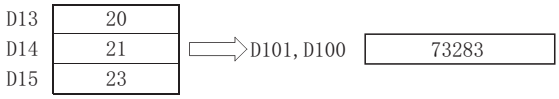
步	指令	軟元件	
0	LD	X20	
1	DATERDP	D10	
3	SECONDP	D13	D100
6	END		

[動作]

· 通過 DATERDP 指令讀取時間資料

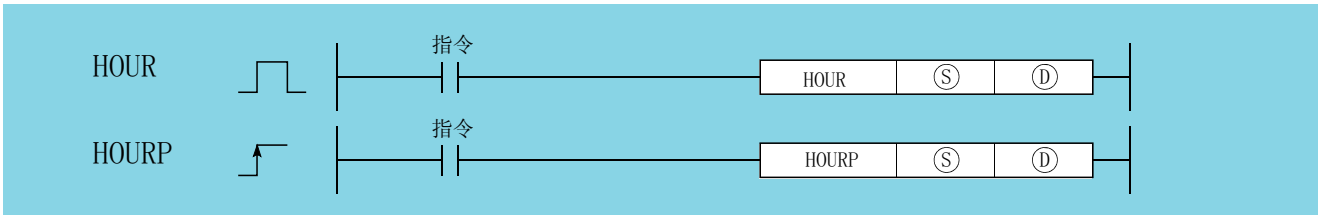


· 通過 SECONDP 指令換算爲秒



7.15.6 HOUR、HOURP

Basic | High performance | Process | Redundant | Universal | LCPU



Ⓢ ：存儲轉換前的時鐘資料的軟元件的起始編號 (BIN16 位)。

Ⓓ ：存儲轉換後的時鐘資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○	○		○				○	---
Ⓓ	---	○		---				---	---

功　能

(1) 將Ⓢ中指定的軟元件編號後面存儲的秒資料換算爲時、分、秒後，將換算結果存儲到Ⓓ中指定的軟元件後面。

例如，指定了 45325 秒時的情況如下所示。

出　錯

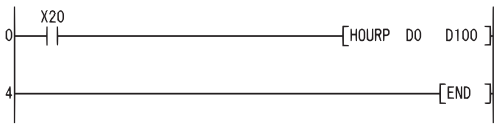
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ 的資料超出了範圍時。	---	---	---	---	○	○
4101	Ⓢ 中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X20 變爲 ON 時，將 D0、D1 中存儲的秒資料換算爲時、分、秒後，存儲到 D100 後面的程式。

[梯形圖模式]

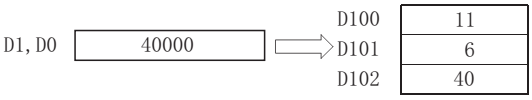


[列表模式]

步	指令	軟元件	
0	LD	X20	
1	HOURP	D0	D100
4	END		

[動作]

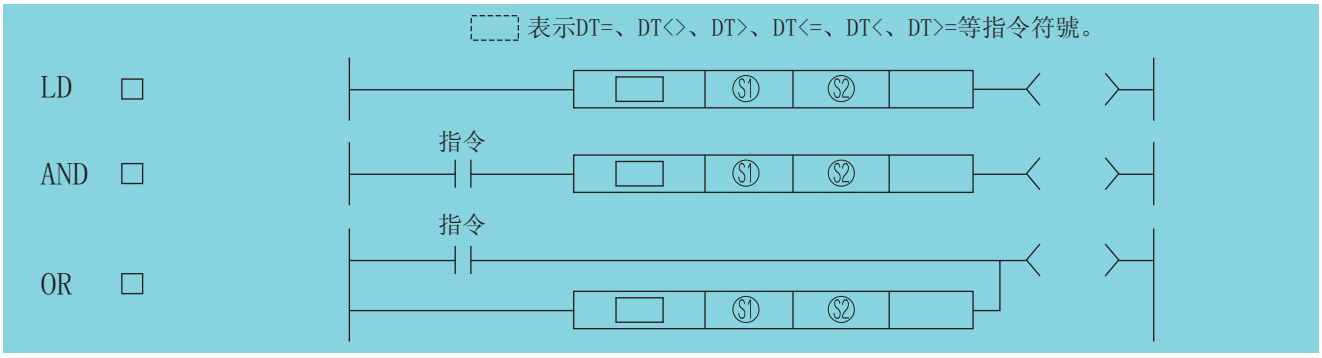
· 通過 HOURP 指令進行至時、分、秒的轉換 (D0、D1 中指定了 40000 秒時)



7.15.7 DT=、DT<>、DT>、DT<=、DT<、DT>=

Basic × | High performance × | Process × | Redundant × | Ver. Universal | LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。



Ⓢ1 ： 存儲進行比較的資料的軟元件的起始編號 (BIN16 位)。

Ⓢ2 ： 存儲進行比較的資料的軟元件的起始編號 (BIN16 位)。

n ： 表示比較物件的值或者存儲比較對照的資料數 (BIN16 位元)。

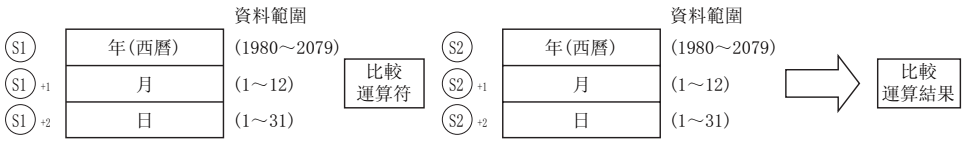
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	---	○		---				---	---
Ⓢ2	---	○		---				---	---
n	---	○		○				○	---

功　能

(1) 對Ⓢ1、Ⓢ2中指定的日期資料進行比較。或者，將Ⓢ1中指定的日期資料與當前日期進行比較。通過 n 可以選擇比較對象。

(a) 與任意日期資料的比較

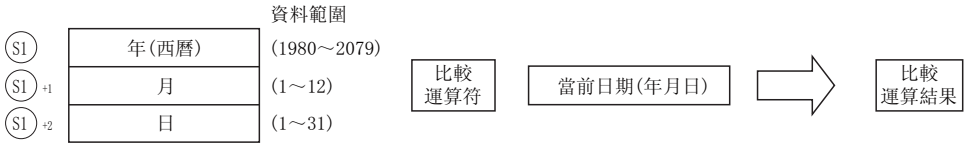
· 將Ⓢ1中指定的日期資料與Ⓢ2中指定的日期資料按照 n 的條件通過 a 觸點進行比較。



(b) 與當前日期資料的比較

· 將Ⓢ1中指定的日期資料與當前的日期資料按照 n 的條件通過 a 觸點進行比較。

· Ⓢ2中指定的時間資料被作爲虛擬資料處理，將被忽略。



要 點

與任意的日期資料比較時，或者與當前的日期資料比較時，Ⓢ1、Ⓢ2中的某一個處於以下狀況時將變爲運算出錯狀態 (出錯代碼： 4101) 或者可能導致誤動作。

· 變址修飾目標的指定範圍超出了軟元件範圍時。

· 在未設置文件寄存器的狀態下指定了文件寄存器時。

(2) 各專案設置是以 BIN 值格式進行設置的。

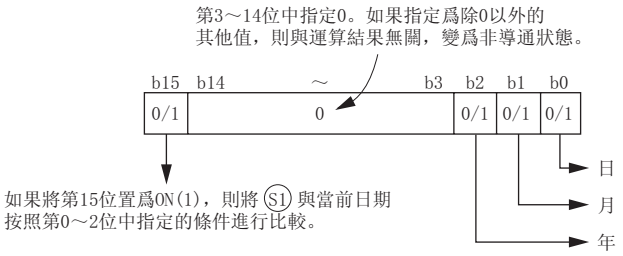
(3) Ⓢ1、Ⓢ2的 “ 年 ” 是以西曆 4 位數在 1980 ～ 2079 的範圍內進行設置。

(4) Ⓢ1+1、Ⓢ2+1 的 “ 月 ” 是在 1 ～ 12(1 月～ 12 月) 的範圍內進行設置。

(5) Ⓢ1+2、Ⓢ2+2 的 “ 日 ” 是在 1 ～ 31(1 日～ 31 日) 的範圍內進行設置。

(6) 通過在 n 中指定下圖的值，可以對比較物件進行詳細指定。

n 的位構成如下所示。



(a) 比較物件日期（第 0 ～ 2 位）

· 0：不對比較物件的日期資料（年 / 月 / 日）進行比較。

· 1：對比較物件的日期資料（年 / 月 / 日）進行比較。

(b) 比較運算物件（第 15 位元）

· 0：將(S1)中指定的日期資料與(S2)中指定的日期資料進行比較。

· 1：將(S1)中指定的日期資料與當前的日期資料進行比較。

(S2)中指定的日期資料將被忽略。

(c) 比較物件位元的處理內容如下所示。

與任意的日期資料進行比較時的 n 值	與當前的日期資料進行比較時的 n 值	比較物件日期	處理內容
0001_H	8001_H	日	僅對日 ((S1)+2) 進行比較。
0002_H	8002_H	月	僅對月 ((S1)+1) 進行比較。
0003_H	8003_H	月、日	對月 ((S1)+1)、日 ((S1)+2) 進行比較。
0004_H	8004_H	年	僅對年 ((S1)) 進行比較。
0005_H	8005_H	年、日	對年 ((S1))、日 ((S1)+2) 進行比較。
0006_H	8006_H	年、月	對年 ((S1))、月 ((S1)+1) 進行比較。
0007_H	8007_H	年、月、日	對年 ((S1))、月 ((S1)+1)、日 ((S1)+2) 進行比較。
除 0001_H ～ 0007_H、8001_H ～ 8007_H 以外		无	對年 ((S1))、月 ((S1)+1)、日 ((S1)+2) 均不進行比較。（變爲非導通狀態。）

(7) 在比較物件軟元件中存儲的資料不能被識別爲日期資料的情況下，執行指令後將 SM709 置爲 ON，變爲非導通狀態。即使是在不能被識別爲日期資料的情況下，只要是在設置範圍內，SM709 將不變爲 ON。

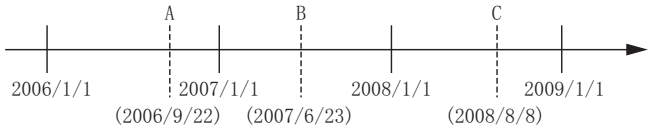
(S1)～(S1)+2 或者(S2)～(S2)+2 超出了指定軟元件範圍時，SM709 也將變爲 ON，變爲非導通狀態。

一旦 SM709 變爲 ON 後，在進行重定 / 電源 OFF 之前將保持 ON 狀態不變，因此應根據需要將其置爲 OFF。

(8) 各指令的比較運算結果如下所示。

內的指令符號	條件	比較運算結果	內的指令符號	條件	比較運算結果
DT=	(S1)=(S2)	導通狀態	DT=	(S1)≠(S2)	非導通狀態
DT<>	(S1)≠(S2)		DT<>	(S1)=(S2)	
DT>	(S1)>(S2)		DT>	(S1)≤(S2)	
DT<=	(S1)≤(S2)		DT<=	(S1)>(S2)	
DT<	(S1)<(S2)		DT<	(S1)≥(S2)	
DT>=	(S1)≥(S2)		DT>=	(S1)<(S2)	

(a) 日期的比較示例如下所示。



上述日期 A、B、C 的比較運算結果如下所示。

即使在相同的條件下進行比較，根據選擇的比較物件其比較運算結果將有所不同。

比較物件	比較條件		
	A<B	B<C	A<C
日	○	×	×
月	×	○	×
月、日	×	○	×
年	○	○	○
年、日	○	○	○
年、月	○	○	○
年、月、日	○	○	○
無	×	×	×

○：導通　×：非導通

(b) 即使在所比較的日期是實際上並不存在的日期的情況下，只要是在允許設置範圍內的日期，將按下述條件進行比較運算。

・日期 A：2006/02/30（該日期實際上並不存在，但在允許設置範圍內。）

・日期 B：2007/03/29

・日期 C：2008/02/31（該日期實際上並不存在，但在允許設置範圍內。）

比較物件	比較條件		
	A<B	B<C	A<C
日	×	×	○
月	×	×	×
月、日	○	×	○
年	○	○	○
年、日	○	○	○
年、月	○	○	○
年、月、日	○	○	○
無	×	×	×

○：導通　×：非導通

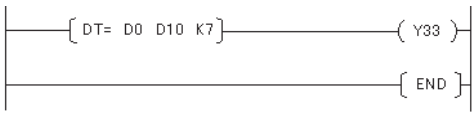
出　錯

(1) DT=、DT<>、DT>、DT<=、DT<、DT>= 指令中無運算出錯。

程式示例

(1) 以下爲將 D0 的資料與 D10 的資料（年、月、日）進行比較，當 D0 的資料與 D10 的資料一致時，將 Y33 變爲導通狀態的程式。

[梯形圖模式]

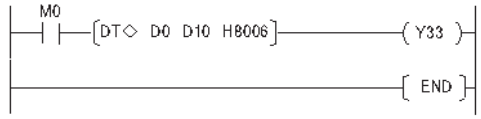


[列表模式]

步	指令	軟元件
0	LDDT=	D0 D10 K7
4	OUT	Y33
5	END	

(2) 以下爲 M0 變爲 ON 時，將 D0 的資料與當前的日期資料（年、月）進行比較，當 D0 的資料與當前的日期資料不一致時，將 Y33 變爲導通狀態的程式。

[梯形圖模式]

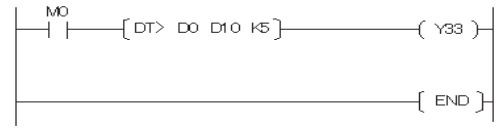


[列表模式]

步	指令	軟元件
0	LD	M0
1	ANDDT◇	D0 D10 H8006
5	OUT	Y33
6	END	

(3) 以下爲 M0 變爲 ON 時，將 D0 的資料與 D10 的資料（年、日）進行比較，當 D10 的資料小於 D0 的資料時，將 Y33 變爲導通狀態的程式。

[梯形圖模式]

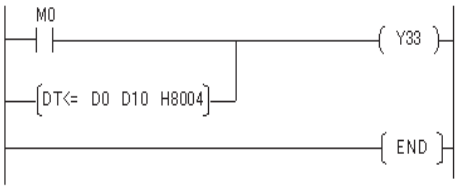


[列表模式]

步	指令	軟元件
0	LD	M0
1	ANDDT>	D0 D10 K5
5	OR	M10
6	ANB	
7	OUT	Y33
8	END	

(4) 以下爲將 D0 的資料與當前的日期資料（年）進行比較，當當前的日期資料大於 D0 的資料時，將 Y33 變爲導通狀態的程式。

[梯形圖模式]



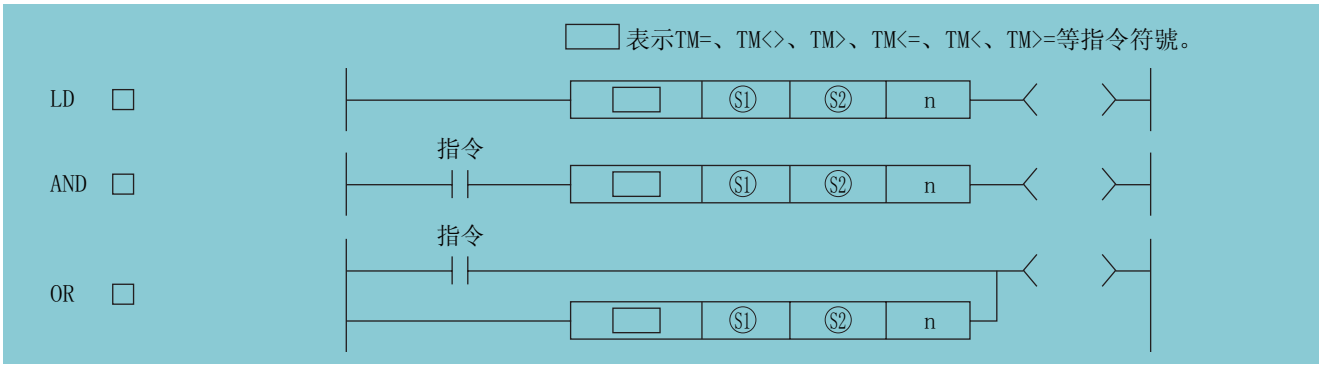
[列表模式]

步	指令	軟元件
0	LD	M0
1	AND	M10
2	ORDT<=	D0 D10 H8004
6	OUT	Y33
7	END	

7.15.8 TM=、TM<>、TM>、TM<=、TM<、TM>=

Basic ✗ | High performance ✗ | Process ✗ | Redundant ✗ | Ver. Universal | LCPU

· 在序列號的前 5 位數爲 “ 10102 ” 以後的 QnU(D)(H)CPU、QnUDE(H)CPU 中可以使用。



Ⓢ1 ： 存儲進行比較的資料的軟元件的起始編號 (BIN16 位)。

Ⓢ2 ： 存儲進行比較的資料的軟元件的起始編號 (BIN16 位)。

n ：表示比較方法的值或者存儲比較方法的資料數 (BIN16 位元)。

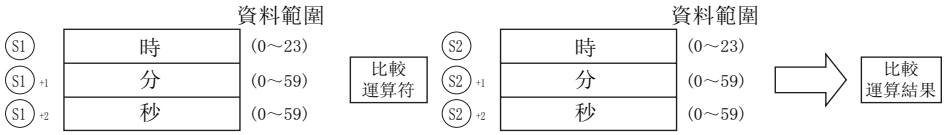
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	---	○		---				---	---
Ⓢ2	---	○		---				---	---
n	---	○		○				○	---

功　能

(1) 對Ⓢ1、Ⓢ2中指定的時間資料進行比較。或者，將Ⓢ1中指定的時間資料與當前時間進行比較。
通過 n 可以選擇比較物件。

(a) 與任意時間資料的比較

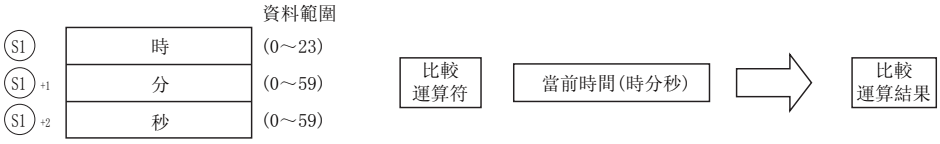
· 將Ⓢ1中指定的時間資料與Ⓢ2中指定的時間資料按照 n 的值通過 a 觸點進行比較。



(b) 與當前時間資料的比較

· 將Ⓢ1中指定的時間資料與當前的時間資料按照 n 的值通過 a 觸點進行比較。

· 將Ⓢ2中指定的時間資料作爲虛擬資料處理，將其忽略。



要 點

與任意的時間資料比較時，或者與當前的時間資料比較時，Ⓢ1、Ⓢ2中的某一個處於以下狀況時將變爲運算出錯狀態（出錯代碼： 4101）或者可能導致誤動作。

· 變址修飾目標的指定範圍超出了軟元件範圍時。
· 在未設置文件寄存器的狀態下指定了文件寄存器時。

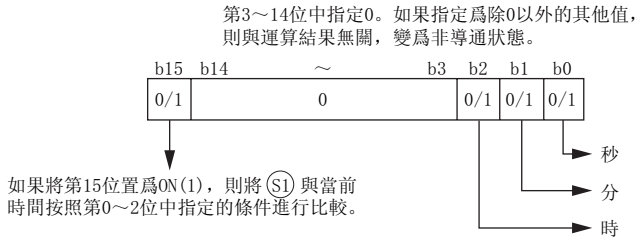
(2) 各專案設置是以 BIN 值格式進行設置的。

(3) Ⓢ1、Ⓢ2的“ 時 ”是在 0 ～ 23(0 時～ 23 時）的範圍內進行設置。

(4) Ⓢ1+1、Ⓢ2+1 的“ 分 ”是在 0 ～ 59(0 分～ 59 分）的範圍內進行設置。

(5) Ⓢ1+2、Ⓢ2+2 的“ 秒 ”是在 0 ～ 59(0 秒～ 59 秒）的範圍內進行設置。

(6) 通過在 n 中指定下圖的值，可以對比較物件進行詳細指定。
n 的位構成如下所示。



(a) 比較物件時間（第 0 ～ 2 位）

· 0：不對比較物件的時間資料（時 / 分 / 秒）進行比較。

· 1：對比較物件的時間資料（時 / 分 / 秒）進行比較。

(b) 比較運算物件（第 15 位元）

· 0：將Ⓢ1中指定的時間資料與Ⓢ2中指定的時間資料進行比較。

· 1：將Ⓢ1中指定的時間資料與當前的時間資料進行比較。

Ⓢ2中指定的時間資料將被忽略。

(c) 比較物件位元的處理內容如下所示。

與任意的時間資料進行比較時的 n 值	與當前的時間資料進行比較時的 n 值	比較物件時間	處理內容
0001_H	8001_H	日	僅對秒 (Ⓢ1+2) 進行比較。
0002_H	8002_H	月	僅對分 (Ⓢ1+1) 進行比較。
0003_H	8003_H	月、日	對分 (Ⓢ1+1)、秒 (Ⓢ1+2) 進行比較。
0004_H	8004_H	年	僅對時 (Ⓢ1) 進行比較。
0005_H	8005_H	年、日	對時 (Ⓢ1)、秒 (Ⓢ1+2) 進行比較。
0006_H	8006_H	年、月	對時 (Ⓢ1)、分 (Ⓢ1+1) 進行比較。
0007_H	8007_H	年、月、日	對時 (Ⓢ1)、分 (Ⓢ1+1)、秒 (Ⓢ1+2) 進行比較。
除 0001_H ～ 0007_H、8001_H ～ 8007_H 以外		無	對時 (Ⓢ1)、分 (Ⓢ1+1)、秒 (Ⓢ1+2) 均不進行比較。(變爲非導通狀態。)

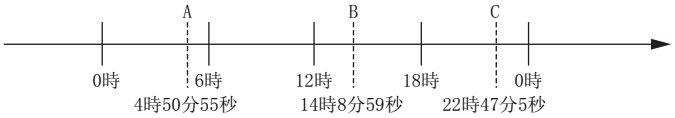
(7) 在比較物件軟元件中存儲的資料不能被識別爲時間資料的情況下，執行指令後將 SM709 置爲 ON，變爲非導通狀態。一旦 SM709 變爲 ON 後，在進行重定 / 電源 OFF 之前將保持 ON 狀態不變，因此應根據需要將其置爲 OFF。

Ⓢ1～Ⓢ1+2 或者Ⓢ2～Ⓢ2+2 超出了指定軟元件範圍時，SM709 也將變爲 ON，變爲非導通狀態。

(8) 各指令的比較運算結果如下所示。

內的指令符號	條件	比較運算結果	內的指令符號	條件	比較運算結果
TM=	Ⓢ1=Ⓢ2	導通狀態	TM=	Ⓢ1≠Ⓢ2	非導通狀態
TM<>	Ⓢ1≠Ⓢ2		TM<>	Ⓢ1=Ⓢ2	
TM>	Ⓢ1>Ⓢ2		TM>	Ⓢ1≤Ⓢ2	
TM<=	Ⓢ1≤Ⓢ2		TM<=	Ⓢ1>Ⓢ2	
TM<	Ⓢ1<Ⓢ2		TM<	Ⓢ1≥Ⓢ2	
TM>=	Ⓢ1≥Ⓢ2		TM>=	Ⓢ1<Ⓢ2	

(a) 時間的比較示例如下所示。



上述時間 A、B、C 的比較運算結果如下所示。

即使在相同的條件下進行比較，根據選擇的比較物件其比較運算結果將有所不同。

比較物件	比較條件		
	A<B	B<C	A<C
秒	○	×	×
分	×	○	×
分、秒	×	○	×
時	○	○	○
時、秒	○	○	○
時、分	○	○	○
時、分、秒	○	○	○
無	×	×	×

○：導通　×：非導通

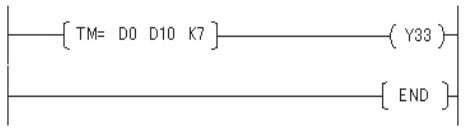
出　錯

(1) 在 TM=、TM<>、TM>、TM<=、TM<、TM>= 指令中無運算出錯。

程式示例

(1) 以下爲將 D0 的資料與 D10 的資料（時、分、秒）進行比較，當 D0 的資料與 D10 的資料一致時，將 Y33 變爲導通狀態的程式。

[梯形圖模式]

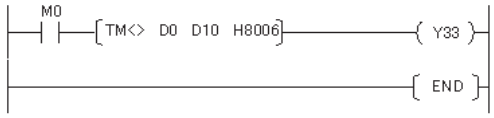


[列表模式]

步	指令	軟元件
0	LDTM=	D0 D10 K7
4	OUT	Y33
5	END	

(2) 以下爲 M0 變爲 ON 時，將 D0 的資料與當前的時間資料（時、分）進行比較，當 D0 的資料與當前的時間資料不一致時，將 Y33 變爲導通狀態的程式。

[梯形圖模式]

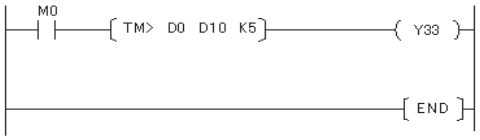


[列表模式]

步	指令	軟元件
0	LD	M0
1	ANDTM<>	D0 D10 H8006
5	OUT	Y33
6	END	

(3) 以下爲 M0 變爲 ON 時，將 D0 的資料與 D10 的資料（時、秒）進行比較，當 D10 的資料小於 D0 的資料時，將 Y33 變爲導通狀態的程式。

[梯形圖模式]

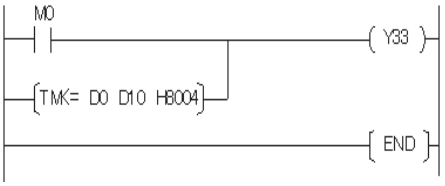


[列表模式]

步	指令	軟元件
0	LD	M0
1	ANDTM>	D0 D10 K5
5	OR	M10
6	ANB	
7	OUT	Y33
8	END	

(4) 以下爲將 D0 的資料與當前的時間資料（時）進行比較，當當前的時間資料大於 D0 的資料時，將 Y33 變爲導通狀態的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	M0
1	AND	M10
2	ORTM<=	D0 D10 H8004
6	OUT	Y33
7	END	

7.16 擴展時鐘指令

Basic ✕ | High performance Ver. | Process Ver. | Redundant Ver. | Universal | LCPU

- 在序列號的前 5 位數爲 “ 07032 ” 以後的高性能型 QCPU 中可以使用。
- 在序列號的前 5 位數爲 “ 07032 ” 以後的過程 CPU 中可以使用。
- 在序列號的前 5 位數爲 “ 07032 ” 以後的冗餘 CPU 中可以使用。

7.16.1 S.DATERD、SP.DATERD

Ⓓ ：存儲讀取的時鐘資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	---	○		---					

功　能

(1) 根據 CPU 模組的時鐘因數讀取 “ 年、月、日、時、分、秒、星期、1/1000 秒 ” 後，以 BIN 值存儲到Ⓓ中指定的軟元件的後面。

時鐘因數 ⇨

Ⓓ	年(西曆)	(1980～2079)
Ⓓ+1	月	(1～12)
Ⓓ+2	日	(1～31)
Ⓓ+3	時(24小時時鐘)	(0～23)
Ⓓ+4	分	(0～59)
Ⓓ+5	秒	(0～59)
Ⓓ+6	星期	(0～6)
Ⓓ+7	1/1000秒	(0～999)

(2) Ⓓ的 “ 年 ” 以西曆 4 位元數存儲。

(3) Ⓓ+6 的 “ 星期 ” 中以 “ 0 ～ 6 ” 存儲 “ 星期日～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
存儲資料	0	1	2	3	4	5	6

(4) 閏年時將進行自動修正。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

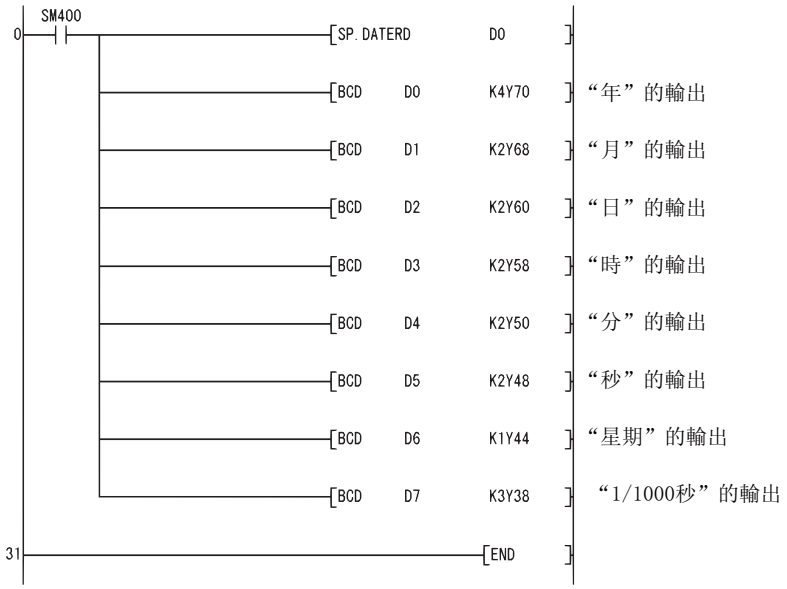
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	Ⓓ中指定的檔案名超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將時鐘資料以 BCD 值進行輸出的程式。

年..... Y70 ～ Y7F
月..... Y68 ～ Y6F
日..... Y60 ～ Y67
時..... Y58 ～ Y5F
分..... Y50 ～ Y57
秒..... Y48 ～ Y4F
星期..... Y44 ～ Y47
1/1000 秒 ..... Y38 ～ Y43

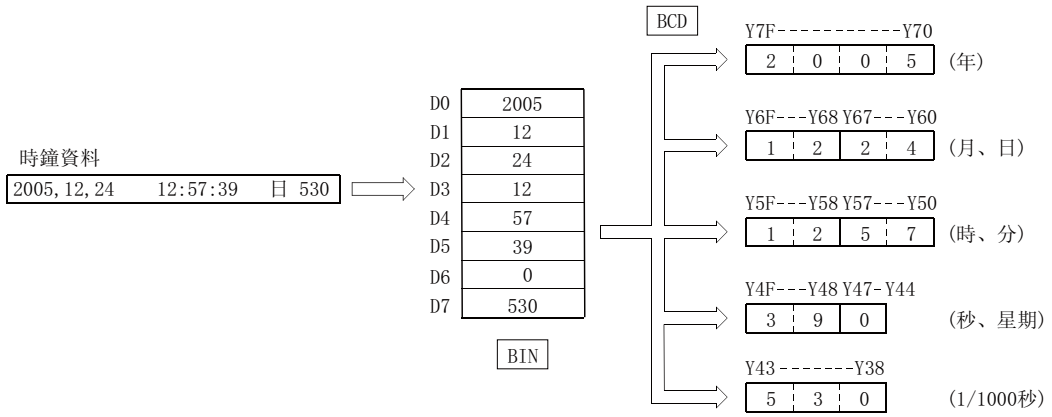
[梯形圖模式]



[列表模式]

步	指令	軟元件	
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		

[動作]



注意事項

(1) 對於本指令，即使在 CPU 模組中設置了錯誤的時鐘資料的情況下，也將讀取時鐘資料並存儲到軟元件中。(例： 2 月 30 日)
通過 DATEWR 指令或 GX Developer 設置時鐘資料時，應設置正確的時鐘資料。

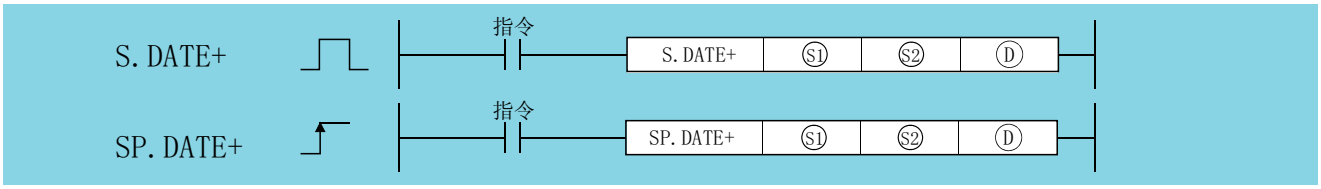
(2) 讀取 1/1000 秒的時鐘資料時的誤差最大爲 2ms。(CPU 模組內部時鐘因數記憶的資料與通過本指令讀取的資料的誤差。)

(3) 對於位軟元件的位數指定，只有在指定滿足下述 (a)、(b) 的條件時才可以使用。
(a) 位數的指定： K4
(b) 軟元件的起始： 16 的倍數
未滿足上述 (a)、(b) 的條件的情況下，將變爲 INSTRCT CODE ERR.(出錯代碼： 4004) 狀態。

Basic × | High performance Ver. | Process Ver. | Redundant Ver. | Universal | LCPU

- 在序列號的前 5 位數爲 “ 07032 ” 以後的高性能型 QCPU 中可以使用。
- 在序列號的前 5 位數爲 “ 07032 ” 以後的過程 CPU 中可以使用。
- 在序列號的前 5 位數爲 “ 07032 ” 以後的冗餘 CPU 中可以使用。

7.16.2 S.DATE+、SP.DATE+



(S1) : 存儲被進行加法運算的時間資料的軟元件的起始編號 (BIN16 位)。
(S2) : 存儲進行加法運算的時間資料的軟元件的起始編號 (BIN16 位)。
(D) : 存儲加法運算結果時間資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
(S1)	---	○		---					
(S2)	---	○		---					
(D)	---	○		---					

功　能

(1) 將(S1)中指定的時間資料與(S2)中指定的時間資料進行加法運算，並將加法運算結果存儲到(D)中指定的軟元件編號的後面。

		設置資料			設置資料			設置資料
(S1)	時	(0～23)	(S2)	時	(0～23)	(D)	時	(0～23)
(S1)+1	分	(0～59)	(S2)+1	分	(0～59)	(D)+1	分	(0～59)
(S1)+2	秒	(0～59) +	(S2)+2	秒	(0～59) ⇒	(D)+2	秒	(0～59)
(S1)+3	－		(S2)+3	－		(D)+3	－	
(S1)+4	1/1000秒	(0～999)	(S2)+4	1/1000秒	(0～999)	(D)+4	1/1000秒	(0～999)

例如，將 6 時 32 分 40 秒 875 與 7 時 48 分 10 秒 500 進行加法運算時的情況如下所示。

(S1)	6時		(S2)	7時		(D)	14時
(S1)+1	32分		(S2)+1	48分		(D)+1	20分
(S1)+2	40秒	+	(S2)+2	10秒	⇒	(D)+2	51秒
(S1)+3	－		(S2)+3	－		(D)+3	－
(S1)+4	875		(S2)+4	500		(D)+4	375

(2) 運算結果的時間超過了 24 小時時，將該值減去 24 小時後的值作爲運算結果。

例如，將 14 時 20 分 30 秒 875 與 20 時 20 分 20 秒 500 進行加法運算時，其結果不是 34 時 40 分 51 秒 375，而是 10 時 40 分 51 秒 375。

ⓢ1	14時		ⓢ2	20時		Ⓓ	10時
ⓢ1+1	20分		ⓢ2+1	20分		Ⓓ+1	40分
ⓢ1+2	30秒	+	ⓢ2+2	20秒	⇨	Ⓓ+2	51秒
ⓢ1+3	–		ⓢ2+3	–		Ⓓ+3	–
ⓢ1+4	875		ⓢ2+4	500		Ⓓ+4	375

要 點

ⓢ1+3、ⓢ2+3、Ⓓ+3 的軟元件不用於運算。

可以將通過 S(P).DATERD 讀取的時鐘資料直接進行加法運算。

Ⓓ	時
Ⓓ+1	分
Ⓓ+2	秒
Ⓓ+3	星期
Ⓓ+4	1/1000秒

通過S(P).DATERD指令讀取時，在秒與1/1000秒之間加入星期。
由於在S(P).DATE+指令中不對星期進行運算，因此可以直接進行加法運算。

出 錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	ⓢ1、ⓢ2的設置資料超出了範圍時。(參閱功能 (1))	---	○	○	○	○	○
4101	ⓢ1、ⓢ2、Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

注意事項

(1) 對於位軟元件的位數指定，只有在指定滿足下述 (a)、(b) 的條件時才可以使用。

(a) 位數的指定 ： K4

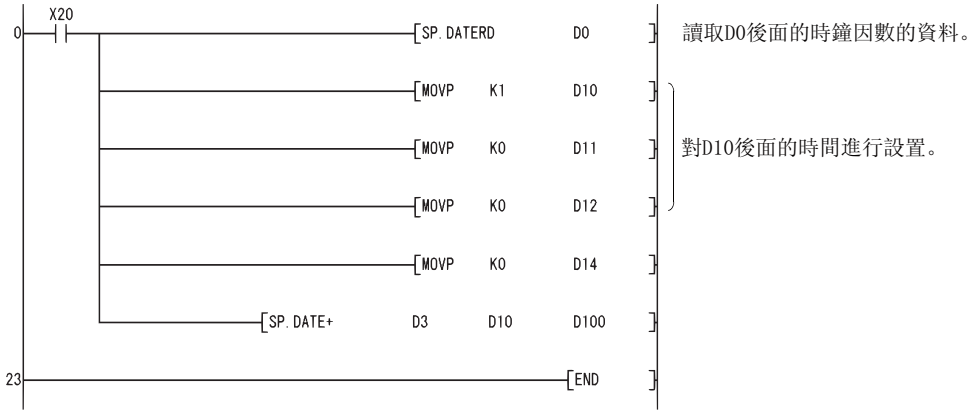
(b) 軟元件的起始： 16 的倍數

未滿足上述 (a)、(b) 的條件的情況下，將變爲 INSTRCT CODE ERR.(出錯代碼： 4004) 狀態。

程式示例

(1) 以下爲 X20 變爲 ON 時，對從時鐘因數中讀取的時間資料進行 1 小時的加法運算後，將運算結果存儲到 D100 後面的程式。

[梯形圖模式]

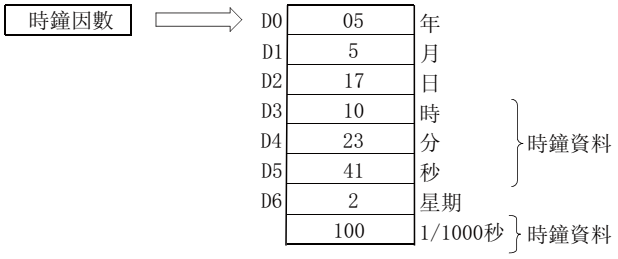


[列表模式]

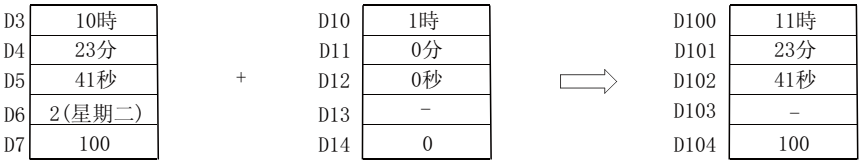
步	指令	軟元件		
0	LD	X20		
1	SP.DATERD		D0	
7	MOVP	K1	D10	
9	MOVP	K0	D11	
11	MOVP	K0	D12	
13	MOVP	K0	D14	
15	SP.DATE+	D3	D10	D100
23	END			

[動作]

・通過 SP.DATERD 指令讀取時間資料



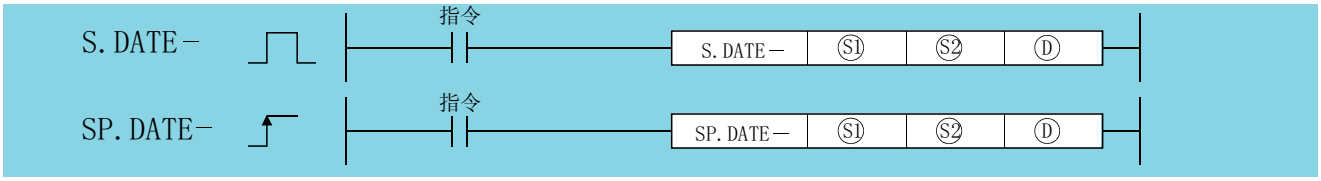
・通過 SP.DATE+ 指令進行加法運算



Basic ✕ | Ver. High performance | Ver. Process | Ver. Redundant | Universal | LCPU

・在序列號的前 5 位數爲 “ 07032 ” 以後的高性能型 QCPU 中可以使用。
・在序列號的前 5 位數爲 “ 07032 ” 以後的過程 CPU 中可以使用。
・在序列號的前 5 位數爲 “ 07032 ” 以後的冗餘 CPU 中可以使用。

7.16.3 S.DATE-、SP.DATE-



(S1) ：存儲被進行減法運算的時間資料的軟元件的起始編號 (BIN16 位)。

(S2) ：存儲進行減法運算的時間資料的軟元件的起始編號 (BIN16 位)。

(D) ：存儲減法運算結果時間資料的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
(S1)	---	○		---					
(S2)	---	○		---					
(D)	---	○		---					

功 能

(1) 將Ⓢ1中指定的時間資料與Ⓢ2中指定的時間資料進行減法運算，並將減法運算結果存儲到Ⓓ中指定的軟元件編號的後面。

		設置資料			設置資料			設置資料
Ⓢ1	時	(0～23)	Ⓢ2	時	(0～23)	Ⓓ	時	(0～23)
Ⓢ1+1	分	(0～59)	Ⓢ2+1	分	(0～59)	Ⓓ+1	分	(0～59)
Ⓢ1+2	秒	(0～59) −	Ⓢ2+2	秒	(0～59) ⇨	Ⓓ+2	秒	(0～59)
Ⓢ1+3	−		Ⓢ2+3	−		Ⓓ+3	−	
Ⓢ1+4	1/1000秒	(0～999)	Ⓢ2+4	1/1000秒	(0～999)	Ⓓ+4	1/1000秒	(0～999)

例如，將 10 時 40 分 20 秒 875 與 3 時 50 分 10 秒 500 進行了減法運算時的情況如下所示。

Ⓢ1	10時		Ⓢ2	3時		Ⓓ	6時
Ⓢ1+1	40分		Ⓢ2+1	50分		Ⓓ+1	50分
Ⓢ1+2	20秒	−	Ⓢ2+2	10秒	⇨	Ⓓ+2	10秒
Ⓢ1+3	−		Ⓢ2+3	−		Ⓓ+3	−
Ⓢ1+4	875		Ⓢ2+4	500		Ⓓ+4	375

(2) 運算結果的時間爲負數時，將該值加上 24 小時後的值作爲運算結果。
例如，將 4 時 50 分 32 秒 875 與 10 時 42 分 12 秒 500 進行了減法運算時，其結果不是 -6 時 8 分 20 秒 375，而是 18 時 8 分 20 秒 375。

Ⓢ1	4時		Ⓢ2	10時		Ⓓ	18時
Ⓢ1+1	50分		Ⓢ2+1	42分		Ⓓ+1	8分
Ⓢ1+2	32秒	−	Ⓢ2+2	12秒	⇨	Ⓓ+2	20秒
Ⓢ1+3	−		Ⓢ2+3	−		Ⓓ+3	−
Ⓢ1+4	875		Ⓢ2+4	500		Ⓓ+4	375

要 點

Ⓢ1+3、Ⓢ2+3、Ⓓ+3 的軟元件不用於運算。
可以將通過 S(P).DATERD 讀取的時鐘資料直接進行減法運算。

Ⓓ	時
Ⓓ+1	分
Ⓓ+2	秒
Ⓓ+3	星期 ◀▮▮▮
Ⓓ+4	1/1000秒

通過S(P).DATERD指令讀取時，在秒與1/1000秒之間加入星期。
由於在S(P).DATE-指令中不對星期進行運算，因此可以直接進行減法運算。

出 錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ1、Ⓢ2的設置資料超出了範圍時。(參閱功能 (1))	---	○	○	○	○	○
4101	Ⓢ1、Ⓢ2、Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

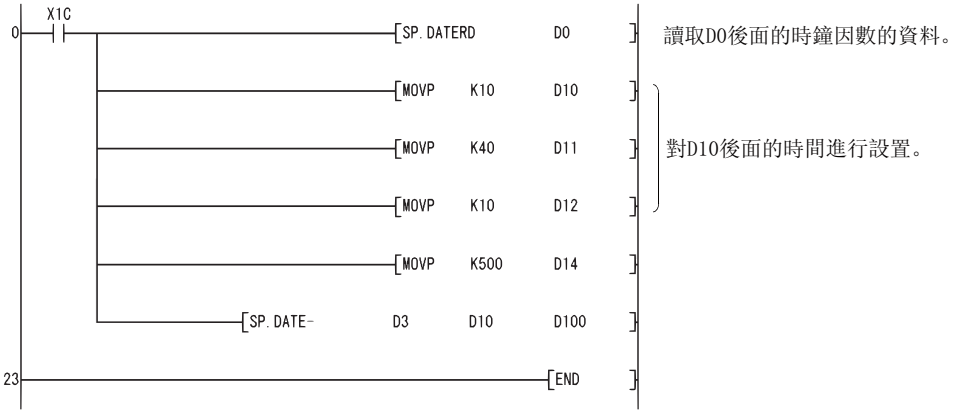
注意事項

(1) 對於位軟元件的位數指定，只有在指定滿足下述 (a)、(b) 的條件時才可以使用。
(a) 位數的指定 ： K4
(b) 軟元件的起始： 16 的倍數
未滿足上述 (a)、(b) 的條件的情況下，將變爲 INSTRCT CODE ERR.(出錯代碼： 4004) 狀態。

程式示例

(1) 以下爲 X1C 將從時鐘因數中讀取的時間資料與 D10 後面存儲的時間資料進行減法運算後，將運算結果存儲到 D100 後面的程式。

[梯形圖模式]

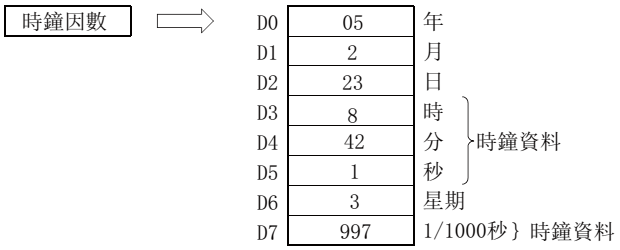


[列表模式]

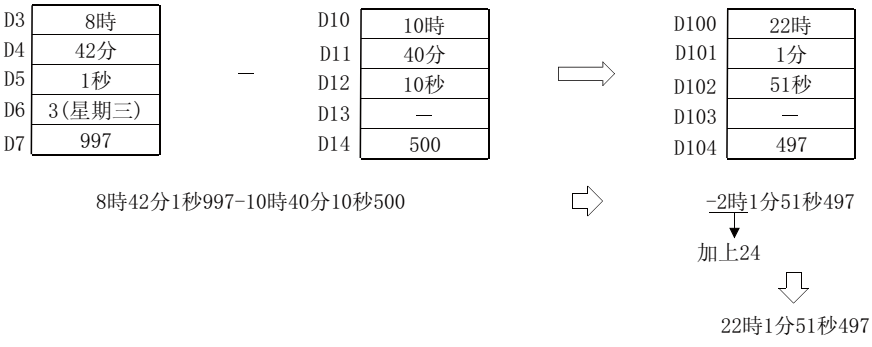
步	指令	軟元件		
0	LD	X1C		
1	SP.DATERD		D0	
7	MOVP	K10	D10	
9	MOVP	K40	D11	
11	MOVP	K10	D12	
13	MOVP	K500	D14	
15	SP.DATE-	D3	D10	D100
23	END			

[動作]

· 通過 SP.DATERD 指令讀取時間資料



· 通過 SP.DATE- 指令進行減法運算



8時42分1秒997−10時40分10秒500 ⇨ −2時1分51秒497

加上24

22時1分51秒497

7.17 程式控制指令

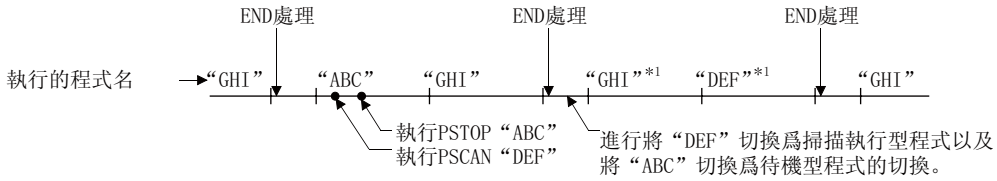
(1) 通過程式控制通指令對執行執行類型進行了切換時的處理如下表所示。

變更前的執行類型	執行指令			
	PSCAN	PSTOP	POFF	PLOW
掃描執行型	保持爲掃描執行型不變。。	變爲待機型。	在下一個掃描中將輸出變爲 OFF。 從下下個掃描以後變爲待機型。	變爲低速執行型。
初始執行型	變爲掃描執行型。			
待機型		保持爲待機型不變。	無處理	
低速執行型	對低速執行型的執行進行中斷後，從下一個掃描開始變爲掃描執行型。 (從 0 步開始執行)	對低速執行型的執行進行中斷後，從下一個掃描開始變爲待機型。	對低速執行型的執行進行中斷後，在下一個掃描中將輸出變爲 OFF。 從下下個掃描以後變爲待機型。	保持爲低速執行型不變。
恒定周期執行型	變爲掃描執行型。	變爲待機型。	在下一個掃描中將輸出變爲 OFF。 從下下個掃描以後變爲待機型。	變爲低速執行型。

要 點

如果將恒定周期執行型程式變更爲其他執行類型的程式，將不能恢復爲恒定周期執行型程式。

(2) 程式的執行類型切換是通過 END 處理進行的。因此在程式的執行過程中不能切換程式的執行類型。
此外，在同一個掃描中對同一個程式設置了不同的類型時，將變爲後執行的執行類型切換指令的執行類型。



*1: “ GHI ”與 “ DEF ”的程式執行順序爲參數的程式設置中所設置的順序。

從恒定執行型程式至其他執行類型的切換時機如下所示。

(a) 對於通用型 QCPU、LCPU
在執行程式控制用指令後的 END 處理中停止恒定周期執行型程式的執行，進行程式類型的變更。

(b) 對於基本型 QCPU、高性能型 QCPU、過程 CPU、冗餘 CPU
在執行程式控制用指令時停止恒定周期執行型程式的執行，在其 END 處理中進行執行類型的變更。

(3) 如果執行了 POFF 指令，將在下一個掃描中將輸出置爲 OFF，在下下個掃描後變爲待機型。
在執行輸出的 OFF 處理之前，即使執行程式控制用指令也將被忽略。

7.17.1 PSTOP、PSTOPP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：置於待機狀態的程式檔案名的字串資料或者存儲字串資料的軟元件的起始編號（字串）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---

功　能

(1) 將Ⓢ中指定的軟元件中存儲的檔案名的程式置爲待機狀態。

(2) 只有驅動器 0（程式記憶體 / 內置 RAM）中存儲的程式才可以被置爲待機狀態。

(3) 指定的程式將在 END 處理中被變爲待機狀態。

(4) 即使在參數中指定了執類型，本指令也將優先。

(5) 檔案名中無需指定副檔名（.QPG）。
（僅以 .QPG 文件作爲物件。）

出　錯

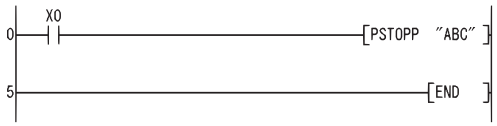
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	Ⓢ中指定的檔案名程式不存在時。	---	○	○	○	○	○
2412	Ⓢ中指定的檔案名的程式類型爲 SFC 程式時。	---	○	○	○	○	○
4101	Ⓢ的檔案名存儲目標軟元件超出了相應軟元件範圍時。	---	○	○	○	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將檔案名爲 ABC 的程式置爲待機狀態的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0
1	PSTOPP	"ABC"
5	END	

7.17.2 POFF、POFFP

Ⓢ ：將輸出置爲 OFF 使之變爲待機型程式的檔案名或者存儲檔案名的軟元件的起始編號（字串）。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---

功　能

(1) 對Ⓢ中指定的軟元件中存儲的檔案名的程式的執行類型進行變更。
- 掃描執行型： 在下一個掃描中將輸出置爲 OFF（非執行處理）。在下下個掃描後變爲待機型。
- 低速執行型： 中斷低速執行型程式的執行，在下一個掃描中將輸出置爲 OFF。在下下個掃描後變爲待機型。

(2) 只有驅動器 0（程式記憶體）中存儲的程式才可以被置爲待機型。

(3) 即使在參數中指定了執行類型，本指令也將優先。

(4) 檔案名中無需指定副檔名（.QPG）。
（僅以 .QPG 文件作爲物件。）

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌（SM0）將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	Ⓢ中指定的檔案名程式不存在時。	---	○	○	○	○	○
4101	Ⓢ的檔案名存儲目標軟元件超出了相應軟元件的範圍時。	---	○	○	○	○	○

備 註

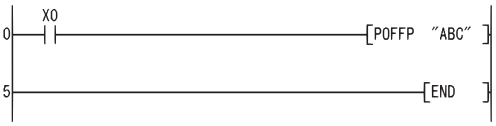
1. 非執行處理是指，執行與各線圈指令的執行條件爲 OFF 狀態時相同的處理。
2. 與條件觸點的 ON/OFF 狀態無關，非執行處理後的各線圈指令的運算結果如下所示。

- OUT 指令 ..... 強制 OFF
- SET 指令
- RST 指令
- SFT 指令
- 基本指令
- 應用指令 ..... 狀態保持
- PLS 指令
- 脈衝化指令（□ P） ..... 執行與條件觸點 OFF 時相同的處理
- 低速 / 高速計時器的當前值 — 0
- 累計計時器的當前值
- 計數器的當前值 ..... 保持

程式示例

(1) 以下爲 X0 變爲 ON 時，將檔案名爲 ABC 的程式置爲非執行後，變爲待機狀態的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0
1	POFFP	"ABC"
5	END	

7.17.3 PSCAN、PSCANP

Ⓢ ：置爲掃描執行型的程式的檔案名或者存儲檔案名的軟元件的起始編號 (字串)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---

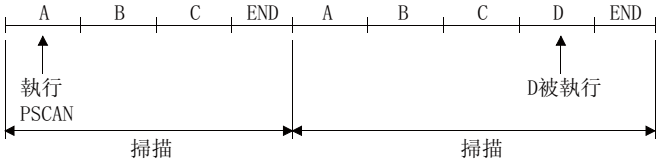
功　能

(1) 將Ⓢ中指定的軟元件中存儲的檔案名的程式置爲掃描執行型。

(2) 只有驅動器 0(程式記憶體 / 內置 RAM) 中存儲的程式才可以被置爲掃描執行型。

(3) 指定的程式將在 END 處理中被變爲掃描執行型。

例 存在程式 A、B、C，通過 A 程式對 D 程式執行了 “ PSCAN ” 時。



(4) 即使在參數中指定了執行類型，本指令也將優先。

(5) 檔案名中無需指定副檔名 (.QPG)。
(僅以 .QPG 文件作爲物件。)

出　錯

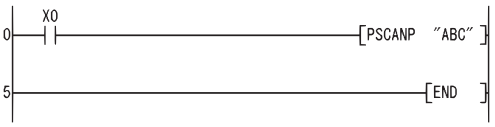
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	Ⓢ 中指定的檔案名程式不存在時。	---	○	○	○	○	○
2504	指定的檔案名的程式類型爲 SFC 程式，但已有其他程式名的 SFC 程式處於已啓動狀態時。(SFC 程式的重復啓動出錯)	---	○	○	○	---	---
4101	Ⓢ 的檔案名存儲目標軟元件超出了相應軟元件範圍時。	---	○	○	○	○	○
4131	指定的檔案名的程式類型爲 SFC 程式，但已有其他程式名的 SFC 程式處於已啓動狀態時。(SFC 程式的重復啓動出錯)	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將檔案名爲 ABC 的程式置爲掃描執行型的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0
1	PSCANP	"ABC"
5	END	

7.17.4 PLOW、PLOWP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ： 置爲低速執行型的程式的檔案名或者存儲檔案名的軟元件的起始編號 (字串)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---

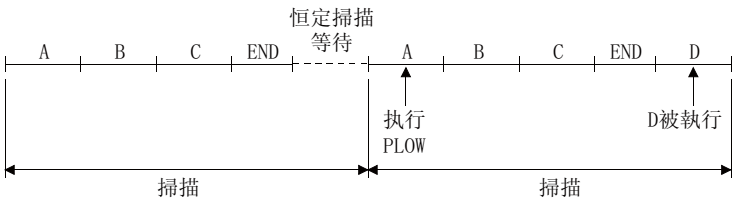
功　能

(1) 將Ⓢ中指定的軟元件中存儲的檔案名的程式置爲低速執行型。

(2) 只有驅動器 0(程式記憶體 / 內置 RAM) 中存儲的程式才可以被置爲低速執行型。

(3) 指定的程式將在 END 處理中被變爲低速執行型。

例 存在程式 A、B、C，通過 A 程式對 D 程式執行了 “ PLOW ” 時。(相當於進行了恒定掃描的設置。)



(4) 即使在參數中指定了執行類型，本指令也將優先。

(5) 檔案名中無需指定副檔名 (.QPG)。
(僅以 .QPG 文件作爲物件。)

出　錯

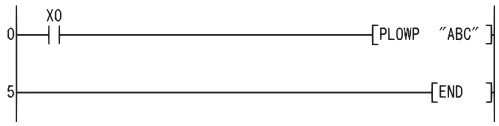
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	指定的檔案名的程式不存在時。	---	○	○	---	---	---
4235	指定的檔案名的程式中存在有 CHK 指令時。	---	○	○	---	---	---

程式示例

(1) 以下爲 X0 變爲 ON 時，將檔案名爲 ABC 的程式置爲低速執行型的程式。

[梯形圖模式]

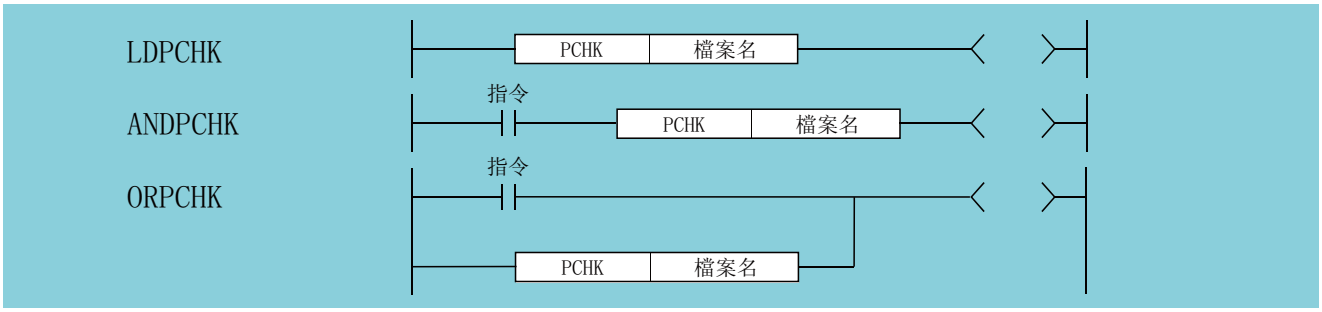


[列表模式]

步	指令	軟元件
0	LD	X0
1	PLOWP	"ABC"
5	END	

7.17.5 PCHK

Basic / High performance / Process / Redundant / Universal / LCPU



Ⓢ ：進行執行狀態檢查的程式的檔案名 (字串)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---							○	---

功　能

(1) 檢查指定的檔案名的程式是處於執行中狀態還是未處於執行中 (非執行) 狀態。

(2) 指定的檔案名的程式處於執行中狀態時，變爲導通，處於非執行中狀態時變爲非導通。

(3) 檔案名指定爲去除了副檔名 (.QPG) 後的檔案名。
例如，檔案名爲 ABC.QPG 時，指定爲 “ ABC ”。

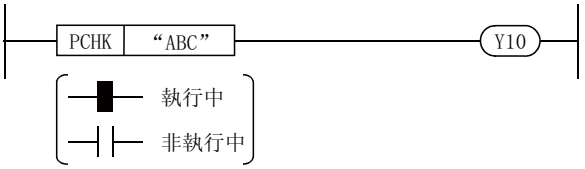
出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	指定的檔案名程式不存在時。	---	○	○	○	---	---

程式示例

(1) 以下爲程式文件 “ ABC.QPG ” 處於執行中狀態時，將 Y10 置爲 ON 的程式。



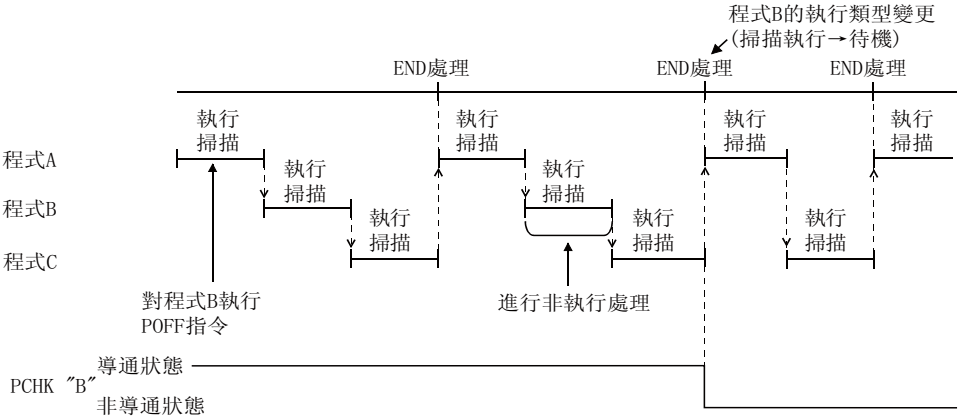
備 註

非執行中表示程式的執行類型爲待機型的狀態。
執行中表示程式的執行類型爲掃描執行型（輸出 OFF 中（非執行處理中）也包含在內）、低速執行型、恒定周期執行型的狀態。

要 點

指定的檔案名的程式（物件程式）處於執行中時，PCHK 指令變爲導通狀態；指定的檔案名的程式（物件程式）處於非執行中時，PCHK 指令變爲非導通狀態；
通過 POFF 指令將物件程式置爲非執行（待機型）狀態時，在進行物件程式的非執行處理期間，PCHK 指令變爲導通狀態。
在非執行處理結束的掃描的 END 處理中，物件程式變爲非執行（待機型）狀態，PCHK 指令變爲非導通狀態。
因此，如果對已通過 POFF 指令結束了非執行處理的程式執行了 PCHK 指令，PCHK 指令有時會變爲導通狀態，應加以注意。

按程式 A →程式 B →程式 C 的順序執行的情況下，程式 A 對程式 B 執行了 POFF 指令，程式 C 對程式 B 執行了 PCHK 指令時的動作如下圖所示。



7.18 其它指令

7.18.1 WDT、WDTP

Basic | High performance | Process | Redundant | Universal | LCPU

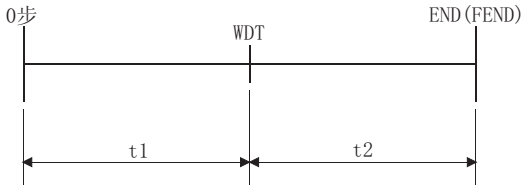
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功　能

(1) 在順控程式的執行過程中進行看門狗計時器的重定。

(2) 根據條件掃描時間超出了看門狗計時器的設置值時使用本指令。
在掃描時間超出了各掃描的看門狗計時器的設置值時，應通過編程工具的參數設置對看門狗計時器的設置值進行變更。

(3) 應使從 0 步開始至 WDT 指令爲止的 t1 及從 WDT 指令開始至 END(FEND) 指令爲止的 t2 均不超過看門狗計時器的設置值。



(4) 雖然在 1 個掃描中可以使用 2 次以上 WDT 指令，但發生異常時至輸出 OFF 爲止需要耗費一定的時間，應加以注意。

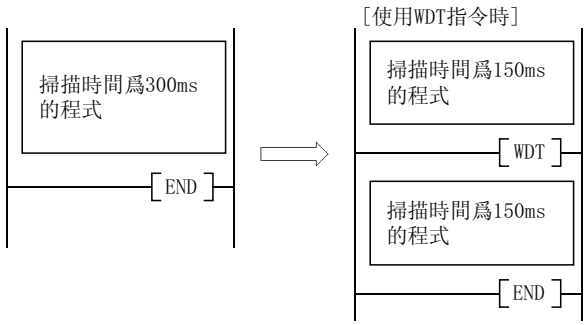
(5) 即使執行了 WDT、WDTP 指令，特殊寄存器中存儲的掃描時間值也不會被清除。
因此，各特殊寄存器的掃描時間值有時會大於參數中設置的看門狗計時器的設置值。

出　錯

(1) WDT(P) 指令中無運算出錯。

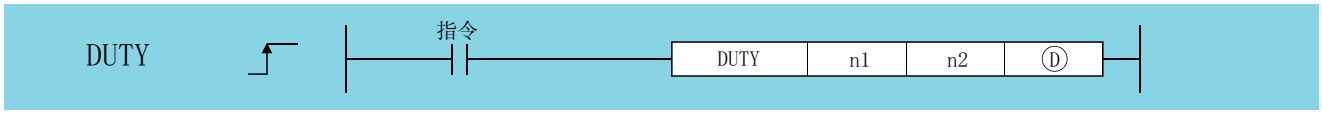
程式示例

(1) 以下爲看門狗計時器的設置爲 200ms，根據程式的執行條件 0 ～ END(FEND) 指令爲止的時間爲 300ms 時的程式。



7.18.2 DUTY

Basic | High performance | Process | Redundant | Universal | LCPU



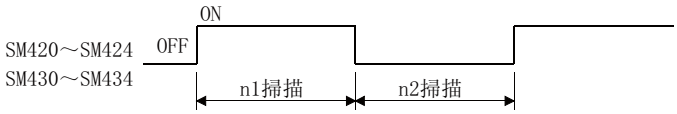
n1 ： 置爲 ON 的掃描數 (BIN16 位)。
n2 ： 置爲 OFF 的掃描數 (BIN16 位)。
Ⓓ ： 用戶用定時時鐘 (SM420 ～ SM424、SM430 ～ SM434)(位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n1	○	○							---
n2	○	○							---
Ⓓ	○ *1	---							---

*1: 僅 SM420 ～ SM424、SM430 ～ SM434 可以使用。

功 能

(1) 將Ⓓ中指定的用戶用定時時鐘 (SM420 ～ SM424、SM430 ～ SM434) 按 n1 中指定的掃描數置爲 ON 後，按 n2 中指定的掃描數置爲 OFF。



(2) 在掃描執行型程式中使用 SM420 ～ SM424，在低速執行型程式中使用 SM430 ～ SM434。

(3) n1、n2 被設置爲 0 時的情況如下所示。

(a) n1=0， n2 ≥ 0 ..... SM420 ～ SM424/SM430 ～ SM434 保持 OFF 狀態不變。

(a) n1<0，n2=0 ..... SM420 ～ SM424/SM430 ～ SM434 保持 ON 狀態不變。

(4) 執行 DUTY 指令時，將 n1、n2、Ⓓ中指定的資料登錄到系統中後，通過 END 處理進行定時脈衝的 ON/OFF。

出 錯

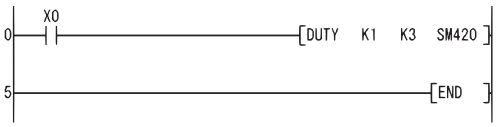
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n1、n2 小於 0 時。	---	---	---	---	○	○
4101	Ⓓ中指定的軟元件爲除 SM420 ～ SM424、SM430 ～ SM434 以外時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 SM420 置爲 1 個掃描 ON，3 個掃描 OFF 的程式。

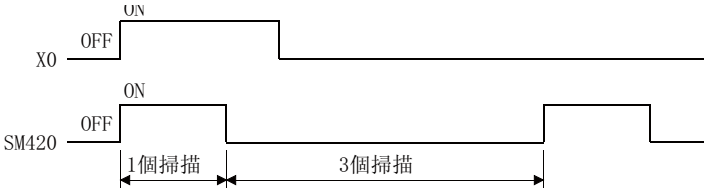
[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X0		
1	DUTY	K1	K3	SM420
5	END			

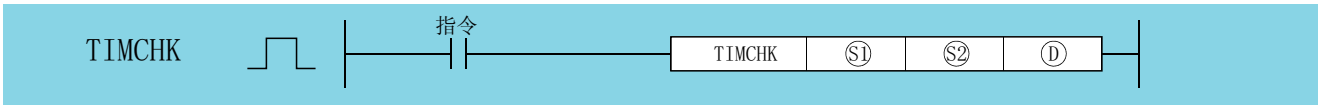
[動作]



Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

7.18.3 TIMCHK



Ⓢ1 ： 存儲計測的當前值的軟元件 (BIN16 位)。

Ⓢ2 ： 存儲計測的設置值的軟元件 (BIN16 位)。

Ⓓ ： 時間到時變爲 ON 的軟元件 (位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ1	---	○		---					---
Ⓢ2	○	○		○					---
Ⓓ	○	---		---					---

功　能

(1) 對條件軟元件的 ON 時間進行計測，如果其連續 ON 的時間超過了Ⓢ2中指定的軟元件中設置的時間，則將Ⓓ中指定的軟元件置爲 ON。

(2) Ⓢ1中指定的軟元件的當前值的清 0 及Ⓓ中指定的軟元件的 OFF 是在執行指令的上升沿時執行。

即使執行指令變爲 OFF 後Ⓢ1中指定的軟元件的當前值及Ⓓ中指定的軟元件的 ON 狀態仍將被保持。

(3) 計測的設置值是以 100ms 爲單位進行設置的。

出　錯

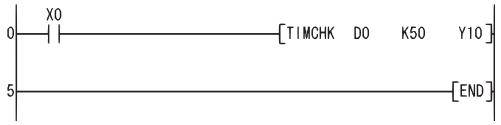
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	指定了不能指定的軟元件時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 X0 的 ON 時間設置爲 5 秒，將當前值存儲軟元件設置爲 D0，將時間到時變爲 ON 的軟元件設置爲 Y10 時的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	X0		
1	TIMCHK	D0	K50	Y10
5	END			

7.18.4 ZRRDB、ZRRDBP

Basic | High performance | Process | Redundant | Universal | LCPU

n　：讀取的文件寄存器的連續位元組號 (BIN32 位元)。

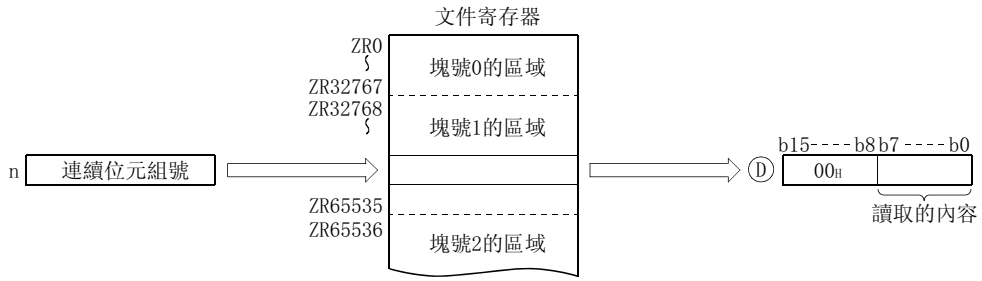
Ⓓ　：存儲讀取的資料的軟元件編號 (BIN16 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
n				○				○	---
Ⓓ				○				---	---

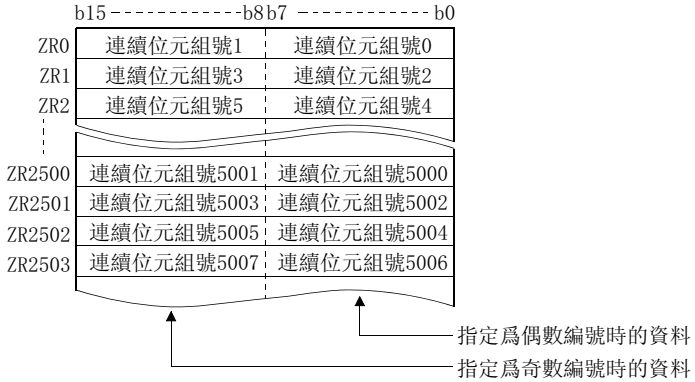
功　能

(1) 在無需理會塊號的情況下，讀取 n 中指定的連續位元組號的文件寄存器的內容，並將其存儲到Ⓓ中指定的軟元件的低 8 位中。

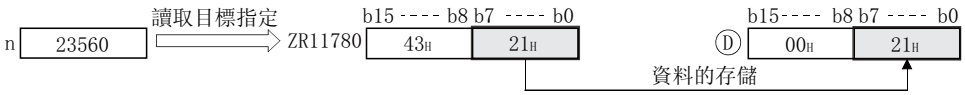
Ⓓ中指定的軟元件的高 8 位將變爲 00_H。



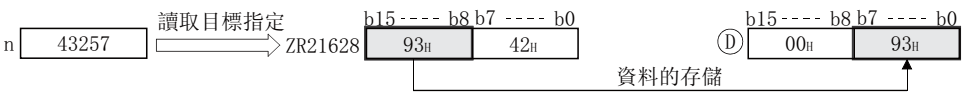
(2) 對應於連續位元組號文件寄存器的編號如下所示。



(a) 指定爲 n=23560 時，讀取 ZR11780 的低 8 位元的資料。



(b) 指定爲 n=41257 時，讀取 ZR21628 的高 8 位元的資料。



出　錯

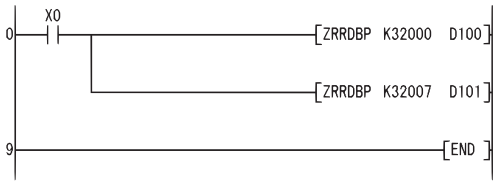
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	指定的軟元件編號 (連續位元組號) 超出了允許指定範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，讀取 R16000 的低位元及 R16003 的高位的內容後，存儲到 D100、D101 中的程式。

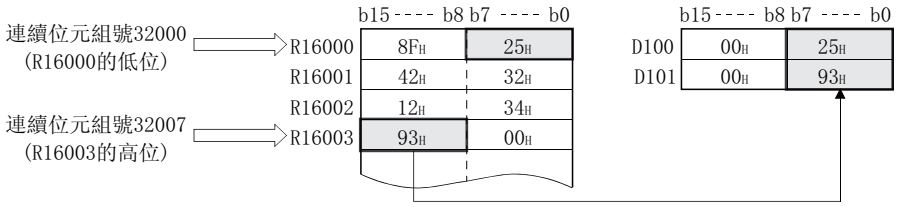
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X0	
1	ZRRDBP	K32000	D100
5	ZRRDBP	K32007	D101
9	END		

[動作]



7.18.5 ZRWRB、ZRWRBP

Basic　High performance　Process　Redundant　Universal　LCPU

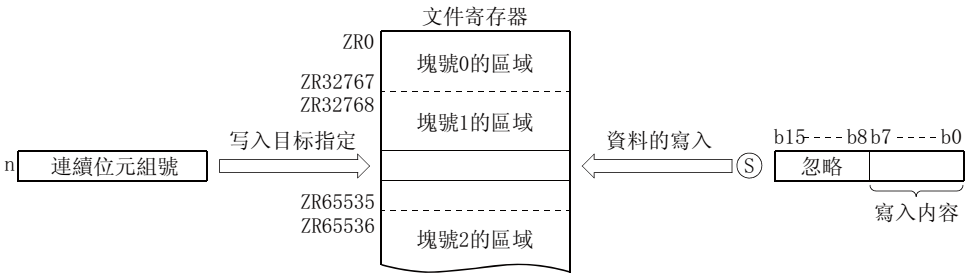
n　：寫入的文件寄存器的連續位元組號 (BIN32 位元)。

Ⓢ　：存儲寫入的資料的軟元件編號 (BIN16 位)。

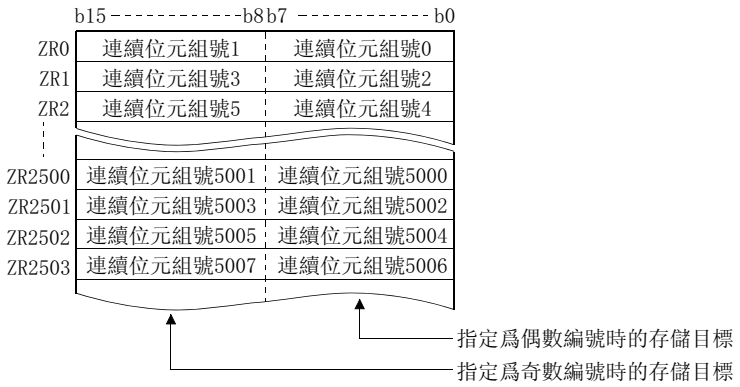
設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
n				○					---
Ⓢ				○					---

功　能

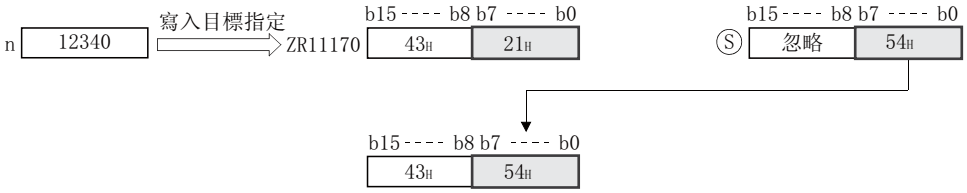
(1) 在無需理會塊號的情況下，將Ⓢ中指定的軟元件中存儲的低位元的內容寫入到 n 中指定的連續位元組號的文件寄存器中。
Ⓢ中指定的軟元件的高 8 位元的資料將被忽略。



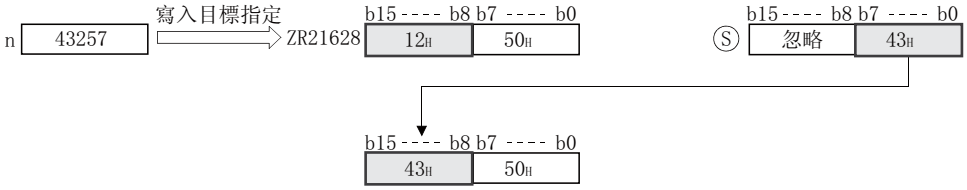
(2) 對應於連續位元組號文件寄存器的編號如下所示。



指定爲 n=12340 時，寫入到 ZR11170 的低 8 位中。



指定爲 n=43257 時，寫入到 ZR21628 的高 8 位中。



出　錯

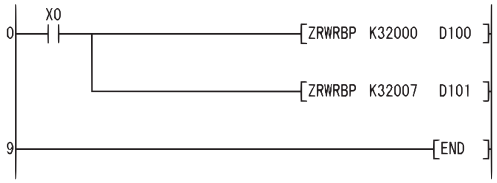
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	指定的軟元件編號（連續位元組號）超出了允許指定範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將 D100、D101 中的低位元資料寫入到 R16000 的低位元及 R16003 的高位中的程式。

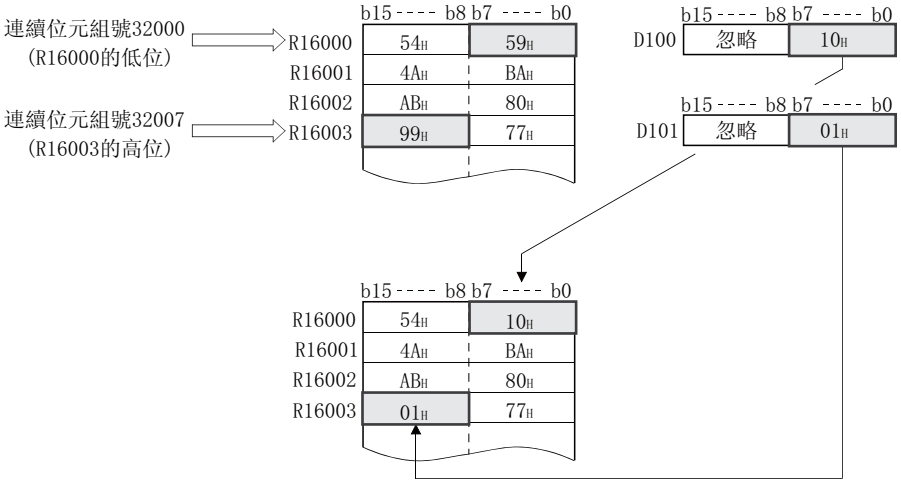
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X0	
1	ZRWRBP	K32000	D100
5	ZRWRBP	K32007	D101
9	END		

[動作]



7.18.6 ADRSET、ADRSETP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ ：讀取間接地址的軟元件編號 (軟元件名)。

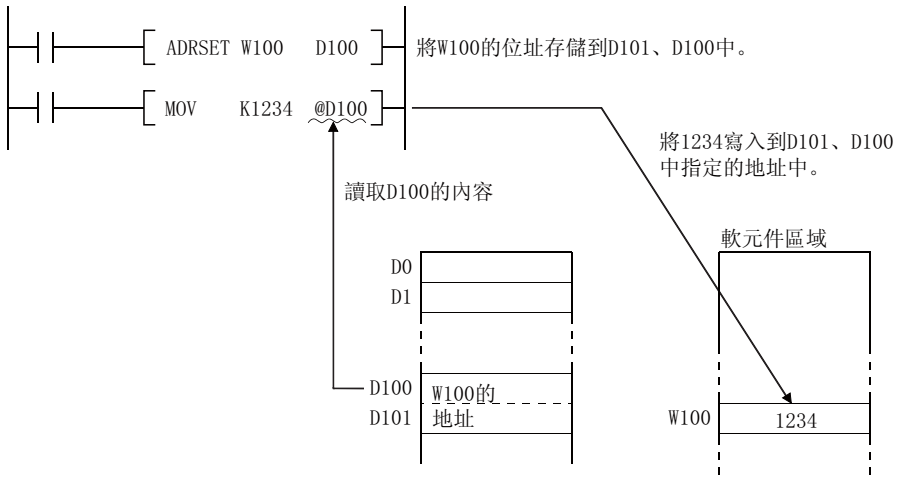
Ⓓ ：存儲Ⓢ中指定的軟元件的間接位址的軟元件編號 (BIN32 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓢ		○				---			
Ⓓ		○				---			

功　能

(1)　將Ⓢ中指定的軟元件的間接位址存儲到Ⓓ+1、Ⓓ的軟元件中。

在順控程式中執行軟元件的間接位址時使用Ⓓ中指定的軟元件中存儲的位址。



(2)　Ⓢ中不能進行位軟元件的位數指定。

出　錯

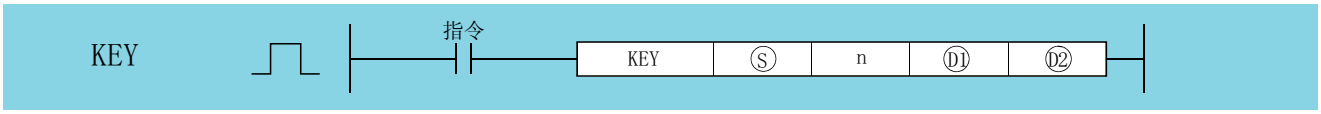
(1)　在 ADRSET(P) 指令中無運算出錯。

備 註

關於間接指定，請參閱 96 頁 3.4 節。

7.18.7 KEY

Basic　High performance　Process　Redundant　Universal　LCPU

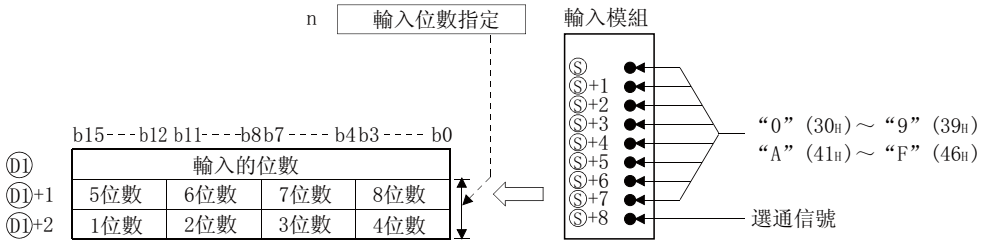


Ⓢ　：進行數位輸入 (X) 的軟元件的起始編號 (位)。
n　：進行數位輸入的位元數 (BIN16 位)。
Ⓓ1　：存儲輸入的數位的軟元件的起始編號 (BIN16 位)。
Ⓓ2　：輸入結束時置爲 ON 的位軟元件編號 (位)。

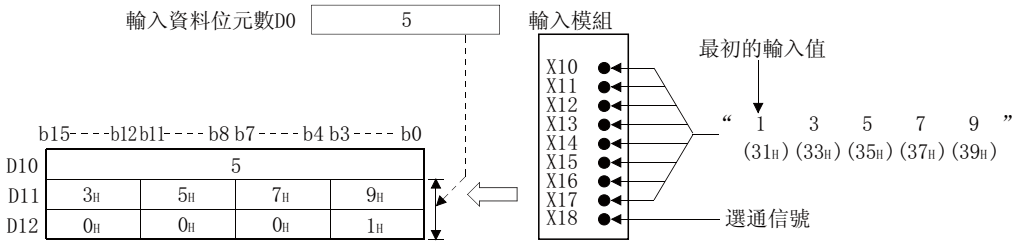
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	○ (僅 X)	---		---		---	---		---
n	○	○		○			○		---
Ⓓ1	---	○		---			---		---
Ⓓ2	○	○		○			---		---

功 能

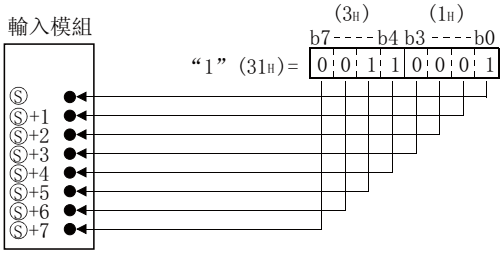
(1) 從Ⓢ中指定的輸入 (X) 中讀取 8 點的 ASCII 資料，轉換爲 16 進制數值後存儲到Ⓓ1中指定的軟元件後面。



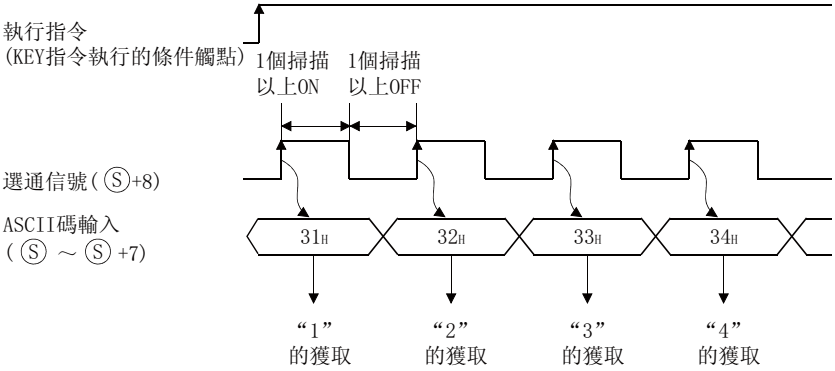
例如，將輸入資料的位元數 (n) 設置爲 5，在輸入模組的 X10 ～ X18 中分別輸入了 “ 31_H ”、“ 33_H ”、“ 35_H ”、“ 37_H ”、“ 39_H ” 時的情況如下所示。



(2) 對Ⓢ中指定的輸入 (X) 進行數位輸入時，將與數位對應的 ASCII 碼執行位展開到Ⓢ～Ⓢ+7 中進行輸入。
可輸入的 ASCII 碼的範圍爲 30_H(0) ～ 39_H(9) 以及 41_H(A) ～ 46_H(F)。

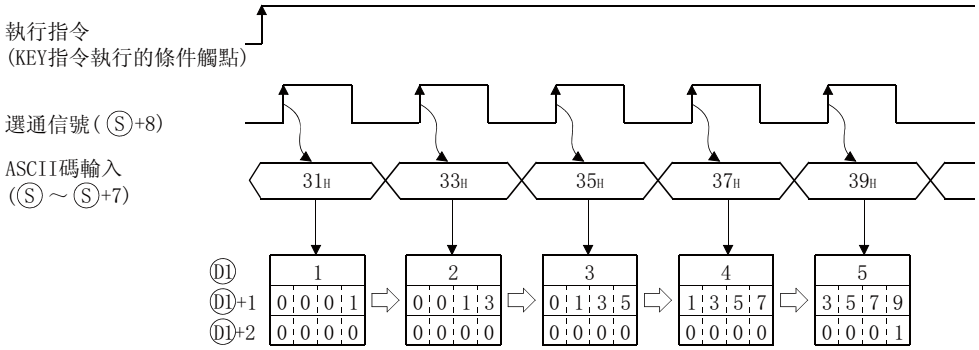


(3) 將 ASCII 碼輸入到Ⓢ～Ⓢ+7 中後，通過將Ⓢ+8 的選通信號置爲 ON，將指定的數位讀取到內部。
選通信號的 ON/OFF 狀態應保持在順控程式的 1 個掃描以上。
如果保持在 1 個掃描以下，有可能無法正常地獲取資料。



(4) 執行指令 (KEY 指令執行的條件觸點) 必須預先置爲 ON，直至指定位數的輸入結束爲止。
如果執行指令爲 OFF 則無法執行 KEY 指令。

(5) 存儲到Ⓓ1中指定的軟元件中時，將實際獲取的數位的位元數存儲到Ⓓ1中，將輸入的 ASCII 碼轉換爲 16 進制 BIN 值後存儲到Ⓓ1+1、Ⓓ1+2 中。



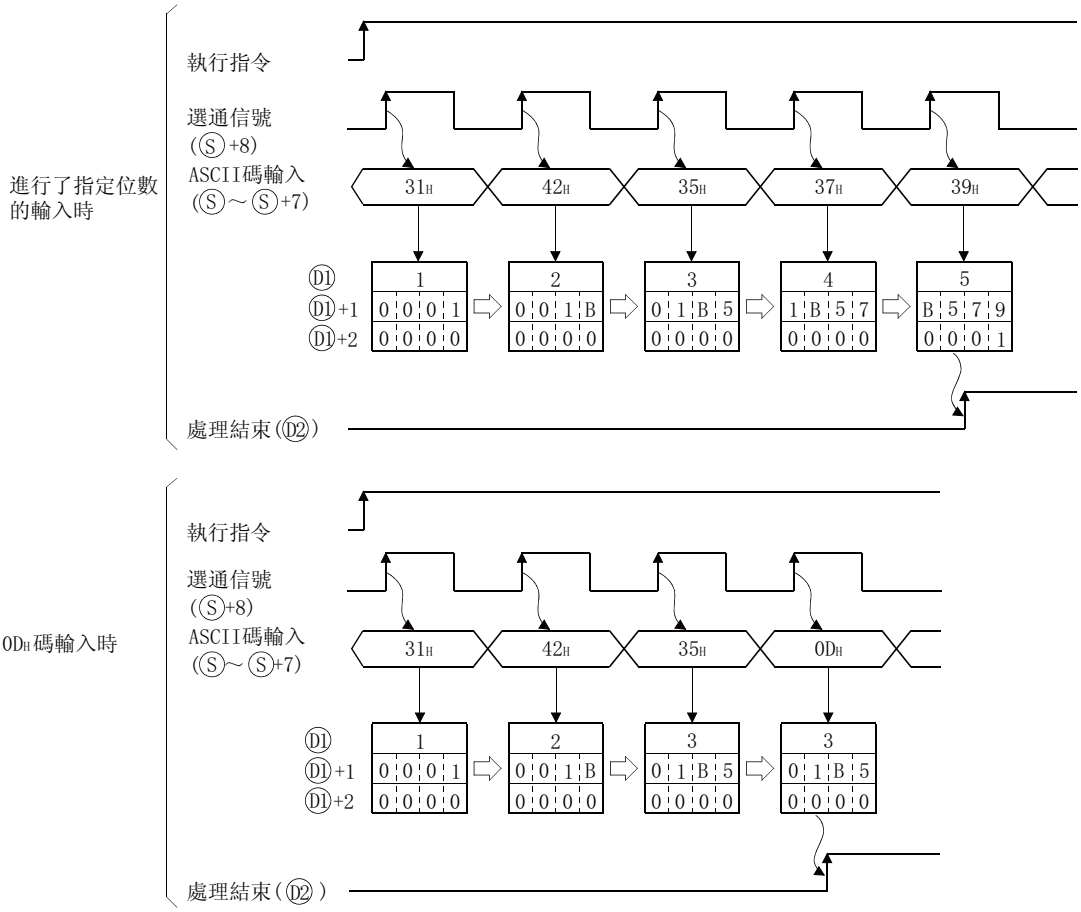
(6) n 中指定的輸入位數範圍爲 1 ～ 8。

(7) 進行輸入資料的內部獲取時，在進行了下述輸入時結束，並將Ⓓ2中指定的位軟元件置爲 ON。

· 進行了 n 中指定的位數的輸入時

· 輸入了 “ $0D_H$ ” 碼時

例如，指定了 n=5 時的動作如下所示。



希望再次進行輸入處理的情況下，需要通過用戶程式對Ⓓ1中存儲的輸入位元數、輸入資料進行清除以及將指定位軟元件置爲 OFF。

如果未進行Ⓓ1的清除以及Ⓓ2的 OFF，將無法進行下一個輸入處理。

出　錯

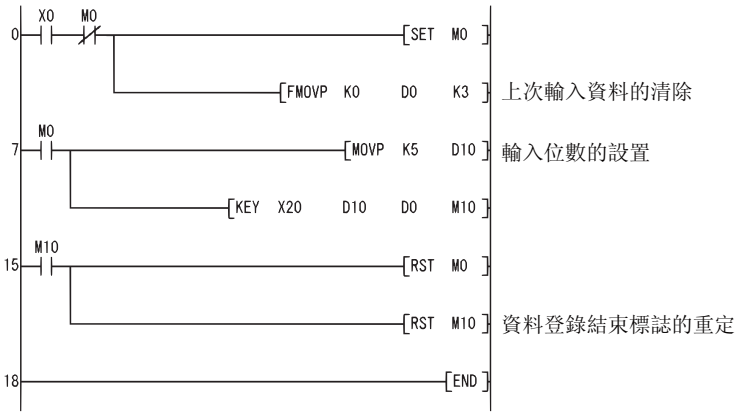
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ 中指定的軟元件不是輸入 (X) 時。 n 中指定的位數超出了 1 ～ 8 的範圍時。	---	○	○	---	---	---

程式示例

(1) 以下爲 X0 變爲 ON 時，通過 X20 ～ X28 連接的數位鍵獲取 5 位元數以內的資料後，存儲到 D0 後面的程式。

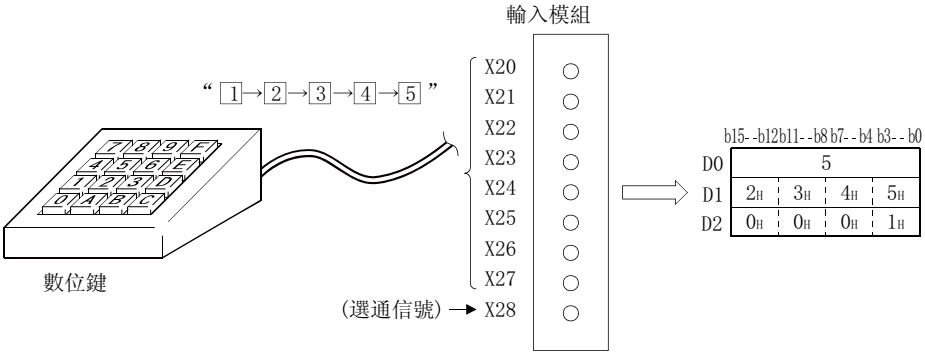
[梯形圖模式]



[列表模式]

步	指令	軟元件			
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				

[動作]



7.18.8 ZPUSH、ZPUSHP、ZPOP、ZPOPP

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓓ ：進行變址寄存器的保存 / 恢復的軟元件的起始編號 (BIN16 位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
Ⓓ	---	○		---					

功　能

ZPUSH

(1) 將下述變址寄存器的內容保存到Ⓓ中指定的軟元件後面。

(如果對變址寄存器的內容進行保存則Ⓓ+0(保存次數) 將被 +1。)

· 基本型 QCPU ..... Z0 ～ Z9

· 高性能型 QCPU、過程 CPU、冗餘 CPU ..... Z0 ～ Z15

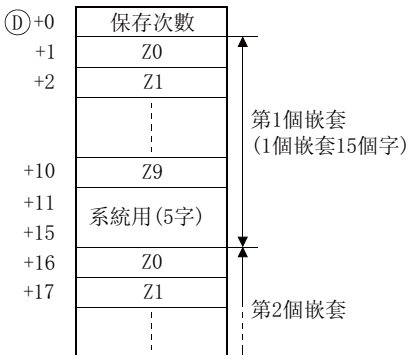
· 通用型 QCPU、LCPU ..... Z0 ～ Z19

(2) 進行資料的恢復時，使用 ZPOP 指令。通過成對使用 ZPUSH 與 ZPOP 指令，可以進行嵌套。

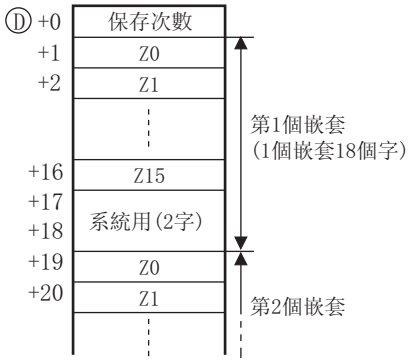
(3) 進行了嵌套時，由於每執行一次 ZPUSH 指令後，將在Ⓓ的後面增加使用的區域，因此應預先預留好與嵌套使用次數相對應的區域。

(4) Ⓓ後面使用的區域的構成如下所示。

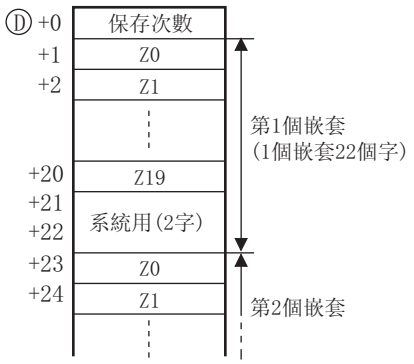
· 使用基本型 QCPU 時



· 使用高性能型 QCPU/ 過程 CPU/ 冗余 CPU 時



・使用通用型 QCPU、LCPU 時



ZPOP

(1) 將Ⓓ中指定的軟元件後面保存資料讀取到變址寄存器中。(對保存的變址寄存器的內容進行讀取時則Ⓓ+0(保存次數) 將被 -1。)

出　錯

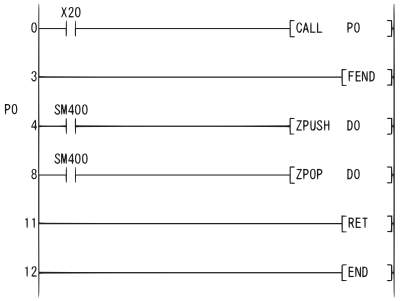
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	ZPOP(P) 指令中Ⓓ+0 的內容 (保存次數) 爲 0 時。	○	○	○	○	○	○
4101	ZPUSH(P) 指令中使用的Ⓓ後面的點數範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

程式示例

(1) 以下爲在 P0 後面的副程式內使用變址寄存器時，將副程式調用之前的變址寄存器的內容保存到 D0 後面的程式。

[梯形圖模式]

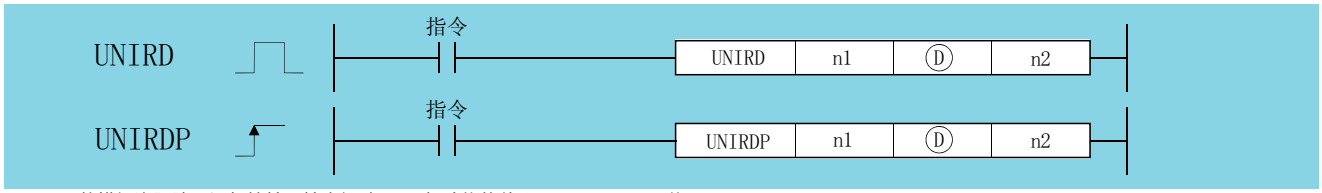


[列表模式]

步	指令	軟元件
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 UNIRD、UNIRDP

Basic | High performance | Process | Redundant | Universal | LCPU



n1　：將模組資訊讀取源起始輸入輸出編號用 16 相除後的值 (0 ～ FFn)(BIN16 位)

Ⓓ　：存儲模組資訊的軟元件的起始編號 (軟元件名)

n2　：讀取資料點數 (0 ～ 256)(BIN16 位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n1	○	○		---				○	---
Ⓓ	---	○		---				---	---
n2	○	○		---				○	---

功　能

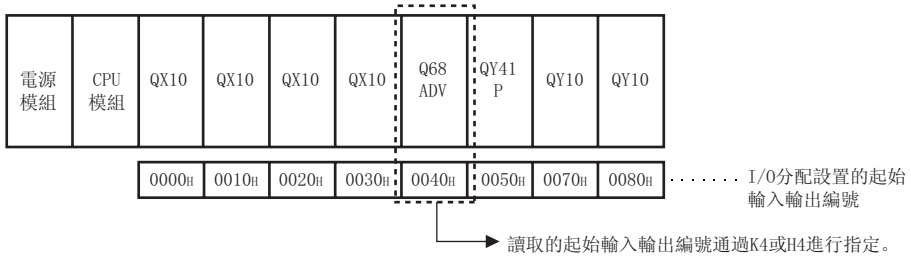
(1) 從 n1(將起始輸入輸出編號用 16 相除後的值) 中指定的模組中，將 n2 中指定的點數的模組資訊存儲到Ⓓ中指定的軟元件後面。

(不是讀取 I/O 分配中指定的模組類型，而是讀取實際安裝的模組狀態。)

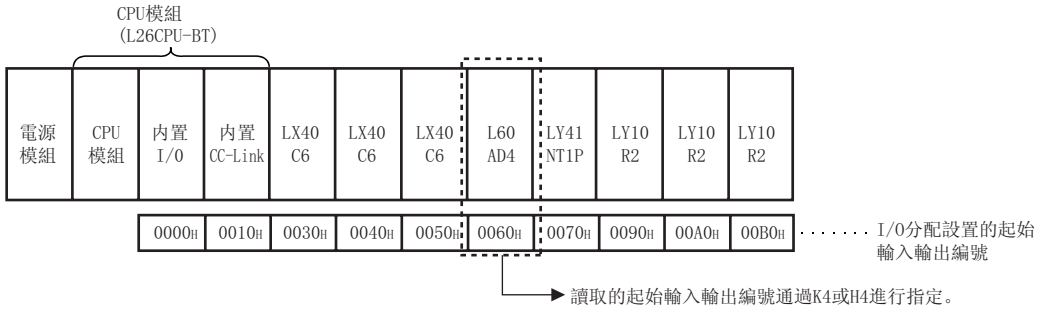
備 註

對於 n1，通過將讀取的模組資訊的插槽的起始輸入輸出編號以 16 進制數 4 位表示時的高 3 位進行指定。

QCPU



LCPU



模組資訊的詳細內容如下所示。

位	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
各模組資訊																

位元	專案名	內容	
		QCPU	LCPU
b0	輸入輸出點數	000：16 點 010：48 點 100：128 點 110：512 點	001：32 點 011：64 點 101：256 點 111：1024 點
b1			
b2			
b3	模組類型	000：輸入模組 001：輸出模組 010：輸入輸出混合模組 011：智慧功能模組	000：輸入模組 001：輸出模組 011：智慧功能模組 111：CPU 內置 I/O
b4			
b5			
b6	外部供應電源狀態 （將來擴展用）	1：外部電源供應中 0：無外部電源供應	0：固定
b7	保險絲熔斷發生有無	1：有保險絲熔斷模組 0：正常	0：固定
b8	在線模組更換狀態 / 從待機系統的執行	1：在線模組更換中或者試圖從冗餘系統的待機系統讀取擴展基板上的模塊資訊。*1 0：除上述以外	0：固定
b9	輕・中度出錯狀態	1：有輕・中度出錯	0：正常
b10	模組出錯狀態	00：無模組出錯 10：中度出錯	01：輕度出錯 11：重度出錯
b11			
b12	模組準備狀態	1：正常	0：有模組出錯
b13	空餘	0：固定	
b14	A/Q 模組	1：Q 系列模組 0：Q 系列模組	0：固定
b15	模組安裝狀態	1：模組已安裝	0：模組未安裝

*1 ： 對於多 CPU 系統中使用的通用型 QCPU，即使其他機號管理的模組處於在線模組更換中的情況下也置爲 ON。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

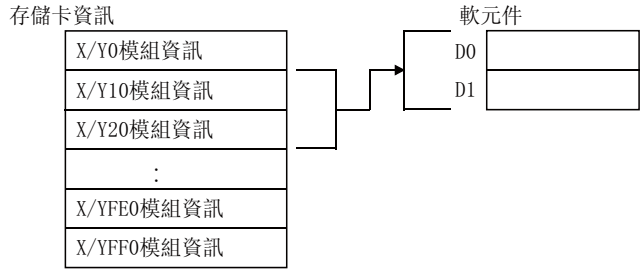
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n1 超出了 0 ～ FF_H 的範圍時。 n2 超出了 0 ～ 256 的範圍時。 n1 與 n2 的合計爲 257 以上時。	---	○	○	○	○	○ *1
	n1 超出了 0 ～ $3F_H$ 的範圍時。 n2 超出了 0 ～ 64 的範圍時。 n1 與 n2 的合計爲 65 以上時。	Q00/Q01	---	---	---	---	○ *2
	n1 超出了 0 ～ F_H 的範圍時。 n2 超出了 0 ～ 16 的範圍時。 n1 與 n2 的合計爲 17 以上時。	Q00J	---	---	---	---	---
4101	ⓓ中指定的軟元件編號後面的 n2 中指定的點數範圍超出了相應軟元件的範圍時。	○	○	○	○	○	○

*1 ： 以 L26CPU-BT、L26CPU-PBT 爲物件。

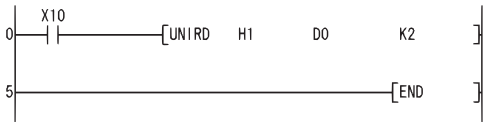
*2 ： 以 L02CPU、L02CPU-P 爲物件。

程式示例

(1) 以下爲將 X10 置爲 ON 時，將輸入輸出編號爲 10H ～ 20H 的模組資訊存儲到 D0 後面的程式。



[梯形圖模式]

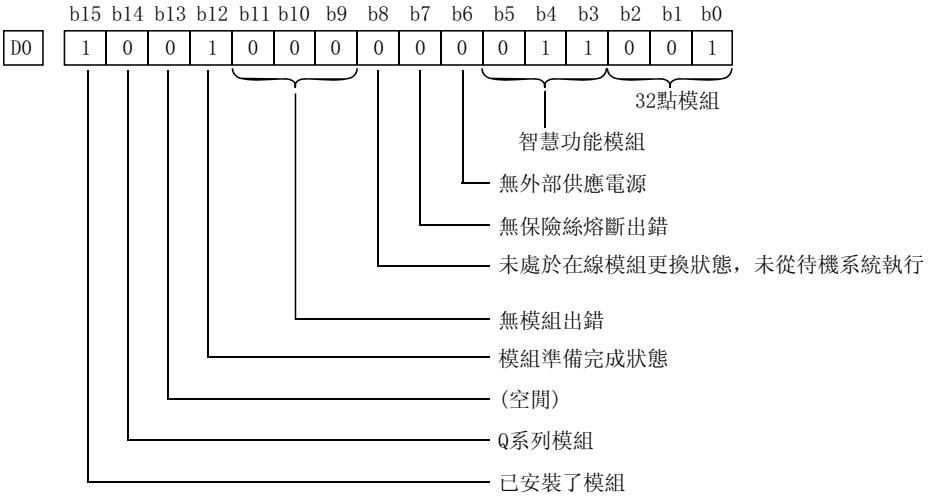


[列表模式]

步	指令	軟元件		
0	LD	X10		
1	UNIRD	H1	D0	K2
5	END			

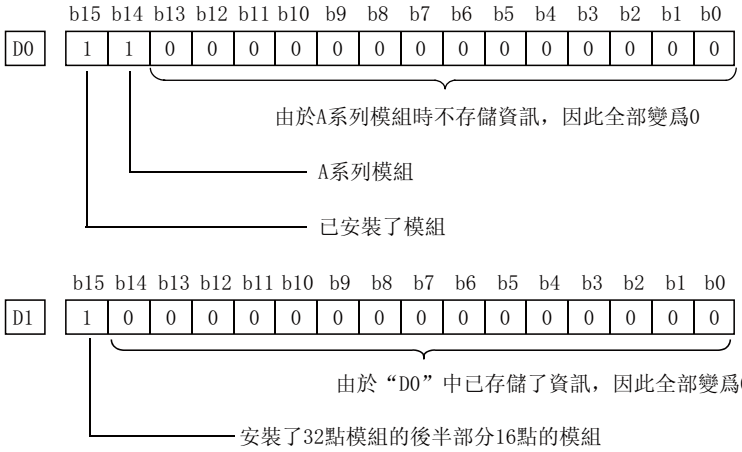
讀取結果側 (讀取到 D0 時)

1) Q 系列 32 點智慧功能模組時



・48 點模組時 D2 中存儲與 D1 相同的內容，64 點模組時 D2 及 D3 中分別存儲與 D1 相同的內容。

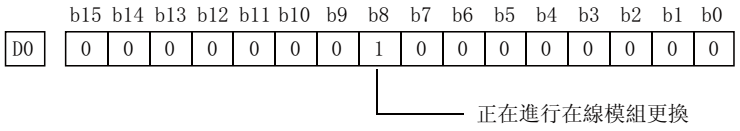
2) A 系列 32 點模組



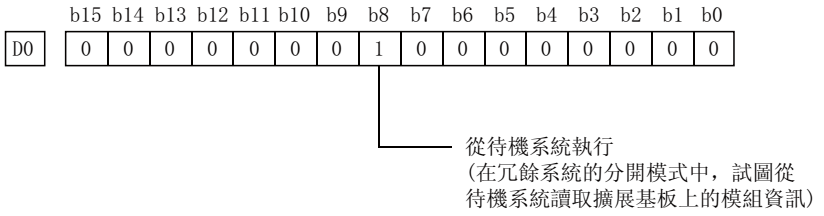
· 48 點模組時 D2 中存儲與 D1 相同的內容，64 點模組時 D2 及 D3 中分別存儲與 D1 相同的內容。

3) 空餘插槽

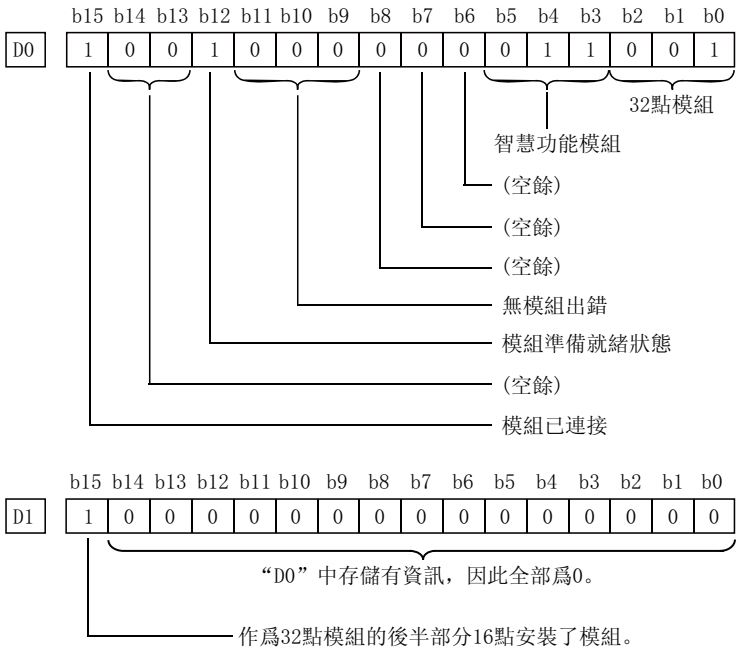
4) 在線模組更換中



5) 在冗餘系統的分開模式中，試圖從待機系統讀取擴展基板上的模組資訊時



6) L 系列 32 點智慧功能模組時



Basic High performance Process Redundant Ver. Universal LCPU

· 在序列號的前 5 位數爲 “ 11043 ” 以後的通用型 QCPU 中可以使用。

7.18.10 TYPERD、TYPERDP

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n	---	○		---				○	---
Ⓓ	---	○		---				---	---

設置資料

設置資料	內容		設置範圍	設置方	資料類型
n	將讀取模組型號的模組的起始輸入輸出編號用 16 相除後的值		0 ～ FF_H、3E0 ～ $3E3_H$ *1	用戶	BIN16 位
Ⓓ	Ⓓ+0	指令執行結果	各軟元件的範圍內	系統	BIN16 位
	Ⓓ+1 ～Ⓓ+9	模組型號			字串

*1: 僅通用型 QCPU。

功　能

(1) 讀取 n 中指定的插槽的模組型號後，存儲到Ⓓ中指定的軟元件後面。

通用型 QCPU 的情況下，物件模組爲下述 6 種。(僅 Q 系列模組)

· CPU 模組
· 輸入模組
· 輸出模組
· 輸入輸出混合模組
· 智慧功能模組
· GOT(匯流排連接時)

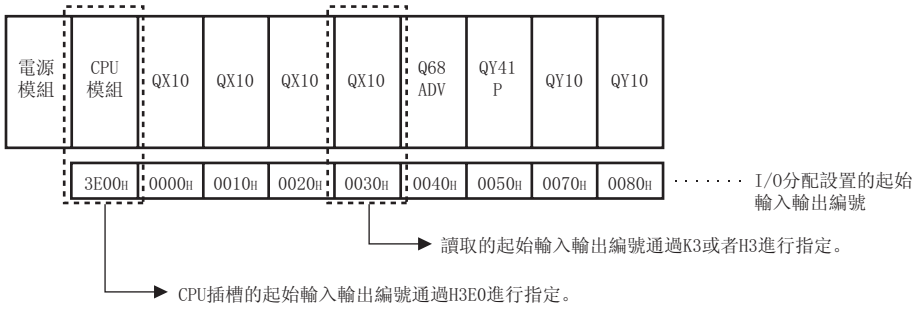
LCPU 的情況下，物件模組爲下述 4 種。

· CPU 模組
· 輸入模組
· 輸出模組
· 智慧功能模組

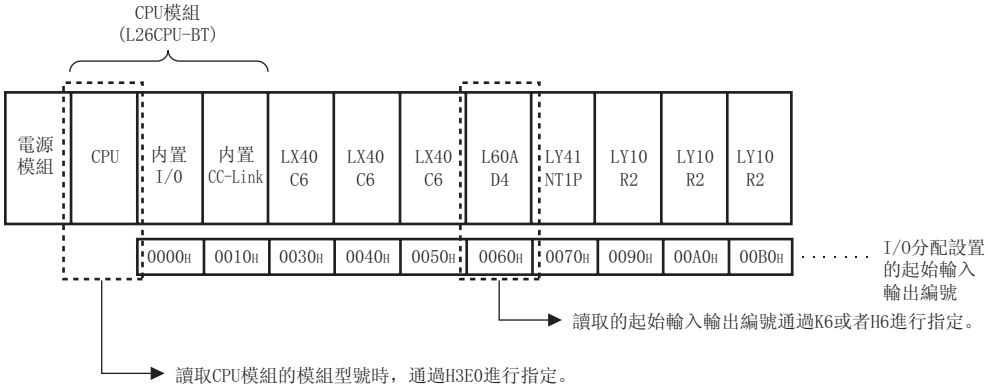
(2) 對於 n，將讀取模組型號的模組的起始輸入輸出編號以 4 位的 16 進制數表示時的高 3 位進行指定。

·指定了 1 個插槽的模組的情況下

通用型 QCPU 的情況下



LCPU



要點

在 LCPU 中，指定了內置 I/O、內置 CC-Link 的起始 I/O 的情況下，CPU 模組的型號將被讀取。

· 指定佔用 2 個插槽的模組的情況下

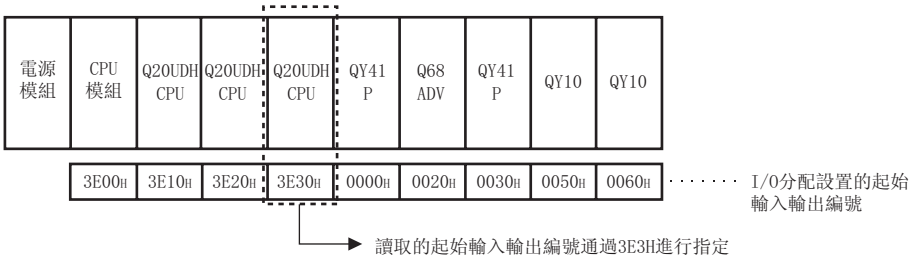
讀取物件模組中指定的起始輸入輸出編號有可能與物件模組的起始輸入輸出編號不相同。

關於指定的起始輸入輸出編號請參閱各模組的手冊。

例 QJ71GP21S-SX 的情況下

指定的起始輸入輸出編號爲，與安裝的模組的起始輸入輸出編號 0010_H 進行了加法運算後的值。

· 多 CPU 系統配置時，讀取 CPU 模組的型號的情況下

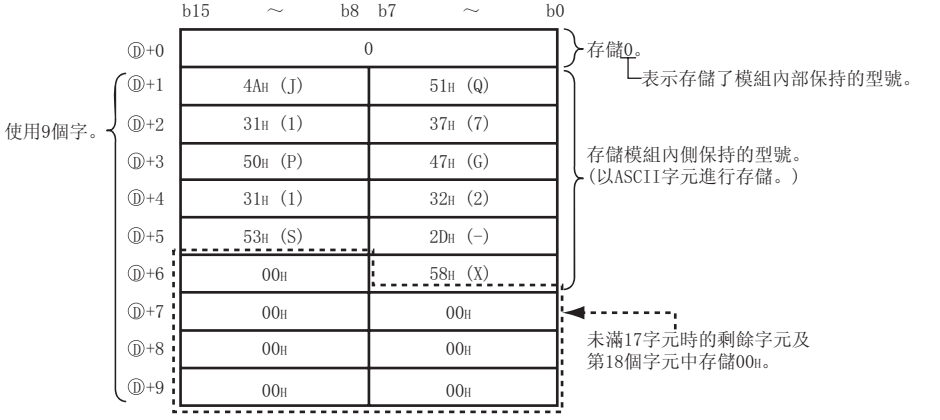


此外，即使指定其他機號 CPU 管理的模組的起始輸入輸出編號也可讀取模組型號。

(3) ⓓ+0 中存儲指令執行結果後，ⓓ+1 ～ⓓ+9 中存儲模組型號。

ⓓ中存儲的值如下所示。

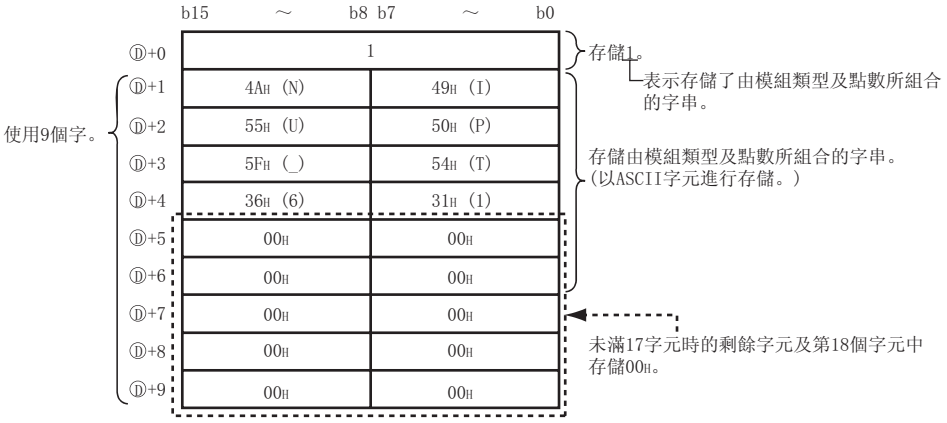
(a) 讀取物件模組內部保持有型號的情況下（例： QJ71GP21-SX）



ⓓ+1 ～ⓓ+9 中存儲的型號示例如下表所示。

物件模組	存儲的型號示例
CPU 模組	Q06UDEHCPU
智慧功能模組	QJ71GP21-SX
GOT	GOT1000

(b) 讀取物件模組內部未保持型號的情況下（例： QX40）



ⓓ+1 ～ⓓ+9 中存儲的字串示例如下表所示。

物件模組	存儲的字串示例
輸入模組 (16 點)	INPUT_16
輸出模組 (32 點)	OUTPUT_32
輸入輸出混合模組 (64 點)	MIXED_64
智慧功能模組 (16 點)	INTELLIGENT_16

[表示模組類型的字串]

・輸入模組： INPUT

・輸出模組： OUTPUT

・輸入輸出混合模組： MIXED

・智慧功能模組 *1: INTELLIGENT

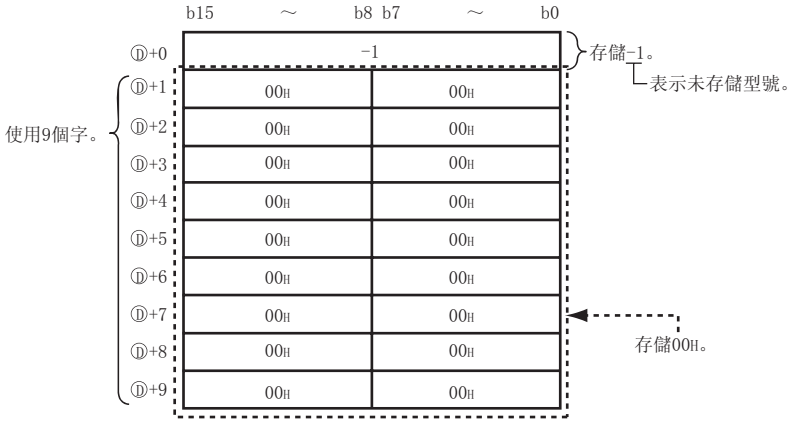
*1: 包括 QI60、GOT。

[表示點數的字串]

· 16 點： _16
· 32 點： _32
· 48 點： _48
· 64 點： _64
· 128 點： _128
· 256 點： _256
· 512 點： _512
· 1024 點： _1024

(c) 其他

· 空餘插槽或者在線模組更換中的情況下
· n 不是模組的起始輸入輸出編號的情況下
· n 的指定在設置範圍以內，但不是可程式控制器參數的 I/O 分配中允許設置的值的情況下



出　錯

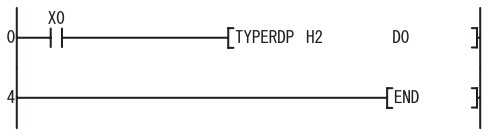
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2110	讀取物件模組由於故障等原因無法通信時。	---	---	---	---	○	○
4101	從ⓓ中指定的軟元件算起的 10 個字超出了允許使用軟元件範圍時。	---	---	---	---	○	○
	n 的指定超出了 0 ～ FF_H、$3E0_H$ ～ $3E3_H$ 的範圍時。	---	---	---	---	○	---
	n 的指定超出了 0 ～ FF_H、$3E0_H$ 的範圍時。	---	---	---	---	---	○

程式示例

(1) 以下爲 X0 變爲 ON 時，將起始輸入輸出編號 0020_H 中安裝的模組的模組型號存儲到 D0 後面的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X0	
1	TYPERD	H2	D0
4	END		

7.18.11 TRACE、TRACER

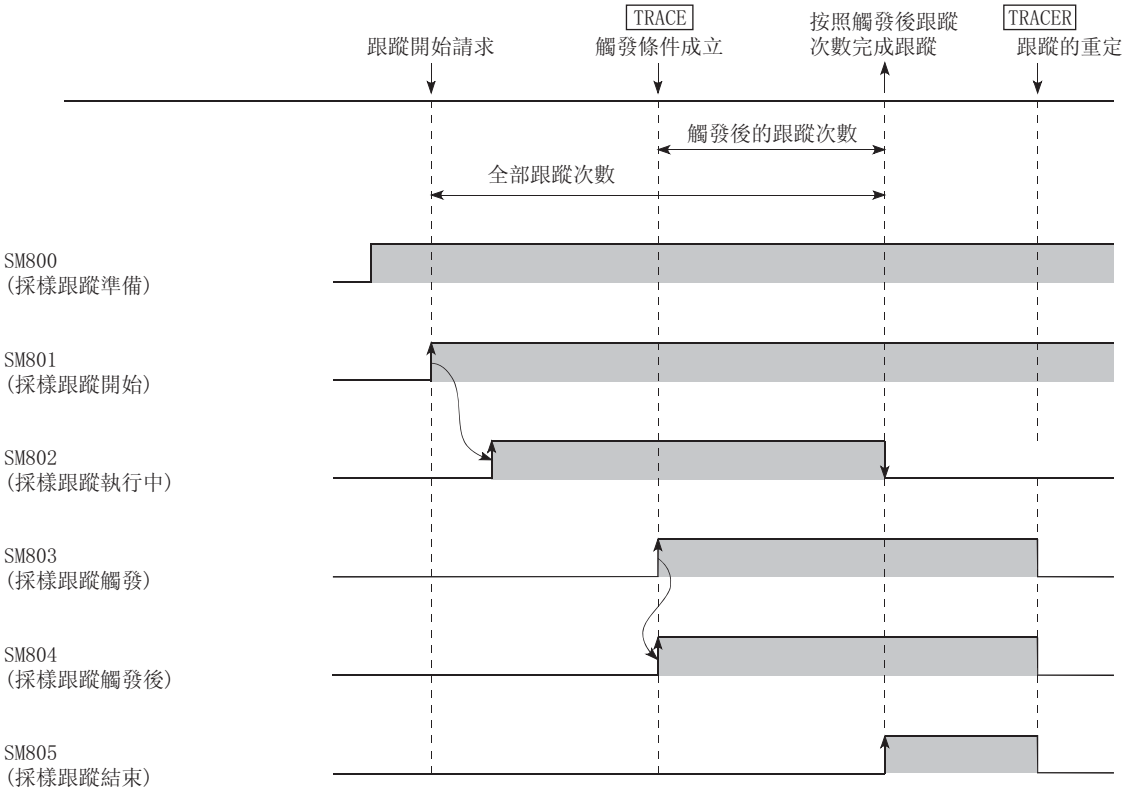
· 在 Q00UJCPU 以外的通用型 QCPU 中可以使用。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

功　能

採樣跟蹤功能是指，對 CPU 模組的軟元件內容進行連續採集的功能。

進行採樣跟蹤時，SM800 爲 ON 時，將 SM801 置爲 ON。



TRACE

(1) TRACE 指令是對採樣跟蹤結果進行鎖存後停止採樣跟蹤的指令。

(2) 在跟蹤執行過程中如果將 SM801 置爲 OFF，採樣將停止。

(3) 執行 TRACE 指令後，如果採樣跟蹤停止，SM805 將變爲 ON。

(4) 如果執行了 1 次 TRACE 指令，從第 2 次開始將被忽略。
如果執行 TRACER 指令，TRACE 指令將再次變爲有效。

TRACER

(1) TRACER 指令是對 TRACE 指令進行重定的指令。
如果執行了 TRACER 指令，TRACE 指令將再次變爲有效狀態。

(2) 如果執行 TRACER 指令，SM803 ～ SM805 將變爲 OFF。

備 註

1. 通過編程工具設置執行採樣跟蹤的軟元件及時機。
   關於採樣跟蹤的詳細內容，請參閱所使用的 CPU 模組的用戶手冊（功能解說 / 程式基礎篇）。
2. 通過編程工具也可執行採樣跟蹤。
   關於通過編程工具進行的採樣跟蹤，請參閱編程工具的操作手冊。

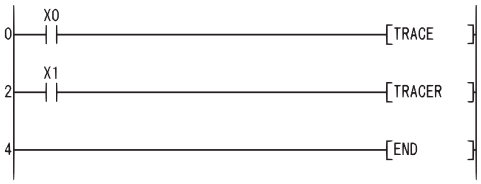
出　錯

(1)　TRACE、TRACER 指令中不存在運算出錯。

程式示例

(1)　以下爲 X0 變爲 ON 時執行 TRACE 指令，X1 變爲 ON 時通過 TRACER 指令對 TRACE 指令進行重定的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0
1	TRACE	
2	LD	X1
3	TRACER	
4	END	

7.18.12 SP.FWRITE

Basic　High performance　Process　Redundant　Universal (Ver.)　LCPU

· 在 Q00UJCPU、Q00UCPU、Q01UCPU 以外的通用型 QCPU 中可以使用。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數		其他
	位	字		位	字			K、H	$	
(S0)	○	○		---				○	---	---
(D0)	---	○		---				---	---	---
(S1)	---	○		---				---	---	---
(S2)	---	○		---				---	○	---
(D1)	△ *1	△ *1		---				---	---	---

*1:　不能使用局部軟元件及各程式中設置的文件寄存器。

<table>
<tr><th>設置資料</th><th colspan="3">內容</th><th>設置範圍</th><th>設置方</th><th>資料類型</th></tr>
<tr><td>U0</td><td colspan="3">虛擬</td><td>---</td><td>---</td><td rowspan="16">BIN16 位元</td></tr>
<tr><td>Ⓢ0</td><td colspan="3">驅動器指定</td><td>2</td><td>用戶</td></tr>
<tr><td rowspan="10">Ⓓ0</td><td colspan="5">存儲控制資料的軟元件的起始編號。
控制資料如下所示。</td></tr>
<tr><td>軟元件</td><td>專案</td><td>內容及設置資料</td><td>設置範圍</td><td>設置方</td></tr>
<tr><td>Ⓓ0</td><td>執行 /
結束類型</td><td>指定執行類型。
0000_H ：二進位寫入
0100_H ：CSV 格式轉換寫入</td><td>0000_H
0100_H</td><td>用戶</td></tr>
<tr><td>Ⓓ0+1</td><td>（未使用）</td><td>系統使用</td><td>---</td><td>系統</td></tr>
<tr><td>Ⓓ0+2</td><td>寫入結果
（資料數）</td><td>放入Ⓢ2中指定資料的實際寫入的資料數。值的單位根據Ⓓ0+7 的資料類型指定。</td><td>---</td><td>系統</td></tr>
<tr><td>Ⓓ0+3</td><td>（未使用）</td><td>---</td><td>---</td><td>---</td></tr>
<tr><td>Ⓓ0+4
Ⓓ0+5</td><td>文件位置</td><td>Ⓓ0中指定了二進位寫入時設置文件位置。
00000000_H：從文件的起始開始
00000001_H ～ $FFFFFFFE_H$
：根據指定位置單位根據資料類型指定。
$FFFFFFFF_H$ ：添加在文件的最後處。
Ⓓ0中指定 CSV 格式寫入時
·對於序列號的前 5 位數爲“01111”以前的高性能型 QCPU，必須設置文件的起始 (0_H)。
·對於序列號的前 5 位數爲“01112”以後的高性能型 QCPU/ 過程 CPU/ 冗餘 CPU/ 通用型 QCPU/LCPU，對文件位置進行設置。
00000000_H ～ $FFFFFFFE_H$
：從文件的起始開始
$FFFFFFFF_H$ ：從文件的最後開始添加</td><td>00000000_H ～
$FFFFFFFF_H$</td><td>用戶</td></tr>
<tr><td>Ⓓ0+6</td><td>列數指定</td><td>Ⓓ0中指定了二進位寫入時必須設置 0。
Ⓓ0中指定了 CSV 格式寫入指定時，對寫入的列數進行設置。
0 ：無列。設置爲 1 行。
0 以外 ：設置爲指定數的列。</td><td>0_H ～ $FFFF_H$
(0 ～ 65535)</td><td>用戶</td></tr>
<tr><td>Ⓓ0+7</td><td>資料類型</td><td>0：字
1：位元組</td><td>0,1</td><td>用戶</td></tr>
<tr><td rowspan="3">Ⓢ1</td><td colspan="5">存儲檔案名的軟元件的起始編號。檔案名如下所示。</td></tr>
<tr><td>軟元件</td><td>專案</td><td>內容及設置資料</td><td>設置範圍</td><td>設置方</td></tr>
<tr><td>Ⓢ1～
Ⓢ1+ □</td><td>檔案名
字串</td><td>指定檔案名的字串。
·省略副檔名時從“.”（點號）開始省略。
·應設置爲 8 個字元 + 點號 +3 個字元以內。
·9 個字元以上時，即使有副檔名也將被忽略，副檔名將變爲“BIN”或“CSV”。</td><td>字串</td><td>用戶</td></tr>
</table>

<table>
<tr><th>設置資料</th><th colspan="3">內容</th><th>設置範圍</th><th>設置方</th><th>資料類型</th></tr>
<tr><td rowspan="4">(S2)</td><td colspan="3">存儲資料的軟元件的起始編號。寫入資料如下所示。</td><td colspan="2"></td><td rowspan="4">BIN16 位元</td></tr>
<tr><td>軟元件</td><td>專案</td><td>內容及設置資料</td><td>設置範圍</td><td>設置方</td></tr>
<tr><td>(S2)</td><td>請求寫入資料數</td><td>指定進行寫入請求的資料數。(字單位)
即使在(D0)+7 中指定位元組時也將換算爲字後以字單位進行設置。</td><td>1 ～ 480
1 ～ 32767*2</td><td rowspan="2">用戶</td></tr>
<tr><td>(S2)+1 ～ (S2)+ □</td><td>寫入資料</td><td>進行寫入請求的資料</td><td>0000_H ～ $FFFF_H$</td></tr>
<tr><td rowspan="4">(D1)</td><td colspan="3">通過處理結束置爲 ON 的位軟元件 (但是異常結束時(D1)+1 也置爲 ON。)</td><td colspan="2"></td><td rowspan="4">位</td></tr>
<tr><td>軟元件</td><td>專案</td><td>內容及設置資料</td><td>設置範圍</td><td>設置方</td></tr>
<tr><td>(D1)</td><td>結束信號</td><td>表示處理結束。
ON: 結束　　OFF: 未結束</td><td>---</td><td rowspan="2">系統</td></tr>
<tr><td>(D1)+1</td><td>異常結束信號</td><td>表示是正常結束還是異常結束。
ON: 異常結束　OFF: 正常結束</td><td>---</td></tr>
</table>

*2 :　變爲通用型 QCPU、LCPU 的範圍。

注意事項

(1) QCPU 的情況下,(S0)(驅動器指定) 中只能設置 ATA 卡驅動器 (2)。
但是，安裝了 Flash 卡的情況下，不能通過 SP.FWRITE 指令進行寫入。
不能設置 SRAM 卡、標準 RAM、標準 ROM 的驅動器。

LCPU 的情況下,(S0)(驅動器指定) 中只能設置 SD 存儲卡的驅動器 (2)。

(2) CSV 設置時寫入的資料爲 10 進制數的值。

例 字元 “ A ” (41_H) → “ 65 ” 將被寫入。
處理範圍： -32768 ～ 32767

(3) 二進位寫入時，字指定中的文件位置的設置範圍爲 00000000_H ～ $7FFFFFFF_H$、$FFFFFFFF_H$。

(4) LCPU 的情況下，SM606(SD 存儲卡強制使用停止指示) 爲 ON 狀態時，不能執行本指令。執行的情況下將變爲無處理。

功　能

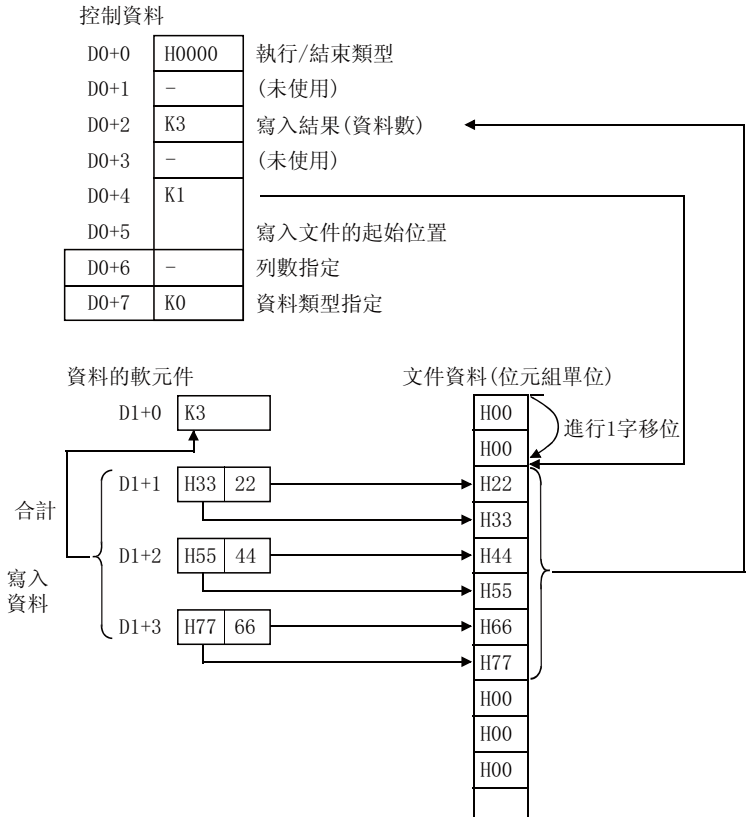
(1) 將指定資料數的資料寫入到指定的文件中。
根據控制資料的執行 / 結束類型，指定是直接寫入二進位資料，還是將二進位資料轉換爲 CSV 格式後進行寫入。
(寫入物件爲 QCPU 的情況下，僅爲 ATA 卡，LCPU 的情況下僅爲 SD 存儲卡。)

(2) 對於處理結束 ((D1)) 的位軟元件，在檢測出本指令的處理結束的 END 指令執行時自動地置爲 ON，通過下一個掃描的 END 指令置爲 OFF。
作爲本指令的執行結束標誌使用。

本指令異常結束時，由於異常結束 ((D1)+1) 元件與處理結束 ((D1)) 軟元件以相同的時機置爲 ON/OFF，因此作爲本指令的異常結束標誌使用。
此外，在指令執行過程中 SM721 置爲 ON。
SM721 爲 ON 的狀態下，不能執行本指令。(執行的情況下將變爲無處理。)

此外，對於執行指令時檢測出的出錯 (SM721 爲 ON 之前)，處理結束 ((D1))、異常結束 ((D1)+1) 及 SM721 不置爲 ON。

(3) 對於資料請求寫入資料數 (Ⓢ②) 及文件位置 (Ⓓ⓪+4、Ⓓ⓪+5) 的處理單位，應設置爲字單位。
二進位寫入時的請求寫入資料數、指定文件位置時的資料的寫入方法如下所示。



(4) 二進位寫入時
(a) 省略了物件文件的副檔名的情況下，副檔名將變爲 “ .BIN ”。
(b) 指定了不存在的文件的情況下，創建相應文件，從起始開始對資料進行添加保存。
此時的新建文件的屬性將被設置爲存檔。
(c) 指定存在的文件時，從存在的文件的起始開始進行保存。
在資料寫入過程中，超出了已有大小的情況下，超出部分的資料將被添加保存。
(d) 指定了大於已有文件容量的文件位置的情況如下所示。
· 對於序列號的前 5 位數爲 “ 01111 ” 以前的高性能型 QCPU，將變爲出錯狀態。
· 對於序列號的前 5 位數爲 “ 01112 ” 以後的高性能型 QCPU/ 過程 CPU/ 冗餘 CPU/ 通用型 QCPU/LCPU，將變爲寫入 0 點而正常結束。
(e) 在對資料進行添加保存的過程中，存儲介質沒有空餘區域的情況下將變爲出錯狀態。
此時，寫入及添加保存成功的部分將保持爲被寫入的狀態不變。
在可添加保存的部分被添加保存後變爲出錯結束。

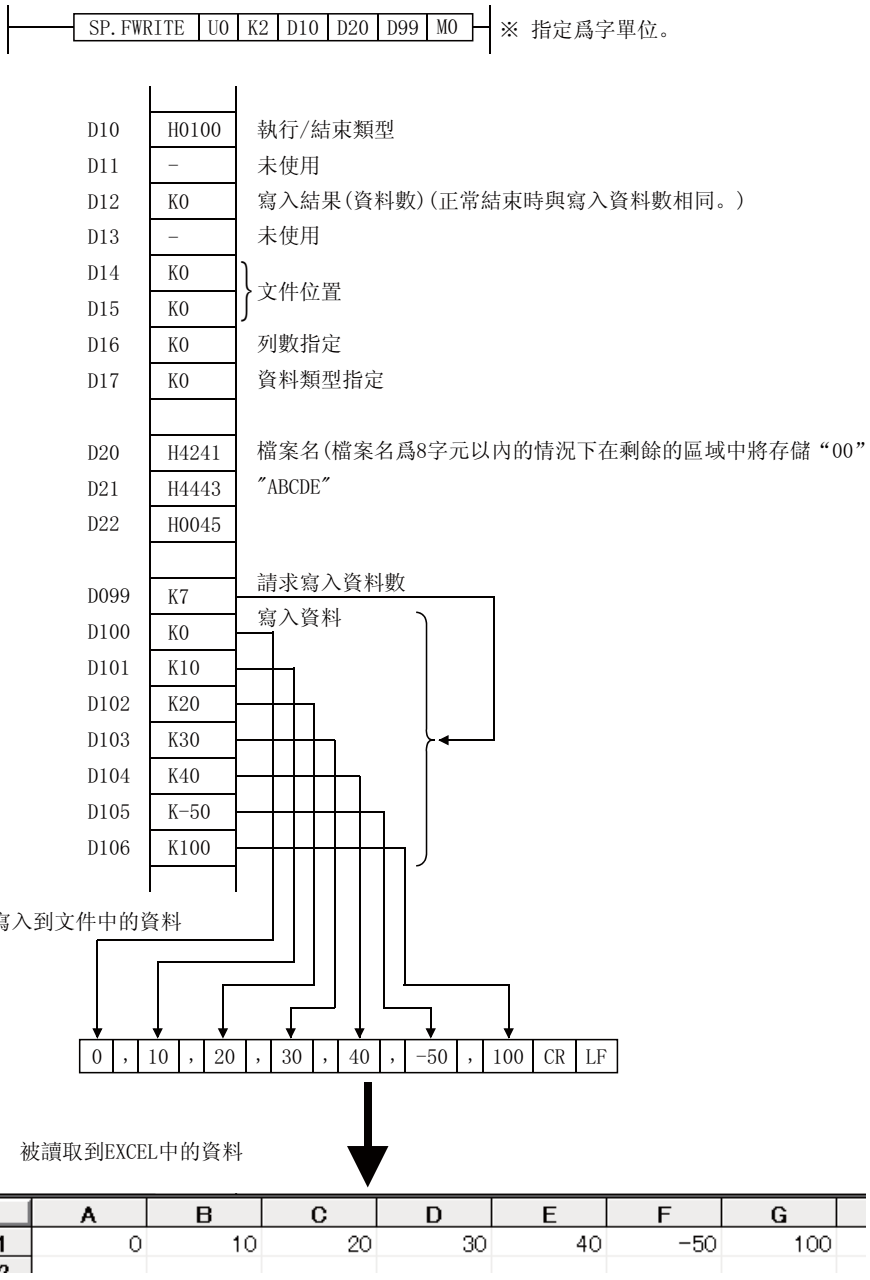
(5) CSV 格式轉換寫入時
(a) 省略副檔名的情況下，副檔名將變爲 “ .CSV ”。
(b) 指定了已存在的文件的情況如下所示。
[序列號的前 5 位數爲 “ 01111 ” 以前的高性能型 QCPU]
將文件內容全部刪除，從起始開始進行資料保存。
[序列號的前 5 位數爲 “ 01112 ” 以後的高性能型 QCPU/ 過程 CPU/ 冗餘 CPU/ 通用型 QCPU/LCPU]
· 在 (Ⓓ⓪+4、Ⓓ⓪+5) 中設置了除 $FFFFFFFF_H$ 以外時，將文件內容全部刪除後從起始開始進行資料保存。
· 在 (Ⓓ⓪+4、Ⓓ⓪+5) 中設置了 $FFFFFFFF_H$ 時，從文件的最後開始進行資料保存。

(c) 指定了不存在的文件的情況下，新建相應文件，從起始開始對資料進行添加保存。
此時新建文件的屬性將被設置爲存檔。

(d) 在對資料進行添加保存的過程中，存儲介質沒有空餘區域的情況下將變爲出錯狀態。
此時，寫入及添加保存成功的部分將保持爲被寫入的狀態不變。
在可添加保存的部分被添加保存後變爲出錯結束。

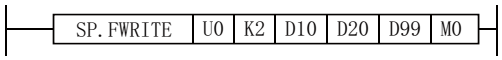
(e) 指定列數爲 0 的情況下，保存爲 1 行的 CSV 格式文件。

例 CSV 格式轉換寫入時指定列數爲 0 的情況下



(f) 在 CSV 格式轉換寫入中，指定列數爲 0 以外的情況下，作爲指定列數的表保存爲 CSV 格式的文件。

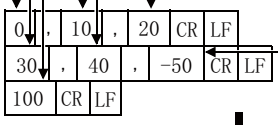
例 CSV 格式轉換寫入時指定列數爲 0 以外的情況下



※ 指定爲字單位。

D10	H0100	執行/結束類型
D11	-	未使用
D12	K0	寫入結果(資料數)(正常結束時與寫入資料數相同。)
D13	-	未使用
D14	K0	文件位置
D15	K0	
D16	K3	列數指定
D17	K0	資料類型指定
D20	H4241	檔案名(檔案名爲8字元以內的情況下在剩餘的區域中將存儲“00”。)
D21	H4443	“ABCD”
D22	H0000	
D099	K7	請求寫入資料數
D100	K0	寫入資料
D101	K10	
D102	K20	
D103	K30	
D104	K40	
D105	K-50	
D106	K100	

被寫入到文件中的資料



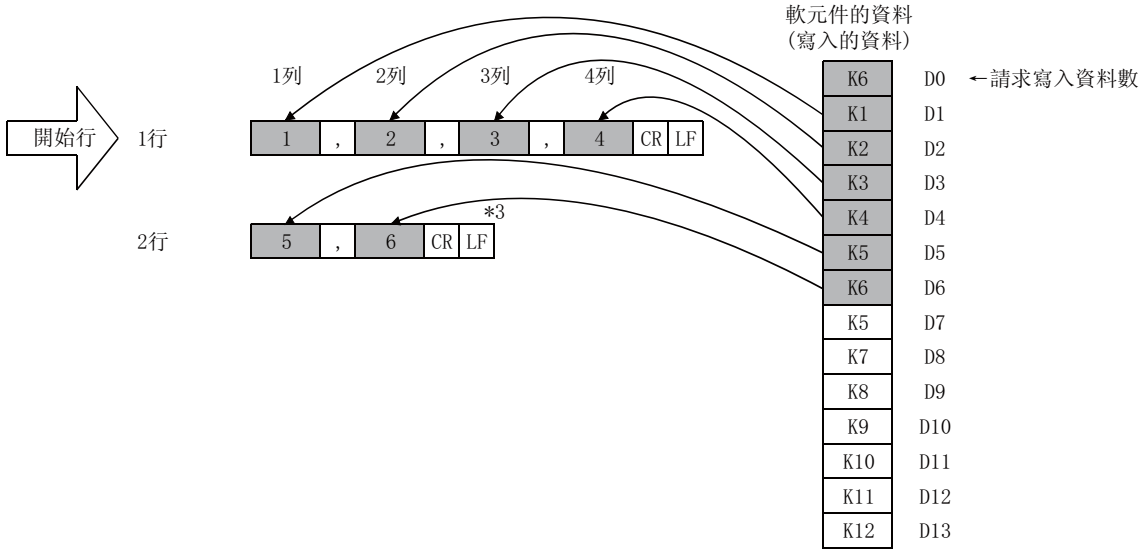
被讀取到EXCEL中的資料

	A	B	C
1	0	10	20
2	30	40	-50
3	100		

(g) 在序列號的前 5 位數爲 “ 01112 ” 以後的高性能型 QCPU/ 過程 CPU/ 冗餘 CPU/ 通用型 QCPU/LCPU 中，進行資料添加時的情況如下所示。

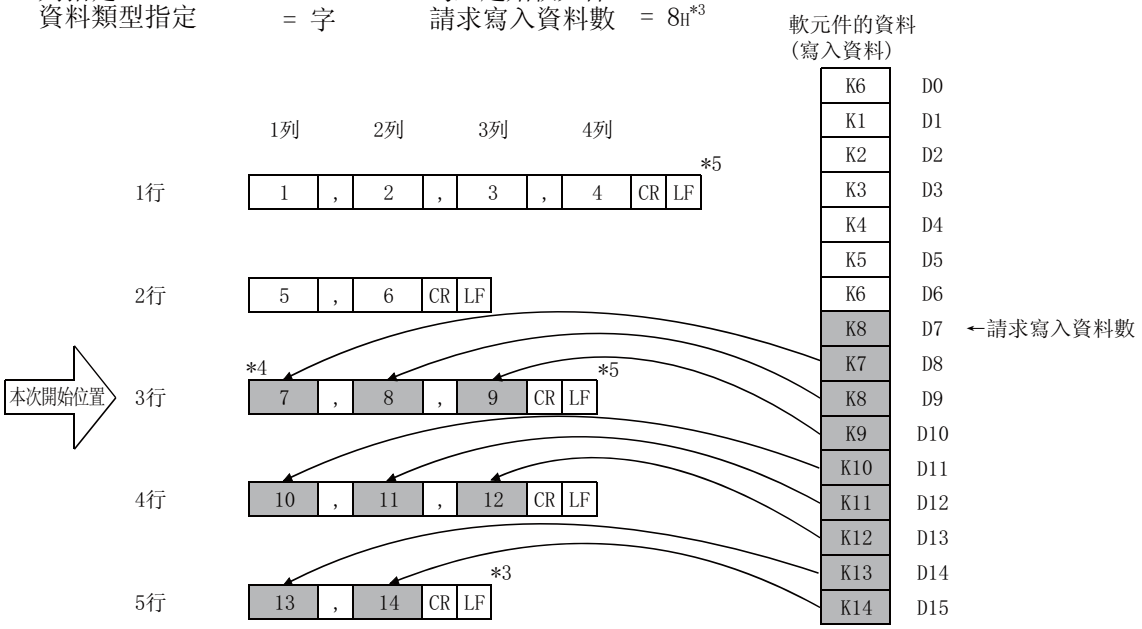
[指定進行寫入的文件。](即使文件已存在也將其刪除後重新創建。)

執行類型 = CSV格式　文件位置 = 0H(新建文件)
列指定 = 4H *3*5　寫入起始軟元件 = D0
資料類型指定 = 字　請求寫入資料數 = 6H *3



[通過添加模式添加到文件的最後處。]

執行類型 = CSV格式　文件位置 = FFFFFFFFH(繼續模式)
列指定 = 3H *3*5　寫入起始軟元件 = D7
資料類型指定 = 字　請求寫入資料數 = 8H*3



*3: “ 寫入點數 ” 不爲 “ 列指定 ” 的整數倍時列將變爲散亂狀態。
*4: 最後資料的後面必須放入換行代碼，因此通常添加模式時從新行的起始開始進行添加。
*5: 添加模式時，如果對上次寫入時的 “ 列指定 ” 進行了更改，列將變爲錯亂狀態。

(h) 不要在中斷程式中執行本指令。
(在中斷程式中執行的情況下將無法保證動作正常。)

(i) 將 CSV 格式文件寫入到 ATA 卡中的情況下，文件大小（合計位元組數）的計算方法如下所示。

合計位元組數 = 最終行以外的合計位元組數 + 最終行的位元組數

（各行的位元組數 = 列數 *1 +1+ 行內各資料值的位元組數的合計 *2 ）

*1: 最終行以外爲指定的列數。對於最終行的列數，根據寫入資料數有可能與指定的列數有所不同，可按下述方法進行計算。
(1) 計算除去最終行後的行數。
除去最終行後的行數 = 請求寫入資料數÷列數（餘數舍去）
(2) 計算最終行的列數。
最終行的列數 = 請求寫入資料數 -（除去最終行後的行數×列數）

*2: 各資料值的位元組數按下述方法計算。

資料值的符號	各資料值的位元組數	位元組數的範圍	示例
正	位數	1 ～ 5(字指定時) 1 ～ 3(位元組指定時)	12345: 5 位元組 67: 2 位元組
負	位數 +1	2 ～ 6(字指定時) 2 ～ 4(位元組指定時)	-12345: 6 位元組 -67: 3 位元組

出 錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

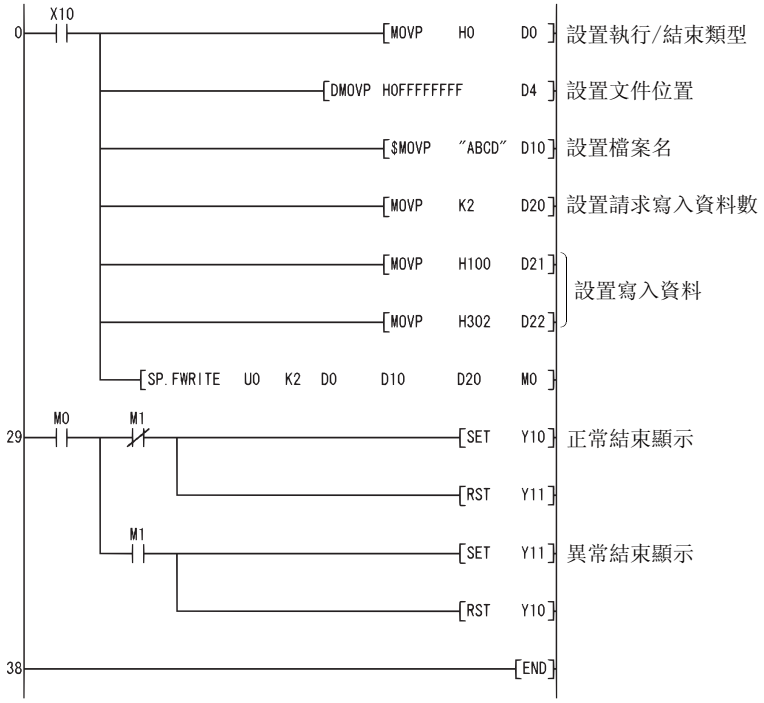
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4004	指定了不能指定的軟元件時。	---	○	○	○	○	○
4100	設置到控制資料 (⑩) 後面的值超出了設置範圍時。 新建文件時，可用空間不足時。 檔案名 (⑤1) 中設置了不能使用的值時。 檔案名 (⑤1) 的屬性爲唯讀時。	---	○	○	○	○	○
	通過驅動器指定 (⑤0) 指定的驅動器不是 ATA 卡時。 ATA 卡的可用空間不足時。 ATA 卡內發生了訪問異常時。	---	○	○	○	○	---
	通過驅動器指定 (⑤0) 指定的驅動器不是 SD 卡時。 SD 卡的可用空間不足時。 SD 卡內發生了訪問異常時。	---	---	---	---	---	○
4101	請求寫入資料數 (⑤2) 中指定的值超出了設置範圍時，或者超出了 (⑤2+1) 後面的軟元件範圍時。	---	○	○	○	○	○
	⑩或者⑪中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 X10 置爲 ON 時，在驅動器 2 中安裝的存儲卡的文件 “ ABCD.BIN ” 中添加 00_H、01_H、02_H、03_H 的 4 位元組的二進位資料的程式。

· 對於控制資料用軟元件，從Ⓓ開始預留 8 點。

[梯形圖模式]



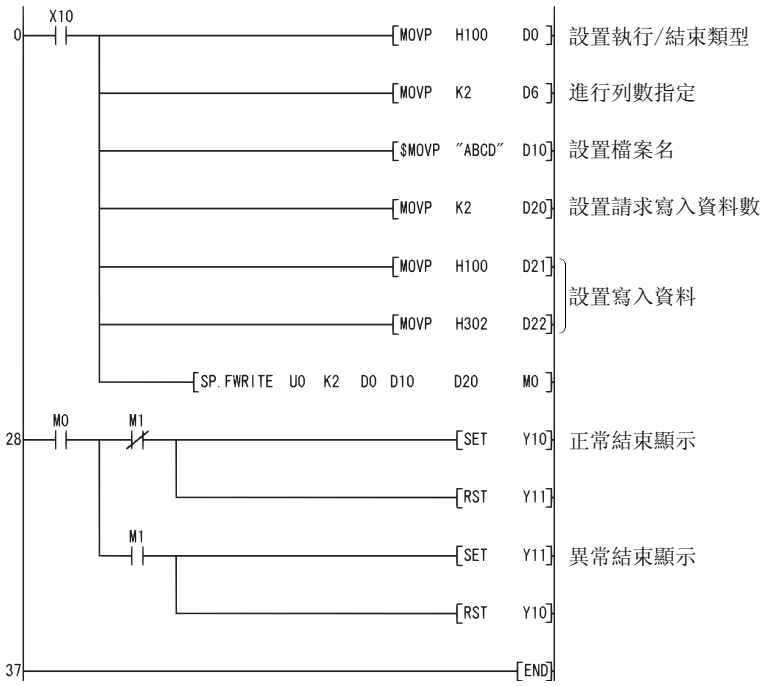
[列表模式]

步	指令	軟元件					
0	LD	X10					
1	MOVP	H0	D0				
3	DMOVP	HOFFFFFFFF	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) 以下爲將 X10 置爲 ON 時，將 00_H、01_H、02_H、03_H 的位元組以 2 列的 CSV 格式文件以檔案名 “ ABCD.CSV ” 創建到驅動器 1 中安裝的存儲卡中的程式。

· 對於控制資料用軟元件，從Ⓓ0 開始預留 8 點。

[梯形圖模式]



[列表模式]

步	指令	軟元件					
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	AND	M1					
31	SET	Y10					
32	RST	Y11					
33	MPP						
34	AND	M1					
35	SET	Y11					
36	RST	Y10					
37	END						

· 寫入的文件如下所示。

0	,	1	,	CR	LF
2	,	3	,	CR	LF

被寫入的文件內容

被讀取到EXCEL中的資料

	A	B
1	0	1
2	2	3

7.18.13 SP.FREAD

Basic High performance Process Redundant Universal Ver. LCPU

· 在 Q00UJCPU、Q00UCPU、Q01UCPU 以外的通用型 QCPU 中可以使用。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數		其他
	位	字		位	字			K、H	$	
(S0)	○	○		---				○	---	---
(D0)	---	○		---				---	---	---
(S1)	---	○		---				---	○	---
(D1)	---	○		---				---	---	---
(D2)	△*1	△*1		---				---	---	---

*1: 不能使用局部軟元件及各程式中設置的文件寄存器。

設置資料	內容			設置範圍	設置方	資料類型
U0	虛擬			---	---	BIN16 位元
(S0)	驅動器指定			2	用戶	
(D0)	存儲控制資料的軟元件的起始編號。 控制資料如下所示。					
	軟元件	專案	內容及設置資料	設置範圍	設置方	
	(D0)	執行 / 結束類型	指定執行類型。 0000_H ：二進位讀取 0100_H ：CSV 格式轉換讀取	0000_H 0100_H	用戶	
	(D0)+1	（未使用）	系統使用	---	系統	
	(D0)+2	請求讀取資料數	指定希望讀取的資料數。 （字單位） (D0)+7 中通過資料類型指定設置爲位元組指定時也將進行字換算並以字單位進行設置。	1～480 1～32767*2	用戶	
	(D0)+3	（未使用）	---	---	---	

*2: 變爲通用型 QCPU、LCPU 的範圍。

設置資料	內容			設置範圍	設置方	資料類型
ⓓ0	ⓓ0 +4 ⓓ0 +5	文件位置	對在ⓓ0中指定了二進位讀取時的文件位置進行設置。 00000000_H：從文件的起始開始 00000001_H ～ $FFFFFFFE_H$ ：根據指定位置單位根據資料類型指定。 $FFFFFFFF_H$：不能設置 在ⓓ0中指定爲 CSV 格式讀取時 ・對於序列號的前 5 位數爲“ 01111 ”以前的高性能型 QCPU，必須設置文件的起始 (0_H) 設。 ・對於序列號的前 5 位數爲“ 01112 ”以後的高性能型 QCPU/ 過程 CPU/ 冗餘 CPU/ 通用型 QCPU/LCPU，設置文件位置 (行)。 00000000_H：從文件的起始開始讀取 00000001_H ～ $FFFFFFFE_H$ ：從指定行開始讀取 $FFFFFFFF_H$ ：從上次的讀取位置開始繼續進行讀取	00000000_H ～ $FFFFFFFF_H$	用戶	BIN16 位元
	ⓓ0 +6	列數指定	ⓓ0中指定了二進位讀取時必須設置爲 0。 ⓓ0中指定了 CSV 格式讀取時，設置進行讀取的列數。 0 ：無列。視爲 1 行。 0 以外 ：視爲指定數的列。	0_H ～ $FFFF_H$ (0 ～ 65535)	用戶	
	ⓓ0 +7	資料類型指定	0：字 1：位元組	0,1	用戶	
ⓢ1	存儲檔案名的軟元件的起始編號。檔案名如下所示。					
	軟元件	專案	內容及設置資料	設置範圍	設置方	
	ⓢ1～ ⓢ1+ □	檔案名 字串	指定檔案名的字串。 ・省略副檔名的情況下須從“ . ”(點號) 開始省略。 ・必須爲 8 字元 + 點號 +3 字元以內。 ・9 字元以上時，即使有副檔名也將被忽略，變爲“ BIN ”或者“ CSV ”。	字串	用戶	
ⓓ1	存儲讀取的資料的軟元件的起始編號。					
	軟元件	專案	內容及設置資料	設置範圍	設置方	
	ⓓ1	讀取結果資料數	對ⓓ0+2 中指定的資料數設置實際讀取的資料數。值的單位根據資料類型指定。	---	系統	
	ⓓ1 +1 ～ ⓓ1 + □	讀取資料	讀取的資料	---	系統	
ⓓ2	通過處理結束置爲 ON 的位軟元件 (但是異常結束時ⓓ2+1 也置爲 ON。)					位
	軟元件	專案	內容及設置資料	設置範圍	設置方	
	ⓓ2	結束信號	表示處理結束。 ON：結束 OFF：未結束	---	系統	
	ⓓ2 +1	異常結束信號	表示是正常結束還是異常結束。 ON：異常結束 OFF：正常結束	---		

注意事項

(1) QCPU 的情況下，ⓢ0 (驅動器指定) 中只能設置 ATA 卡驅動器 (2)。
但是，安裝了 Flash 卡的情況下，不能通過 SP.FREAD 指令進行讀取。
不能設置 SRAM 卡、標準 RAM、標準 ROM 的驅動器。

LCPU 的情況下，ⓢ0 (驅動器指定) 中只能設置 SD 存儲卡的驅動器 (2)。

(2) CSV 設置時寫入的資料爲 10 進制數的值。

例 字元 “ A ” (41_H) → “ 65 ” 將被寫入。

處理範圍： -32768 ～ 32767

(3) 二進位讀取時，字指定中的文件位置的設置範圍爲 00000000_H ～ $7FFFFFFF_H$。

(4) LCPU 的情況下，SM606(SD 存儲卡強制使用停止指示) 爲 ON 狀態時，不能執行本指令。執行的情況下將變爲無處理。

功　能

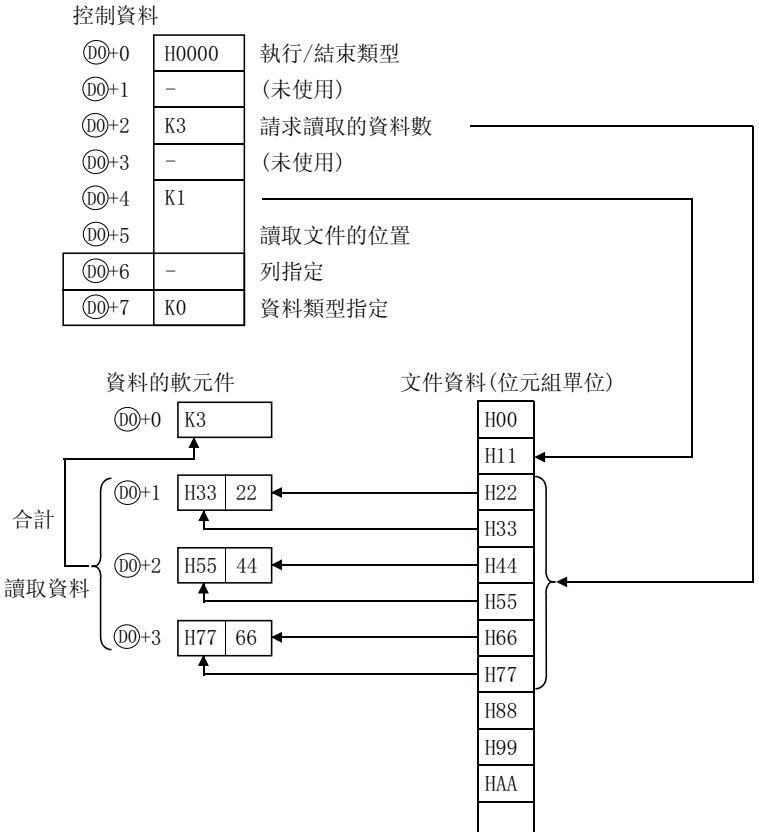
(1) 從指定文件中進行資料讀取。
根據控制資料的執行 / 結束類型，指定是將文件內容以二進位資料進行讀取，還是將文件內容轉換爲 CSV 格式後進行讀取。(讀取物件爲 QCPU 的情況下，僅爲 ATA 卡，讀取物件爲 LCPU 的情況下，僅爲 SD 存儲卡。)

(2) 對於處理結束 (ⓓ2) 的位軟元件，在檢測出本指令的處理結束的 END 指令執行時自動地置爲 ON，通過下一個掃描的 END 指令置爲 OFF。
作爲本指令的執行結束標誌使用。

本指令異常結束時，由於異常結束 (ⓓ2+1) 軟元件與處理結束 (ⓓ2) 的軟元件以相同的時機置爲 ON/OFF，因此作爲本指令的異常結束標誌使用。
此外，在指令執行過程中 SM721 置爲 ON。
SM721 爲 ON 的狀態下，不能執行本指令。(執行的情況下將變爲無處理。)

此外，對於執行指令時檢測出的出錯 (SM721 爲 ON 之前)，處理結束 (ⓓ1)、異常結束 (ⓓ1+1) 及 SM721 不置爲 ON。

(3) 資料請求讀取資料數 (Ⓓ0+2)、文件位置 (Ⓓ0+4、Ⓓ0+5) 以及讀取的結果資料數 (Ⓓ1) 的處理單位設置以字單位被指定。
二進位讀取時的讀取方法如下所示。



(4) 二進位讀取時

(a) 省略物件文件的副檔名的情況下，副檔名將被視爲 “ .BIN ”。

(b) 指定了不存在的文件的情況下，將變爲出錯狀態。

(c) 指定了大於已有文件大小的位置的情況如下所示。

- 對於序列號的前 5 位數爲 “ 01111 ” 以前的高性能型 QCPU，將變爲出錯狀態。
- 對於序列號的前 5 位數爲 “ 01112 ” 以後的高性能型 QCPU/ 過程 CPU/ 冗餘 CPU/ 通用型 QCPU/LCPU，將變爲 0 點讀取而正常結束。

(5) CSV 格式轉換讀取時

(a) 將 CSV 格式文件的要素 (EXCEL 的單格) 以行方向的順序進行讀取，將數值字串轉換爲二進位值後，存儲到軟元件中。

(b) 省略了副檔名的情況下，副檔名將變爲 “ .CSV ”。

(c) 指定了不存在的文件的情況下將變爲出錯狀態。

(d) 從文件的起始開始請求讀取資料數 (Ⓓ0+2) 中指定的要素將被讀取。
在讀取指定資料數的資料之前，達到了文件的最終資料時的情況如下所示。

- 對於序列號的前 5 位數爲 “ 01111 ” 以前的高性能型 QCPU，將變爲出錯狀態。
- 對於序列號的前 5 位數爲 “ 01112 ” 以後的高性能型 QCPU/ 過程 CPU/ 冗餘 CPU/ 通用型 QCPU/LCPU，讀取可以讀取的資料。

(e) 指定列數爲 0 的情況下，將忽略 CSV 格式文件的行分割進行讀取。

例 CSV 格式轉換讀取時指定列數爲 0 的情況下

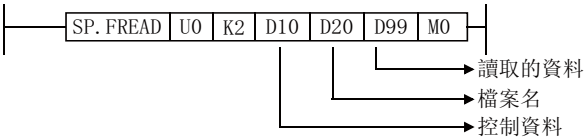
EXCEL中創建的資料

	A	B	C
1	大/小專案		測定值
2	長度	1	3
3	溫度	-21	

以CSV格式保存的資料

大/小專案	,	,	測定值	CR	LF	
長度	,	1	,	3	CR	LF
溫度	,	-21	,	CR	LF	

讀取到軟元件中的資料



控制資料

D10	H0100	執行/結束類型
D11	-	未使用讀取資料數
D12	K9	請求讀取的資料數
D13	-	未使用
D14	K0	讀取文件的位置
D15	K0	
D16	K0	列數指定
D17	K0	資料類型指定
D20	H4241	檔案名
D21	H4443	"ABCDE"
D22	H0045	

讀取的資料

放入讀取的資料數

	D099	K9	結果讀取資料數
讀取資料 大/小專案	D100	K0	“大/小專案”爲非數值資料，因此存儲轉換資料(0)。
讀取資料 ,與, 之間的資料	D101	K0	“”爲非數值資料，因此存儲轉換資料(0)。
讀取資料 測量值	D102	K0	“測量值”爲非數值資料，因此存儲轉換資料(0)。
讀取資料 長度	D103	K0	“長度”爲非數值資料，因此存儲轉換資料(0)。
讀取資料 1	D104	K1	“1”爲數值，因此被轉換爲二進位值。
讀取資料 3	D105	K3	“3”爲數值，因此被轉換爲二進位值。
讀取資料 溫度	D106	K0	“溫度”爲非數值資料，因此存儲轉換資料(0)。
讀取資料 K-21	D107	K-21	“-21” 爲數值，因此被轉換爲二進位值。
讀取資料 ,與CR 之間的資料	D108	K0	“”爲非數值資料，因此存儲轉換資料(0)。

即使各個行的列數不相同的情況下，也將忽略行進行讀取。

要 點

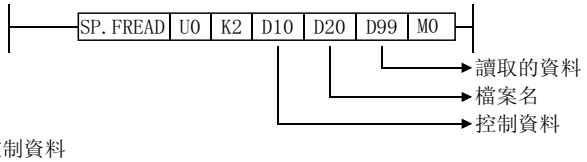
在 EXCEL 中不能創建此類文件。在用戶對 CSV 文件進行了修改的情況下才會發生此類現象。

例 讀取時各個行的列數不相同的情況下

以CSV格式保存的資料

大/小專案	,	,	測定值	CR	LF	
長度	,	1	,	3	CR	LF
溫度	,	-21	,		CR	LF

讀取到軟元件中的資料



控制資料

D10	H0100	執行/結束類型
D11	-	未使用
D12	K7	請求讀取資料數
D13	-	未使用
D14	K0	讀取文件的位置
D15	K0	
D16	K0	列數指定
D17	K0	資料類型指定
D20	H4241	檔案名
D21	H4443	"ABCD"
D22	H0000	

被讀取的資料

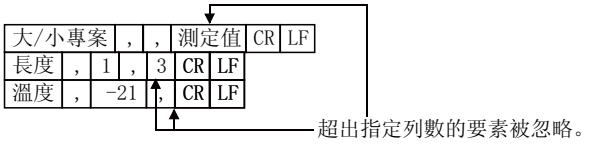
(f) 在 CSV 格式轉換讀取中，指定列數爲 0 以外的情況下，作爲指定列數的表對 CSV 格式文件進行讀取。超出指定列數的要素將被忽略。

例 CSV 格式轉換讀取時指定列數爲 0 以外的情況下

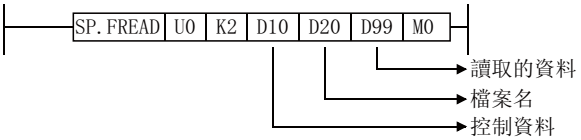
EXCEL中創建的資料

	A	B	C
1	大/小專案		測定值
2	長度	1	3
3	溫度	-21	

以CSV格式保存的資料



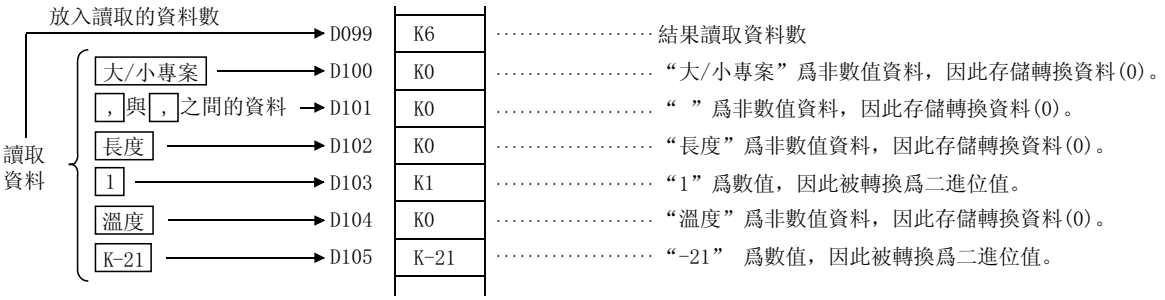
讀取到軟元件中的資料



控制資料

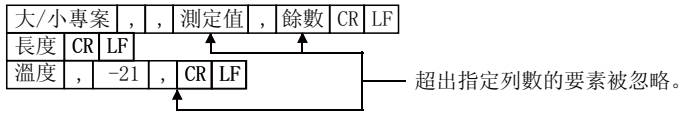
D10	H0100	執行/結束類型
D11	-	未使用
D12	K6	請求讀取資料數
D13	-	未使用
D14	K0	讀取文件的位置
D15	K0	
D16	K2	列數指定
D17	K0	資料類型指定
D20	H4241	檔案名
D21	H4443	"ABCD"
D22	H0000	

讀取的資料

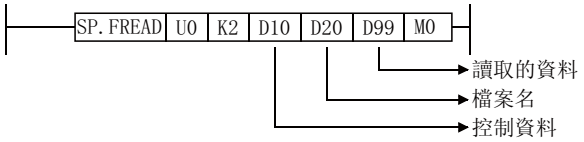


即使各個行的列數不相同的情況下，超出指定列數的要素將被忽略，未達到指定列數的列將被補 0。

例 讀取時各個行的列數不相同的情況下



讀取到軟元件中的資料



控制資料

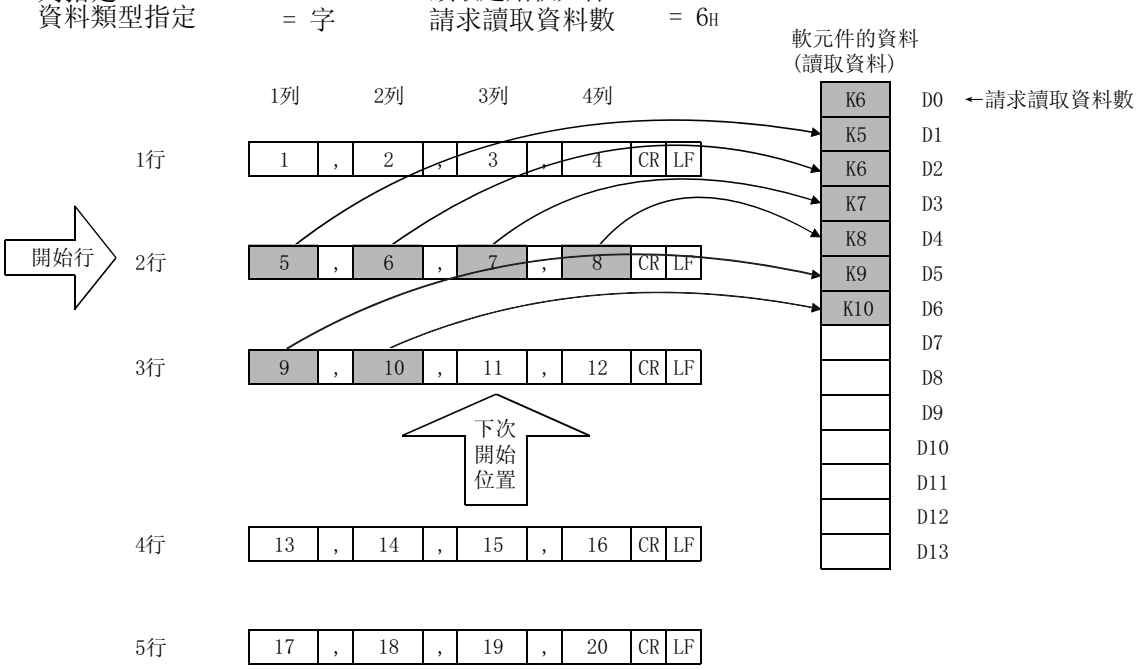
D10	H0100	執行/結束類型
D11	-	未使用
D12	K6	請求讀取資料數
D13	-	未使用
D14	K0	讀取文件的位置
D15	K0	
D16	K2	列數指定
D17	K0	資料類型指定
D20	H4241	檔案名
D21	H4443	"ABCD"
D22	H0000	

讀取的資料

(g) 在序列號的前 5 位數爲 “ 01112 ” 以後的高性能型 QCPU/ 過程 CPU/ 冗餘 CPU/ 通用型 QCPU/LCPU 中，可以分爲多次進行讀取。

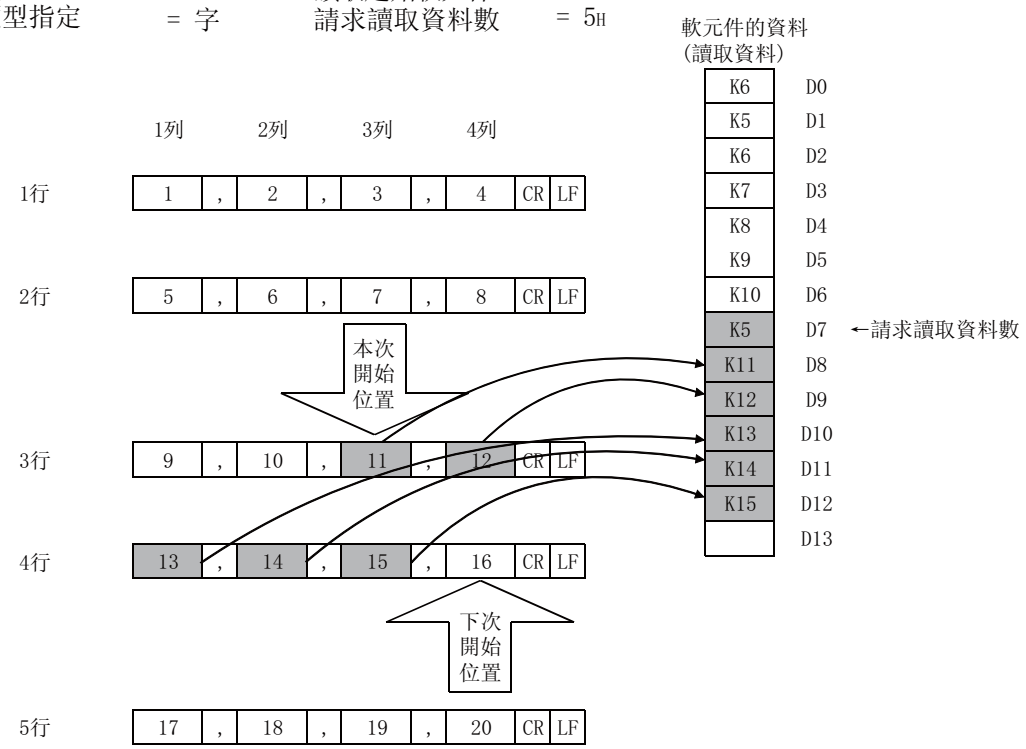
[指定希望開始讀取的行。]

執行類型 = CSV格式　開始行數 = 2H
列指定 = 4H　讀取起始軟元件 = D0
資料類型指定 = 字　請求讀取資料數 = 6H



[在上次繼續模式中，從上次讀取的位置開始繼續進行讀取。]

執行類型 = CSV格式　開始行數 = FFFFFFFFH(繼續模式)
列指定 = 4H　讀取起始軟元件 = D7
資料類型指定 = 字　請求讀取資料數 = 5H



· 以繼續模式進行讀取時，如果對 “ 執行類型 ”、“ 列數指定 ”、“ 資料類型指定 ” 進行了與上次不同的設置，將無法正常地從上次位置開始添加。

· 在通過繼續模式繼續進行資料讀取的過程中，如果執行了其他設置的 SP.FREAD 指令或 SP.FWRITE 指令，將無法正常地從上次位置開始添加。

(h) 在 CSV 格式轉換讀取中，讀取 CSV 格式文件中包含有超出範圍的數值或者非數值要素的情況下，將被轉換爲 0_H。

(i) 在 CSV 格式轉換讀取中，讀取時數值按以下方式進行值的轉換。

CSV 內數值		-32768 ～ -1	0 ～ 32767	32768 ～ 65535
字軟元件	無符號	32768 ～ 65535	0 ～ 32767	32768 ～ 65535
	有符號	-32768 ～ -1	0 ～ 32767	-32768 ～ -1

(j) 不要在中斷程式中執行本指令。
(在中斷程式中執行的情況下，有可能導致誤動作。)

出　錯

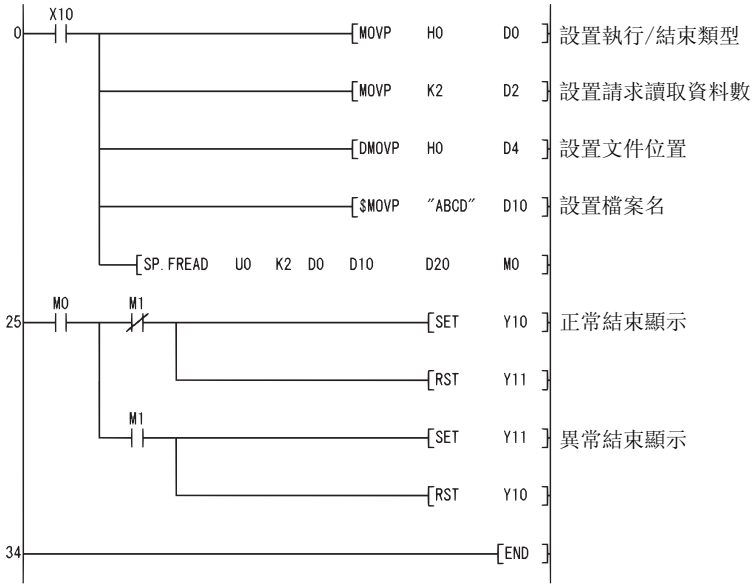
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	檔案名字串 (Ⓢ1) 後面指定的檔案名在指定的驅動器中不存在時。	---	○	○	○	○	○
4004	指定了不能指定的軟元件時。	---	○	○	○	○	○
4100	設置到控制資料 (Ⓓ0) 後面的值超出了設置範圍時。(Ⓓ0+2 除外)	---	○	○	○	○	○
	通過驅動器指定 (Ⓢ0) 指定的驅動器不是 ATA 卡時。 ATA 卡內發生了訪問異常時。	---	○	○	○	○	---
	二進位讀取時文件的資料數小於請求讀取資料數 (Ⓓ0+2) 中指定的容量時。	---	○	---	---	---	---
	通過驅動器指定 (Ⓢ0) 指定的驅動器不是 SD 卡時。 SD 卡內發生了訪問異常時。	---	---	---	---	---	○
4101	讀取的資料數 (Ⓓ0+2) 中指定的值超出了設置範圍時。 讀取的資料容量超出了讀取軟元件的容量時。	---	○	○	○	○	○
	Ⓓ0或者Ⓓ2中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲將 X10 置爲 ON 時，從驅動器 2 中安裝的存儲卡的文件 “ ABCD.BIN ” 的起始開始，以二進位讀取 4 位元組的程式。
· 對於控制資料用軟元件，從 D0 開始預留 8 點。
· 對於讀取用軟元件，從 D20 開始預備留 100 位元組。

[梯形圖模式]



[列表模式]

步	指令	軟元件					
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	AND	M1					
28	SET	Y10					
29	RST	Y11					
30	MPP						
31	AND	M1					
32	SET	Y11					
33	RST	Y10					
34	END						

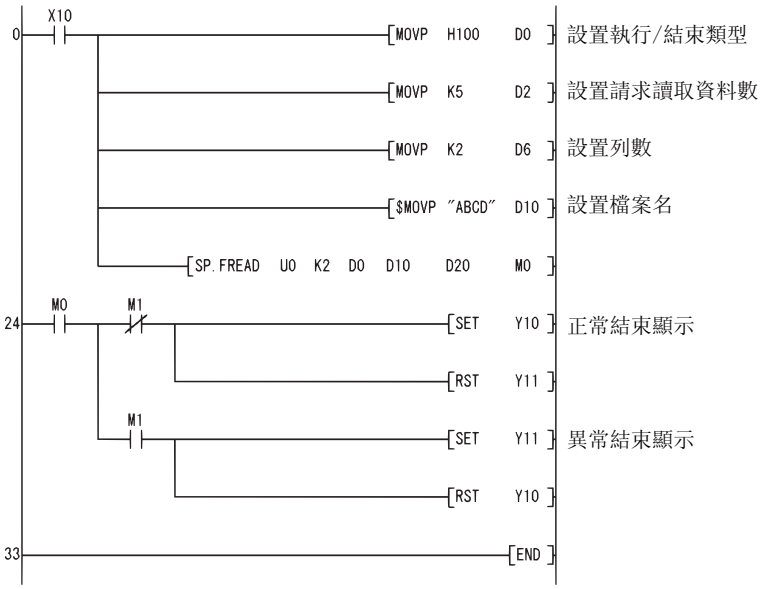
(2) 以下爲將 X10 置爲 ON 時，將檔案名 “ ABCD.CSV ” 以 2 列的 CSV 格式讀取到安裝在驅動器 2 中的存儲卡中的程式。

· 對於控制資料用軟元件，從 D0 開始預留 8 點。

· 對於讀取用軟元件，從 D20 開始預留 100 位元組。

· 讀取 CSV 格式文件中，不存在非數值的要素。

[梯形圖模式]

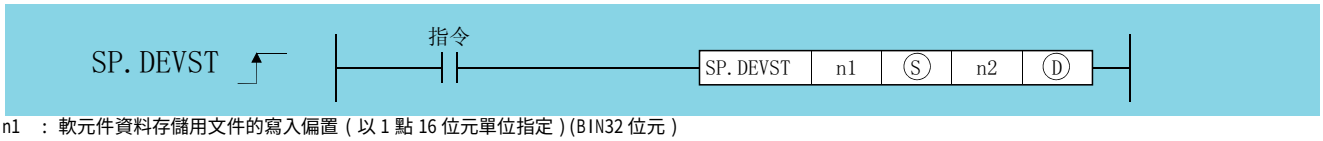


[列表模式]

步	指令	軟元件					
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	AND	M1					
27	SET	Y10					
28	RST	Y11					
29	MPP						
30	AND	M1					
31	SET	Y11					
32	RST	Y10					
33	END						

7.18.14 SP.DEVST

Basic High performance Process Redundant Universal LCPU



n1 ：軟元件資料存儲用文件的寫入偏置（以 1 點 16 位元單位指定）(BIN32 位元）

Ⓢ ：寫入標準 ROM 中的軟元件的起始編號（軟元件名）

n2 ：寫入點數 (BIN16 位）

Ⓓ ： Ⓓ+0: 結束軟元件（位）

： Ⓓ +1: 異常結束軟元件（位）

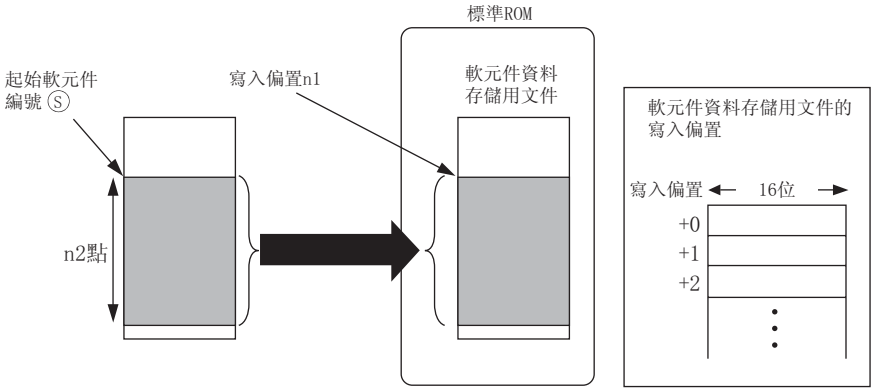
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n1	---	○	○	---				○	---
Ⓢ	---	○	○	---				---	---
n2	---	○	○	---				○	---
Ⓓ	△*1	---	△*1	---				---	---

*1: 不能使用局部軟元件及各程式中設置的文件寄存器。

功 能

(1) 將Ⓢ中指定的軟元件的 n2 中指定的點數的軟元件資料，寫入到標準 ROM 上的軟元件資料存儲用文件的 n1 中指定的寫入偏置中。

n1 是從軟元件資料存儲用文件起始開始的偏置，通過字偏置（每 16 位元 +1 單位）進行指定。



(2) 對於標準 ROM 的軟元件資料寫入位置結束軟元件 (Ⓓ+0)，在檢測出本指令的處理結束的 END 指令執行時自動地置爲 ON，通過下一個掃描的 END 指令置爲 OFF，因此作爲本指令的執行結束標誌使用。

(3) 本指令異常結束時，異常結束軟元件 (Ⓓ+1) 與結束軟元件 (Ⓓ+0) 以相同的時機置爲 ON/OFF，因此作爲本指令的異常結束標誌使用。

(4) 本指令的執行過程中 SM721 置爲 ON。

SM721 已處於 ON 狀態的情況下，不能執行本指令。（執行的情況下將變爲無處理。）

(5) 執行指令時檢測出出錯的情況下，結束軟元件 (Ⓓ+0)、異常結束軟元件 (Ⓓ+1)、SM721 不置爲 ON。

出　錯

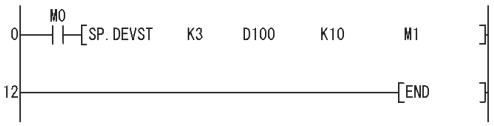
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	在可程式控制器參數的可程式控制器文件設置中，未對軟元件存儲用文件進行設置時。	---	---	---	---	○	○
4100	n1 中指定的寫入偏置超出了軟元件資料存儲用文件的範圍時。 從 n1 中指定的寫入偏置開始的 n2 點超出了軟元件資料存儲用文件的範圍時。	---	---	---	---	○	○
4101	從Ⓢ中指定的軟元件開始的 n2 點的範圍超出了相應軟元件的範圍時。 Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 M0 變爲 ON 時，將從 D100 開始的 10 點寫入到標準 ROM 的軟元件資料存儲用文件中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	M0
1	SP.DEVST	K3 D100 K10 M1
12	END	

注意事項

(1) 寫入到標準 ROM 中值將成爲執行本指令時的值。

(2) 由於 SP.DEVST 指令的執行，標準 ROM 寫入次數指標 (SD687、SD688) 將增加。標準 ROM 寫入次數指標超過了 10 萬次時，將變爲 FLASH ROM ERROR(出錯代碼： 1610) 狀態。

(3) 爲了防止不經意的指令執行導致 ROM 寫入次數的增加，對至標準 ROM 的寫入指令執行次數指定 (SD695) 進行設置，對 1 日內的寫入次數進行限制。
超過了設置的寫入次數 (預設值： 36 次) 時，將變爲 OPERATION ERROR(出錯代碼： 4113) 狀態。

7.18.15 S.DEVLD、SP.DEVLD

Basic　High performance　Process　Redundant　Universal　LCPU

n1 ： 從軟元件資料存儲用文件中的讀取偏置 (以 1 點 16 位元單位指定) (BIN32 位元)

Ⓓ ： 從標準 ROM 中讀取的軟元件的起始編號 (軟元件名)

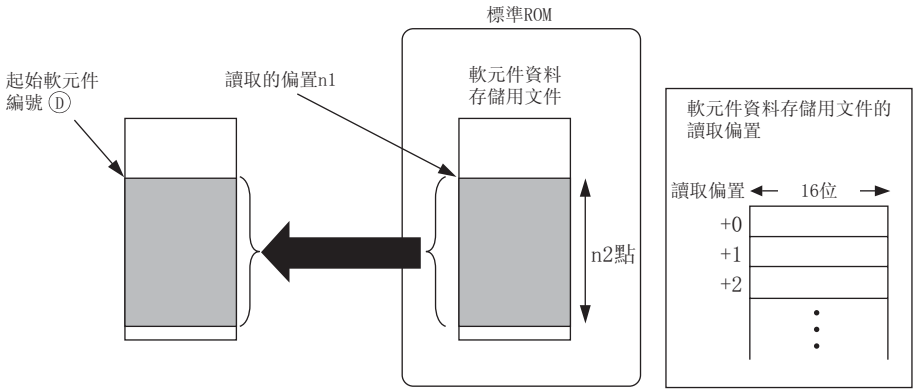
n2 ： 讀取點數 (BIN16 位)

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n1	---	○		---				○	---
Ⓓ	---	○		---				---	---
n2	---	○		---				○	---

功　能

(1) 從標準 ROM 上的軟元件資料存儲用文件的 n1 中指定的讀取偏置中，讀取 n2 中指定的點數的軟元件資料後，存儲到ⓓ中指定的軟元件中。

n1 爲軟元件資料存儲用文件從起始開始的偏置，通過字偏置（以每 16 位元 +1 爲單位）進行指定。



出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	在可程式控制器參數的可程式控制器文件設置中，未進行軟元件存儲用文件的設置時。	---	---	---	---	○	○
4100	n1 中指定的地址超出了標準 ROM 的範圍時。 從 n1 中指定的地址開始的 n2 點超出了標準 ROM 的範圍時。	---	---	---	---	○	○
4101	從ⓓ中指定的軟元件開始的 n2 點的範圍超出了相應軟元件範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 M0 變爲 ON 時，從標準 ROM 的軟元件資料存儲用文件中將 10 點數據讀取到 D100 中的程式。

[梯形圖模式]

[列表模式]

步	指令	軟元件		
0	LD	M0		
1	SP. DEVLD	K3	D100	K10
11	END			

7.18.16 PLOADP

Basic High performance Process Redundant Universal LCPU

Ⓢ ：存儲裝載的程式的驅動器號、檔案名的字串資料或者存儲字串資料的軟元件的起始編號 (BIN16 位)。*1

Ⓓ ：指令結束時使其 1 個掃描 ON 的軟元件 (位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 $	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	△*2	---		---				---	---

*1: 以 “ (驅動器編號): (檔案名) ” 格式進行指定。例 1: MAIN

*2: 局部軟元件不能使用。

功 能

(1) 將存儲卡、標準 ROM 中存儲的程式傳送到程式記憶體 (驅動器 0) 中。
即使傳送的程式未在可程式控制器參數的程式設置中登錄，也在 CPU 模組內部的程式設置中將其設置爲待機類型。
此時可程式控制器參數的程式設置不變化。
(通過 PLOADP 指令傳送程式時，程式記憶體中需要有連續的可用空間。)

(2) 通過 PLOADP 指令添加的程式將被分配爲未使用的程式號中最小的號。
(由用戶指定的情況下，應將程式號存儲到 SD720 中。)
例如，通過 PLOADP 指令添加 “ MAIN6 ” 的情況如下所示。

(a) 在程式號以連續方式設置的情況下，被添加到所設置程式號的最後。
程式號 1 ～ 5 已被設置了程式時，將新增程式號。

程式號	程式名
1	MAIN1
2	MAIN2
3	MAIN3
4	MAIN4
5	MAIN5

通過PLOADP指令添加 “MAIN6” 。

程式號	程式名
1	MAIN1
2	MAIN2
3	MAIN3
4	MAIN4
5	MAIN5
6	MAIN6

←被添加到最後處。

(b) 在存在有多個未放入程式的空閒程式號的情況下，通過 PLOADP 指令指定的程式將被添加到最小的空閒程式號中。
(通過 PUNLOADP 指令刪除了程式時將會產生空閒的程式號。)
在存在有空閒的程式號 2 及 4 的情況下，將被添加到程式號 2 中。

程式號	程式名
1	MAIN1
2	空闲
3	MAIN3
4	空闲
5	MAIN5

通過PLOADP指令添加 “MAIN6” 。

程式號	程式名
1	MAIN1
2	MAIN6
3	MAIN3
4	空闲
5	MAIN5

←被添加到最小的空閒號中。

(3) 驅動器號可指定爲 1、2、4。(不能指定爲驅動器 3)
· 驅動器 1: 存儲卡 (RAM)
· 驅動器 2: 存儲卡 (ROM)
· 驅動器 4: 標準 ROM

(4) 檔案名中無需指定副檔名 (.QPG)。

(5) 在本指令結束的掃描的 END 處理中，將Ⓓ中指定的位軟元件置爲 ON，在下一個 END 處理中置爲 OFF。

(6) PLOADP/PUNLOADP/PSWAPP 指令不能同時執行。
如果同時執行了多個上述指令，後執行的指令將變爲非執行。
使用上述指令時，用戶應設置互鎖以防止上述指令同時被執行。

(7) 不要在中斷程式中執行本指令。
(如果在中斷程式中執行了本指令，有可能導致誤動作。)

(8) 希望執行通過本指令傳送到程式記憶體中的程式時，應通過 PSCAN 指令將其設置爲 “ 掃描執行型 ”。
(參閱 581 頁 7.17.3 項)

(9) 通過本指令傳送的程式的可程式控制器文件設置方法如下所示。

(a) 各程式文件的使用方法
通過本指令傳送的程式的文件寄存器、軟元件初始值、注釋、局部軟元件的使用方法均爲 “ 按照可程式控制器文件設置 ”。
但是，通過本指令進行程式傳送時，如果滿足下述兩個條件將會變爲出錯狀態。
· 在可程式控制器文件設置中設置了 “ 使用局部軟元件 ”
· 程式記憶體的程式個數超過了參數中設置的程式個數時
在通過本指令傳送的程式中使用局部軟元件時，應在參數中登錄一個虛擬的程式文件。然後通過 PUNLOADP 指令將該虛擬程式文件刪除後，通過 PLOADP 指令進行裝載。

(b) I/O 刷新設置
在通過本指令傳送的程式的 I/O 刷新設置中，輸入、輸出均爲 “ 無設置 ”。

(10) “ PLOADP 指令 ” 與 “ RUN 中寫入 ” 的處理不能同時執行。

(a) 在 PLOADP 指令的處理過程中，如果發生了 RUN 中寫入請求，則 RUN 中寫入將被等待。
PLOADP 指令的處理結束後，開始執行 RUN 中寫入。

(b) 如果在進行 RUN 中寫入的過程中執行 PLOADP 指令，則 PLOADP 指令的處理將被等待。
在 RUN 中寫入處理結束後，開始進行 PLOADP 指令的處理。

出 錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

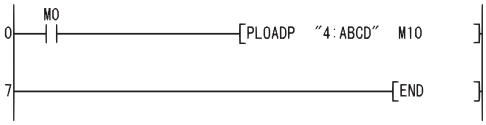
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2401	預留的可用空間小於局部軟元件的文件容量時。	---	○	○	---	---	---
2410	Ⓢ 中指定的驅動器號的檔案名不存在時。 存在有與要裝載的程式文件相同名稱的程式文件時。	---	○	○	---	---	---
2413	驅動器 0 中預留的可用空間小於相應程式的裝載容量時。	---	○	○	---	---	---
4100	Ⓢ 中指定的驅動器號爲禁止指定的驅動器號時。	---	○	○	---	---	---
4101	在程式記憶體中登錄的程式個數已達到下表的文件個數時。 存儲到 SD720 中的程式號已被使用，或者超出了下表的程式號範圍時。	---	○	○	---	---	---

CPU 型號	程式記憶體 (文件個數)	最大程式號
Q02(H)CPU	28 個	28
Q06HCPU	60 個	60
Q12HCPU	124 個	124
Q25HCPU	124 個	124
Q12PHCPU	124 個	124
Q25PHCPU	124 個	124

程式示例

(1) 以下爲 M0 變爲 ON 時，將驅動器 4 中存儲的 “ ABCD.QPG ” 傳送到驅動器 0 中，並置爲待機狀態的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	M0
1	PLOADP	"4:ABCD" M10
7	END	

7.18.17 PUNLOADP

Basic　High performance　Process　Redundant　Universal　LCPU

Ⓢ ： 要卸載的程式檔案名的字串資料或者存儲字串資料的軟元件的起始編號 (BIN16 位)。

Ⓓ ： 指令結束時使其 1 個掃描 ON 的軟元件 (位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 $	其他
Ⓢ	---	○		---				○	---
Ⓓ	△ *1	---		---				---	---

*1: 局部軟元件不能使用。

功　能

(1) 將程式記憶體 (驅動器 0) 中存儲的待機程式從程式記憶體中刪除。
(不能刪除通過 PSCAN 指令設置爲 “ 掃描執行型 ” 的程式或者通過 PLOW 指令設置爲 “ 低速執行型 ” 的程式。)

(2) 通過 PUNLOADP 指令刪除的程式的程式號將變爲 “ 空閒 ”。
在可程式控制器參數的程式設置中設置了程式號 1 ～ 5 時，如果通過本指令刪除了程式號 2 的程式，則程式號將變爲空閒。

程式號	程式名
1	MAIN1
2	MAIN2
3	MAIN3
4	MAIN4
5	MAIN5

通過PLOADP指令刪除 “MAIN2”。

程式號	程式名
1	MAIN1
2	空闲
3	MAIN3
4	MAIN4
5	MAIN5

←程式號2的程式被刪除。

(3) 檔案名中無需指定副檔名 (.QPG)。

(4) 在本指令結束的掃描的 END 處理中，將Ⓓ中指定的位軟元件置爲 ON，在下一個 END 處理中置爲 OFF。

(5) PLOADP/PUNLOADP/PSWAPP 指令不能同時執行。
如果同時執行了多個上述指令，後執行的指令將變爲非執行。
使用上述指令時，用戶應設置互鎖以防止上述指令同時被執行。

(6) 執行 PUNLOADP 指令後，進行可程式控制器的電源 OFF → ON 或者 CPU 模組的重定時的情況如下所示。

(a) 在可程式控制器參數中進行了引導設置的情況下，被進行了引導設置的程式將被傳送到程式記憶體中。
不再執行通過 PUNLOADP 指令刪除的程式的情況下，應將相應程式名從可程式控制器參數的引導設置及程式設置中刪除。

(b) 未在可程式控制器參數中進行引導設置的情況下，將變爲 “ FILE SET ERROR(出錯代碼 2400) ” 的出錯狀態。

1) 不再執行通過 PUNLOADP 指令刪除的程式的情況下，應將相應程式名從可程式控制器參數的引導設置中刪除。

2) 需要再次執行通過 PUNLOADP 指令刪除的程式的情況下，應將相應程式寫入到 CPU 模組中。

(7) 不要在中斷程式中執行本指令。
(如果在中斷程式中執行了本指令將無法保證動作正常。)

(8) 通過本指令從程式記憶體中刪除程式時，應事先通過 PSTOP 指令將其設置爲 “ 待機執行型 ”。
(參閱 579 頁 7.17.1 項)

(9) “ PUNLOADP 指令 ” 與 “ RUN 中寫入 ” 的處理不能同時執行。

(a) 在 PUNLOADP 指令的處理過程中，如果發生了 RUN 中寫入請求，則 RUN 中寫入將被等待。
PUNLOADP 指令的處理結束後，開始執行 RUN 中寫入。

(b) 如果在進行 RUN 中寫入的過程中執行 PUNLOADP 指令，則 PUNLOADP 指令的處理將被等待。
在 RUN 中寫入處理結束後，開始進行 PUNLOADP 指令的處理。

出　錯

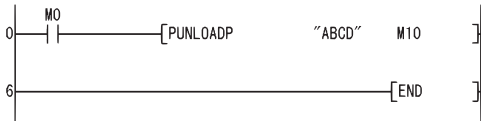
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	Ⓢ 中指定的檔案名不存在時。	---	○	○	---	---	---
4101	Ⓢ 中指定的程式不是待機程式，或者是當前正在執行中的程式時。	---	○	○	---	---	---

程式示例

(1) 以下爲 M0 由 OFF 變爲 ON 時，將存儲在驅動器 0 中的 “ ABCD.QPG ” 從記憶體中刪除的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	M0
1	PUNLOADP	"ABCD" M10
6	END	

7.18.18 PSWAPP

Basic　High performance　Process　Redundant　Universal　LCPU

Ⓢ1 ： 要卸載的程式檔案名的字串資料或者存儲字串資料的軟元件的起始編號 (BIN16 位)。

Ⓢ2 ： 存儲要裝載的程式的驅動器號、檔案名的字串資料或者存儲字串資料的軟元件的起始編號 (BIN16 位)。*1

Ⓓ ： 指令結束時使其 1 個掃描 ON 的軟元件 (位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 $	其他
Ⓢ1	---	○		---				○	---
Ⓢ2	---	○		---				○	---
Ⓓ	△*2	---		---				---	---

*1:　以 “ (驅動器編號) : (檔案名) ” 格式進行指定。例 1: MAIN

*2:　局部軟元件不能使用。

功 能

(1) 將Ⓢ1中指定的程式記憶體 (驅動器 0) 中存儲的待機型程式從程式記憶體中刪除。然後，將Ⓢ2中指定的存儲卡、標准 ROM 中存儲的程式傳送到程式記憶體中，並置爲待機狀態。
(將程式傳送到程式記憶體中時，程式記憶體中需要有連續的可用空間。)
此外，不能刪除通過 PSCAN 指令設置爲 “ 掃描執行型 ” 的程式或者通過 PLOW 指令設置爲 “ 低速執行型 ” 的程式。

(2) 通過 PSWAPP 指令傳送到程式記憶體中的程式的程式號將變爲從程式記憶體中被刪除的程式的程式號。
(即使在從程式記憶體中被刪除的程式的程式號前面存在有空閒的程式號，也不會分配給傳送到程式記憶體中的程式。)
例如，程式號 2 爲 “ 空閒 ” 時，如果通過本指令對程式號 3 的程式進行了置換，傳送到程式記憶體中的程式將被登錄爲程式號 3。

程式號	程式名
1	MAIN1
2	空闲
3	MAIN3
4	MAIN4
5	MAIN5

通過PSWAPP指令將“MAIN3”與“MAIN6”進行交換。

程式號	程式名	
1	MAIN1	
2	空闲	
3	MAIN6	←變爲MAIN6。
4	MAIN4	
5	MAIN5	

(3) 驅動器號可指定爲 1、2、4。(不能指定爲驅動器 3)
· 驅動器 1: 存儲卡 (RAM)
· 驅動器 2: 存儲卡 (ROM)
· 驅動器 4: 標準

(4) 檔案名中無需指定副檔名 (.QPG)。

(5) 在本指令執行結束的掃描的 END 處理中，將Ⓓ中指定的位軟元件置爲 ON，在下一個 END 處理中置爲 OFF。

(6) PLOADP/PUNLOADP/PSWAPP 指令不能同時執行。
如果同時執行了多個上述指令，後執行的指令將變爲非執行。
使用上述指令時，用戶應設置互鎖以防止上述指令同時被執行。

(7) 執行 PSWAPP 指令後，進行可程式控制器的電源 OFF → ON 或者 CPU 模組的重定時的情況如下所示。
(a) 在可程式控制器參數中進行了引導設置的情況下，被進行了引導設置的程式將被傳送到程式記憶體中。
執行通過 PSWAPP 指令置換的程式的情況下，應將可程式控制器參數的引導設置及程式設置變更爲相應的程式名。
(b) 未在可程式控制器參數中進行引導設置的情況下，將變爲 “ FILE SET ERROR(出錯代碼 : 2400) ” 的出錯狀態。
1) 執行通過 PSWAPP 指令置換的程式的情況下，應將可程式控制器參數的程式設置變更爲相應的程式名。
2) 執行可程式控制器參數的程式設置中設置的程式的情況下，應將相應程式再次寫入到 CPU 模組中。

(8) 不要在中斷程式中執行本指令。
(如果在中斷程式中執行了本指令將無法保證動作正常。)

(9) 執行了 PSWAPP 指令的程式的可程式控制器文件設置情況如下所示。
(a) 各程式文件的使用方法
執行了 PSWAPP 指令後的程式的文件寄存器、軟元件初始值、注釋、局部軟元件的使用方法均爲 “ 按照可程式控制器文件設置 ”。

(b) I/O 刷新設置

在執行了 PSWAPP 指令後的程式的 I/O 刷新設置中，輸入、輸出均爲 “ 無設置 ”。

(10) “ PSWAPP 指令 ” 與 “ RUN 中寫入 ” 的處理不能同時執行。

(a) 在 PSWAPP 指令的處理過程中，如果發生了 RUN 中寫入請求，則 RUN 中寫入將被等待。
PSWAPP 指令的處理結束後，開始執行 RUN 中寫入。

(b) 如果在進行 RUN 中寫入的過程中執行 PSWAPP 指令，則 PSWAPP 指令的處理將被等待。
在 RUN 中寫入處理結束後，開始進行 PSWAPP 指令的處理。

出　錯

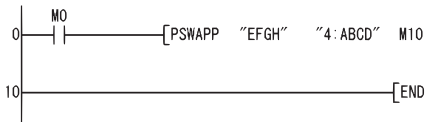
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2410	ⓢ1、ⓢ2中指定的驅動器號的檔案名不存在時。	---	○	○	---	---	---
2413	驅動器 0 中預留的可用空間小於相應程式的裝載容量時。	---	○	○	---	---	---
4100	ⓢ1中指定的驅動器號爲禁止指定的驅動器號時。	---	○	○	---	---	---
4101	ⓢ1中指定的程式不是待機程式，或者是當前正在執行中的程式時。	---	○	○	---	---	---

程式示例

(1) 以下爲 M0 由 OFF 變爲 ON 時，將存儲在驅動器 0 中的 “ EFGH.QPG ” 從記憶體中刪除的同時，將驅動器 4 中存儲的 “ ABCD.QPG ” 傳送到驅動器 0 中的程式。

[梯形圖模式]



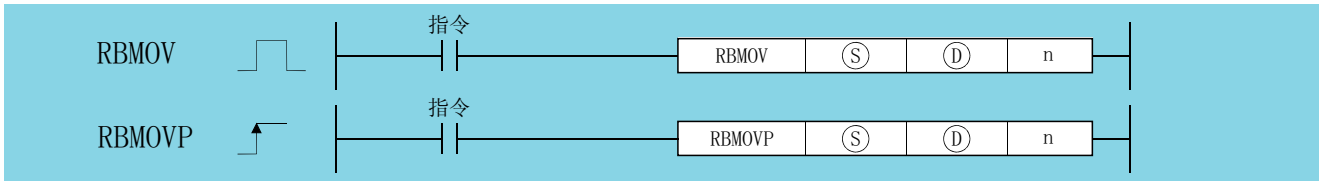
[列表模式]

步	指令	軟元件
0	LD	M0
1	PSWAPP	"EFGH" "4:ABCD" M10
10	END	

7.18.19 RBMOV、RBMOVP

Basic　High performance　Process　Redundant　Universal Ver.　LCPU

· 在 Q00UJCPU 以外的通用型 QCPU 中可以使用。



Ⓢ　：存儲傳送資料的軟元件的起始編號 (BIN16 位)。

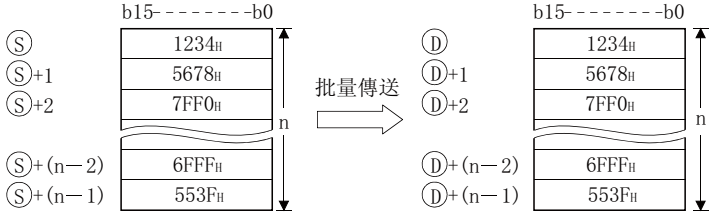
Ⓓ　：傳送目標的軟元件的起始編號 (BIN16 位)。

n　：傳送數 (BIN16 位)。

設置資料	內部軟元件 位	內部軟元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常數 K、H	其他
Ⓢ	○						---		---
Ⓓ	○						---		---
n	○						○		---

功　能

(1) 將從Ⓢ中指定的軟元件開始的 n 點的 16 位元資料，批量地傳送到Ⓓ中指定的軟元件開始的 n 點中。



(2) 即使傳送源與傳送目標的軟元件重復時，也可進行傳送。

向軟元件編號較小的一方傳送時，從Ⓢ開始進行傳送，向軟元件編號較大的一方傳送時，從Ⓢ+(n-1) 開始進行傳送。

但是，在從 R 傳送至 ZR 或者從 ZR 傳送至 R 中時，應按如下所示那樣注意避免 ZR 與 Z 的各傳送範圍重復。

· ZR 的傳送範圍 ((指定的 ZR 起始號) ～ (指定的 ZR 起始號 + 傳送數 -1))

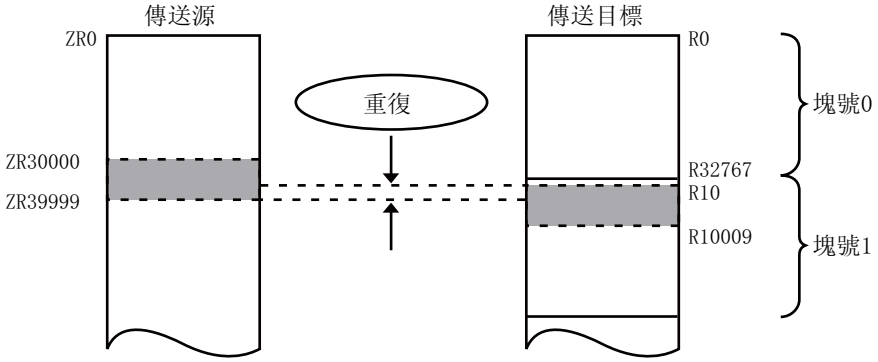
· R 的傳送範圍 ((指定的 R 起始號 + 文件寄存器號× 32768) ～ (指定的 R 起始號 + 文件寄存器號× 32768+ 傳送數 -1))

例　將 10000 點的資料從傳送源 ZR30000 傳送至傳送目的程式號 1 的 R10 中時，傳送範圍重復。

· ZR 的傳送範圍→ (30000) ～ (30000+10000-1) → (30000) ～ (39999)

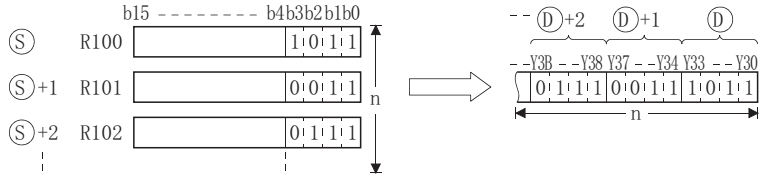
· R 的傳送範圍 → (10+(1 × 32768)) ～ (10+(1 × 32768)+10000-1)→ (32778) ～ (42777)

因此，從 32778 起至 39999 爲止的範圍重復。



(3) 在Ⓢ爲字軟元件而Ⓓ爲位元軟元件的情況下，字軟元件將成爲位軟元件的位數指定中指定的位元數物件。

在Ⓓ中指定了 K1Y30 的情況下，Ⓢ中指定的字軟元件的低 4 位元將成爲物件。



出　錯

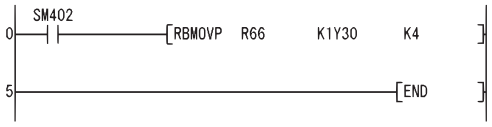
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4101	從Ⓢ、Ⓓ開始的 n 點的軟元件範圍超出了相應軟元件的範圍時。 Ⓢ、Ⓓ中的某一個中未指定文件寄存器時。	---	○	○	○	○	---

程式示例

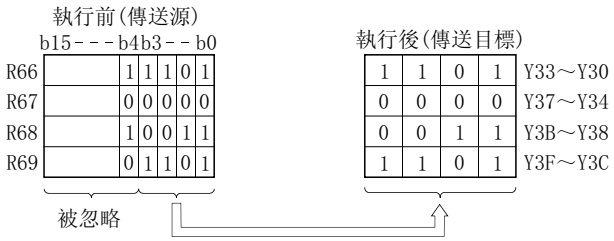
(1)　以下爲將 R66 ～ R99 的低 4 位元的資料以 4 點爲單位從 Y30 輸出到 Y3F 中的程式。

[梯形圖模式]



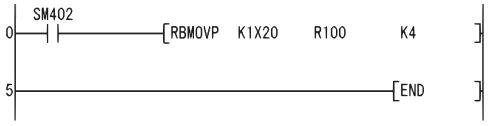
[列表模式]

步	指令	軟元件		
0	LD	SM402		
1	RBMOVP	R66	K1Y30	K4
5	END			



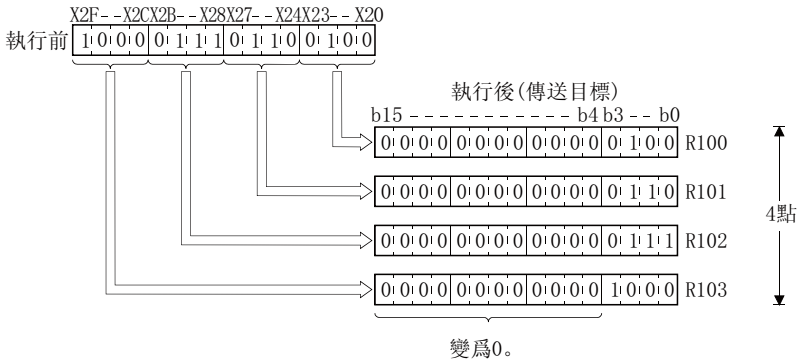
(2)　以下爲將 X20 ～ X2F 的資料以 4 點爲單位輸出到 R100 ～ R103 中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	SM402		
1	RBMOVP	K1X20	R100	K4
5	END			



要 點

本指令在 QnHCPU/QnPHCPU/QnPRHCPU 中對文件寄存器資料進行大批量的批量傳送時有效。在 QnUCPU 的情況下，處理速度與 BMOV 指令相同。

與 BMOV 指令的處理速度相比的情況如下所示。

(1) 文件寄存器→內部軟元件 / 內部軟元件→文件寄存器的情況下。

CPU	指令	文件寄存器存儲物件記憶體	1 字		1000 字		10000 字	
			最小	最大	最小	最大	最小	最大
QnHCPU QnPHCPU QnPRHCPU	RBMOV	標準 RAM	20.0 μs		91.0 μs		775.0 μs	
		SRAM 卡	22.0 μs		305.0 μs		2900.0 μs	
		Flash 卡 *1	22.5 μs		405.0 μs		3950.0 μs	
	BMOV	標準 RAM	7.5 μs		76.2 μs		720.0 μs	
		SRAM 卡	8.0 μs		384.0 μs		3900.0 μs	
		Flash 卡 *1			418.0 μs		4250.0 μs	
QnCPU	RBMOV	標準 RAM	45.5 μs		215.0 μs		1850.0 μs	
		SRAM 卡	49.5 μs		540.0 μs		5150.0 μs	
		Flash 卡 *1						
	BMOV	標準 RAM	17.5 μs		177.0 μs		1700.0 μs	
		SRAM 卡	18.0 μs		500.0 μs		5050.0 μs	
		Flash 卡 *1			572.0 μs		5800.0 μs	
Q00UCPU Q01UCPU	RBMOV	標準 RAM	12.2 μs	34.9 μs	121.5 μs	145.1 μs	1111.5 μs	1135.1 μs
		SRAM 卡 *2	---	---	---	---	---	---
		Flash 卡 *2	---	---	---	---	---	---
	BMOV	標準 RAM	7.3 μs	13.8 μs	116.5 μs	124.2 μs	1106.5 μs	1114.2 μs
		SRAM 卡 *2	---	---	---	---	---	---
		Flash 卡 *2	---	---	---	---	---	---
Q02UCPU	RBMOV	標準 RAM	9.4 μs	31.3 μs	118.5 μs	141.3 μs	1108.5 μs	1131.3 μs
		SRAM 卡	9.4 μs	31.4 μs	178.5 μs	201.3 μs	1708.5 μs	1731.3 μs
		Flash 卡 *1	9.4 μs	32.1 μs	278.5 μs	301.3 μs	2708.5 μs	2731.3 μs
	BMOV	標準 RAM	5 μs	11.6 μs	114.5 μs	122.3 μs	1104.5 μs	1112.3 μs
		SRAM 卡	5.1 μs	11.7 μs	174.5 μs	182.3 μs	1704.5 μs	1712.3 μs
		Flash 卡 *1	5 μs	11.6 μs	274.5 μs	282.3 μs	2704.5 μs	2712.3 μs
Q03UD(E)CPU	RBMOV	標準 RAM	11.3 μs	16.8 μs	120.7 μs	127.1 μs	1110.7 μs	1117.1 μs
		SRAM 卡	11.2 μs	16.7 μs	180.7 μs	187.1 μs	1710.7 μs	1717.1 μs
		Flash 卡 *1	11.3 μs	16.8 μs	280.7 μs	287.1 μs	2710.7 μs	2717.1 μs
	BMOV	標準 RAM	4.8 μs	6.6 μs	114.7 μs	117.1 μs	1104.7 μs	1107.1 μs
		SRAM 卡	4.8 μs	6.6 μs	174.7 μs	177.1 μs	1704.7 μs	1707.1 μs
		Flash 卡 *1	4.8 μs	6.5 μs	274.7 μs	277.1 μs	2704.7 μs	2707.1 μs
Q04UD(E)HCPU Q06UD(E)HCPU Q10UD(E)HCPU Q13UD(E)HCPU Q20UD(E)HCPU Q26UD(E)HCPU Q50UDEHCPU Q100UDEHCPU	RBMOV	標準 RAM	9.2 μs	15.1 μs	61.0 μs	68.6 μs	531.0 μs	538.6 μs
		SRAM 卡	9.4 μs	15.6 μs	165.0 μs	172.6 μs	1576.0 μs	1583.6 μs
		Flash 卡 *1	9.4 μs	15.7 μs	260.0 μs	267.6 μs	2526.0 μs	2533.6 μs
	BMOV	標準 RAM	4.1 μs	5.6 μs	56.0 μs	58.6 μs	526.0 μs	528.6 μs
		SRAM 卡	4.5 μs	6.1 μs	160.0 μs	162.6 μs	1571.0 μs	1573.6 μs
		Flash 卡 *1	4.3 μs	6.2 μs	255.0 μs	257.6 μs	2521.0 μs	2523.6 μs

*1: 將文件寄存器存儲到 Flash 卡中時，內部軟元件→文件寄存器的模式將無處理。

*2: 在 Q00UCPU、Q01UCPU 中，不能使用存儲卡。

(2) 文件寄存器→文件寄存器的情況下

CPU	指令	文件寄存器存儲物件記憶體	1字		1000字		10000字	
			最小	最大	最小	最大	最小	最大
QnHCPU QnPHCPU QnPRHCPU	RBMOV	標準 RAM	20.0 μs		91.0 μs		775.0 μs	
		SRAM 卡	22.5 μs		545.0 μs		5300.0 μs	
	BMOV	標準 RAM	7.5 μs		77.0 μs		720.0 μs	
		SRAM 卡	8.5 μs		692.0 μs		7050.0 μs	
QnCPU	RBMOV	標準 RAM	45.5 μs		215.0 μs		1850.0 μs	
		SRAM 卡	50.0 μs		870.0 μs		8350.0 μs	
	BMOV	標準 RAM	17.5 μs		179.0 μs		1700.0 μs	
		SRAM 卡	18.5 μs		839.0 μs		8600.0 μs	
Q00UCPU Q01UCPU	RBMOV	標準 RAM	12.6 μs	35.3 μs	232.5 μs	256.1 μs	2211.5 μs	2235.1 μs
		SRAM 卡 *1	---	---	---	---	---	---
	BMOV	標準 RAM	7.7 μs	14.2 μs	227.5 μs	234.2 μs	2206.5 μs	2214.2 μs
		SRAM 卡 *1	---	---	---	---	---	---
Q02UCPU	RBMOV	標準 RAM	9.6 μs	31.5 μs	228.5 μs	252.3 μs	2208.5 μs	2231.3 μs
		SRAM 卡	9.6 μs	31.5 μs	378.5 μs	401.3 μs	3708.5 μs	3731.3 μs
	BMOV	標準 RAM	5.2 μs	11.8 μs	224.5 μs	232.3 μs	2204.5 μs	2212.3 μs
		SRAM 卡	5.2 μs	11.8 μs	374.5 μs	382.3 μs	3704.5 μs	3712.3 μs
Q03UD(E)CPU	RBMOV	標準 RAM	11.2 μs	16.7 μs	230.7 μs	237.1 μs	2210.7 μs	2217.1 μs
		SRAM 卡	11.6 μs	16.7 μs	380.7 μs	387.1 μs	3710. μs	3717.1 μs
	BMOV	標準 RAM	4.9 μs	6.7 μs	224.7 μs	227.1 μs	2204.7 μs	2207.1 μs
		SRAM 卡	5.2 μs	6.7 μs	374.7 μs	377.1 μs	3704.7 μs	3707.1 μs
Q04UD(E)HCPU Q06UD(E)HCPU Q10UD(E)HCPU Q13UD(E)HCPU Q20UD(E)HCPU Q26UD(E)HCPU Q50UDEHCPU Q100UDEHCPU	RBMOV	標準 RAM	9.3 μs	15.5 μs	118.0 μs	124.6 μs	1102.0 μs	1107.6 μs
		SRAM 卡	9.7 μs	15.5 μs	365.0 μs	371.6 μs	3571.0 μs	3578.6 μs
	BMOV	標準 RAM	4.3 μs	6.2 μs	113.0 μs	115.6 μs	1096.0 μs	1098.6 μs
		SRAM 卡	4.5 μs	6.1 μs	360.0 μs	362.6 μs	3566.0 μs	3568.6 μs

*1: 在 Q00UCPU、Q01UCPU 中，不能使用存儲卡。

7.18.20 UMSG

Basic High performance Process Redundant Universal LCPU

Ⓢ : 顯示模組中顯示的字串或者存儲顯示字串的軟元件的起始編號（字串）

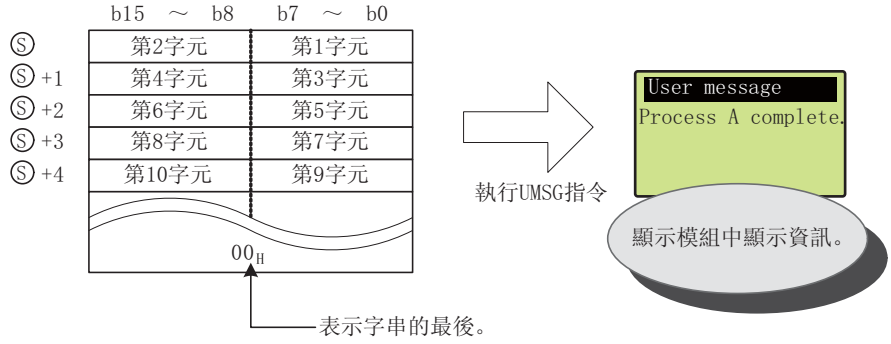
設置資料	內部軟元件		R、ZR	間接指令	J□\□		U□\G□	Zn	常數		其他
	位	字			位	字			K、H	字串	
Ⓢ	---	○		○	---					△*1	---

*1: 僅字串可以使用。

功　能

(1) 將Ⓢ中指定的字串資料作爲至顯示模組的用戶資訊進行顯示。

顯示Ⓢ中直接指定的字串（用“ ”（雙引號）圍住進行指定），或者顯示Ⓢ中指定的軟元件編號開始至存儲了“ 00_H ”的軟元件編號爲止的字串。



(2) 顯示模組可顯示的字串最多爲半形 128 字元（全形 64 字元）。

(3) 在 UMSG 指令的上升沿時將顯示用戶資訊。
指令爲 ON 的狀態下，如果對字串進行更改，顯示模組中將顯示更改後的用戶資訊。

(4) 在 END 處理時進行 UMSG 指令中指定的字串的顯示。執行了多個 UMSG 指令的情況下，END 之前執行的 UMSG 指令將有效。
執行了多個程式時，最後執行的 UMSG 指令將有效。

(5) 在未安裝顯示模組的狀態下執行了指令時，將變爲無處理。

(6) 在用戶資訊的顯示過程中，如果按壓顯示模組的“ ESC ”，顯示中的資訊將消失。
再次顯示資訊時，通過顯示模組功能表畫面執行“ 用戶資訊 ”。

(7) 指令的引數中指定了 NULL 代碼（00_H）的情況下，在資訊顯示狀態下顯示的資訊將消失。
在指令的引數中指定 NULL 代碼（00_H）的方法如下所示。

關於顯示模組的詳細內容，請參閱 MELSEC-L CPU 模組用戶手冊（功能解說 / 程式基礎篇）。

出　錯

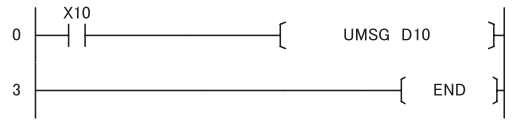
(1)　在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	Ⓢ的字串的字元數超過了半形 128 字元 (全形 64 字元) 時。	---	---	---	---	---	○
4101	Ⓢ中指定的軟元件編號以後的相應軟元件的範圍內不存在 NULL 代碼 (00_H) 時。	---	---	---	---	---	○

程式示例

(1)　以下爲將 X10 置爲 ON 時，將 D10 後面存儲的字串顯示到顯示模組中的程式。

[梯形圖模式]



[列表模式]

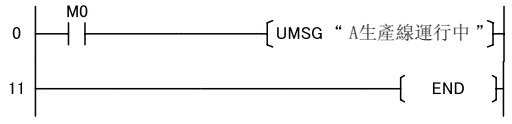
步	指令	軟元件
0	LD	X10
1	UMSG	D10
3	END	

[動作]

	b15 ～ b8	b7 ～ b0	
D10	81_H	83_H	A
D11	93_H	83_H	生
D12	65_H	83_H	産
D13	69_H	83_H	線
D14	93_H	83_H	運
D15	58_H	83_H	行
D16	86_H	92_H	中
D17	0000_H		

(2)　以下爲將 M0 置爲 ON 時，在顯示模組中顯示 “ A 生產線運行中 ” 的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	M0
1	UMSG	“A生產線運行中”
11	END	

[動作]

b15 ～ b8	b7 ～ b0	
60_H	82_H	A
89_H	83_H	生
43_H	83_H	産
93_H	83_H	線
40_H	81_H	運
$5E_H$	89_H	行
$5D_H$	93_H	中
86_H	92_H	
0000_H		

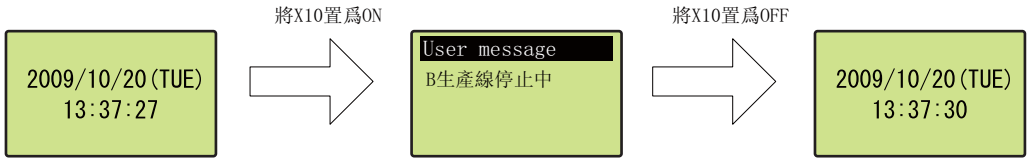
(3) 以下爲將 X10 置爲 ON 時，在顯示模組中顯示 “ B 生產線停止中 ”，將 X10 置爲 OFF 時資訊消失的程式。

[梯形圖模式]

[列表模式]

步	指令	軟元件
0	LD	X10
1	UMSG	“B生產線停止中”
11	LDI	X10
12	UMSG	""
14	END	

[動作]



第 8 章 資料鏈接指令

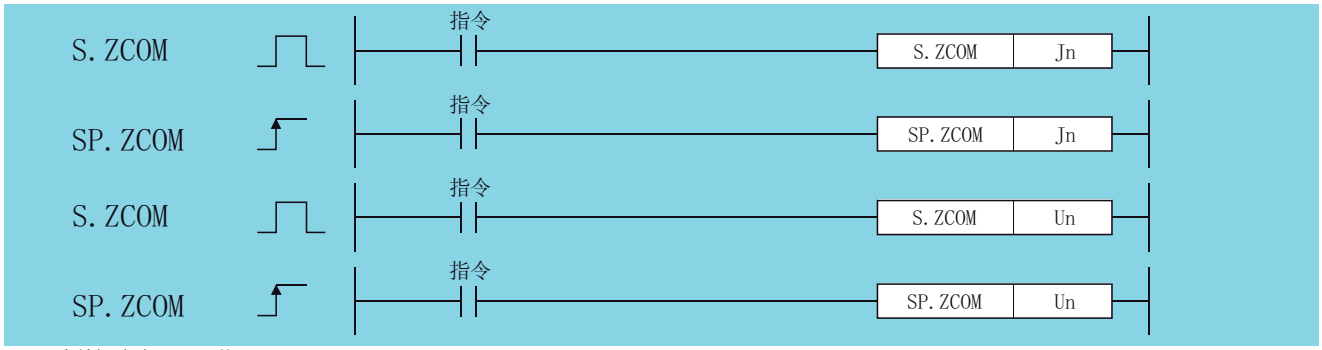
8.1 網路刷新指令

備 註

對於本章中記述的各指令的名稱，在未特別指定的情況下，將按下述方式進行省略。
- S(P).ZCOM → ZCOM
- S(P).RTREAD → RTREAD
- S(P).RTWRITE → RTWRITE

8.1.1 S.ZCOM、SP.ZCOM

Basic | High performance | Process | Redundant | Universal | LCPU



Jn ：自站網路號 (BIN16 位元)。
Un ：自站網路模組的起始 I/O 編號 (BIN16 位)。

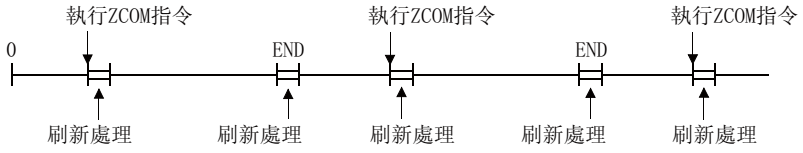
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字				
---	---			---					

ZCOM 指令用於在順控程式的執行過程中以任意時機進行刷新。

通過 ZCOM 指令進行刷新的物件如下所示。
- CC-Link IE 控制網路的刷新 (刷新參數設置時) (僅 QCPU)
- CC-Link IE 現場網路的刷新 (刷新參數設置時) (僅序列號的前 5 位數爲 “ 12012 ” 以後的通用型 QCPU、序列號的前 5 位數爲 “ 13012 ” 以後的 LCPU)
- MELSECNET/H 的刷新 (刷新參數設置時) (僅 QCPU)
- CC-Link 的自動刷新 (刷新參數設置時)
- 智慧功能模組的自動刷新 (刷新參數設置時)

功 能

(1) 執行 ZCOM 指令時，CPU 模組將暫時中斷順控程式的處理，進行 Jn/Un 中指定的網路模組的刷新處理。
(序列號的前 5 位數爲 “ 13011 ” 以前的 LCPU 的情況下，不能通過 Jn 進行指定。)



(2) 在 ZCOM 指令中，不執行下述處理。

(a) CPU 模組與編程工具的通信處理。

(b) 其他站監視處理。

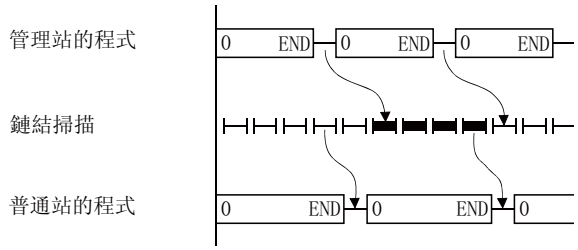
(c) 在串列通信模組中，對其他智慧功能模組的緩衝記憶體的讀取處理。

(d) MELSECNET/H 的低速迴圈傳送資料。

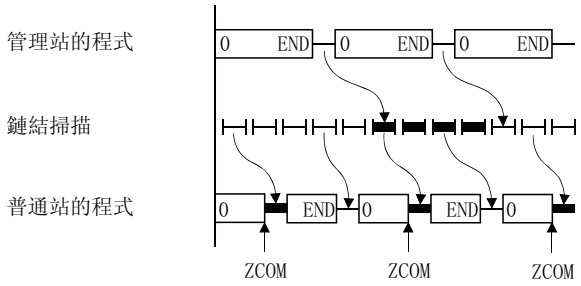
(3) 對於 CC-Link IE 控制網路及 MELSECNET/H(可程式控制器網路)

(a) 自站的順控程式的掃描時間長於其他站的掃描時間時，爲了切實地執行從其他站的資料獲取，可以使用 ZCOM 指令。

1) 未使用 ZCOM 指令時的資料發送接收示例



2) 使用了 ZCOM 指令時的資料發送接收示例

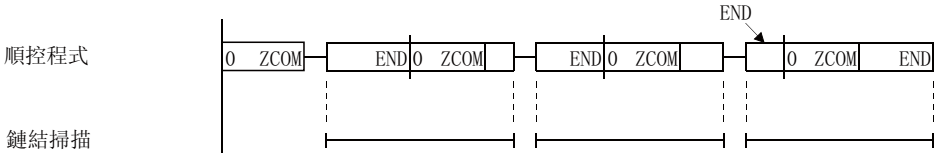


關於 CC-Link IE 控制網路及 MELSECNET/H(可程式控制器網路) 的傳送延遲時間的詳細內容，請參閱以下手冊：

· CC-Link IE 控制網路參考手冊

· Q 系列 MELSECNET/H 網路系統參考手冊 (可程式控制器網路篇)

(b) 在鏈結掃描時間長於順控程式的掃描時間的情況下，即使使用 ZCOM 指令也不能提高資料的發送接收速度。



(4) 對於 MELSECNET/H(遠端 I/O 網路)

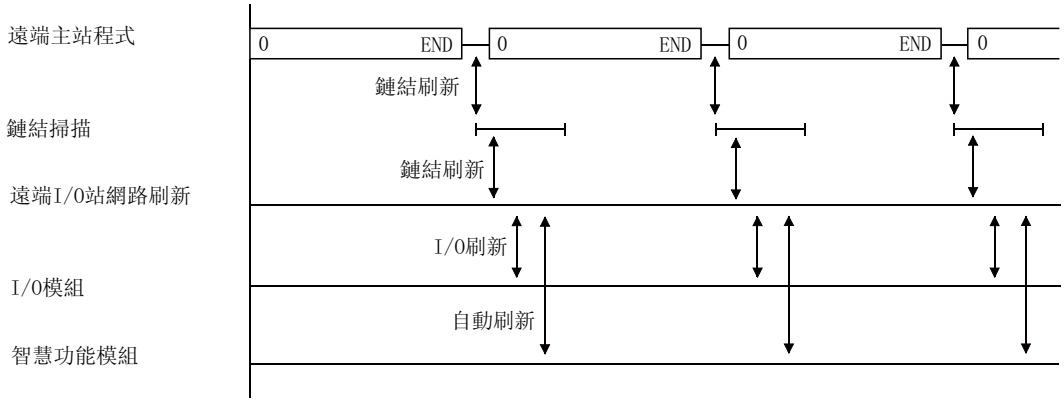
遠端主站的鏈結刷新是在 CPU 模組的 “ END 處理 ” 時進行的。

由於鏈結刷新結束時進行掃描，因此鏈結掃描與 CPU 模組的程式 “ 同步 ”。

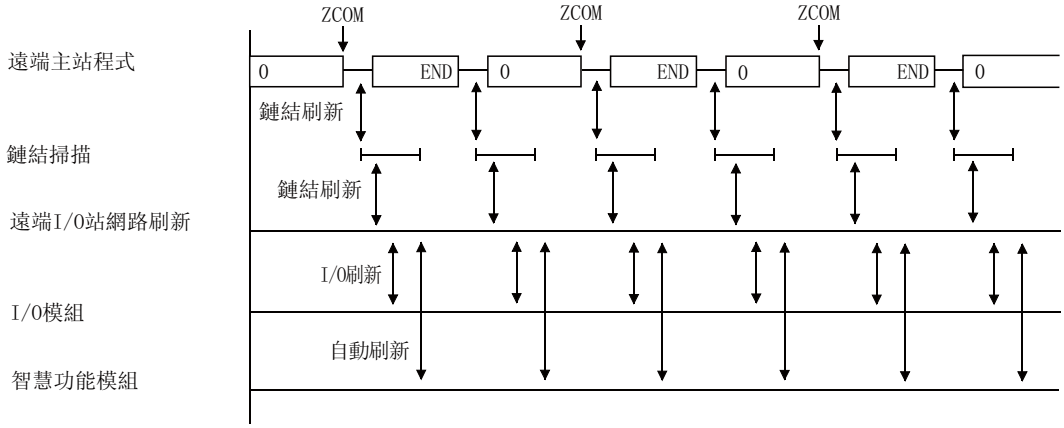
在遠端主站中使用 ZCOM 指令時，將在 ZCOM 指令執行的時點進行鏈結刷新，鏈結刷新結束時進行鏈結掃描。

因此，如果在遠端主站中使用 ZCOM 指令，可以提高與遠端 I/O 站的發送接收處理速度。

1) 未使用 ZCOM 指令時



2) 使用了 ZCOM 指令時



關於遠端 MELSECNET/H(遠端 I/O 網路) 的傳送延遲時間的詳細內容，請參閱以下手冊：

・Q 系列 MELSECNET/H 網路系統參考手冊 (可程式控制器網路篇)

(5) ZCOM 指令在順控程式中的使用次數無限制。

但是，順控程式的掃描時間將會有相當於刷新時間的延遲，因此應加以注意。

(6) 通過將 Un 指定爲引數，不僅可將網路模組指定爲物件，也可將智慧功能模組指定爲物件。

此時，進行智慧功能模組的緩衝記憶體的自動刷新。(變爲 FROM/TO 指令的替代指令。)

(7) 對於通用型 QCPU 及 LCPU，在 ZCOM 指令實施過程中，將處於中斷允許狀態。在中斷程式等中使用刷新資料的情況下，有可能會發生資料背離現象，應加以注意。

要 點

1.ZCOM 指令不能用於恒定周期執行型程式、中斷程式。
2. 在冗餘 CPU 中，使用 ZCOM 指令時是有限制的。
有關詳細內容請參閱下述手冊。
・QnPRHCPU 用戶手冊 (冗餘系統篇)

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2111	起始 I/O 編號號中指定的模組不是網路模組 / 智慧功能模組時。	○	○	○	○	---	---
4102	指定的網路號未與自站相連接時。	○	○	○	○	○	○仏 1
	起始 I/O 編號中指定的模組不是網路模組 / 智慧功能模組時。	---	---	---	---	○	○

*1:　以序列號的前 5 位數爲 “ 13012 ” 以後的模組爲物件。

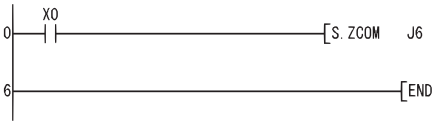
要 點

希望僅與週邊設備進行通信時，應使用 COM 指令 (參閱 395 頁 7.6.9 項、397 頁 7.6.10 項)

程式示例

(1) 以下爲 X0 變爲 ON 時，對網路號 6 的網路模組進行鏈結刷新的程式。

[梯形圖模式]

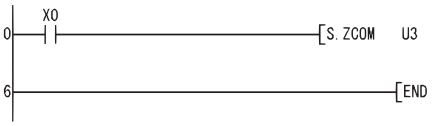


[列表模式]

步	指令	軟元件
0	LD	X0
1	S. ZCOM	J6
6	END	

(2) 以下爲 X0 變爲 ON 時，對安裝在起始 I/O 編號爲 X/Y30 ～ X/Y4F 位置上的網路模組進行鏈結刷新的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0
1	S. ZCOM	U3
6	END	

8.2 路由資訊的讀取 / 寫入

8.2.1 S.RTREAD、SP.RTREAD

· 在序列號的前 5 位數爲 “ 13012 ” 以後的 LCPU 中可以使用。

n ：傳送目標網路號 (1 ～ 239) (BIN16 位)。

Ⓓ ：存儲讀取的資料的軟元件的起始編號 (軟元件名)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n	○	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 根據路由參數中設置的路由資訊，對 n 中指定的傳送目標網路號的資料進行讀取後，存儲到Ⓓ的後面。

(2) n 中指定的傳送目標網路號的資料未在路由參數中進行設置時，在Ⓓ的後面將存儲 0。

(3) Ⓓ後面存儲的資料內容如下所示。

(各資料的範圍)

Ⓓ+0	中繼目標網路號	(1～239)
+1	中繼目標站號	參閱下表
+2	虛擬	

[中繼站目標站號的指定範圍]

網路類型	指定範圍
MELSECNET/H	1 ～ 64
CC-Link IE 控制網路	1 ～ 120
CC-Link IE 現場網路	主站： 固定爲 125(存儲固定值。) 本地站： 1 ～ 120(存儲站號。)

出 錯

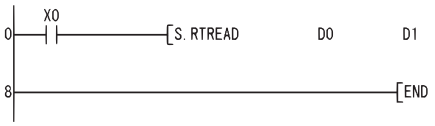
(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 的資料超出了 1 ～ 239 的範圍時。	---	○	○	○	○	---
4101	Ⓓ中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

(1) 以下爲 X0 變爲 ON 時，對 D0 中指定的網路號的路由資訊進行讀取的程式。

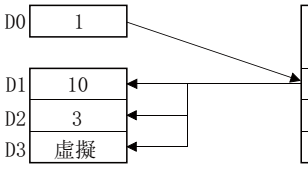
[梯形圖模式]



[列表模式]

步	指令	軟元件	
0	LD	X0	
1	S.RTREAD	D0	D1
8	END		

[動作]



[路由參數的設置內容]

傳送目標網路號	中繼目標網路號	中繼目標站號
1	10	3
2	10	2
3	10	1

8.2.2 S.RTWRITE、SP.RTWRITE

Basic / High performance / Process / Redundant / Universal / LCPU (Ver.)

· 在序列號的前 5 位數爲 “ 13012 ” 以後的 LCPU 中可以使用。

n ： 傳送目標網路號 (1 ～ 239)(BIN16 位)。

Ⓢ ： 存儲寫入資料的軟元件的起始編號 (軟元件名)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n	○	○		---				○	---
Ⓢ	---	○		---				---	---

功　能

(1) 將Ⓢ後面的路由資料登錄到路由參數的 n 中指定的傳送目標網路號的區域中。

(2) Ⓢ後面設置的資料內容如下所示。

(各資料的範圍)

Ⓢ+0	中繼目標網路號	(0～239)
+1	中繼目標站號	參閱下表
+2	虛擬	

[中繼站目標站號的指定範圍]

網路類型	指定範圍
MELSECNET/H	1 ～ 64
CC-Link IE 控制網路	1 ～ 120
CC-Link IE 現場網路	主站： 固定爲 125 本地站： 1 ～ 120

(3) n 中指定的傳送目標網路號的資料未在路由參數中進行設置時，將被變更爲Ⓢ後面的資料。

(4) Ⓢ後面的資料 (Ⓢ+0 ～Ⓢ+2) 全部爲 0 時，將 n 中指定的傳送目標網路號的資料從路由參數中刪除。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4100	n 的資料超出了 1 ～ 239 的範圍時。 Ⓢ 後面的資料超出了各設置範圍時。 網路參數的路由參數中登錄的路由資訊的登錄數與 RTWRITE 指令中登錄的路由資訊的登錄數的合計超過了 64 時。	---	○	○	○	○	---
4101	Ⓢ 中指定的軟元件超出了相應軟元件的範圍時。	---	---	---	---	○	○

程式示例

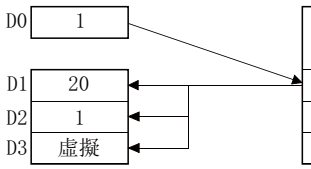
(1) 以下爲 X0 變爲 ON 時，將 D1 ～ D3 中指定的路由資訊寫入到 D0 中指定的網路號的網路模組中的程式。

[梯形圖模式]

[列表模式]

步	指令	軟元件	
0	LD	X0	
1	S.RTWRITE	D0	D1
9	END		

[動作]



[路由參數的設置內容]

傳送目標網路號	中繼目標網路號	中繼目標站號
1	20	1
2	10	2
3	10	1

第 9 章 多 CPU 專用指令

9.1 寫入自站 CPU 共用記憶體

在多 CPU 系統中對自站 CPU 共用記憶體進行寫入時，通過 S.TO 指令或者 TO 指令進行。

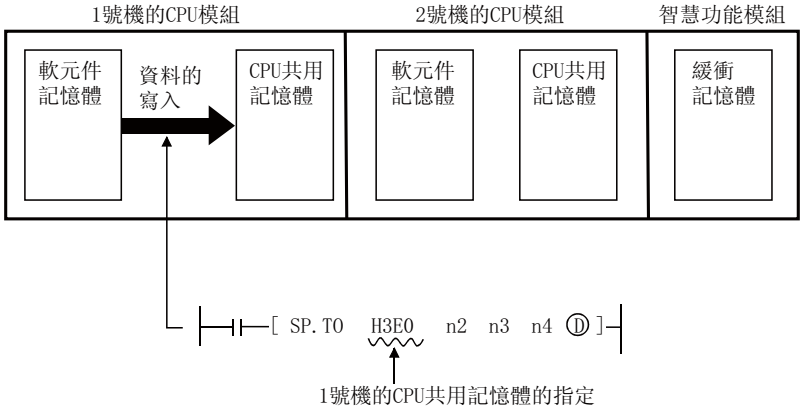
S.TO 指令及 TO 指令的使用可否如下表所示。

CPU 模組		S.TO 指令	TO 指令
基本型 QCPU	Q00JCPU	不能使用	不能使用
	Q00CPU、Q01CPU	可以使用	可以使用
高性能型 QCPU	Q02CPU、Q02HCPU、 Q06HCPU、Q12HCPU、 Q25HCPU	可以使用	不能使用
過程 CPU	Q02PHCPU、Q06PHCPU、 Q12PHCPU、Q25PHCPU	可以使用	不能使用
冗餘 CPU	Q12PRHCPU、Q25PRHCPU	不能使用	不能使用
通用型 QCPU	Q00UJCPU	不能使用	不能使用
	Q00UCPU、Q01UCPU、Q02UCPU、Q03UDCPU、Q04UDHCPU、 Q06UDHCPU、Q10UDHCPU， Q13UDHCPU、Q20UDHCPU， Q26UDHCPU、 Q03UDECPU、Q04UDEHCPU、Q06UDEHCPU、Q10UDEHCPU、 Q13UDEHCPU、Q20UDEHCPU、Q26UDEHCPU、Q50UDEHCPU、 Q100UDEHCPU	可以使用	可以使用
LCPU	L02CPU、L26CPU-BT、L02CPU-P、L26CPU-PBT	不能使用	不能使用

(1) S.TO 指令的動作

通過 S.TO 指令，可以將資料寫入到自站 CPU 模組的 CPU 共用記憶體中。

在 1 號機的 CPU 模組中，執行 S.TO 指令時的處理如下圖所示。



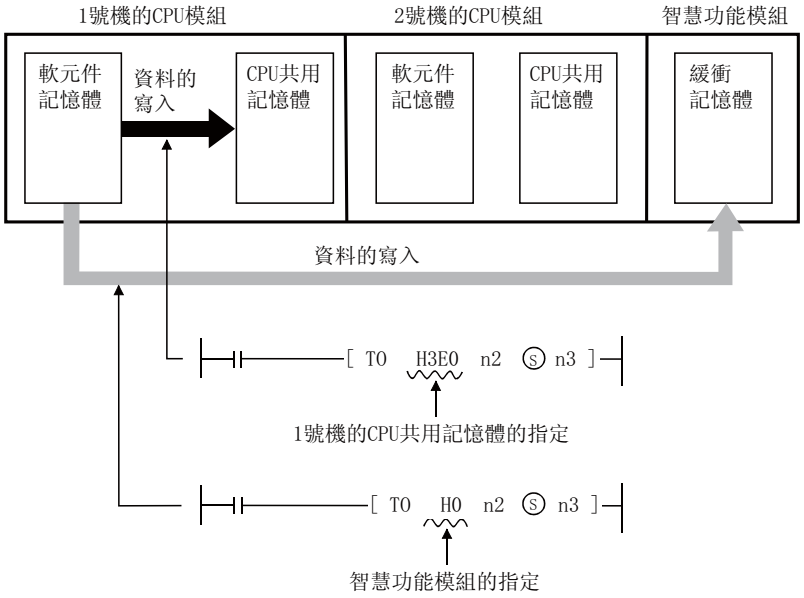
(2) TO 指令的動作

通過 TO 指令，可以將軟元件記憶體的資料寫入到下述記憶體中。

・自站 CPU 模組的 CPU 共用記憶體

・智慧功能模組的緩衝記憶體

在 1 號機的 CPU 模組中，執行了 TO 指令時的處理如下圖所示。



要 點

在基本型 QCPU 及通用型 QCPU 中，使用 S.TO 指令或 TO 指令均可對 CPU 共用記憶體進行寫入，但對自站 CPU 的共用記憶體進行寫入時，建議使用 TO 指令，因爲可以減少步數及處理時間。

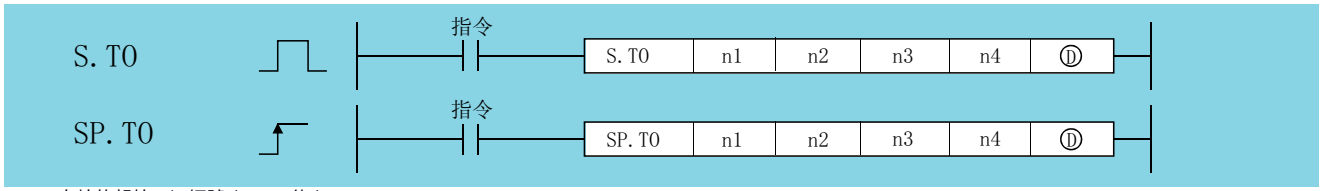
備 註

通過 TO 指令對智慧功能模組的緩衝記憶體進行寫入時，請參閱 415 頁 7.8.2 項。

Ver. Basic　Ver. High performance　Process　Redundant　Universal　LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的 Q00CPU、Q01CPU 中可以使用。
· 在功能版本 B 以後的高性能型 QCPU 中可以使用。

9.1.1 S.TO、SP.TO

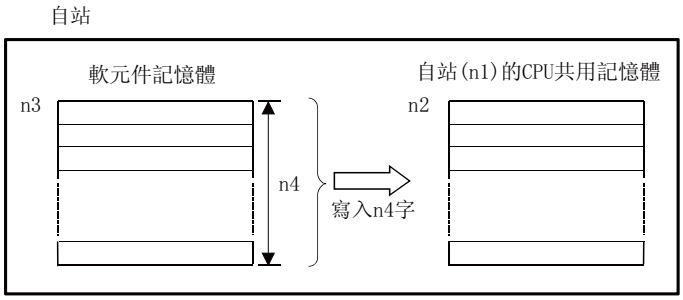


n1 ：自站的起始 I/O 編號 (BIN16 位)。
n2 ：寫入目標的自站 CPU 共用記憶體位址 (BIN16 位)。
· 基本型 QCPU：0 ～ 511
· 高性能型 QCPU、過程 CPU、通用型 QCPU：0 ～ 4095
n3 ：存儲寫入資料的軟元件的起始編號 (BIN16 位)。
n4 ：寫入資料數 (BIN16 位元)。
· 基本型 QCPU：1 ～ 320
· 高性能型 QCPU、過程 CPU：1 ～ 256
· 通用型 QCPU：0 ～ 2048
Ⓓ ：寫入結束時使其 1 個掃描 ON 的軟元件 (位)。

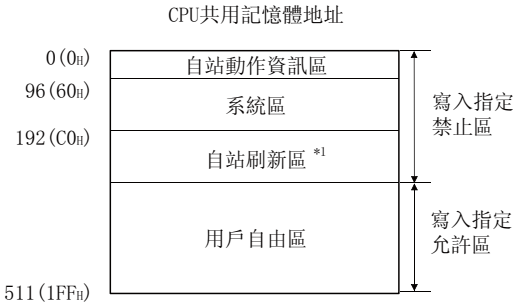
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
n1	---	○		---				○	---
n2	---	○		---				○	---
n3	---	○		---				---	---
n4	---	○		---				○	---
Ⓓ	○	○		---				---	---

功 能

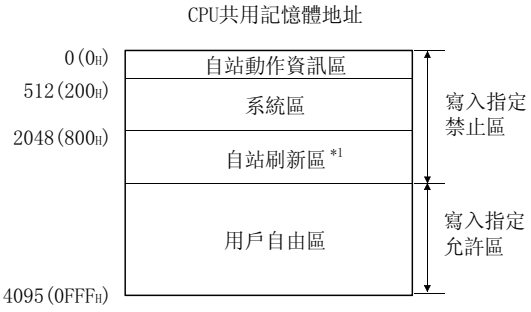
(1) 從自站 CPU 模組的 n3 開始，將 n4 字的軟元件資料寫入到自站 CPU 模組的 n2 中指定的 CPU 共用記憶體位址的後面。
寫入結束時，ⓓ中指定的結束位將變爲 ON。



(a) 基本型 QCPU 時的 CPU 共用記憶體地址



(b) 高性能型 QCPU、過程 CPU、通用型 QCPU*2 時的 CPU 共用記憶體地址



*1: 未進行自動刷新設置的情況下，可以作爲用戶自由區使用。
此外，即使進行了自動刷新設置，自動刷新發送範圍後面的區域也可作爲用戶自由區使用。

*2: 不能通過 S(P).TO 指令對通用型 QCPU 的多 CPU 高速通信區進行寫入。

(2) 寫入點數爲 0 時，將變爲無處理且結束軟元件也不變爲 ON。

(3) 每個站的 1 個掃描中只能執行一個 S.TO 指令。
在有 2 處以上的執行條件同時成立的情況下，由於將自動進行握手，因此後執行的 S.TO 指令將不進行處理。

(4) 寫入資料數根據物件 CPU 模組而有所不同。

CPU 模組	寫入點數
基本型 QCPU	1 ～ 320
高性能型 QCPU、過程 CPU	1 ～ 256
通用型 QCPU	1 ～ 2048

要 點

對 CPU 共用記憶體進行資料寫入時，也可使用智慧功能模組軟元件進行寫入。
關於智慧功能模組軟元件，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

出 錯

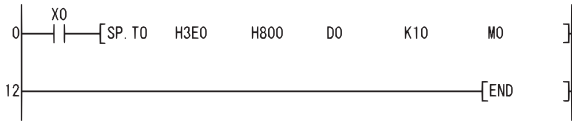
在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2107	自站的起始 I/O 編號 (n1) 中指定了自站以外時。	---	○	○	---	---	---
2110	CPU 模組的起始 I/O 編號中指定的位置上 CPU 模組不存在時。	○	○	○	---	○	---
4002	指定的指令不正確時。	○	○	○	---	○	---
4003	軟元件數有誤時。	○	○	○	---	○	---
4004	指定了不能使用的軟元件時。	○	○	○	---	○	---
4100	自站的起始 I/O 編號 (n1) 中，指定了除 $3E0_H/3E1_H/3E2_H/3E3_H$ 以外時。	○	○	○	---	○	---
4101	寫入目標的自站 CPU 共用記憶體位址 (n2) 中指定了自站動作資訊區、系統區或自站刷新區時。	---	○	○	---	---	---
	寫入點數 (n4) 超出了設置資料的指定範圍時。 寫入目標的自站 CPU 共用記憶體位址 (n2) 的起始超出了 CPU 共用記憶體地址的範圍時。 寫入目標的自站 CPU 共用記憶體位址 (n2)+ 寫入點數 (n4) 超出了 CPU 共用記憶體地址的範圍時。 存儲寫入資料的起始軟元件編號 (n3)+ 寫入點數 (n4) 超出了軟元件的範圍時。	○	○	○	---	○	---
4111	寫入目標的自站 CPU 共用記憶體位址 (n2) 中指定了自站動作資訊區、系統區或自站刷新區時。	○	---	---	---	○	---
4112	自站的起始 I/O 編號 (n1) 中，指定了除自站以外時。	○	---	---	---	○	---

程式示例

(1) 以下爲 X0 由 OFF 變爲 ON 時，將從 D0 開始 10 點的資料存儲到 1 號機的 CPU 共用記憶體的 800H 地址號中的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件
0	LD	X0
1	SP.TO	H3E0 H800 D0 K10 M0
12	END	

備 註

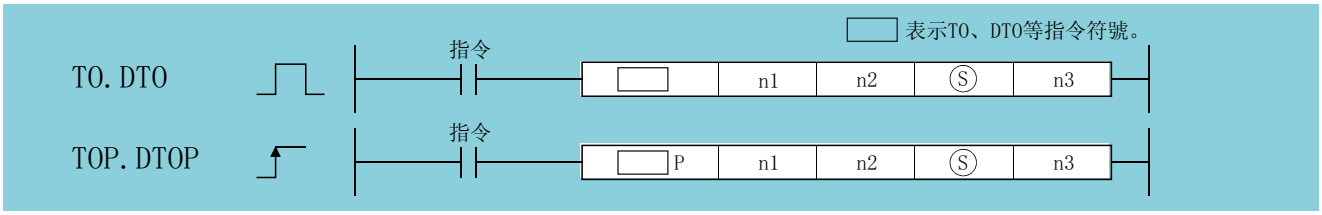
進行 n1 指定時，是將安裝了 CPU 模組的插槽的起始 I/O 編號以 4 位數的 16 進制數表示時的高 3 位進行指定。

	CPU 插槽	插槽 0	插槽 1	插槽 2
起始 I/O 編號	3E00	3E10	3E20	3E30
n1	3E0	3E1	3E2	3E3

Ver. Basic | High performance | Process | Redundant | Universal | LCPU

· 在序列號的前 5 位數爲 “ 04122 ” 以後的基本型 QCPU 中可以使用。

9.1.2 TO、TOP、DTO、DTOP



n1 ： 自站的起始 I/O 編號 (BIN16 位元)。
· 基本型 QCPU： $3E0_H$
· 通用型 QCPU： $3E0_H$ ～ $3E3_H$

n2 ： 寫入目標的自站 CPU 共用記憶體位址 (BIN16 位元)。
· 基本型 QCPU： 192 ～ 511
· 通用型 QCPU： 2048 ～ 4095、10000 ～ 24335*1

Ⓢ ： 寫入資料或者存儲寫入資料的軟元件的起始編號 (BIN16 位元)。

n3 ： 寫入資料數 (BIN16 位元)。
· 基本型 QCPU： TO(P)： 1 ～ 320、DTO(P)： 1 ～ 160
· 通用型 QCPU： TO(P)： 1 ～ 14336*2、DTO(P)： 1 ～ 7168*1

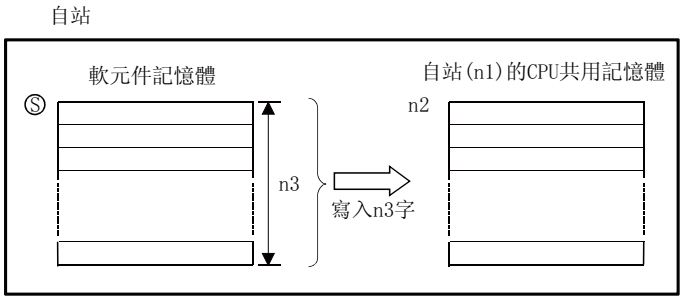
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他 U
	位	字		位	字				
n1		○			○			○	○
n2		○			○			○	---
Ⓢ		○			---			○	---
n3		○			○			○	---

*1: 設置範圍取決於多 CPU 高速通信功能的自動刷新設置範圍。

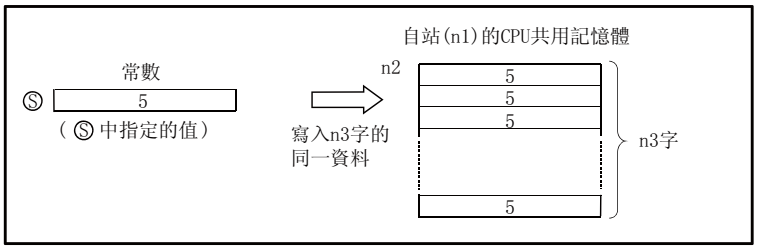
功 能

TO

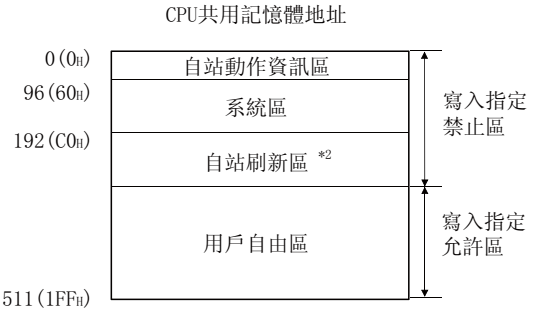
(1) 將自站 CPU 模組的Ⓢ開始的 n3 字的軟元件資料，寫入到自站 CPU 模組的 n2 中指定的 CPU 共用記憶體位址的後面。



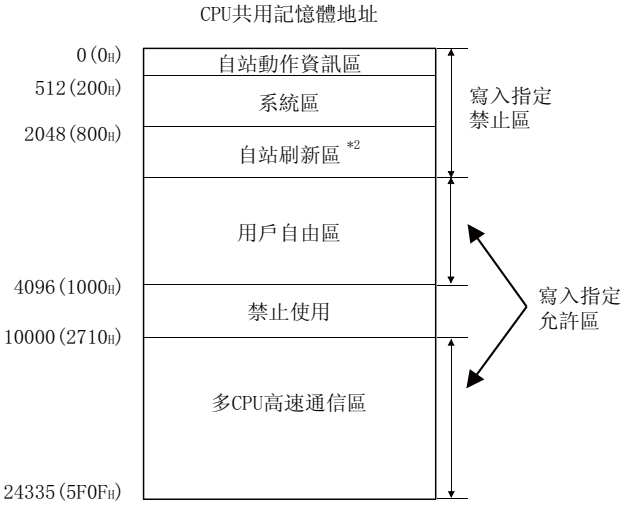
Ⓢ中指定了常數時，將 n3 字的同一資料 (Ⓢ中指定的值) 寫入到從指定 CPU 共用記憶體開始的 n3 字區域中。



(a) 基本型 QCPU 時的 CPU 共用記憶體地址



(b) 高性能型 QCPU 時的 CPU 共用記憶體地址 *3



*2: 未進行自動刷新設置的情況下，可以作爲用戶自由區使用。
此外，即使進行了自動刷新設置，自動刷新發送範圍後面的區域也可作爲用戶自由區使用。

*3: 在下述 CPU 中，不能對多 CPU 高速通信區進行寫入。
- Q00UCPU
- Q01UCPU
- Q02UCPU

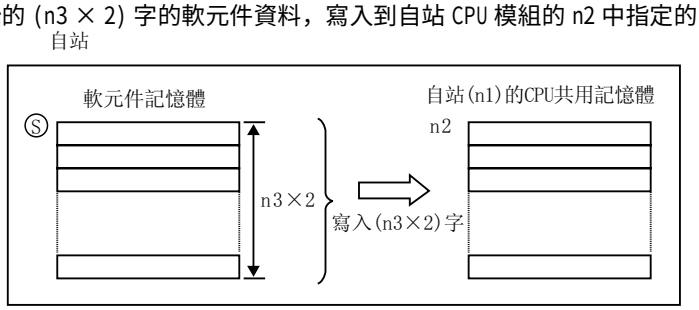
(2) 寫入點數爲 0 時，將變爲無處理。

(3) 寫入資料數根據物件 CPU 模組而有所不同。

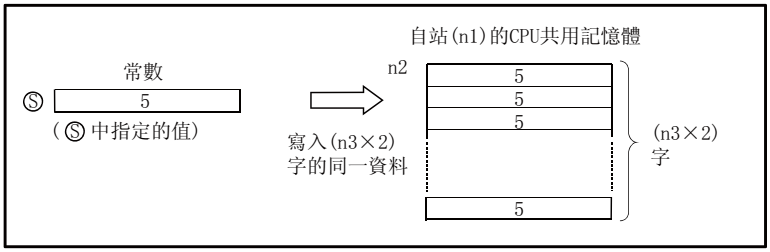
CPU 模組	寫入點數
基本型 QCPU	1 ～ 320
通用型 QCPU	1 ～ 14336

DTO

(1) 將自站 CPU 模組的Ⓢ開始的 (n3 × 2) 字的軟元件資料，寫入到自站 CPU 模組的 n2 中指定的 CPU 共用記憶體位址的後面。



Ⓢ中指定了常數時，將 (n3 × 2) 字的同一資料 (Ⓢ中指定的值) 寫入到從指定 CPU 共用記憶體開始的 (n3 × 2) 字區域中。



(2) 寫入點數爲 0 時，將變爲無處理。

(3) 寫入資料數根據物件 CPU 模組而有所不同。

CPU 模組	寫入點數
基本型 QCPU	1 ～ 160
通用型 QCPU	1 ～ 7168

要 點

對 CPU 共用記憶體進行資料寫入時，也可使用智慧功能模組軟元件進行寫入。
關於智慧功能模組軟元件，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

出　錯

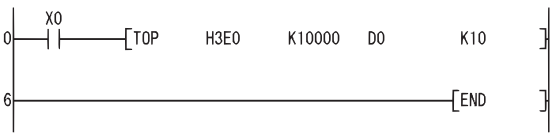
在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2110	CPU 模組的起始 I/O 編號中指定的位置上 CPU 模組不存在時。	○	---	---	---	○	---
4101	寫入點數 (n3) 超出了設置資料的指定範圍時。 寫入目標的自站 CPU 共用記憶體位址 (n2)+ 寫入點數 (n3) 超出了 CPU 共用記憶體地址的範圍時。 存儲寫入資料的起始軟元件編號 (Ⓢ)+ 寫入點數 (n3) 超出了軟元件的範圍時。 寫入目標的自站 CPU 共用記憶體位址 (n2) 的起始值的指定超出了允許寫入的區域時。	○	---	---	---	○	---
4111	寫入目標的自站 CPU 共用記憶體位址 (n2) 的起始值不是有效值時。	○	---	---	---	○	---
4112	(n1) 中指定了除自站以外時。(但是，指定了其他站的多 CPU 高速通信區時除外。)	○	---	---	---	○	---

程式示例

(1) 以下爲 X0 變爲 ON 時，將從 D0 開始 10 點的資料存儲到 1 號機的 CPU 共用記憶體的 10000 號後面的程式。

[梯形圖模式]

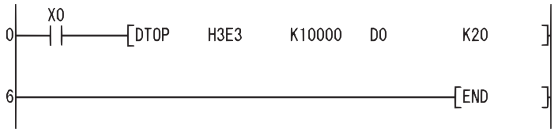


[列表模式]

步	指令	軟元件			
0	LD	X0			
1	TOP	H3E0	K10000	D0	K10
6	END				

(2) 以下爲 X0 變爲 ON 時，將從 D0 開始 20 點的資料存儲到 4 號機的 CPU 共用記憶體的 10000 號後面的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X0			
1	DTOP	H3E3	K10000	D0	K20
6	END				

備 註

進行 n1 指定時，是將安裝了 CPU 模組的插槽的起始 I/O 編號以 4 位數的 16 進制數表示時的高 3 位進行指定。

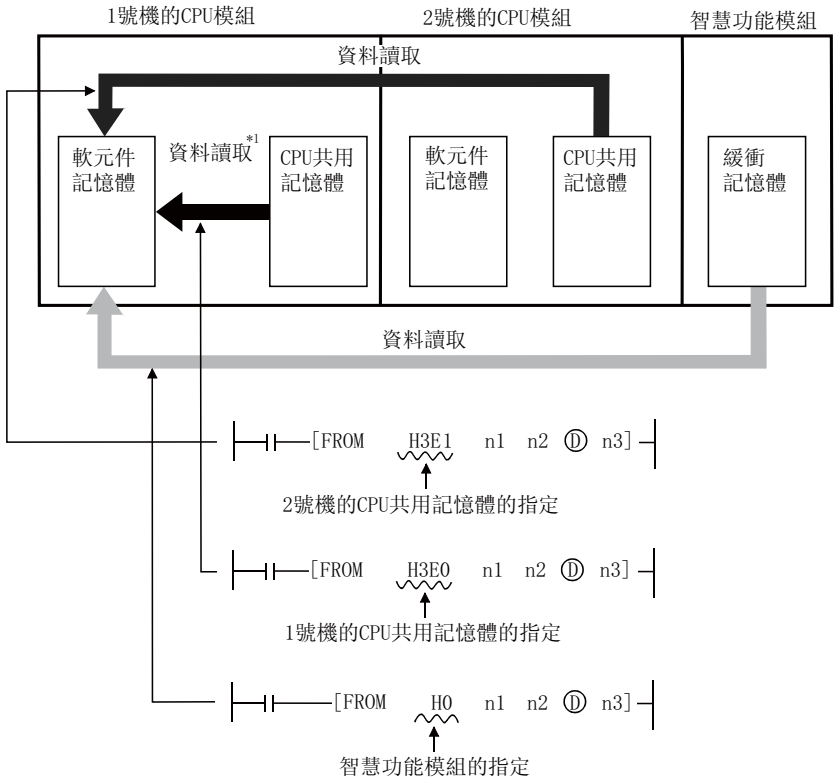
	CPU 插槽	插槽 0	插槽 1	插槽 2
起始 I/O 編號	3E00	3E10	3E20	3E30
n1	3E0	3E1	3E2	3E3

9.2 從其他站 CPU 共用記憶體中讀取資料

通過在多 CPU 系統中使用 FROM(P)/DFRO(P) 指令，可以從下述記憶體中進行讀取。

・智慧功能模組的緩衝記憶體
・其他站 CPU 模組的 CPU 共用記憶體
・自站 CPU 模組的 CPU 共用記憶體 (只在基本型 QCPU、通用型 QCPU 中可以執行)

在 1 號機中執行了 FROM(P) 指令時的處理如下圖所示。



*1: 在基本型 QCPU、通用型 QCPU 中可以執行。

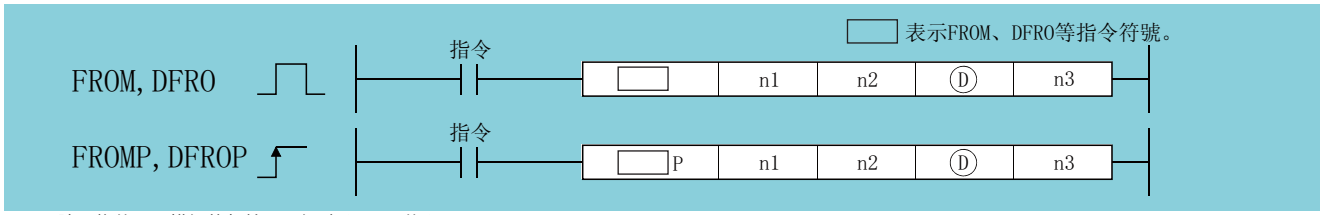
備 註

通過 FROM/DFRO 指令對智慧功能模組的緩衝記憶體進行讀取時，請參閱 413 頁 7.8.1 項。

· 在序列號的前 5 位數爲 “ 04122 ” 以後的 Q00CPU、Q01CPU 中可以使用。
· 在功能版本 B 以後的高性能型 QCPU 中可以使用。

9.2.1 FROM、FROMP、DFRO、DFROP

1 使用基本型 QCPU、通用型 QCPU 時



n1 ：讀取物件 CPU 模組的起始 I/O 編號 (BIN16 位)。
· 基本型 QCPU：$3E0_H$ ～ $3E2_H$
· 通用型 QCPU：$3E0_H$ ～ $3E3_H$

n2 ：讀取目標的自站 CPU 共用記憶體位址 (BIN16 位)。
· 基本型 QCPU：0 ～ 512
· 通用型 QCPU：0 ～ 4095,10000 ～ 24335*1

Ⓓ ：存儲讀取資料的軟元件的起始編號 (BIN16 位)。

n3 ：讀取資料數 (BIN16 位元)。
· 基本型 QCPU：FROM(P)：1 ～ 512、DFRO(P)：1 ～ 256
· 通用型 QCPU：FROM(P)：1 ～ 14336*1、DFRO(P)：1 ～ 7168*1

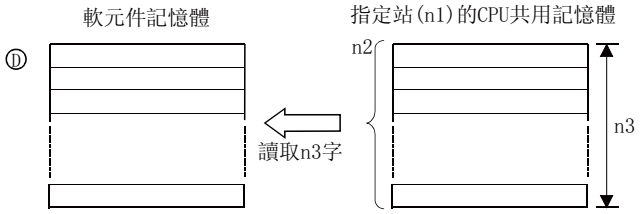
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數	其他
	位	字		位	字			K、H	U
n1	---	○		○				○	○
n2	---	○		○				○	---
Ⓓ	---	○		---				---	---
n3	---	○		○				○	---

*1: 設置範圍取決於多 CPU 高速通信功能的自動刷新設置範圍。

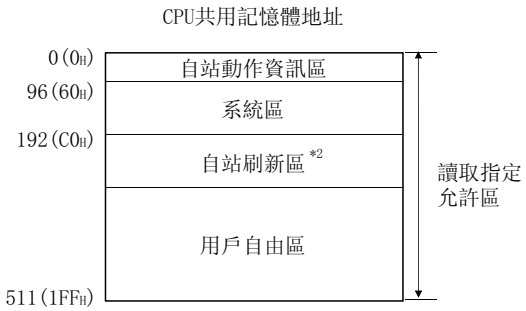
功 能

FROM

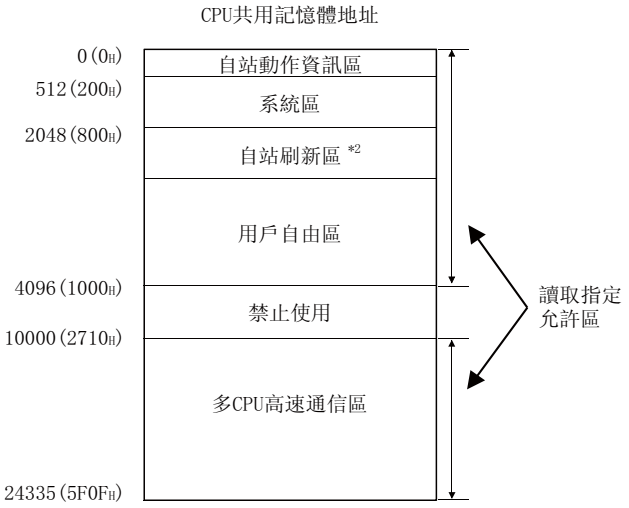
(1) 從 n1 中指定的 CPU 模組的 n2 中指定的 CPU 共用記憶體位址開始，讀取 n3 字的資料後，存儲到Ⓓ中指定的軟元件後面。



(a) 基本型 QCPU 時的 CPU 共用記憶體地址



(b) 通用型 QCPU 時的 CPU 共用記憶體地址 *3



*2: 未進行自動刷新設置的情況下，可以作爲用戶自由區使用。
此外，即使進行了自動刷新設置，自動刷新發送範圍後面的區域也可作爲用戶自由區使用。

*3: 在下述 CPU 中，不能從多 CPU 高速通信區中讀取。
· Q00UCPU
· Q01UCPU
· Q02UCPU

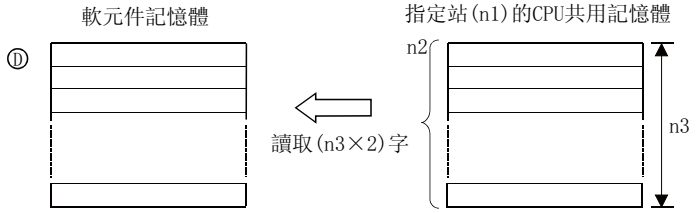
(2) 讀取資料 n3 爲 0 時，將變爲無處理。

(3) 讀取資料數根據物件 CPU 模組而有所不同。

CPU 模組	寫入點數
基本型 QCPU	1 ～ 512
通用型 QCPU	1 ～ 14336

DFRO

(1) 將 n1 中指定的 CPU 模組的 n2 中指定的 CPU 共用記憶體位址中，讀取 (n3 × 2) 字的資料後，存儲到ⓓ中指定的軟元件後面。



(2) 讀取資料 n3 爲 0 時，將變爲無處理。

(3) 讀取資料數根據物件 CPU 模組而有所不同。

CPU 模組	寫入點數
基本型 QCPU	1 ～ 256
通用型 QCPU	1 ～ 7168

要 點

對 CPU 共用記憶體進行資料寫入時，也可使用智慧功能模組軟元件進行寫入。
關於智慧功能模組軟元件，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

出　錯

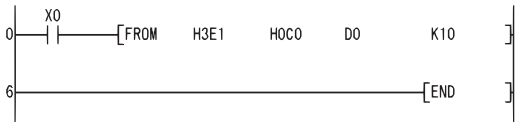
在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2110	CPU 模組的起始 I/O 編號中指定的位置上 CPU 模組不存在時。	○	---	---	---	○	---
4101	進行讀取的 CPU 共用記憶體地址 (n2) 的起始超出了 CPU 共用記憶體的範圍時。 進行讀取的 CPU 共用記憶體地址 (n2)+ 讀取點數 (n3) 超出了 CPU 共用記憶體的範圍時。 讀取資料存儲軟元件編號 (ⓓ)+ 讀取點數 (n3) 超出了指定軟元件的範圍時。 進行讀取的 CPU 共用記憶體地址 (n2) 的起始值不是有效值時。(4097 ～ 9999)	○	---	---	---	○	---

程式示例

(1) 以下爲 X0 變爲 ON 時，將 2 號機的 CPU 共用記憶體的 C0H 位址號開始的 10 點數據存儲到 D0 後面的程式。

[梯形圖模式]

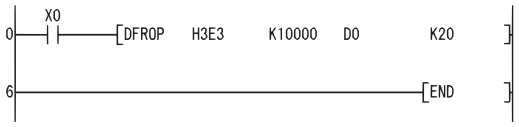


[列表模式]

步	指令	軟元件			
0	LD	X0			
1	FROM	H3E1	H0C0	D0	K10
6	END				

(2) 以下爲 X0 變爲 ON 時，將 4 號機的 CPU 共用記憶體的 10000 位址號開始的 20 點數據存儲到 D0 後面的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件			
0	LD	X0			
1	DFROP	H3E3	K10000	D0	K20
6	END				

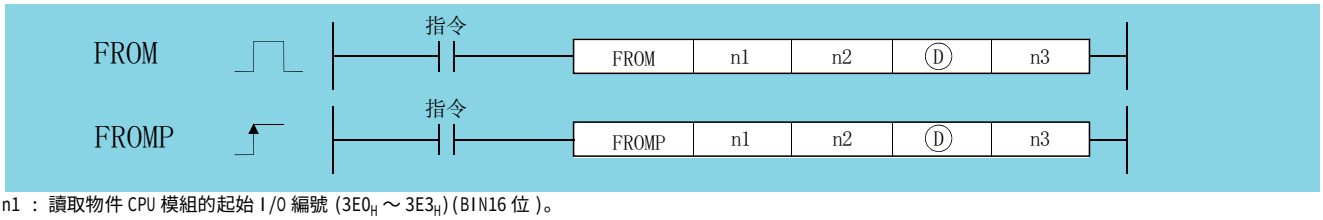
備 註

1. 進行 n1 指定時，是將安裝了 CPU 模組的插槽的起始 I/O 編號以 4 位數的 16 進制數表示時的高 3 位進行指定。

	CPU 插槽	插槽 0	插槽 1	插槽 2
起始 I/O 編號	3E00	3E10	3E20	3E30
n1	3E0	3E1	3E2	3E3

2. 對 FROM/TO 指令進行了自動互鎖。

2 使用高性能型 QCPU、過程 CPU 時



n1 ： 讀取物件 CPU 模組的起始 I/O 編號 ($3E0_H$ ～ $3E3_H$)(BIN16 位)。

n2 ： 讀取目標的 CPU 共用記憶體位址 (0 ～ 4095)(BIN16 位)。

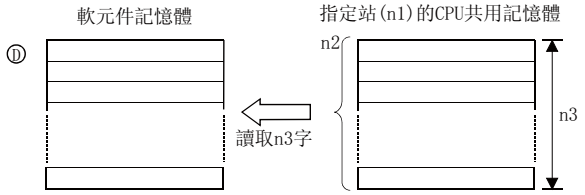
Ⓓ ： 存儲讀取的資料的軟元件的起始編號 (BIN16 位)。

n3 ： 讀取資料數 (1 ～ 4096)(BIN16 位)。

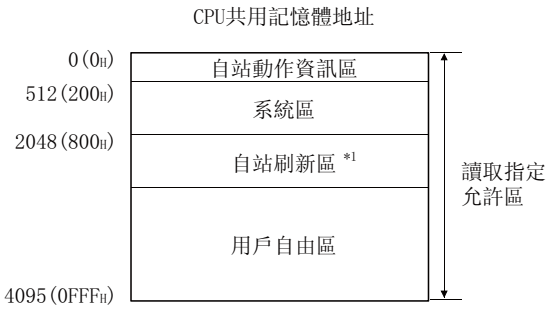
設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他 U
	位	字		位	字				
n1	---	○		○				○	○
n2	---	○		○				○	---
Ⓓ	---	○		---				---	---
n3	---	○		○				○	---

功 能

(1) 從 n1 中指定的 CPU 模組的 n2 中指定的 CPU 共用記憶體位址開始，讀取 n3 字的資料後，存儲到Ⓓ中指定的軟元件後面。



高性能型 QCPU、過程 CPU 時的 CPU 共用記憶體地址



*1: 未進行自動刷新設置的情況下，可以作爲用戶自由區使用。
此外，即使進行了自動刷新設置，自動刷新發送範圍後面的區域也可作爲用戶自由區使用。

(2) 讀取資料 n3 爲 0 時，將變爲無處理。

(3) 讀取資料數根據物件 CPU 模組而有所不同。

CPU 模組	寫入點數
高性能型 QCPU 過程 CPU	1 ～ 4096

要 點

對 CPU 共用記憶體進行資料讀取時，也可使用智慧功能模組軟元件進行寫入。
關於智慧功能模組軟元件，請參閱所使用的 CPU 模組的用戶手冊 (功能解說 / 程式基礎篇)。

出　錯

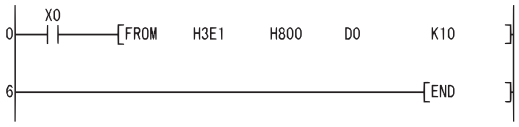
在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
2110	CPU 模組的起始 I/O 編號中指定的位置上 CPU 模組不存在時。	○	---	---	---	○	---
4101	進行讀取的 CPU 共用記憶體地址 (n2) 的起始超出了 CPU 共用記憶體的範圍時。 進行讀取的 CPU 共用記憶體地址 (n2)+ 讀取點數 (n3) 超出了 CPU 共用記憶體的範圍時。 讀取資料存儲軟元件編號 (Ⓓ)+ 讀取點數 (n3) 超出了指定軟元件的範圍時。 進行讀取的 CPU 共用記憶體地址 (n2) 的起始值不是有效值時。(4097 ～ 9999)	○	---	---	---	○	---

程式示例

(1) 以下爲 X0 變爲 ON 時，將 2 號機的 CPU 共用記憶體的 800_H 地址號開始的 10 點數據存儲到 D0 後面的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件				
0	LD	X0				
1	FROM	H3E1	H800	D0	K10	
6	END					

備 註

1. 進行 n1 指定時，是將安裝了 CPU 模組的插槽的起始 I/O 編號以 4 位數的 16 進制數表示時的高 3 位進行指定。

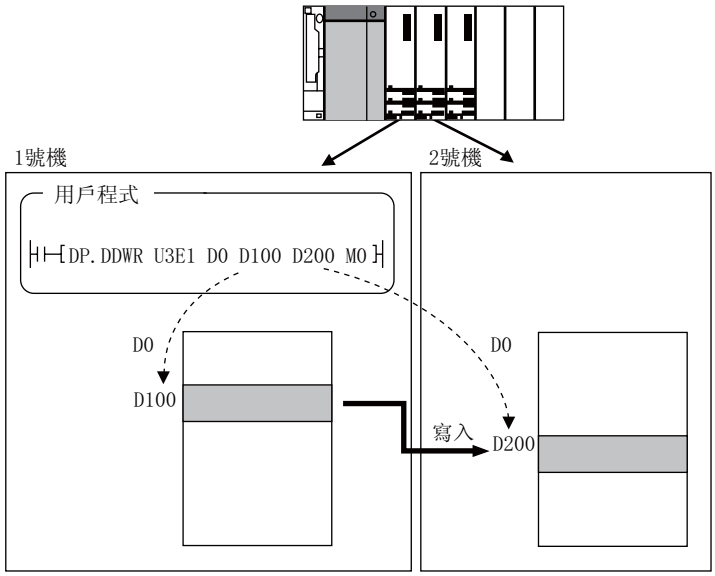
	CPU 插槽	插槽 0	插槽 1	插槽 2
起始 I/O 編號	3E00	3E10	3E20	3E30
n1	3E0	3E1	3E2	3E3

2. 對 FROM/TO 指令進行了自動互鎖。

第 10 章　多 CPU 高速通信專用指令

10.1 概要

多 CPU 高速通信專用指令是從通用型 QCPU 中對其他站的通用型 QCPU 進行軟元件資料的寫入 / 讀取的指令。
根據多 CPU 高速通信專用指令，從 1 號機對 2 號機進行寫入時的動作如下圖所示。



要 點

使用多 CPU 高速通信專用指令時，自站、其他站 (指令的執行物件站) 均只能使用下述 CPU 模組。
・序列號的前 5 位數爲 “ 10012 ” 以後的 Q03UDCPU、Q04UDHCPU、Q06UDHCPU
・Q10UDHCPU、Q13UDHCPU、Q20UDHCPU、Q26UDHCPU
・QnUDE(H)CPU

(1) 用於執行多 CPU 高速通信專用指令的系統、參數設置
多 CPU 高速通信專用指令功能在如下所示的系統配置、參數設置時可以執行。
・1 號機中使用的是 QnUD(H)CPU 或者 QnUDE(H)CPU。
・使用了多 CPU 高速通信主基板 (Q3 □ DB)。
・可程式控制器參數的多 CPU 設置中設置了 “ 使用多 CPU 高速通信功能 ”。

(2) 可進行寫入 / 讀取的軟元件

(a) 可進行寫入 / 讀取的軟元件名

通過多 CPU 高速通信專用指令可對其他站的通用型 QCPU 進行寫入 / 讀取的軟元件如下表所示。

分類	類型	軟元件名	物件軟元件設置 允許 / 禁止	備註
內部用戶軟元件	位元軟元件	X、Y、M、L、B、F、SB	仮	設置時的必要條件 ・進行了 16 位 (4 位數) 的位數指定。 ・開始位軟元件爲 16(10_H) 的倍數。
	字軟元件	T、ST、C、D、W、SW	○	-
內部系統軟元件	位元軟元件	SM	仮	設置時的必要條件 ・進行了 16 位 (4 位數) 的位數指定。 ・開始位軟元件爲 16(10_H) 的倍數。
	字軟元件	SD	○	-
文件寄存器	字軟元件	R、ZR	○	-

○：可以設置　△：帶條件可設置

要 點

SB、SW、SM、SD 中包含有系統資訊區。
通過多 CPU 高速通信專用指令的 D(P).DDWR 指令，對上述軟元件進行寫入時，應注意防止破壞系統資訊。

(3) 軟元件的指定方法及可進行寫入 / 讀取的範圍

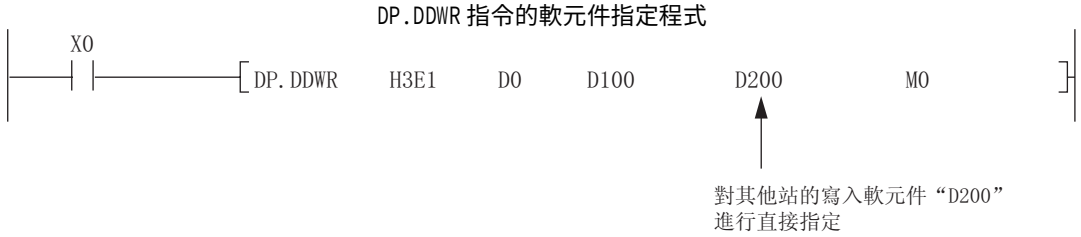
在其他站 CPU 軟元件的指定方法中，有軟元件指定及字串指定這 2 種類型。

在軟元件指定及字串指定中，可對其他站進行寫入 / 讀取的軟元件範圍是不相同的。

(a) 軟元件指定

軟元件指定是對執行寫入 / 讀取的其他站軟元件進行直接指定的方法。

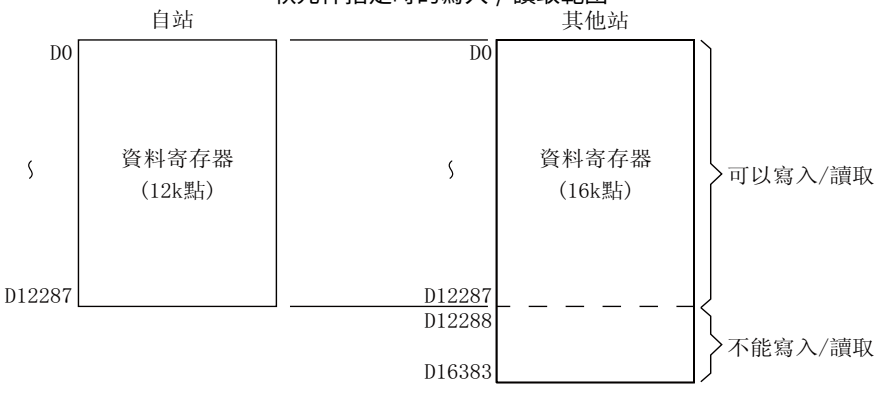
DP.DDWR 指令的軟元件指定程式



在軟元件指定中，可在自站的軟元件範圍內進行寫入 / 讀取。

例如，自站的資料寄存器爲 12k 點，其他站的資料寄存器爲 16k 點時，可對其他站的資料寄存器的起始開始的 12k 點的資料進行寫入 / 讀取。

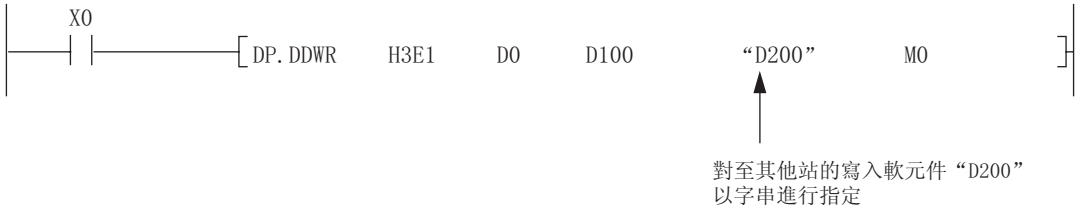
軟元件指定時的寫入 / 讀取範圍



(b) 字串指定

字串指定是對進行寫入 / 讀取的其他站的軟元件以字串進行指定的方法。

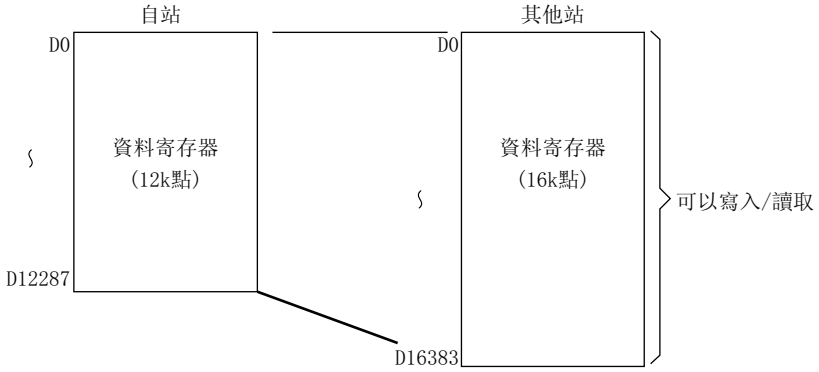
DP.DDWR 指令的字串指定程式



在字串指定中，可對其他站軟元件的所有範圍進行寫入 / 讀取。

例如，自站的資料寄存器爲 12k 點，其他站的資料寄存器爲 16k 點時，可對其他站的資料寄存器的起始開始的 16k 點的資料進行寫入 / 讀取。

字串指定時的寫入 / 讀取範圍



備 註

字串指定的注意事項如下所示。

- 字串指定中最多可指定 32 個字元。
- 無論在軟元件號的高位中是否附加了 “ 0 ”，均作爲相同的軟元件進行處理。
  例如，“ D1 ”與 “ D0001 ”均作爲 D1 處理。
- 以大寫字母指定的軟元件與以小寫字母字母指定的軟元件均作爲相同的軟元件進行處理。
  例如，“ D1 ”與 “ d1 ”均作爲 D1 進行處理。
- 通過字串對其他站 CPU 中不存在的軟元件進行了指定時，指令將變爲異常結束。

(4) 多 CPU 高速通信區的管理

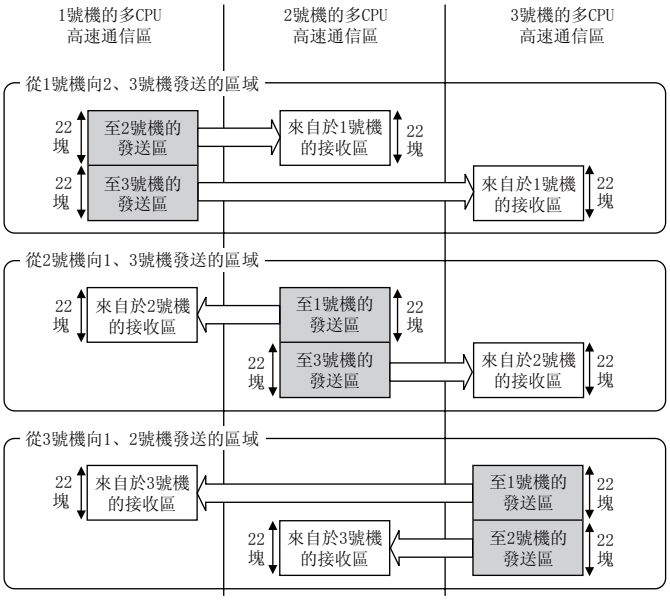
(a) 多 CPU 高速通信區是以 16 字作爲最小單位的塊進行管理的。

各站中可使用的塊數、指令中使用的塊數如下表所示。

CPU 個數	系統區 *1	
	1k 點	2k 點
2	46	110
3	22	54
4	14	35

*1: 關於系統區的設置，請參閱 QCPU 用戶手冊 (多 CPU 系統篇)。

(b) 在由 3 個 CPU 模組構成的多 CPU 系統中，系統區大小爲 1k 字時的多 CPU 高速通信區的構成如下所示。



(5) 指令中使用的塊數

指令中使用的塊數根據進行寫入的點數而不同。

指令中使用的塊數如下表所示。

指令中指定的寫入 / 讀取點數	D(P).DDWR 指令	D(P).DDRD 指令
1 ～ 4	1	1
5 ～ 20	2	
21 ～ 36	3	
37 ～ 52	4	
53 ～ 68	5	
69 ～ 84	6	
85 ～ 100	7	

(6) 可同時執行的多 CPU 高速通信專用指令

在通用型 CPU 中，可同時執行以下範圍內數量的多 CPU 高速通信專用指令。

[各站中可使用的塊數] ≧ [同時執行指令使用的塊數的合計]

由於執行多 CPU 高速通信專用指令，多 CPU 高速通信專用指令使用的塊數超過了多 CPU 高速通信區的總塊數時，在該掃描中不執行本指令（變爲無處理），在下一個掃描中再次執行本指令。但是，執行了本指令時，多 CPU 高速通信區的空閒塊數少於 SD796 ～ SD799（多 CPU 高速通信專用指令最多塊數設置）的設置值的情況下，本指令將變爲異常結束。

多 CPU 高速通信區的空閒塊數少於多 CPU 高速通信專用指令使用的塊數時或者少於 SD796 ～ SD799 的設置值時，多 CPU 高速通信專用指令的執行可否情況如下所示。

指令使用塊數*1與空閒的大小關係 / SD設置值與空閒塊數的大小關係	指令使用塊數*1 ≤空閒塊數*2	指令使用塊數*1＞空閒塊數*2
SD設置值*3≤空閒塊數*2	執行	不執行(執行無處理)
SD設置值*3＞空閒塊數*2	異常結束	

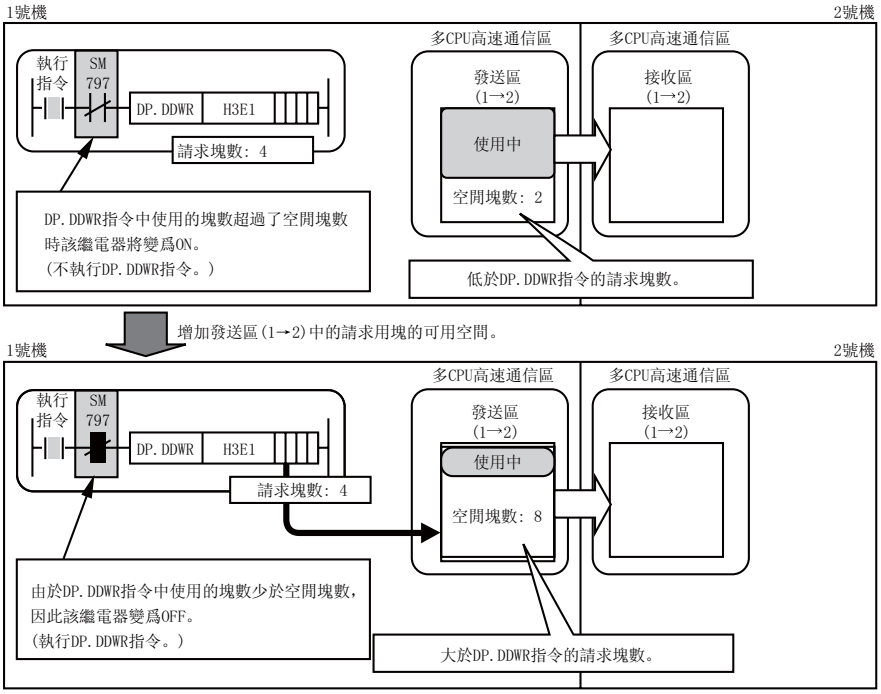
*1：多CPU高速通信專用指令使用的塊數。
*2：多CPU高速通信區的空閒塊數。
*3：SD796～SD799的設置值。

(7) 使用多 CPU 高速通信專用指令時的互鎖

(a) 作爲多 CPU 高速通信專用指令的互鎖，配備了特殊繼電器 SM796 ～ SM799(多 CPU 高速通信專用指令使用塊資訊)。
同時執行多個多 CPU 高速通信專用指令時，應將 SM796 ～ SM799 作爲 CPU 高速通信專用指令的互鎖使用。

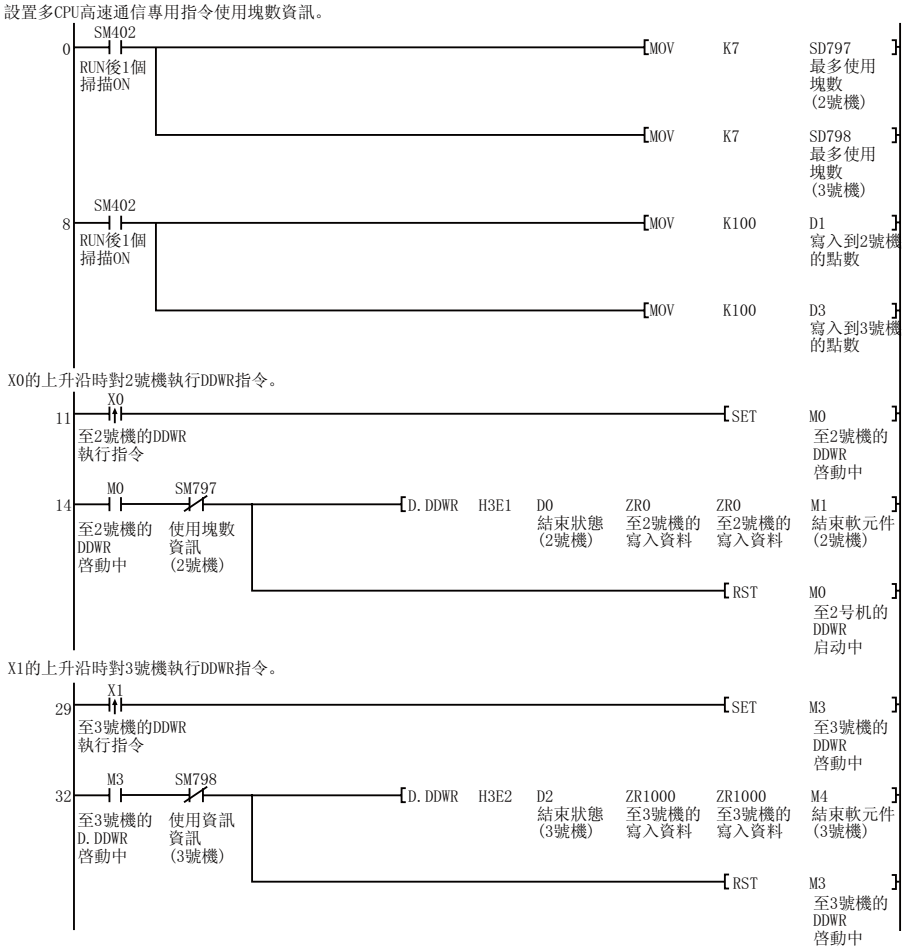
要 點

使用特殊繼電器 SM796 ～ SM799 時，應將各站中可使用指令的最多塊數設置到特殊寄存器 SD796 ～ SD799 中。(例如，對 3 號機執行多 CPU 高速通信專用指令的塊數的最大值爲 5 時，在 SD798 中設置 5。)
多 CPU 高速通信區的空閒塊數低於 SD796 ～ SD799 中設置的塊數時，相應特殊繼電器 (SM796 ～ SM799) 將變爲 ON。



(b) 將 SM796 ～ SM799 作爲互鎖使用時的程式示例

X0 的上升沿時對 2 號機執行 D.DDWR 指令，X1 的上升沿時對 3 號機執行 D.DDWR 指令的程式如下所示。



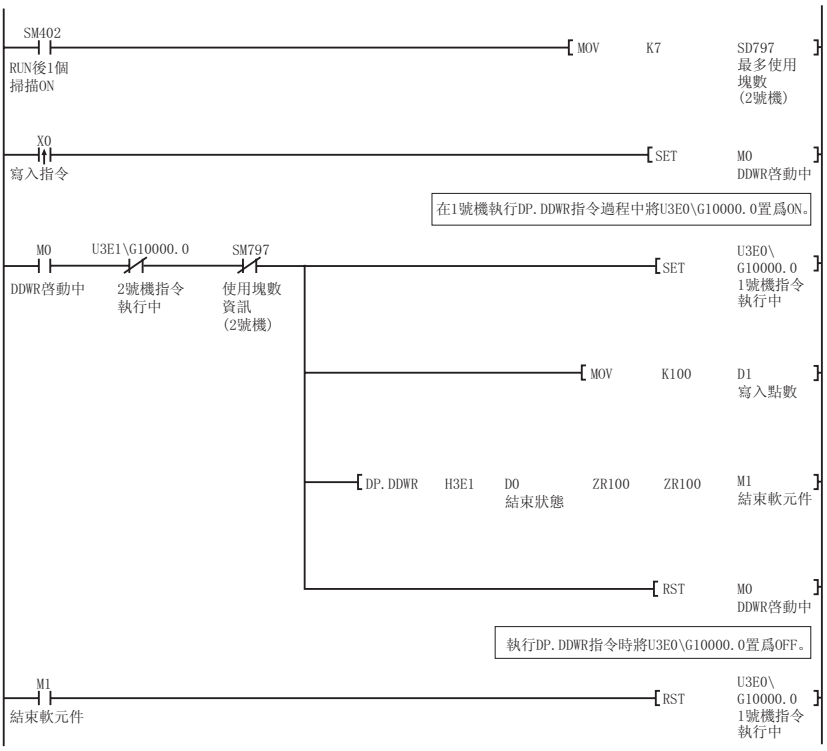
(8) 在多個 CPU 模組中相互執行多 CPU 高速通信專用指令時的程式示例

在通用型 QCPU 之間相互執行多 CPU 高速通信專用指令時，應採取互鎖以防止同時執行多 CPU 高速通信專用指令。

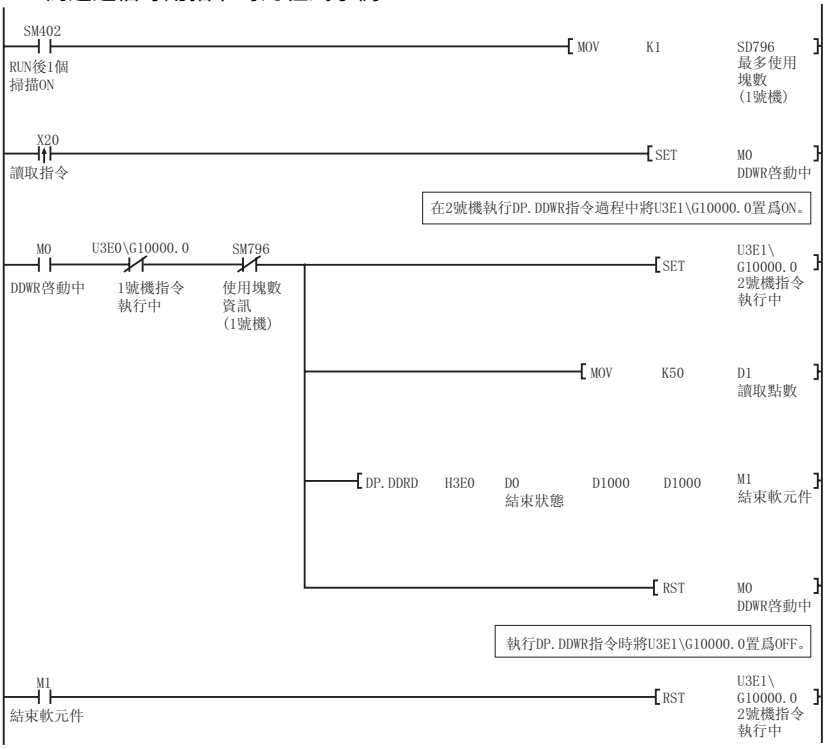
在互鎖中使用多 CPU 共用軟元件 (U3En\G10000 ～)。

在 1 號機與 2 號機之間相互執行多 CPU 高速通信專用指令時的程式示例如下所示。

在 1 號機中執行多 CPU 高速通信專用指令時的程式示例



在 2 號機中執行多 CPU 高速通信專用指令時的程式示例



(9) 通過多 CPU 高速通信專用指令對超過 100 字的資料進行寫入 / 讀取時的程式示例

多 CPU 高速通信專用指令可處理的寫入 / 讀取點數最多爲 100 字。對超過 100 字的資料進行寫入 / 讀取時，通過多次執行多 CPU 高速通信專用指令來實現。

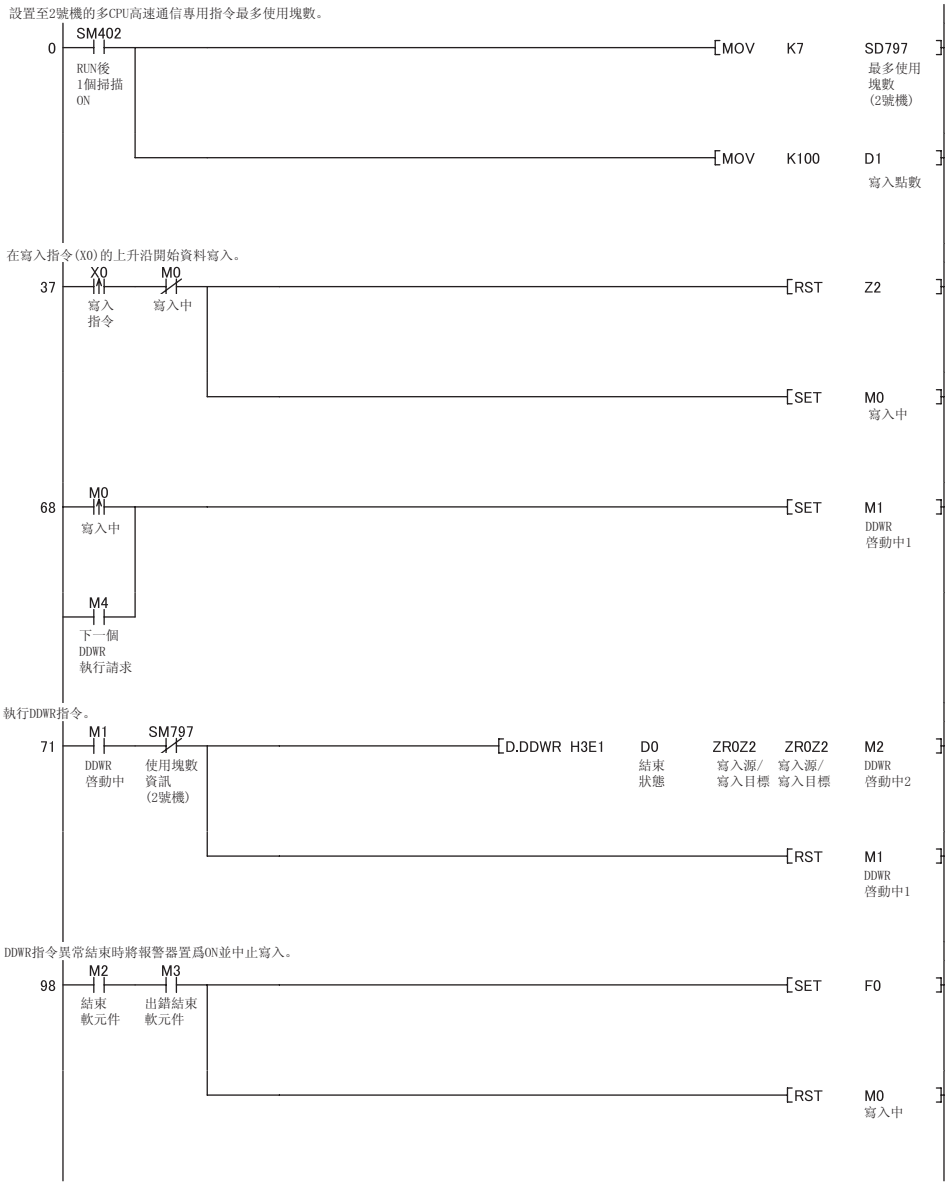
此外，下圖所示爲使用了多 CPU 高速通信專用指令的 D(P).DDWR 指令的程式示例，使用多 CPU 高速通信專用指令的 D(P).DDRD 指令時，可以使用與下圖的程式示例相同構成的程式。

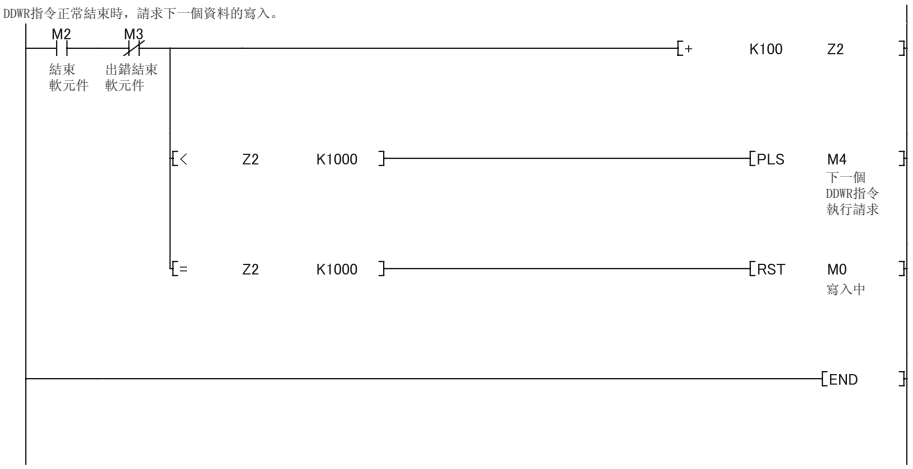
(a) 僅啓動 1 個 D(P).DDWR 指令的程式示例

使用 D.DDWR 指令，將 1 號機的 ZR0 ～ ZR999(1000 點) 寫入到 2 號機的 ZR0 ～ ZR999 中的程式示例如下所示。

在下圖的示例中，通過 D.DDWR 指令的結束軟元件 (M2) 的 ON，啓動下一個 D.DDWR 指令，以實現每次僅執行 1 個 D.DDWR 指令。

僅啓動 1 個 D(P).DDWR 指令的程式示例



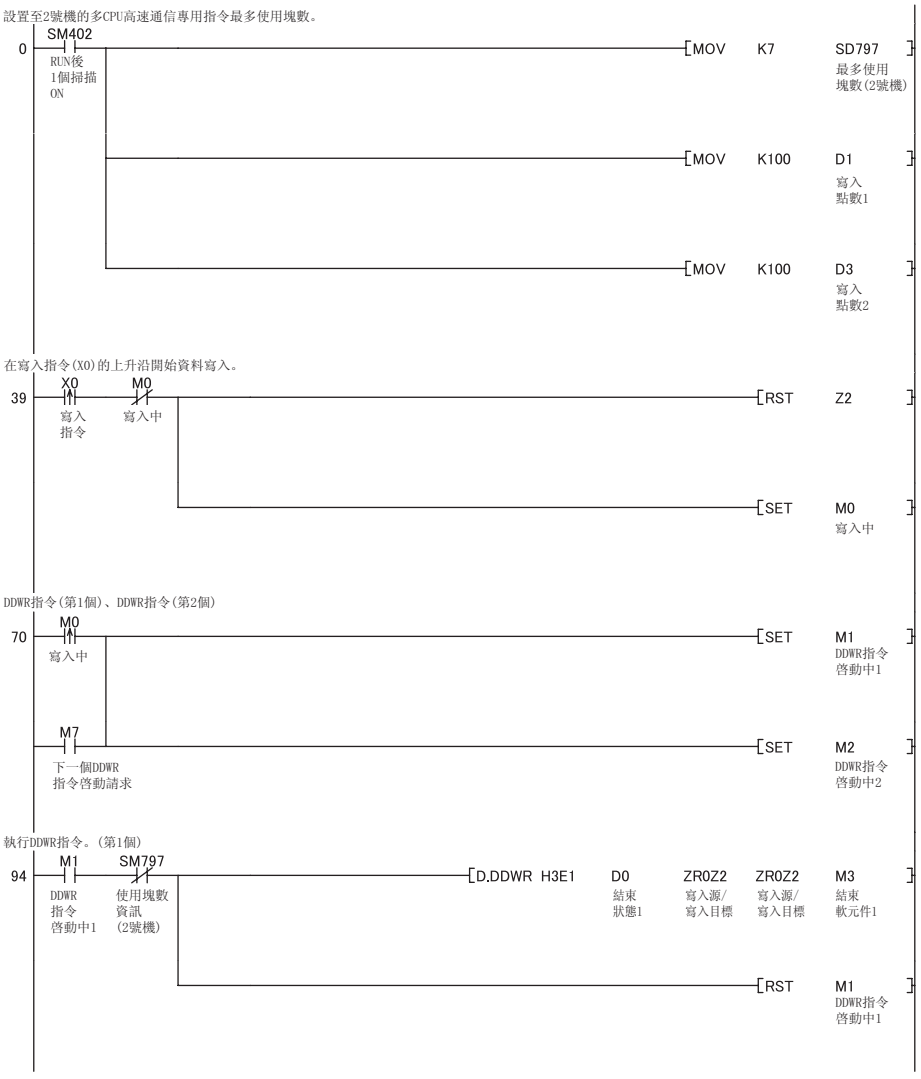


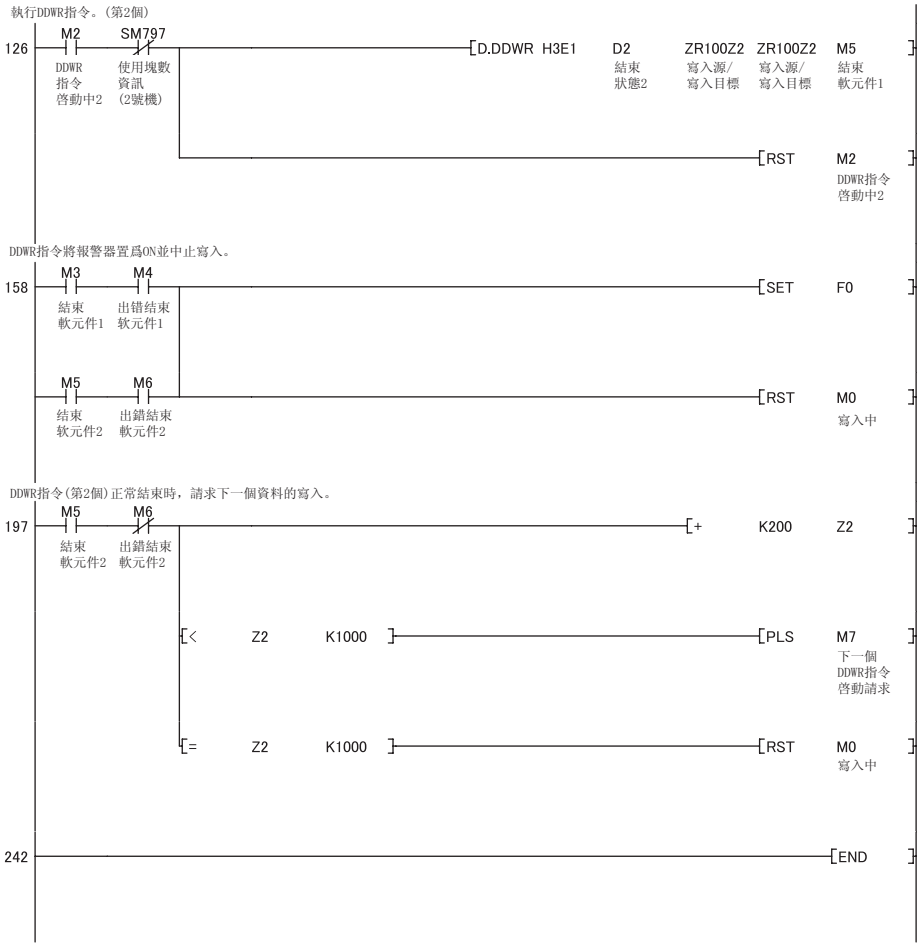
(b) 同時啓動 2 個以上 D(P).DDWR 指令的程式示例

使用 D.DDWR 指令，將 1 號機的 ZR0 ～ ZR999(1000 點) 寫入到 2 號機的 ZR0 ～ ZR999 中的程式示例如下所示。

如下圖程式示例所示，同時啓動 2 個以上多 CPU 高速通信專用指令的軟元件寫入 / 讀取時，多 CPU 高速通信區 (發送區) 的總塊數越多，則至多 CPU 高速通信專用指令的寫入 / 讀取結束爲止的時間便可越短。

同時啓動多個 D(P).DDWR 指令的程式示例

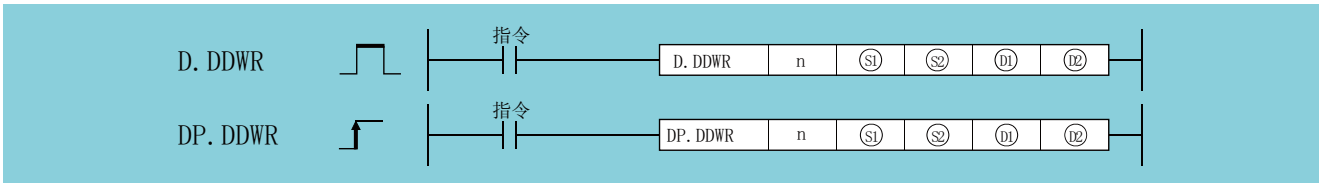



執行DDWR指令。(第2個)
126
M2
DDWR
指令
啓動中2
SM797
使用塊數
資訊
(2號機)
D.DDWR H3E1
D2
結束
狀態2
ZR100Z2
寫入源/
寫入目標
ZR100Z2
寫入源/
寫入目標
M5
結束
軟元件1
RST
M2
DDWR指令
啓動中2
DDWR指令將報警器置爲ON並中止寫入。
158
M3
結束
軟元件1
M4
出错結束
软元件1
SET
F0
M5
結束
软元件2
M6
出錯結束
軟元件2
RST
M0
寫入中
DDWR指令(第2個)正常結束時，請求下一個資料的寫入。
197
M5
結束
軟元件2
M6
出錯結束
軟元件2
+
K200
Z2
<
Z2
K1000
PLS
M7
下一個
DDWR指令
啓動請求
=
Z2
K1000
RST
M0
寫入中
242
END

Basic | High performance | Process | Redundant | Ver. Universal | LCPU

· 在序列號的前 5 位數爲 “ 10012 ” 以後的通用型 QCPU 中可以使用。
· 在 Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 中不能使用。

10.2 D.DDWR、DP.DDWR



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字 *5		位	字				
n *1	---	○	○	---				○	---
ⓢ1 *2	-	△ *3	△ *4	---				---	---
ⓢ2 *2	-	○	○	---				---	---
ⓓ1 *2	-	○	○	---				---	---
ⓓ2 *2	△ *6	-	△ *4	---				---	---

*1: 設置資料 n 不能進行變址修飾。
*2: 設置資料ⓢ1～ⓓ2可以進行變址修飾。
*3: 不能使用局部軟元件。
*4: 不能使用各程式的文件寄存器。
*5: 不能使用 FD、@□ (間接指定)。
*6: 不能使用 FX、FY。

設置資料

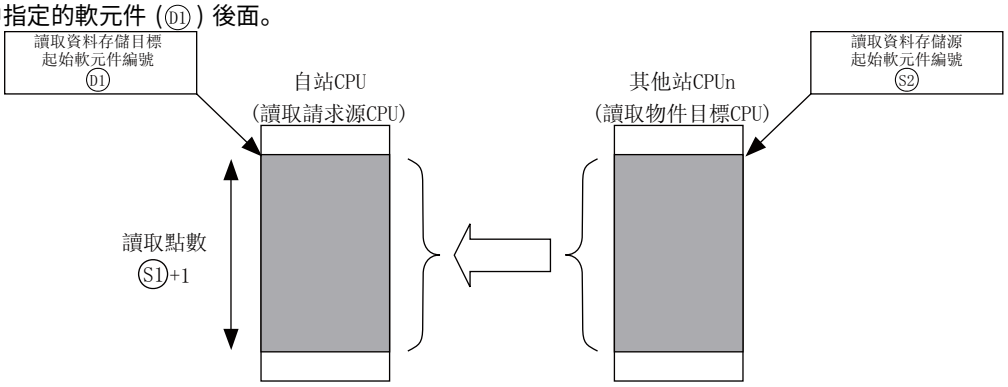
設置資料	內容	資料類型
n	其他站 CPU 的起始 I/O 編號÷ 16 1 號機 :$3E0_H$、2 號機： $3E1_H$、3 號機： $3E2_H$、4 號機： $3E3_H$	BIN16 位
ⓢ1	存儲控制資料的自站 CPU 的起始軟元件	軟元件名
ⓢ2	存儲寫入資料的自站 CPU 的起始軟元件	
ⓓ1	存儲寫入資料的其他站 CPU 的起始軟元件	軟元件名 *7 字串 *8*9
ⓓ2	結束軟元件	位

*7: 指定了文件寄存器 (R、ZR) 的情況下，在自站 CPU 中也可向超出範圍的其他站 CPU 的軟元件進行寫入。
*8: 通過用字串 “ ” 指定起始軟元件，在執行本指令的自站 CPU 中也可向超出範圍的其他站 CPU 的軟元件進行寫入。
*9: 不能指定進行了變址修飾後的軟元件。(例 D0Z0 等)

軟元件	專案	設置資料	設置範圍	設置方
ⓢ1+0	結束狀態	存儲指令結束時的執行結果。 $0000_{(H)}$: 無出錯 (正常結束) $0000_{(H)}$ 以外： 出錯代碼 (異常結束)	-	系統
ⓢ1+1	寫入資料點數	以字爲單位設置寫入資料點數。	1 ～ 100	用戶

功　能

(1) 多 CPU 系統配置時，將自站 CPU 中指定的軟元件 (Ⓢ2) 後面的資料，以 (Ⓢ1+1) 中指定的寫入資料點數，存儲到其他站 CPU(n) 中指定的軟元件 (Ⓓ1) 後面。



(2) D(P).DDWR 指令的正常 / 異常結束可以通過結束軟元件 (Ⓓ2+0)、結束時的狀態顯示軟元件 (Ⓓ2+1) 進行確認。

(a) 結束軟元件 (Ⓓ2+0)
在指令結束的掃描的 END 處理時變爲 ON。在下一個 END 處理時變爲 OFF。

(b) 結束時的狀態顯示軟元件 (Ⓓ2+1)
根據指令結束時的狀態而 ON/OFF。
- 正常結束時： OFF
- 異常結束時：在指令結束的掃描的 END 處理時變爲 ON。在下一個 END 處理時變爲 OFF。(異常結束時在控制資料 (Ⓢ1+0)：結束狀態) 中存儲出錯代碼。)

(3) 指令中使用的塊數取決於寫入資料點數。(參閱 665 頁 10.1 節)

指令中使用的塊數

指令中指定的寫入點數	D(P).DDWR 指令
1 ～ 4	1
5 ～ 20	2
21 ～ 36	3
37 ～ 52	4
53 ～ 68	5
69 ～ 84	6
85 ～ 100	7

(4) 在多 CPU 高速通信區中無空閒塊的情況下，即使執行指令也將異常結束。通過在特殊寄存器 (SD796 ～ SD799) 中設置指令中使用的塊數，將特殊繼電器 (SM796 ～ SM799) 用作互鎖，可以防止異常結束。(參閱 665 頁 10.1 節)

出　錯

在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4350	指定的其他站 CPU 有錯誤時。或者，進行了多 CPU 高速通信專用指令不能使用的設置時。 · 指定了已進行了預約設置的機號。 · 指定了未安裝的機號。 · 其他站 CPU 起始 I/O 編號÷ 16n 超出了 3E0H ～ 3E3H 的範圍。 · 在設置爲 “ 不使用多 CPU 高速通信功能 ” 的情況下執行了本指令。 · 在不能使用本指令的 CPU 模組中執行了本指令。 · 指定了自站 CPU。 · 指定了不能執行指令的 CPU。	---	---	---	---	○	---
4351	其他站 CPU 不支援本指令。	---	---	---	---	○	---
4352	軟元件數有錯誤時。	---	---	---	---	○	---
4353	指定了不能使用的軟元件時。	---	---	---	---	○	---
4354	以不能處理的字串指定了軟元件時。	---	---	---	---	○	---
4355	寫入資料點數 (Ⓢ1+1) 超出了 0 ～ 100 的範圍時。	---	---	---	---	○	---

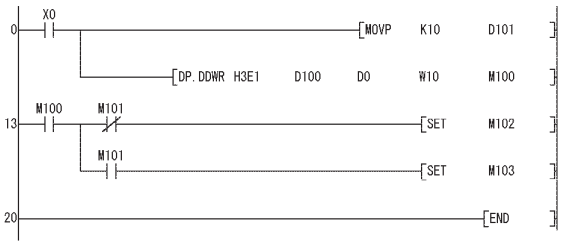
在以下情況下，將變爲異常結束，出錯代碼將被存儲到結束狀態存儲軟元件 (Ⓢ1+0) 中指定的軟元件中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
0010_H	至物件目標 CPU 的指令請求超出了允許值時。(多 CPU 高速通信區中無空閒塊)	---	---	---	---	○	---
1001_H	Ⓓ1中指定的其他站 CPU 的軟元件是在其他站 CPU 中不能使用的軟元件，或者超出了軟元件範圍。	---	---	---	---	○	---
1003_H	未從其他站 CPU 模組返回指令回應。(多 CPU 高速通信區中無空閒塊)	---	---	---	---	○	---
1080_H	D(P).DDWR 指令中設置的寫入資料點數爲 0 時。	---	---	---	---	○	---

程式示例

(1) 以下爲 X0 變爲 ON 時，將自站 CPU 的 D0 開始的 10 字資料存儲到 2 號機 CPU 的 W10 後面的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件				
0	LD	X0				
1	MOVP	K10	D101			
3	DP.DDWR	H3E1	D100	D0	W10	M100
13	LD	M100				
14	MPS					
15	ANI	M101				
16	SET	M102				
17	MPP					
18	AND	M101				
19	SET	M103				
20	END					

注意事項

(1) n、Ⓢ2以及Ⓓ1中可以進行位軟元件的位數指定。但是，在Ⓢ2以及Ⓓ1中進行位軟元件的位元數指定時，需要滿足下述條件。
· 是 16 位 (4 位數) 的位數指定。
· 開始位軟元件爲 16(10_H) 的倍數。

(2) 應在寫入物件的 CPU 處於啓動的狀態下執行本指令。
如果在寫入物件 CPU 處於未啓動狀態時執行本指令，將變爲無處理。

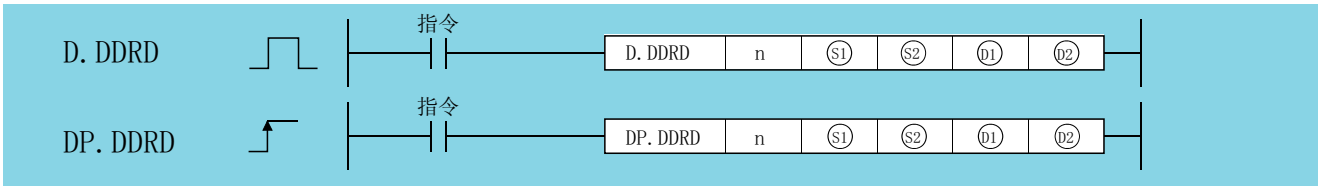
(3) 執行本指令後，在結束軟元件變爲 ON 之前，如果對設置資料中指定的軟元件的範圍等進行了變更，系統中存儲的資料 (結束狀態、結束軟元件) 將無法正常存儲。

(4) SB、SW、SM、SD 中包含有系統資訊區。
通過多 CPU 高速通信專用指令的 D(P).DDWR 指令對上述軟元件進行寫入時，應注意防止破壞系統資訊。

Basic × | High performance × | Process × | Redundant × | Ver. Universal | LCPU ×

· 在序列號的前 5 位數爲 “ 10012 ” 以後的通用型 QCPU 中可以使用。
· 在 Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 中不能使用。

10.3 D.DDRD、DP.DDRD



設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字 *5		位	字				
n *1	---	○	○	---				○	---
S1 *2	-	△ *3	△ *4	---				---	---
S2 *2	-	○	○	---				---	---
D1 *2	-	○	○	---				---	---
D2 *2	△ *6	-	△ *4	---				---	---

*1: 設置資料 n 不能進行變址修飾。
*2: 設置資料Ⓢ1～Ⓓ2可以進行變址修飾。
*3: 不能使用局部軟元件。
*4: 不能使用各程式的文件寄存器。
*5: 不能使用 FD、@□ (間接指定)。
*6: 不能使用 FX、FY。

設置資料

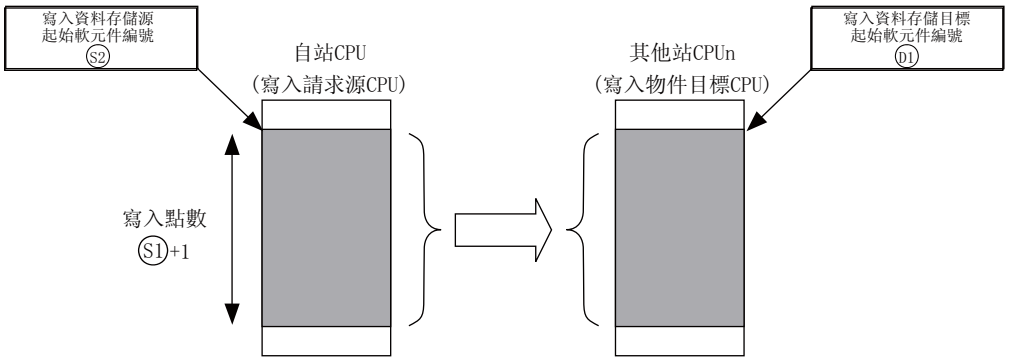
設置資料	內容	資料類型
n	其他站 CPU 的起始 I/O 編號÷ 16 1 號機 :$3E0_H$、2 號機： $3E1_H$、3 號機： $3E2_H$、4 號機： $3E3_H$	BIN16 位元
S1	存儲控制資料的自站 CPU 的起始軟元件	軟元件名
S2	存儲讀取資料的其他站 CPU 的起始軟元件	軟元件名 *7 字串 *8*9
D1	存儲讀取資料的自站 CPU 的起始軟元件	軟元件名
D2	結束軟元件	位元

*7: 指定了文件寄存器 (R、ZR) 的情況下，在自站 CPU 中也可向超出範圍的其他站 CPU 的軟元件進行讀取。
*8: 通過用字串 “ ” 指定起始軟元件，在執行本指令的自站 CPU 中也可向超出範圍的其他站 CPU 的軟元件進行讀取。
*9: 不能指定進行了變址修飾的軟元件。(例 D0Z0 等)

軟元件	專案	設置資料	設置範圍	設置方
S1+0	結束狀態	存儲指令結束時的執行結果。 $0000_{(H)}$: 無出錯 (正常結束) $0000_{(H)}$ 以外： 出錯代碼 (異常結束)	-	系統
S1+1	讀取資料點數	以字爲單位設置讀取資料點數。	1 ～ 100	用戶

功 能

(1) 多 CPU 系統配置時，將其他站 CPU(n) 中指定的軟元件 (Ⓓ1) 後面的資料，以 (Ⓢ1)+1) 中指定的讀取資料點數，存儲到自站 CPU 中指定的軟元件 (Ⓢ2) 後面。



(2) D(P).DDRD 指令的正常 / 異常結束可以通過結束軟元件 (Ⓓ2+0)、結束時的狀態顯示軟元件 (Ⓓ2+1) 進行確認。

(a) 結束軟元件 (Ⓓ2+0)

在指令結束的掃描的 END 處理時變爲 ON。在下一個 END 處理時變爲 OFF。

(b) 結束時的狀態顯示軟元件 (Ⓓ2+1)

根據指令結束時的狀態而 ON/OFF。

- 正常結束時： OFF
- 異常結束時：在指令結束的掃描的 END 處理時變爲 ON。在下一個 END 處理時變爲 OFF。(異常結束時在控制資料 (Ⓢ1+0: 結束狀態) 中存儲出錯代碼。)

(3) 指令中使用的塊數取決於讀取的資料點數。

指令中使用的塊數

指令中指定的讀取點數	D(P).DDRD 指令
1 ～ 100	1

(4) 在多 CPU 高速通信區中無空閒塊的情況下，即使執行指令也將異常結束。通過在特殊寄存器 (SD796 ～ SD799) 中設置指令中使用的塊數，將特殊繼電器 (SM796 ～ SM799) 用作互鎖，可以防止異常結束。(參閱 665 頁 10.1 節)

出　錯

在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4350	指定的其他站 CPU 有錯誤時。或者，進行了多 CPU 高速通信專用指令不能使用的設置時。 · 指定了已進行了預約設置的機號。 · 指定了未安裝的機號。 · 其他站 CPU 起始 I/O 編號÷ 16n 超出了 3E0H ～ 3E3H 的範圍。 · 在設置爲 “ 不使用多 CPU 高速通信功能 ” 的情況下執行了本指令。 · 在不能使用本指令的 CPU 模組中執行了本指令。 · 指定了自站 CPU。 · 指定了不能執行指令的 CPU。	---	---	---	---	○	---
4351	其他站 CPU 不支援本指令時。	---	---	---	---	○	---
4352	軟元件數有錯誤時。	---	---	---	---	○	---
4353	指定了不能使用的軟元件時。	---	---	---	---	○	---
4354	以不能處理的字串指定了軟元件時。	---	---	---	---	○	---
4355	讀取資料點數 (Ⓢ1+1) 超出了 0 ～ 100 的範圍時。	---	---	---	---	○	---

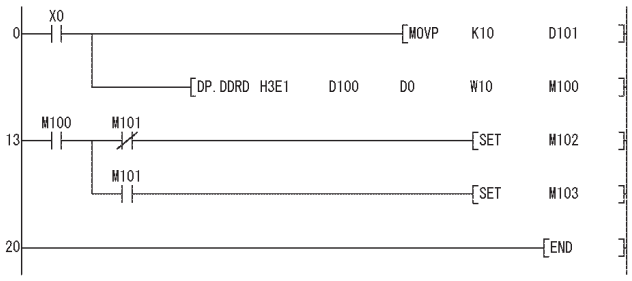
在以下情況下，將變爲異常結束，出錯代碼將被存儲到結束狀態存儲軟元件 (Ⓢ1+0) 中指定的軟元件中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
0010_H	至物件目標 CPU 的指令請求超出了允許值時。(多 CPU 高速通信區中無空閒塊)	---	---	---	---	○	---
1001_H	Ⓢ2中指定的其他站 CPU 的軟元件是在其他站 CPU 中不能使用的軟元件，或者超出了軟元件範圍時。	---	---	---	---	○	---
1003_H	未從其他站 CPU 模組返回指令回應。(多 CPU 高速通信區中無空閒塊)	---	---	---	---	○	---
1081_H	D(P).DDRD 指令中設置的讀取資料點數爲 0 時。	---	---	---	---	○	---

程式示例

(1) 以下爲 X0 變爲 ON 時，將 2 號機 CPU 的 D0 開始的 10 字資料存儲到自站 CPU 的 W10 後面的程式。

[梯形圖模式]



[列表模式]

步	指令	軟元件				
0	LD	X0				
1	MOVP	K10	D101			
3	DP.DDRD	H3E1	D100	D0	W10	M100
13	LD	M100				
14	MPS					
15	AN!	M101				
16	SET	M102				
17	MPP					
18	AND	M101				
19	SET	M103				
20	END					

注意事項

(1) n、Ⓢ2以及Ⓓ1中可以進行位軟元件的位數指定。但是，在Ⓢ2以及Ⓓ1中進行位軟元件的位元數指定時，需要滿足下述條件。
· 是 16 位 (4 位數) 的位數指定。
· 開始位軟元件爲 16(10_H) 的倍數。

(2) 應在讀取物件的 CPU 處於啓動的狀態下執行本指令。 如果在讀取物件 CPU 處於未啓動狀態時執行本指令，將變爲無處理。

(3) 執行本指令後，在結束軟元件變爲 ON 之前，如果對設置資料中指定的軟元件的範圍等進行了變更，系統中存儲的資料 (結束狀態、結束軟元件) 將無法正常存儲。

第 11 章 冗餘系統指令 (用於冗餘 CPU)

11.1 SP.CONTSW

Basic | High performance | Process | Redundant | Universal | LCPU

Ⓢ : 用於指定發出系統切換請求處理的 0 以外的值 (BIN16 位)。

Ⓓ : 異常結束軟元件編號 (位)。

設置資料	內部軟元件		R、ZR	J□\□		U□\G□	Zn	常數 K、H	其他
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	○	○ *1		---				---	---

*1: 可以使用字軟元件的位指定。

功 能

(1) 在執行了 SP.CONTSW 指令的掃描的 END 處理時，進行控制系統與待機系統的切換。

(2) 通過 SP.CONTSW 指令進行系統切換時，需要預先將 “ 手動切換允許標誌 (SM1592) ” 置爲 ON(允許)。

(3) 使用了多個 SP.CONTSW 指令時，通過對Ⓢ發生了系統切換的程式的處理塊進行掌控。

Ⓢ中的值的允許指定範圍爲 -32768 ～ -1，1 ～ 32767(1_H ～ $FFFF_H$)。

通過 SP.CONTSW 指令指定的Ⓢ的值在系統切換正常結束時將被存儲到出錯公共資訊的 “ 系統切換指令變數 (SD6) ” 中。*2
在同一個掃描中執行了多個 SP.CONTSW 指令時，最先執行的 SP.CONTSW 指令的變數將被存儲到系統切換指令變數 (SD6) 中。

(4) SP.CONTSW 指令中指定的Ⓢ的值在系統切換正常結束時，將被存儲到新控制系統 CPU 模組的 “ 系統切換指令變數 (SD1602) ” 中。*3
通過在新控制系統 CPU 模組中讀取 SD1602，可以確認通過哪個 SP.CONTSW 指令進行了系統切換。

*2: 通過 SP.CONTSW 指令指定Ⓢ的值可以在 GX Developer 的可程式控制器診斷的出錯公共資訊中確認。

*3: 新控制系統 CPU 模組是指，通過 SP.CONTSW 指令進行系統切換後，由待機系統被切換爲控制系統的 CPU 模組。

(5) 在通過 SP.CONTSW 指令未能完成系統切換時，在控制系統 CPU 模組中異常結束軟元件將變爲 ON。

(a) 執行 SP.CONTSW 指令的情況下，如果由於下述原因檢測出 OPERATION ERROR，執行 SP.CONTSW 指令時異常結束軟元件將變爲 ON。

・執行的 SP.CONTSW 指令的Ⓢ中指定了 0 時。
・“ 手動切換允許標誌 (SM1592) ” 處於 OFF 狀態時。
・在分開模式下的待機系統中執行了 SP.CONTSW 指令時。
・在調試模式下執行了 SP.CONTSW 指令時。

(b) 在由於下表中的原因導致系統未能完成切換的情況下，在 END 處理的系統切換執行時異常結束軟元件將變爲 ON。

系統切換失敗原因號	系統切換失敗原因
0	正常結束。
1	熱備電纜脫落或者熱備電纜異常。
2	發生了待機系統硬體異常、電源 OFF、處於重定模式、看門狗計時器出錯。
3	控制系統發生了看門狗計時器出錯。
4	熱備傳送準備中。
5	通信超時。
6	待機系統中發生停止出錯。(看門狗計時器出錯除外)
7	控制系統及待機系統的動作狀態異常。
8	正在從控制系統向待機系統進行記憶體複製。
9	正在進行 RUN 中寫入。
10	待機系統中檢測出網路異常。

系統切換未能完成，異常結束軟元件變爲 ON 時，在 “ 系統切換原因 (SD1588) ” 中將存儲 16，在 “ 系統切換失敗原因 (SD1589) ” 中將存儲上表中的系統切換失敗原因號。

(6) 對於處於 ON 狀態的異常結束位元，應通過用戶程式或者 GX Developer 將其變爲 OFF。

在異常結束軟元件處於 ON 的狀態下，如果執行 SP.CONTSW 指令後正常地進行了系統切換，在新待機系統 CPU 模組的異常結束軟元件也將變爲 OFF。

但是，在由於 SP.CONTSW 指令以外的原因導致系統切換未能完成時，異常結束軟元件將不變爲 OFF。

出　錯

(1) 在以下情況下將變爲出錯狀態，出錯標誌 (SM0) 將 ON，出錯代碼將被存儲到 SD0 中。

出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
4110	執行 SP.CONTSW 指令時，在Ⓢ中指定了 0 時。	---	---	---	○	---	---
4120	執行 SP.CONTSW 指令時，手動切換允許標誌 (SM1592) 處於 OFF(禁止) 狀態時。	---	---	---	○	---	---
4121	在分開模式下的待機系統 CPU 模組中執行了 SP.CONTSW 指令時。 在調試模式下執行了 SP.CONTSW 指令時。	---	---	---	○	---	---

(2) 在系統切換未能完成的情況下，出錯標誌 (SM0) 將變爲 ON，出錯代碼將被存儲到 SD0 中。

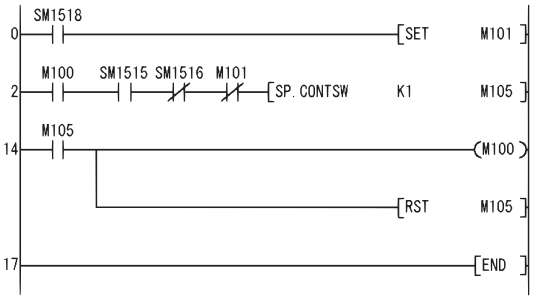
出錯代碼	出錯內容	Q00J/Q00/Q01	QnH	QnPH	QnPRH	QnU	LCPU
6220	熱備電纜脫落或者熱備電纜異常時。 待機系統中發生了硬體異常、電源 OFF、處於重定模式、看門狗計時器出錯時。 控制系統中發生了看門狗計時器出錯時。 處於熱備傳送準備中狀態時。 發生了通信超時時。 待機系統中發生了除看門狗計時器出錯以外的停止型出錯時。 控制系統與待機系統的動作狀態不匹配時。 正在從控制系統向待機系統進行記憶體複製時。 正在進行 RUN 中寫入時。 待機系統中檢測出網路異常時。	---	---	---	○	---	---

程式示例

(1) 以下爲在系統切換指令 (M100) 的上升沿進行系統切換的程式。

如果系統切換指令 (M100) 保持爲 ON 狀態不變，系統切換後新控制系統 CPU 模組中也將執行 SP.CONTSW 指令，因此將 M101 附加到執行條件中作爲防止連續切換標誌使用。

[梯形圖模式]



[列表模式]

步	指令	軟元件		
0	LD	SM1518		
1	SET	M101		
2	LD	M100		
3	AND	SM1515		
4	ANI	SM1516		
5	ANI	M101		
6	SP. CONTSW		K1	M105
14	LD	M105		
15	OUT	M100		
16	RST	M105		
17	END			

附錄

附錄 1 運算處理時間

附錄 1.1 運算處理時間的思路

(1) QCPU、LCPU 的處理時間是下述處理時間的合計。
- 各指令的處理時間的合計
- END 處理時間（包括 I/O 刷新時間）
- 掃描時間延遲功能的處理時間

(2) 各指令的處理時間
是 687 頁附錄 1.2、701 頁附錄 1.3、723 頁附錄 1.4 中記述的各指令的處理時間的合計時間。

(3) END 處理時間、I/O 刷新時間、掃描時間延遲功能的處理時間
關於 END 處理時間、I/O 刷新時間、掃描時間延遲功能的處理時間的內容，請參閱以下手冊。
- QnUCPU 用戶手冊（功能解說 / 程式基礎篇）
- Qn(H)/QnPH/QnPRHCPU 用戶手冊（功能解說 / 程式基礎篇）
- MELSEC-LCPU 用戶手冊（功能解說 / 程式基礎篇）

附錄 1.2 基本型 QCPU 的運算處理時間

各指令的運算處理時間如本頁以後的表中所示。

由於運算處理時間根據源、目標的內容而有所不同，因此表中的值只應作爲處理時間的大致參考。

要 點

使用文件寄存器 (ZR)、模組訪問軟元件 (Un\G □、U3En\G0 ～ G511)、鏈結直接軟元件 (Jn\ □) 的情況下，請參閱 687 頁 附錄 1.2(6) 的加法運算時間，對各指令的處理時間進行加法運算。

(1) 順控程式指令

指令		條件 (軟元件)		處理時間 (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		未執行時		0.20	0.16	0.10
		執行時				
MEP MEF		未執行時		0.30	0.24	0.15
		執行時				
EGP		未執行時	(OFF → OFF) (ON → ON)	0.20	0.16	0.10
		執行時	(OFF → ON) (ON → OFF)			
EGF		未執行時	(OFF → OFF) (ON → ON)	17	9.5	9.4
		執行時	(OFF → ON) (ON → OFF)	18	14	14
OUT	Y	無變化時	(OFF → OFF) (ON → ON)	0.20	0.16	0.10
		變化時	(OFF → ON) (ON → OFF)	0.20	0.16	0.10
	D0.0	無變化時	(OFF → OFF) (ON → ON)	0.40	0.32	0.20
		變化時	(OFF → ON) (ON → OFF)	0.40	0.32	0.20
	F	OFF 時		24	20	19
		ON 時	顯示時	260	210	200
			顯示結束	205	165	155

指令		條件（軟元件）			處理時間（μs）		
					Q00JCPU	Q00CPU	Q01CPU
OUT	T	未執行時			1.1	0.88	0.55
		執行時	時間到後		1.1	0.88	0.55
			加法運算時	K	1.1	0.88	0.55
				D	1.2	0.96	0.60
	C	未執行時			1.1	0.88	0.55
		執行時	時間到後		1.1	0.88	0.55
			加法運算時	K	1.1	0.88	0.55
				D	1.2	0.96	0.60
OUTH	T	未執行時			1.1	0.88	0.55
		執行時	時間到後		1.1	0.88	0.55
			加法運算時	K	1.1	0.88	0.55
				D	1.2	0.96	0.60
SET	Y	未執行時			0.20	0.16	0.10
		執行時	無變化時 (ON → ON)		0.20	0.16	0.10
			變化時 (OFF → ON)		0.20	0.16	0.10
	D0.0	未執行時			0.40	0.32	0.20
		執行時	無變化時 (ON → ON)		0.40	0.32	0.20
			變化時 (OFF → ON)		0.40	0.32	0.20
	F	未執行時			0.50	0.44	0.25
		執行時	顯示時		255	205	195
			顯示結束		195	160	150
RST	Y	未執行時			0.20	0.16	0.10
		執行時	無變化時 (OFF → OFF)		0.20	0.16	0.10
			變化時 (ON → OFF)		0.20	0.16	0.10
	D0.0	未執行時			0.40	0.32	0.20
		執行時	無變化時 (ON → ON)		0.40	0.32	0.20
			變化時 (OFF → ON)		0.40	0.32	0.20
	SM	未執行時			0.20	0.16	0.10
		執行時			0.20	0.16	0.10
	F	未執行時			0.48	0.44	0.25
		執行時	顯示時		75	69	65
			顯示結束		43	35	33
	T,C	未執行時			0.80	0.64	0.40
		執行時			1.0	0.80	0.50
	D	未執行時			0.40	0.32	0.20
		執行時			0.60	0.48	0.30
	Z	未執行時			0.50	0.40	0.25
		執行時			9.4	7.9	7.4
	R	未執行時			---	0.32	0.20
		執行時			---	0.48	0.30
PLS					12	9.5	9.2
PLF					11	9.5	8.9
FF	Y	未執行時			0.68	0.40	0.25
		執行時			7.5	6.2	5.7
DELTA	DY0	未執行時			0.50	0.40	0.25
		執行時			26	21	21
DELTAP	DY0	未執行時			0.48	0.40	0.25
		執行時			58	45	43
SFT SFTP		未執行時			0.50	0.34	0.25
		執行時			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		進行出錯檢查			660	600	520
		不進行出錯檢查 （・電池檢查） （・保險絲熔斷檢查） （・I/O 模組校驗）			660	600	520
NOP		---			0.20	0.16	0.10
NOPLF PAGE		---			0.20	0.16	0.10

(2) 基本指令

未執行時的處理時間如下所示。

Q00JCPU ..... 0.20 × (各指令的步數 +1) μs

Q00CPU ..... 0.16 × (各指令的步數 +1) μs

Q01CPU ..... 0.10 × (各指令的步數 +1) μs

指令	條件 (軟元件)		處理時間 (μs)		
			Q00JCPU	Q00CPU	Q01CPU
LD=	導通時		0.80	0.64	0.40
	非導通時		0.80	0.64	0.40
AND=	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
OR=	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
LD< >	導通時		0.80	0.64	0.40
	非導通時		0.80	0.64	0.40
AND< >	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
OR< >	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
LD>	導通時		0.80	0.64	0.40
	非導通時		0.80	0.64	0.40
AND>	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
OR>	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
LD<=	導通時		0.80	0.64	0.40
	非導通時		0.80	0.64	0.40
AND<=	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
OR<=	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
LD<	導通時		0.80	0.64	0.40
	非導通時		0.80	0.64	0.40
AND<	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
OR<	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
LD>=	導通時		0.80	0.64	0.40
	非導通時		0.80	0.64	0.40
AND>=	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
OR>=	未執行時		0.70	0.56	0.35
	執行時	導通時	0.80	0.64	0.40
		非導通時	0.80	0.64	0.40
LDD=	導通時		1.0	0.80	0.50
	非導通時		1.0	0.80	0.50
ANDD=	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50

指令	條件（軟元件）		處理時間（μs）		
			Q00JCPU	Q00CPU	Q01CPU
ORD=	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50
LDD< >	導通時		1.0	0.80	0.50
	非導通時		1.0	0.80	0.50
ANDD< >	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50
ORD< >	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50
LDD>	導通時		1.0	0.80	0.50
	非導通時		1.0	0.80	0.50
ANDD>	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50
ORD>	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50
LDD<=	導通時		1.0	0.80	0.50
	非導通時		1.0	0.80	0.50
ANDD<=	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50
ORD<=	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50
LDD<	導通時		1.0	0.80	0.50
	非導通時		1.0	0.80	0.50
ANDD<	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50
ORD<	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50
LDD>=	導通時		1.0	0.80	0.50
	非導通時		1.0	0.80	0.50
ANDD>=	未執行時		0.80	0.64	0.40
	執行時	導通時	1.0	0.80	0.50
		非導通時	1.0	0.80	0.50
ORD>=	未執行時		0.80	0.64	0.40
	執行時	导通时	1.0	0.80	0.50
		非导通时	1.0	0.80	0.50
BKCMP= ⓢ1 ⓢ2 Ⓓ n	n=1		130	105	97
BKCMP=P ⓢ1 ⓢ2 Ⓓ n	n=96		205	175	165
BKCMP< > ⓢ1 ⓢ2 Ⓓ n	n=1		130	105	98
BKCMP< >P ⓢ1 ⓢ2 Ⓓ n	n=96		210	180	165
BKCMP> ⓢ1 ⓢ2 Ⓓ n	n=1		130	105	97
BKCMP>P ⓢ1 ⓢ2 Ⓓ n	n=96		210	180	165
BKCMP>= ⓢ1 ⓢ2 Ⓓ n	n=1		130	105	98
BKCMP>=P ⓢ1 ⓢ2 Ⓓ n	n=96		205	175	165
BKCMP< ⓢ1 ⓢ2 Ⓓ n	n=1		130	105	98
BKCMP<P ⓢ1 ⓢ2 Ⓓ n	n=96		210	180	165
BKCMP<= ⓢ1 ⓢ2 Ⓓ n	n=1		130	105	97
BKCMP<=P ⓢ1 ⓢ2 Ⓓ n	n=96		205	175	165

指令	條件（軟元件）	處理時間（μs）		
		Q00JCPU	Q00CPU	Q01CPU
+ Ⓢ Ⓓ +P Ⓢ Ⓓ	執行時	1.0	0.80	0.50
+ Ⓢ1 Ⓢ2 Ⓓ +P Ⓢ1 Ⓢ2 Ⓓ	執行時	1.2	0.96	0.60
- Ⓢ Ⓓ -P Ⓢ Ⓓ	執行時	1.0	0.80	0.50
- Ⓢ1 Ⓢ2 Ⓓ -P Ⓢ1 Ⓢ2 Ⓓ	執行時	1.2	0.96	0.60
D+ Ⓢ Ⓓ D+P Ⓢ Ⓓ	執行時	1.3	1.04	0.65
D+ Ⓢ1 Ⓢ2 Ⓓ D+P Ⓢ1 Ⓢ2 Ⓓ	執行時	1.5	1.2	0.75
D- Ⓢ Ⓓ D-P Ⓢ Ⓓ	執行時	1.3	1.04	0.65
D- Ⓢ1 Ⓢ2 Ⓓ D-P Ⓢ1 Ⓢ2 Ⓓ	執行時	1.5	1.2	0.75
* Ⓢ1 Ⓢ2 Ⓓ *P Ⓢ1 Ⓢ2 Ⓓ	執行時	1.1	0.88	0.55
/ Ⓢ1 Ⓢ2 Ⓓ /P Ⓢ1 Ⓢ2 Ⓓ	---	19	16	15
D* Ⓢ1 Ⓢ2 Ⓓ D*P Ⓢ1 Ⓢ2 Ⓓ	---	41	34	31
D/ Ⓢ1 Ⓢ2 Ⓓ D/P Ⓢ1 Ⓢ2 Ⓓ	---	28	23	21
B+ Ⓢ Ⓓ B+P Ⓢ Ⓓ	---	34	28	26
B+ Ⓢ1 Ⓢ2 Ⓓ B+P Ⓢ1 Ⓢ2 Ⓓ	---	47	39	37
B- Ⓢ Ⓓ B-P Ⓢ Ⓓ	---	34	28	26
B- Ⓢ1 Ⓢ2 Ⓓ B-P Ⓢ1 Ⓢ2 Ⓓ	---	48	40	38
DB+ Ⓢ Ⓓ DB+P Ⓢ Ⓓ	---	58	48	44
DB+ Ⓢ1 Ⓢ2 Ⓓ DB+P Ⓢ1 Ⓢ2 Ⓓ	---	60	49	46
DB- Ⓢ Ⓓ DB-P Ⓢ Ⓓ	---	59	48	45
DB- Ⓢ1 Ⓢ2 Ⓓ DB-P Ⓢ1 Ⓢ2 Ⓓ	---	60	51	45
B* Ⓢ1 Ⓢ2 Ⓓ B*P Ⓢ1 Ⓢ2 Ⓓ	---	42	35	33
B/ Ⓢ1 Ⓢ2 Ⓓ B/P Ⓢ1 Ⓢ2 Ⓓ	---	48	40	37

指令	條件（軟元件）	處理時間（μs）		
		Q00JCPU	Q00CPU	Q01CPU
DB* ⓢ1 ⓢ2 ⓓ DB*P ⓢ1 ⓢ2 ⓓ	---	140	120	110
DB/ ⓢ1 ⓢ2 ⓓ DB/P ⓢ1 ⓢ2 ⓓ	---	83	69	65
BK+ ⓢ1 ⓢ2 ⓓ n	n=1	105	86	80
BK+P ⓢ1 ⓢ2 ⓓ n	n=96	185	155	140
BK- ⓢ1 ⓢ2 ⓓ n	n=1	105	86	80
BK-P ⓢ1 ⓢ2 ⓓ 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
BKBCD ⓢ ⓓ n	n=1	78	63	57
BKBCDP ⓢ ⓓ n	n=96	315	275	250
BKBIN ⓢ ⓓ n	n=1	74	61	57
BKBINP ⓢ ⓓ n	n=96	285	255	230
MOV	ⓢ=D0、ⓓ=D1	0.70	0.56	0.35
MOVP	ⓢ=D0、ⓓ=J1\W1	155	130	120
DMOV	ⓢ=D0、ⓓ=D1	0.90	0.72	0.45
DMOVP	ⓢ=D0、ⓓ=J1\W1	165	135	120
$MOV	0 字元	46	38	35
$MOVP	32 字元	98	80	73
CML CMLP	---	0.70	0.56	0.35
DCML DCMLP	---	0.90	0.72	0.45

指令		條件（軟元件）	處理時間（μs）		
			Q00JCPU	Q00CPU	Q01CPU
BMOV Ⓢ Ⓓ n		n=1	27	21	20
BMOVP Ⓢ Ⓓ n		n=96	72	62	53
FMOV Ⓢ Ⓓ n		n=1	23	19	17
FMOVP Ⓢ Ⓓ n		n=96	48	41	36
XCH XCHP		---	7.6	6.3	5.7
DXCH DXCHP		---	9.5	8.0	7.1
BXCH Ⓓ1 Ⓓ2 n		n=1	62	51	48
BXCHP Ⓓ1 Ⓓ2 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) 應用指令

未執行時的處理時間如下所示。

Q00JCPU ..... 0.20 × (各指令的步數 +1) μs

Q00CPU ..... 0.16 × (各指令的步數 +1) μs

Q01CPU ..... 0.10 × (各指令的步數 +1) μs

指令	條件（軟元件）	處理時間（μs）		
		Q00JCPU	Q00CPU	Q01CPU
WAND Ⓢ Ⓓ WANDP Ⓢ Ⓓ	執行時	1.0	0.80	0.50
WAND Ⓢ1 Ⓢ2 Ⓓ WANDP Ⓢ1 Ⓢ2 Ⓓ	執行時	1.2	0.96	0.60
DAND Ⓢ Ⓓ DANDP Ⓢ Ⓓ	執行時	1.3	1.04	0.65
DAND Ⓢ1 Ⓢ2 Ⓓ DANDP Ⓢ1 Ⓢ2 Ⓓ	執行時	1.5	1.2	0.75
BKAND Ⓢ1 Ⓢ2 Ⓓ n	n=1	110	87	79
BKANDP Ⓢ1 Ⓢ2 Ⓓ n	n=96	185	155	140
WOR Ⓢ Ⓓ WORP Ⓢ Ⓓ	執行時	1.0	0.80	0.50
WOR Ⓢ1 Ⓢ2 Ⓓ WORP Ⓢ1 Ⓢ2 Ⓓ	執行時	1.2	0.96	0.60
DOR Ⓢ Ⓓ DORP Ⓢ Ⓓ	執行時	1.3	1.04	0.65

指令	條件（軟元件）	處理時間（μs）		
		Q00JCPU	Q00CPU	Q01CPU
DOR Ⓢ1 Ⓢ2 Ⓓ DORP Ⓢ1 Ⓢ2 Ⓓ	執行時	1.5	1.2	0.75
BKOR Ⓢ1 Ⓢ2 Ⓓ n	n=1	110	87	81
BKORP Ⓢ1 Ⓢ2 Ⓓ n	n=96	185	155	140
WXOR Ⓢ Ⓓ WXORP Ⓢ Ⓓ	執行時	1.0	0.80	0.50
WXOR Ⓢ1 Ⓢ2 Ⓓ WXORP Ⓢ1 Ⓢ2 Ⓓ	執行時	1.2	0.96	0.60
DXOR Ⓢ Ⓓ DXORP Ⓢ Ⓓ	執行時	1.3	1.04	0.65
DXOR Ⓢ1 Ⓢ2 Ⓓ DXORP Ⓢ1 Ⓢ2 Ⓓ	執行時	1.5	1.2	0.75
BKXOR Ⓢ1 Ⓢ2 Ⓓ n	n=1	110	87	81
BKXORP Ⓢ1 Ⓢ2 Ⓓ n	n=96	185	155	140
WXNR Ⓢ Ⓓ WXNRP Ⓢ Ⓓ	執行時	1.0	0.80	0.50
WXNR Ⓢ1 Ⓢ2 Ⓓ WXNRP Ⓢ1 Ⓢ2 Ⓓ	執行時	1.2	0.96	0.60
DXNR Ⓢ Ⓓ DXNRP Ⓢ Ⓓ	執行時	1.3	1.04	0.65
DXNR Ⓢ1 Ⓢ2 Ⓓ DXNRP Ⓢ1 Ⓢ2 Ⓓ	執行時	1.5	1.2	0.75
BKXNR Ⓢ1 Ⓢ2 Ⓓ n	n=1	110	87	82
BKXNRP Ⓢ1 Ⓢ2 Ⓓ n	n=96	185	155	140
ROR Ⓓ n	n=1	13	11	9.7
RORP Ⓓ n	n=15	13	11	9.7
RCR Ⓓ n	n=1	15	12	12
RCRP Ⓓ n	n=15	15	13	12
ROL Ⓓ n	n=1	13	11	10
ROLP Ⓓ n	n=15	13	11	10
RCL Ⓓ n	n=1	15	13	12
RCLP Ⓓ n	n=15	16	13	12
DROR Ⓓ n	n=1	15	12	12
DRORP Ⓓ n	n=31	15	13	12
DRCR Ⓓ n	n=1	17	14	14
DRCRP Ⓓ n	n=31	18	16	15
DROL Ⓓ n	n=1	14	13	12
DROLP Ⓓ n	n=31	14	13	12
DRCL Ⓓ n	n=1	18	15	14
DRCLP Ⓓ n	n=31	20	17	16
SFR Ⓓ n	n=1	13	10	9.7
SFRP Ⓓ n	n=15	13	11	9.5
SFL Ⓓ n	n=1	12	10	9.5
SFLP Ⓓ n	n=15	12	9.8	9.5

指令	條件（軟元件）		處理時間（μs）		
			Q00JCPU	Q00CPU	Q01CPU
BSFR Ⓓ n	n=1		42	35	33
BSFRP Ⓓ n	n=96		69	58	54
BSFL Ⓓ n	n=1		41	34	32
BSFLP Ⓓ n	n=96		63	53	50
DSFR Ⓓ n	n=1		19	16	15
DSFRP Ⓓ n	n=96		71	61	53
DSFL Ⓓ n	n=1		19	16	15
DSFLP Ⓓ n	n=96		70	60	52
BSET Ⓓ n	n=1		27	22	20
BSETP Ⓓ n	n=15		27	22	20
BRST Ⓓ n	n=1		27	22	21
BRSTP Ⓓ n	n=15		27	22	21
TEST Ⓢ1 Ⓢ2 Ⓓ TESTP Ⓢ1 Ⓢ2 Ⓓ	---		35	30	27
DTEST Ⓢ1 Ⓢ2 Ⓓ DTESTP Ⓢ1 Ⓢ2 Ⓓ	---		37	31	28
BKRST Ⓓ n	n=1		49	41	38
BKRSTP Ⓓ n	n=96		64	54	50
SER Ⓢ1 Ⓢ2 Ⓓ n	n=1	全部一致	56	54	42
		全部不一致	56	54	42
SERP Ⓢ1 Ⓢ2 Ⓓ n	n=96	全部一致	280	240	220
		全部不一致	280	240	220
DSER Ⓢ1 Ⓢ2 Ⓓ n	n=1	全部一致	71	67	53
		全部不一致	71	67	54
DSERP Ⓢ1 Ⓢ2 Ⓓ n	n=96	全部一致	495	415	375
		全部不一致	500	415	375
SUM SUMP	Ⓢ=0		32	26	25
	Ⓢ=$FFFF_H$		27	22	21
DSUM DSUMP	Ⓢ=0		54	44	42
	Ⓢ=$FFFFFFFF_H$		54	44	42
DECO Ⓢ Ⓓ n	n=2		60	50	46
DECOP Ⓢ Ⓓ n	n=8		80	65	61
ENCO Ⓢ Ⓓ n	n=2	M1=ON	66	55	51
		M4=ON	66	54	51
ENCOP Ⓢ Ⓓ n	n=8	M1=ON	90	76	71
		M256=ON	76	74	71
SEG SEGP	---		8.0	6.8	6.1
DIS Ⓢ Ⓓ n	n=1		47	39	36
DISP Ⓢ Ⓓ n	n=4		53	43	40
UNI Ⓢ Ⓓ n	n=1		54	44	41
UNIP Ⓢ Ⓓ n	n=4		60	49	46
NDIS Ⓢ1 Ⓓ Ⓢ2 NDISP Ⓢ1 Ⓓ Ⓢ2	---		92	76	38
NUNI Ⓢ1 Ⓓ Ⓢ2 NUNIP Ⓢ1 Ⓓ Ⓢ2	---		47	39	36
WTOB Ⓢ Ⓓ n	n=1		56	46	42
WTOBP Ⓢ Ⓓ n	n=96		190	155	145

指令	條件（軟元件）	處理時間（μs）		
		Q00JCPU	Q00CPU	Q01CPU
BTOW Ⓢ Ⓓ n	n=1	56	46	42
BTOWP Ⓢ Ⓓ n	n=96	190	155	145
MAX Ⓢ Ⓓ n	n=1	48	40	36
MAXP Ⓢ Ⓓ n	n=96	300	240	235
MIN Ⓢ Ⓓ n	n=1	48	40	36
MINP Ⓢ Ⓓ n	n=96	300	240	235
DMAX Ⓢ Ⓓ n	n=1	52	43	39
DMAXP Ⓢ Ⓓ n	n=96	600	490	460
DMIN Ⓢ Ⓓ n	n=1	52	43	39
DMINP Ⓢ Ⓓ n	n=96	585	475	445
SORT Ⓢ1 n Ⓢ2 Ⓓ1 Ⓓ2	n=1、Ⓢ2 =1	66	55	50
	n=96、Ⓢ2 =16	329	270	252
DSORT Ⓢ1 n Ⓢ2 Ⓓ1 Ⓓ2	n=1、Ⓢ2 =1	98	57	52
	n=96、Ⓢ2 =16	386	317	294
WSUM Ⓢ Ⓓ n	n=1	52	43	40
WSUMP Ⓢ Ⓓ n	n=96	175	140	135
DWSUM Ⓢ Ⓓ n	n=1	61	51	46
DWSUMP Ⓢ Ⓓ 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 Ⓢ1 ～ Ⓢ5 CALLP Pn Ⓢ1 ～ Ⓢ5	---	245	200	190
RET	返回至自身程式	16	13	12
FCALL Pn FCALLP Pn	---	29	24	22
FCALL Pn Ⓢ1 ～ Ⓢ5 FCALLP Pn Ⓢ1 ～ Ⓢ5	---	250	205	190
COM	---	110	77	72
IX	---	65	54	51
IXEND	---	30	26	25
IXDEV + IXSET	觸點數 1	145	120	110
	觸點數 14	770	630	585
FIFW	資料數 0	36	32	28
FIFWP	資料數 96	36	32	28
FIFR	資料數 1	45	41	36
FIFRP	資料數 96	93	82	70
FPOP	資料數 1	40	37	32
FPOPP	資料數 96	40	37	32
FINS	資料數 0	53	44	38
FINSP	資料數 96	100	89	76
FDEL	資料數 1	60	50	43
FDELP	資料數 96	110	95	82
FROM n1 n2 Ⓓ n3	n3=1	125	105	93
FROMP n1 n2 Ⓓ n3 *1	n3=1000	740	695	685

*1: FROM/TO 指令的處理時間根據插槽數及安裝的模組而有所不同。
（根據擴展基板的類型其處理時間也有所不同。）

指令	條件（軟元件）	處理時間（μs）		
		Q00JCPU	Q00CPU	Q01CPU
DFRO n1 n2 Ⓓ n3	n3=1	130	110	100
DFROP n1 n2 Ⓓ n3 *1	n3=500	745	695	675
TO n1 n2 Ⓢ n3	n3=1	120	105	92
TOP n1 n2 Ⓢ n3 *1	n3=1000	735	680	645
DTO n1 n2 Ⓢ n3	n3=1	130	110	99
DTOP n1 n2 Ⓢ n3 *1	n3=500	740	680	640
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+	無進位	47	39	36
DATE+P	有進位	50	42	38
DATE-	無進位	47	40	36
DATE-P	有進位	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

*1: FROM/TO 指令的處理時間根據插槽數及安裝的模組而有所不同。
（根據擴展基板的類型其處理時間也有所不同。）

(4) QCPU 用指令處理時間 (QCPU 專用）

指令	條件（軟元件）	處理時間（μs）		
		Q00JCPU	Q00CPU	Q01CPU
UNIRD	n=1	96	80	74
UNIRDP	n=16	440	370	340

(5) 序列號的前 5 位數爲 “ 04122 ” 以後中可使用的指令

指令	條件（軟元件）			處理時間（μs）		
				Q00JCPU	Q00CPU	Q01CPU
LDE=	單精度	導通時		43.0	35.5	33.0
		非導通時		46.0	38.0	35.5
ANDE=	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	35.5	29.5	26.5
			非導通時	42.0	35.0	32.5
ORE=	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	42.0	35.0	32.5
			非導通時	37.0	31.0	28.5
LDE< >	單精度	導通時		46.0	38.0	35.5
		非導通時		43.5	36.0	33.0
ANDE< >	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	38.5	31.5	29.0
			非導通時	39.5	33.0	30.5
ORE< >	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	45.0	37.5	35.0
			非導通時	34.5	29.0	26.5
LDE>	單精度	導通時		46.0	37.5	35.5
		非導通時		46.0	38.5	35.0
ANDE>	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	38.5	32.0	29.0
			非導通時	42.0	35.0	32.5
ORE>	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	45.0	37.5	34.5
			非導通時	37.0	31.0	29.0
LDE<=	單精度	導通時		45.5	37.5	35.0
		非導通時		46.5	38.5	35.5
ANDE<=	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	38.5	31.5	29.0
			非導通時	42.5	35.5	32.5
ORE<=	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	45.0	37.5	34.5
			非導通時	37.5	31.5	28.5
LDE<	單精度	導通時		45.5	37.5	35.0
		非導通時		46.5	38.5	35.5
ANDE<	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	38.0	31.5	29.0
			非導通時	42.5	35.5	32.5
ORE<	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	45.0	37.5	34.5
			非導通時	37.5	31.5	29.0
LDE>=	單精度	導通時		45.5	38.0	35.5
		非導通時		46.5	38.0	35.0
ANDE>=	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	38.5	32.0	29.0
			非導通時	42.5	35.5	32.5

指令	條件（軟元件）			處理時間（μs）		
				Q00JCPU	Q00CPU	Q01CPU
ORE>=	單精度	未執行時		1.5	1.2	1.0
		執行時	導通時	45.0	38.5	34.5
			非導通時	37.5	31.0	28.5
E+ Ⓢ Ⓓ	單精度	Ⓢ=0、Ⓓ=0		29.5	25.0	23.0
E+P Ⓢ Ⓓ		Ⓢ=2^{127}、Ⓓ=2^{127}		65.5	60.5	49.5
E+ Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0		31.0	27.0	24.0
E+P Ⓢ1 Ⓢ2 Ⓓ		Ⓢ1=2^{127}、Ⓢ2=2^{127}		66.5	56.0	51.0
E- Ⓢ Ⓓ	單精度	Ⓢ=0、Ⓓ=0		29.5	25.0	23.0
E-P Ⓢ Ⓓ		Ⓢ=2^{127}、Ⓓ=2^{127}		48.5	41.0	37.5
E- Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0		31.0	27.0	24.0
E-P Ⓢ1 Ⓢ2 Ⓓ		Ⓢ1=2^{127}、Ⓢ2=2^{127}		50.5	42.5	38.5
E* Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0		30.0	25.5	23.0
E*P Ⓢ1 Ⓢ2 Ⓓ		Ⓢ1=2^{127}、Ⓢ2=2^{127}		65.5	55.0	49.5
E/ Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=1		30.0	26.0	23.0
E/P Ⓢ1 Ⓢ2 Ⓓ		Ⓢ1=2^{127}、Ⓢ2=-2^{126}		69.5	57.5	53.0
INT INTP	單精度	Ⓢ=0		21.5	18.5	16.0
		Ⓢ=32766.5		38.0	32.0	29.5
DINT DINTP	單精度	Ⓢ=0		23.0	19.5	17.5
		Ⓢ=1234567890.3		42.0	35.5	32.0
FLT FLTP	單精度	Ⓢ=0		22.5	19.5	17.0
		Ⓢ=$7FFF_H$		26.5	23.0	20.0
DFLT DFLTP	單精度	Ⓢ=0		23.0	20.0	17.5
		Ⓢ=$7FFFFFFF_H$		26.0	23.5	19.5
ENEG ENEGP	Ⓢ=0			20.5	17.0	15.5
	Ⓢ=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	小數點形式全 2 位數指定			138.0	148.0	196.0
EVALP	指數形式全 6 位數指定			164.0	177.0	214.0
SIN SINP	單精度			204.0	173.0	157.0
COS COSP	單精度			187.0	158.0	144.0
TAN TANP	單精度			224.0	190.0	173.0
RAD RADP	單精度			51.0	43.0	39.0
DEG DEGP	單精度			51.0	43.0	39.0
SQR SQRP	單精度			60.0	51.0	46.5
EXP EXPP	單精度	Ⓢ=-10		306.0	259.0	235.0
		Ⓢ=1		306.0	259.0	235.0
LOG LOGP	單精度	Ⓢ=1		73.0	61.5	56.0
		Ⓢ=10		301.0	255.0	232.0

指令	條件（軟元件）	處理時間（μs）		
		Q00JCPU	Q00CPU	Q01CPU
RND RNDP	---	12.5	11.0	10.0
SRND SRNDP	---	13.5	12.0	11.0

指令名	條件・處理點數		處理時間（μs）		
			Q00JCPU	Q00CPU	Q01CPU
COM *2	有 CPU 共用記憶體的自動刷新	刷新範圍： 2k 字（各 CPU 0.5k 字的均等分配）	---	920	880
	無 CPU 共用記憶體的自動刷新	---	---	150	135
FROM	本機 CPU 共用記憶體讀取	n3=1	---	100	90
		n3=320	---	440	420
	其他機號 CPU 共用記憶體	n3=1	---	110	105
		n3=320	---	305	290
TO	至本機 CPU 共用記憶體的寫入	n3=1	---	100	95
		n3=320	---	440	425
S.TO	至本機 CPU 共用記憶體的寫入	n4=1	---	205	195
		n4=320	---	545	525

*2: 在多 CPU 系統中，與其他機號 CPU 的處理重疊時，將最多延遲下述時間。

僅主基板系統時

（指令的延遲時間）= 4 × 0.54 ×（處理點數）×（其他機號 CPU 個數）(μs)

包括擴展基板時

（指令的延遲時間）= 4 × 1.30 ×（處理點數）×（其他機號 CPU 個數）(μs)

(6) 使用文件寄存器、模組訪問軟元件、鏈結直接軟元件時的加法運算時間一覽表

軟元件名	資料	軟元件指定位置	處理時間（μs）		
			Q00JCPU	Q00CPU	Q01CPU
文件寄存器 (ZR)	位	源	---	34	32
		目標	---	23	22
	字	源	---	13	12
		目標	---	9	8
	雙字	源	---	14	13
		目標	---	10	9
模組訪問軟元件 (Un\G □、U3En\G0 ～ G511)	位	源	99	82	77
		目標	167	137	129
	字	源	74	61	58
		目標	72	60	56
	雙字	源	76	63	59
		目標	92	75	71
鏈結直接軟元件 (Jn\ □)	位	源	178	147	137
		目標	303	248	233
	字	源	154	126	118
		目標	153	125	117
	雙字	源	155	127	119
		目標	163	133	125

附錄 1.3 高性能型 QCPU/ 過程 CPU/ 冗餘 CPU 的運算處理時間

各指令的運算處理時間如本頁以後的表中所示。

由於運算處理時間根據源、目標的內容而有所不同，因此表中的值只應作爲處理時間的大致參考。

要 點

使用文件寄存器 (ZR)、模組訪問軟元件 (Un\G □、U3En\G0 ～ G4095)、鏈結直接軟元件 (Jn\ □) 的情況下，請參閱 701 頁附錄 1.3(5) 的加法運算時間，對各指令的處理時間進行加法運算。

(1) 順控程式指令

指令	條件 (軟元件)				處理時間 (μs)			
					Qn	QnH	QnPH	QnPRH
LD LDI AND ANI OR ORI	---				0.079	0.034	0.034	0.034
LDP LDF ANDP ANDF ORP ORF					0.158	0.068	0.068	0.068
ANB ORB MPS MRD MPP	---				0.079	0.034	0.034	0.034
INV	未執行時				0.079	0.034	0.034	0.034
	執行時							
MEP MEF	未執行時				0.173	0.073	0.073	0.073
	執行時							
EGP EGF	未執行時	(OFF → OFF) (ON → ON)			0.158	0.068	0.068	0.068
	執行時	(OFF → ON) (ON → OFF)						
OUT	無變化時	(OFF → OFF) (ON → ON)			0.158	0.068	0.068	0.068
	變化時	(OFF → ON) (ON → OFF)			0.158	0.068	0.068	0.068
	F	OFF 時			2.8	1.2	1.2	1.2
		ON 時	顯示時		162	69.7	69.7	69.7
			顯示結束		126	54	54	54
	T	未執行時			0.63	0.27	0.27	0.27
		執行時	時間到後		0.63	0.27	0.27	0.27
			加法運算時	K	0.63	0.27	0.27	0.27
				D	0.63	0.27	0.27	0.27
	C	未執行時			0.63	0.27	0.27	0.27
		執行時	時間到後		0.63	0.27	0.27	0.27
			加法運算時	K	0.63	0.27	0.27	0.27
				D	0.63	0.27	0.27	0.27

指令	條件（軟元件）				處理時間（μs）			
					Qn	QnH	QnPH	QnPRH
OUTH	T	未執行時			0.63	0.27	0.27	0.27
		執行時	時間到後		0.63	0.27	0.27	0.27
			加法運算時	K	0.63	0.27	0.27	0.27
				D	0.63	0.27	0.27	0.27
SET	未執行時				0.158	0.068	0.068	0.068
	執行時	無變化時 (ON → ON)			0.158	0.068	0.068	0.068
		變化時 (OFF → ON)			0.158	0.068	0.068	0.068
	F	未執行時			0.47	0.20	0.20	0.20
		執行時	顯示時		161	69	69	69
			顯示結束		0.47	0.20	0.20	0.20
RST	未執行時				0.158	0.068	0.068	0.068
	執行時	無變化時 (OFF → OFF)			0.158	0.068	0.068	0.068
		變化時 (ON → OFF)			0.158	0.068	0.068	0.068
	SM	未執行時			0.158	0.068	0.068	0.068
		執行時			0.158	0.068	0.068	0.068
	F	未執行時			0.47	0.20	0.20	0.20
		執行時	顯示時		90	38	38	38
			顯示結束		0.47	0.20	0.20	0.20
	T,C	未執行時			0.63	0.27	0.27	0.27
		執行時			0.63	0.27	0.27	0.27
	D	未執行時			0.24	0.10	0.10	0.10
		執行時			0.24	0.10	0.10	0.10
	Z	未執行時			0.47	0.20	0.20	0.20
		執行時			4.3	1.9	1.9	1.9
	R	未執行時			0.40	0.17	0.17	0.17
		執行時			0.40	0.17	0.17	0.17
PLS PLF	---				1.0	0.44	0.44	0.44
FF	Y	未執行時			0.47	0.20	0.20	0.20
		執行時			0.47	0.20	0.20	0.20
DELTA DELTAP	DY0	未執行時			0.47	0.20	0.20	0.20
		執行時			5.9	2.6	2.6	2.6
SFT SFTP	未執行時				0.47	0.20	0.20	0.20
	執行時				1.66	0.71	0.71	0.71
MC	---				0.24	0.10	0.10	0.10
MCR	---				0.079	0.034	0.034	0.034
FEND END	進行出錯檢查				380	150	150	500
	不進行出錯檢查 （・電池檢查） （・保險絲熔斷檢查） （・I/O 模組校驗）				380	150	150	500
NOP	---				0.079	0.034	0.034	0.034
NOPLF PAGE	---				0.079	0.034	0.034	0.034

(2) 基本指令

未執行時的處理時間如下所示。

Q02CPU ..... 0.079 ×（各指令的步數 +1）μs

Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU、Q02PHCPU、Q06PHCPU、Q12PHCPU、Q25PHCPU、Q12PRHCPU、Q25PRHCPU

..... 0.034 ×（各指令的步數 +1）μs

指令	條件（軟元件）		處理時間（μs）			
			Qn	QnH	QnPH	QnPRH
LD=	導通時		0.24	0.10	0.10	0.10
	非導通時		0.24	0.10	0.10	0.10
AND=	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10

指令	條件（軟元件）		處理時間（μs）			
			Qn	QnH	QnPH	QnPRH
OR=	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
LD< >	導通時		0.24	0.10	0.10	0.10
	非導通時		0.24	0.10	0.10	0.10
AND< >	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
OR< >	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
LD>	導通時		0.24	0.10	0.10	0.10
	非導通時		0.24	0.10	0.10	0.10
AND>	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
OR>	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
LD<=	導通時		0.24	0.10	0.10	0.10
	非導通時		0.24	0.10	0.10	0.10
AND<=	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
OR<=	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
LD<	導通時		0.24	0.10	0.10	0.10
	非導通時		0.24	0.10	0.10	0.10
AND<	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
OR<	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
LD>=	導通時		0.24	0.10	0.10	0.10
	非導通時		0.24	0.10	0.10	0.10
AND>=	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
OR>=	未執行時		0.24	0.10	0.10	0.10
	執行時	導通時	0.24	0.10	0.10	0.10
		非導通時	0.24	0.10	0.10	0.10
LDD=	導通時		0.55	0.24	0.24	0.24
	非導通時		0.39	0.17	0.17	0.17
ANDD=	未執行時		0.39	0.17	0.17	0.17
	執行時	導通時	0.55	0.24	0.24	0.24
		非導通時	0.39	0.17	0.17	0.17
ORD=	未執行時		0.39	0.17	0.17	0.17
	執行時	導通時	0.55	0.24	0.24	0.24
		非導通時	0.55	0.24	0.24	0.24
LDD< >	導通時		0.55	0.24	0.24	0.24
	非導通時		0.55	0.24	0.24	0.24
ANDD< >	未執行時		0.39	0.17	0.17	0.17
	執行時	導通時	0.55	0.24	0.24	0.24
		非導通時	0.55	0.24	0.24	0.24
ORD< >	未執行時		0.39	0.17	0.17	0.17
	執行時	導通時	0.55	0.24	0.24	0.24
		非導通時	0.55	0.24	0.24	0.24

指令	條件（軟元件）			處理時間（μs）			
				Qn	QnH	QnPH	QnPRH
LDD>	導通時			0.55	0.24	0.24	0.24
	非導通時			0.55	0.24	0.24	0.24
ANDD>	未執行時			0.39	0.17	0.17	0.17
	執行時		導通時	0.55	0.24	0.24	0.24
			非導通時	0.55	0.24	0.24	0.24
ORD>	未執行時			0.39	0.17	0.17	0.17
	執行時		導通時	0.55	0.24	0.24	0.24
			非導通時	0.55	0.24	0.24	0.24
LDD<=	導通時			0.55	0.24	0.24	0.24
	非導通時			0.55	0.24	0.24	0.24
ANDD<=	未執行時			0.39	0.17	0.17	0.17
	執行時		導通時	0.55	0.24	0.24	0.24
			非導通時	0.55	0.24	0.24	0.24
ORD<=	未執行時			0.39	0.17	0.17	0.17
	執行時		導通時	0.55	0.24	0.24	0.24
			非導通時	0.55	0.24	0.24	0.24
LDD<	導通時			0.55	0.24	0.24	0.24
	非導通時			0.55	0.24	0.24	0.24
ANDD<	未執行時			0.39	0.17	0.17	0.17
	執行時		導通時	0.55	0.24	0.24	0.24
			非導通時	0.55	0.24	0.24	0.24
ORD<	未執行時			0.39	0.17	0.17	0.17
	執行時		導通時	0.55	0.24	0.24	0.24
			非導通時	0.55	0.24	0.24	0.24
LDD>=	導通時			0.55	0.24	0.24	0.24
	非導通時			0.55	0.24	0.24	0.24
ANDD>=	未執行時			0.39	0.17	0.17	0.17
	執行時		導通時	0.55	0.24	0.24	0.24
			非導通時	0.55	0.24	0.24	0.24
ORD>=	未執行時			0.39	0.17	0.17	0.17
	執行時		導通時	0.55	0.24	0.24	0.24
			非導通時	0.55	0.24	0.24	0.24
LDE= *1	單精度	導通時		93	40	6.4	6.4
				14.9	6.4		
		非導通時		92	40	6.4	6.4
				14.9	6.4		
	雙精度	導通時		93	40	-	-
				14.9	6.4		
		非導通時		92	40	-	-
				14.9	6.4		
ANDE= *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	93	40	6.4	6.4
				14.9	6.4		
			非導通時	92	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		-	-	-	-
		執行時	導通時	93	40	-	-
				14.9	6.4		
			非導通時	92	40	-	-
				14.9	6.4		

*1: Qn/QnH 的處理時間根據 CPU 模組的序列號的不同而有所不同。
上欄： 序列號的前 5 位數爲 “ 05031 ” 以前
下欄： 序列號的前 5 位數爲 “ 05032 ” 以後
但是，未執行時的條件不存在上下欄的區別。

指令	條件（軟元件）			處理時間（μs）			
				Qn	QnH	QnPH	QnPRH
ORE= *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	93	40	6.4	6.4
				14.9	6.4		
			非導通時	92	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	-	-
		執行時	導通時	93	40	-	-
				14.9	6.4		
			非導通時	92	40	-	-
				14.9	6.4		
LDE< > *1	單精度	導通時		92	40	6.4	6.4
				14.9	6.4		
		非導通時		92	40	6.4	6.4
				14.9	6.4		
	雙精度	導通時		92	40	---	---
				14.9	6.4		
		非導通時		92	40	---	---
				14.9	6.4		
ANDE< > *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	92	40	6.4	6.4
				14.9	6.4		
			非導通時	93	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	---	---
		執行時	導通時	92	40	---	---
				14.9	6.4		
			非導通時	92	40	---	---
				14.9	6.4		
ORE< > *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	93	40	6.4	6.4
				14.9	6.4		
			非導通時	92	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	---	---
		執行時	導通時	93	40	---	---
				14.9	6.4		
			非導通時	92	40	---	---
				14.9	6.4		
LDE> *1	單精度	未執行時		92	40	6.4	6.4
		導通時		14.9	6.4		
				92	40	6.4	6.4
				14.9	6.4		
	雙精度	非導通時		92	40	---	---
				14.9	6.4		
				92	40	---	---
				14.9	6.4		
ANDE> *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	92	40	6.4	6.4
				14.9	6.4		
			非導通時	93	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	---	---
		執行時	導通時	92	40	---	---
				14.9	6.4		
			非導通時	92	40	---	---
				14.9	6.4		

*1: Qn/QnH 的處理時間根據 CPU 模組的序列號的不同而有所不同。
上欄： 序列號的前 5 位數爲 “ 05031 ” 以前
下欄： 序列號的前 5 位數爲 “ 05032 ” 以後
但是，未執行時的條件不存在上下欄的區別。

指令	條件（軟元件）			處理時間（μs）			
				Qn	QnH	QnPH	QnPRH
ORE> *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	93	40	6.4	6.4
				14.9	6.4		
			非導通時	92	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	---	---
		執行時	導通時	93	40	---	---
				14.9	6.4		
			非導通時	92	40	---	---
				14.9	6.4		
LDE<= *1	單精度	導通時		93	40	6.4	6.4
				14.9	6.4		
		非導通時		92	40	6.4	6.4
				14.9	6.4		
	雙精度	導通時		93	40	---	---
				14.9	6.4		
		非導通時		92	40	---	---
				14.9	6.4		
ANDE<= *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	92	40	6.4	6.4
				14.9	6.4		
			非導通時	92	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	---	---
		執行時	導通時	92	40	---	---
				14.9	6.4		
			非導通時	92	40	---	---
				14.9	6.4		
ORE<= *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	92	40	6.4	6.4
				14.9	6.4		
			非導通時	92	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	---	---
		執行時	導通時	92	40	---	---
				14.9	6.4		
			非導通時	92	40	---	---
				14.9	6.4		
LDE< *1	單精度	導通時		92	40	6.4	6.4
				14.9	6.4		
		非導通時		92	40	6.4	6.4
				14.9	6.4		
	雙精度	導通時		92	40	---	---
				14.9	6.4		
		非導通時		92	40	---	---
				14.9	6.4		
ANDE< *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	92	40	6.4	6.4
				14.9	6.4		
			非導通時	92	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	---	---
		執行時	導通時	92	40	---	---
				14.9	6.4		
			非導通時	92	40	---	---
				14.9	6.4		

*1: Qn/QnH 的處理時間根據 CPU 模組的序列號的不同而有所不同。
上欄： 序列號的前 5 位數爲 “ 05031 ” 以前
下欄： 序列號的前 5 位數爲 “ 05032 ” 以後
但是，未執行時的條件不存在上下欄的區別。

指令	條件（軟元件）			處理時間（μs）			
				Qn	QnH	QnPH	QnPRH
ORE< *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	93	40	6.4	6.4
				14.9	6.4		
			非導通時	92	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	---	---
		執行時	導通時	93	40	---	---
				14.9	6.4		
			非導通時	92	40	---	---
				14.9	6.4		
LDE>= *1	單精度	導通時		93	40	6.4	6.4
				14.9	6.4		
		非導通時		92	40	6.4	6.4
				14.9	6.4		
	雙精度	導通時		93	40	---	---
				14.9	6.4		
		非導通時		92	40	---	---
				14.9	6.4		
ANDE>= *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	92	40	6.4	6.4
				14.9	6.4		
			非導通時	92	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	---	---
		執行時	導通時	92	40	---	---
				14.9	6.4		
			非導通時	92	40	---	---
				14.9	6.4		
ORE>= *1	單精度	未執行時		0.55	0.24	0.24	0.24
		執行時	導通時	92	40	6.4	6.4
				14.9	6.4		
			非導通時	92	40	6.4	6.4
				14.9	6.4		
	雙精度	未執行時		0.55	0.24	---	---
		執行時	導通時	92	40	---	---
				14.9	6.4		
			非導通時	92	40	---	---
				14.9	6.4		
LD$=	導通時			38	16	16	16
	非導通時			34	15	15	15
AND$=	未執行時			0.56	0.23	0.23	0.23
	執行時		導通時	39	17	17	17
			非導通時	32	14	14	14
OR$=	未執行時			0.56	0.24	0.24	0.24
	執行時		導通時	40	17	17	17
			非導通時	33	14	14	14
LD$< >	導通時			32	14	14	14
	非導通時			40	17	17	17
AND$< >	未執行時			0.56	0.23	0.23	0.23
	執行時		導通時	33	14	14	14
			非導通時	39	17	17	17
OR$< >	未執行時			0.56	0.24	0.24	0.24
	執行時		導通時	32	14	14	14
			非導通時	39	17	17	17
LD$>	導通時			32	14	14	14
	非導通時			40	17	17	17

*1: Qn/QnH 的處理時間根據 CPU 模組的序列號的不同而有所不同。
上欄： 序列號的前 5 位數爲 “ 05031 ” 以前
下欄： 序列號的前 5 位數爲 “ 05032 ” 以後
但是，未執行時的條件不存在上下欄的區別。

指令	條件（軟元件）		處理時間（μs）			
			Qn	QnH	QnPH	QnPRH
AND$>	未執行時		0.56	0.23	0.23	0.23
	執行時	導通時	33	14	14	14
		非導通時	39	17	17	17
OR$>	未執行時		0.56	0.24	0.24	0.24
	執行時	導通時	32	14	14	14
		非導通時	39	17	17	17
LD$<=	導通時		40	17	17	17
	非導通時		32	14	14	14
AND$<=	未執行時		0.56	0.23	0.23	0.23
	執行時	導通時	39	17	17	17
		非導通時	32	14	14	14
OR$<=	未執行時		0.56	0.24	0.24	0.24
	執行時	導通時	40	17	17	17
		非導通時	33	14	14	14
LD$<	導通時		32	14	14	14
	非導通時		40	17	17	17
AND$<	未執行時		0.56	0.23	0.23	0.23
	執行時	導通時	32	14	14	14
		非導通時	39	16	16	16
OR$<	未執行時		0.56	0.24	0.24	0.24
	執行時	導通時	32	14	14	14
		非導通時	39	16	16	16
LD$>=	導通時		40	17	17	17
	非導通時		32	14	14	14
AND$>=	未執行時		0.56	0.23	0.23	0.23
	執行時	導通時	39	16	16	16
		非導通時	32	14	14	14
OR$>=	未執行時		0.56	0.24	0.24	0.24
	執行時	導通時	39	17	17	17
		非導通時	32	14	14	14

指令	條件（軟元件）	處理時間（μs）			
		Qn	QnH	QnPH	QnPRH
BKCMP= Ⓢ1 Ⓢ2 Ⓓ n	n=1	48	21	21	21
BKCMP=P Ⓢ1 Ⓢ2 Ⓓ n	n=96	142	61	61	61
BKCMP< > Ⓢ1 Ⓢ2 Ⓓ n	n=1	48	21	21	21
BKCMP< >P Ⓢ1 Ⓢ2 Ⓓ n	n=96	150	65	65	65
BKCMP> Ⓢ1 Ⓢ2 Ⓓ n	n=1	48	21	21	21
BKCMP>P Ⓢ1 Ⓢ2 Ⓓ n	n=96	142	61	61	61
BKCMP>= Ⓢ1 Ⓢ2 Ⓓ n	n=1	48	21	21	21
BKCMP>=P Ⓢ1 Ⓢ2 Ⓓ n	n=96	150	65	65	65
BKCMP< Ⓢ1 Ⓢ2 Ⓓ n	n=1	48	21	21	21
BKCMP<P Ⓢ1 Ⓢ2 Ⓓ n	n=96	158	68	68	68
BKCMP<= Ⓢ1 Ⓢ2 Ⓓ n	n=1	48	21	21	21
BKCMP<=P Ⓢ1 Ⓢ2 Ⓓ n	n=96	150	65	65	65
+ Ⓢ Ⓓ +P Ⓢ Ⓓ	執行時	0.39	0.17	0.17	0.17
+ Ⓢ1 Ⓢ2 Ⓓ +P Ⓢ1 Ⓢ2 Ⓓ	執行時	0.47	0.20	0.20	0.20
- Ⓢ Ⓓ -P Ⓢ Ⓓ	執行時	0.39	0.17	0.17	0.17
- Ⓢ1 Ⓢ2 Ⓓ -P Ⓢ1 Ⓢ2 Ⓓ	執行時	0.47	0.20	0.20	0.20
D+ Ⓢ Ⓓ D+P Ⓢ Ⓓ	執行時	0.71	0.31	0.31	0.31
D+ Ⓢ1 Ⓢ2 Ⓓ D+P Ⓢ1 Ⓢ2 Ⓓ	執行時	0.79	0.34	0.34	0.34
D- Ⓢ Ⓓ D-P Ⓢ Ⓓ	執行時	0.71	0.30	0.30	0.30
D- Ⓢ1 Ⓢ2 Ⓓ D-P Ⓢ1 Ⓢ2 Ⓓ	執行時	0.79	0.34	0.34	0.34
* Ⓢ1 Ⓢ2 Ⓓ *P Ⓢ1 Ⓢ2 Ⓓ	執行時	0.47	0.20	0.20	0.20
/ Ⓢ1 Ⓢ2 Ⓓ /P Ⓢ1 Ⓢ2 Ⓓ	---	2.7	1.2	1.2	1.2
D* Ⓢ1 Ⓢ2 Ⓓ D*P Ⓢ1 Ⓢ2 Ⓓ	---	7.9	3.4	3.4	3.4
D/ Ⓢ1 Ⓢ2 Ⓓ D/P Ⓢ1 Ⓢ2 Ⓓ	---	14	6.1	6.1	6.1
B+ Ⓢ Ⓓ B+P Ⓢ Ⓓ	---	2.2	1.0	1.0	1.0
B+ Ⓢ1 Ⓢ2 Ⓓ B+P Ⓢ1 Ⓢ2 Ⓓ	---	5.0	2.2	2.2	2.2
B- Ⓢ Ⓓ B-P Ⓢ Ⓓ	---	2.0	0.9	0.9	0.9
B- Ⓢ1 Ⓢ2 Ⓓ B-P Ⓢ1 Ⓢ2 Ⓓ	---	4.9	2.1	2.1	2.1

指令	條件（軟元件）		處理時間（μs）			
			Qn	QnH	QnPH	QnPRH
DB+ Ⓢ Ⓓ DB+P Ⓢ Ⓓ	---		12	5.0	5.0	5.0
DB+ Ⓢ1 Ⓢ2 Ⓓ DB+P Ⓢ1 Ⓢ2 Ⓓ	---		12	5.3	5.3	5.3
DB- Ⓢ Ⓓ DB-P Ⓢ Ⓓ	---		11	4.8	4.8	4.8
DB- Ⓢ1 Ⓢ2 Ⓓ DB-P Ⓢ1 Ⓢ2 Ⓓ	---		12	5.2	5.2	5.2
B* Ⓢ1 Ⓢ2 Ⓓ B*P Ⓢ1 Ⓢ2 Ⓓ	---		3.7	1.6	1.6	1.6
B/ Ⓢ1 Ⓢ2 Ⓓ B/P Ⓢ1 Ⓢ2 Ⓓ	---		3.8	1.6	1.6	1.6
DB* Ⓢ1 Ⓢ2 Ⓓ DB*P Ⓢ1 Ⓢ2 Ⓓ	---		24	10	10	10
DB/ Ⓢ1 Ⓢ2 Ⓓ DB/P Ⓢ1 Ⓢ2 Ⓓ	---		27	12	12	12
E+ Ⓢ Ⓓ E+P Ⓢ Ⓓ	單精度	Ⓢ=0、Ⓓ=0	1.8	0.78	0.78	0.78
		Ⓢ=2^{127}、Ⓓ=2^{127}	1.8	0.78	0.78	0.78
	雙精度	Ⓢ=0、Ⓓ=0	203	87	---	---
		Ⓢ=2^{127}、Ⓓ=2^{127}	203	87	---	---
E+ Ⓢ1 Ⓢ2 Ⓓ E+P Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0	2.4	1.1	1.1	1.1
		Ⓢ1=2^{127}、Ⓢ2=2^{127}	2.4	1.1	1.1	1.1
	雙精度	Ⓢ1=0、Ⓢ2=0	209	90	---	---
		Ⓢ1=2^{127}、Ⓢ2=2^{127}	209	90	---	---
E- Ⓢ Ⓓ E-P Ⓢ Ⓓ	單精度	Ⓢ=0、Ⓓ=0	1.8	0.78	0.78	0.78
		Ⓢ=2^{127}、Ⓓ=2^{127}	1.8	0.78	0.78	0.78
	雙精度	Ⓢ=0、Ⓓ=0	202	87	---	---
		Ⓢ=2^{127}、Ⓓ=2^{127}	202	87	---	---
E- Ⓢ1 Ⓢ2 Ⓓ E-P Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0	2.4	1.1	1.1	1.1
		Ⓢ1=2^{127}、Ⓢ2=2^{127}	2.4	1.1	1.1	1.1
	雙精度	Ⓢ1=0、Ⓢ2=0	210	90	---	---
		Ⓢ1=2^{127}、Ⓢ2=2^{127}	210	90	---	---
E* Ⓢ1 Ⓢ2 Ⓓ E*P Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0	2.4	1.1	1.1	1.1
		Ⓢ1=2^{126}、Ⓢ2=2^{127}	2.4	1.1	1.1	1.1
	雙精度	Ⓢ1=0、Ⓢ2=0	222	96	---	---
		Ⓢ1=2^{126}、Ⓢ2=2^{127}	222	96	---	---
E/ Ⓢ1 Ⓢ2 Ⓓ E/P Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=1	12	5.2	5.2	5.2
		Ⓢ1=2^{127}、Ⓢ2=-2^{126}	12	5.2	5.2	5.2
	雙精度	Ⓢ1=0、Ⓢ2=1	369	159	---	---
		Ⓢ1=2^{127}、Ⓢ2=-2^{126}	369	159	---	---

指令	條件（軟元件）		處理時間（μs）			
			Qn	QnH	QnPH	QnPRH
$+ Ⓢ Ⓓ $+P Ⓢ Ⓓ	---		68	29	29	29
$+ Ⓢ1 Ⓢ2 Ⓓ $+P Ⓢ1 Ⓢ2 Ⓓ	---		81	35	35	35
INC INCP	---		0.32	0.14	0.14	0.14
DINC DINCP	---		0.47	0.20	0.20	0.20
DEC DECP	---		0.32	0.14	0.14	0.14
DDEC DDECP	---		0.47	0.20	0.20	0.20
BCD BCDP	---		1.1	0.48	0.48	0.48
DBCD DBCDP	---		3.2	1.4	1.4	1.4
BIN BINP	---		1.0	0.44	0.44	0.44
DBIN DBINP	---		1.9	0.82	0.82	0.82
INT INTP	單精度	Ⓢ=0	3.2	1.4	1.4	1.4
		Ⓢ=32766.5	3.2	1.4	1.4	1.4
	雙精度	Ⓢ=0	22	9.3	---	---
		Ⓢ=32766.5	22	9.3	---	---
DINT DINTP	單精度	Ⓢ=0	2.5	1.1	1.1	1.1
		Ⓢ=1234567890.3	2.5	1.1	1.1	1.1
	雙精度	Ⓢ=0	24	10	---	---
		Ⓢ=1234567890.3	24	10	---	---
FLT FLTP	單精度	Ⓢ=0	2.1	0.92	0.92	0.92
		Ⓢ=$7FFF_H$	2.1	0.92	0.92	0.92
	雙精度	Ⓢ=0	22	9.6	---	---
		Ⓢ=$7FFF_H$	22	9.6	---	---
DFLT DFLTP	單精度	Ⓢ=0	2.1	0.88	0.88	0.88
		Ⓢ=$7FFFFFFF_H$	2.1	0.88	0.88	0.88
	雙精度	Ⓢ=0	26	11	---	---
		Ⓢ=$7FFFFFFF_H$	26	11	---	---
DBL DBLP	---		4.5	1.9	1.9	1.9
WORD WORDP	---		4.7	2.0	2.0	2.0
GRY GRYP	---		4.7	2.0	2.0	2.0
DGRY DGRYP	---		5.3	2.3	2.3	2.3
GBIN GBINP	---		18	7.7	7.7	7.7
DGBIN DGBINP	---		32	14	14	14
NEG NEGP	---		3.6	1.6	1.6	1.6

指令	條件（軟元件）	處理時間（μs）			
		Qn	QnH	QnPH	QnPRH
DNEG DNEGP	---	4.3	1.8	1.8	1.8
ENEG ENEGP	---	3.9	1.7	1.7	1.7
BKBCD Ⓢ Ⓓ n	n=1	38	17	17	17
BKBCDP Ⓢ Ⓓ n	n=96	99	43	43	43
BKBIN Ⓢ Ⓓ n	n=1	38	17	17	17
BKBINP Ⓢ Ⓓ n	n=96	99	43	43	43
MOV MOVP	Ⓢ=D0、Ⓓ=D1	0.24	0.10	0.10	0.10
	Ⓢ=D0、Ⓓ=J1\W1	---	---	---	---
		---	---	---	---
		140*1	60*1	60*1	60*1
DMOV DMOVP	Ⓢ=D0、Ⓓ=D1	0.47	0.20	0.20	0.20
	Ⓢ=D0、Ⓓ=J1\W1	---	---	---	---
		---	---	---	---
		147*1	64*1	64*1	64*1
EMOV EMOVP	---	0.63	0.27	0.27	0.27
$MOV $MOVP	---	40	17	17	17
CML CMLP	---	0.40	0.17	0.17	0.17
DCML DCMLP	---	0.55	0.24	0.24	0.24
BMOV Ⓢ Ⓓ n	n=1	17	7.1	7.1	7.1
BMOVP Ⓢ Ⓓ n	n=96	32	14	14	14
FMOV Ⓢ Ⓓ n	n=1	6.7	2.9	2.9	2.9
FMOVP Ⓢ Ⓓ n	n=96	14	6.1	6.1	6.1
XCH XCHP DXCH DXCHP	---	1.3	0.54	0.54	0.54
BXCH Ⓓ1 Ⓓ2 n	n=1	31	13	13	13
BXCHP Ⓓ1 Ⓓ2 n	n=96	84	36	36	36
SWAP SWAPP	---	3.7	1.6	1.6	1.6
CJ	---	3.2	1.4	1.4	1.4
SCJ	---	3.2	1.4	1.4	1.4
JMP	---	3.2	1.4	1.4	1.4
GOEND	---	0.39	0.34	0.34	0.34
DI	---	0.95	0.41	0.41	0.41
EI	---	1.3	0.54	0.54	0.54
IMASK	---	11	4.6	4.6	4.6
IRET	---	1.6	0.68	0.68	0.68
RFS	n=1	6.7	4.7	4.7	4.7
RFSP	n=96	19	13	13	13
UDCNT1	---	15	6.5	6.5	---
UDCNT2	---	16	6.8	6.8	---

*1: 上欄表示使用 A38B/A1S38B 以及擴展基板時的處理時間。
中欄表示使用 A38HB/A1S38HB 時的處理時間。
下欄表示使用 Q312B 時的處理時間。

指令	條件（軟元件）	處理時間（μs）			
		Qn	QnH	QnPH	QnPRH
TTMR	---	10	4.4	4.4	---
STMR	---	20	7.1	7.1	---
ROTC	---	26	11	11	---
RAMP	---	18	7.7	7.7	---
SPD	---	19	8.3	8.3	---
PLSY	---	10	4.5	4.5	---
PWM	---	9.1	3.9	3.9	---
MTR	---	11	4.9	4.9	---

(3) 應用指令

未執行時的處理時間如下所示。

Q02CPU ..... 0.079 × (各指令的步數 +1) μs

Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU、Q02PHCPU、Q06PHCPU、Q12PHCPU、Q25PHCPU、Q12PRHCPU、Q25PRHCPU ..... 0.034 × (各指令的步數 +1) μs

指令	條件（軟元件）	處理時間（μs）			
		Qn	QnH	QnPH	QnPRH
WAND Ⓢ Ⓓ WANDP Ⓢ Ⓓ	執行時	0.39	0.17	0.17	0.17
WAND Ⓢ1 Ⓢ2 Ⓓ WANDP Ⓢ1 Ⓢ2 Ⓓ	執行時	0.47	0.20	0.20	0.20
DAND Ⓢ Ⓓ DANDP Ⓢ Ⓓ	執行時	0.71	0.31	0.31	0.31
DAND Ⓢ1 Ⓢ2 Ⓓ DANDP Ⓢ1 Ⓢ2 Ⓓ	執行時	0.79	0.34	0.34	0.34
BKAND Ⓢ1 Ⓢ2 Ⓓ n	n=1	36	16	16	16
BKANDP Ⓢ1 Ⓢ2 Ⓓ n	n=96	74	32	32	32
WOR Ⓢ Ⓓ WORP Ⓢ Ⓓ	執行時	0.40	0.17	0.17	0.17
WOR Ⓢ1 Ⓢ2 Ⓓ WORP Ⓢ1 Ⓢ2 Ⓓ	執行時	0.47	0.20	0.20	0.20
DOR Ⓢ Ⓓ DORP Ⓢ Ⓓ	執行時	0.71	0.31	0.31	0.31
DOR Ⓢ1 Ⓢ2 Ⓓ DORP Ⓢ1 Ⓢ2 Ⓓ	執行時	0.79	0.34	0.34	0.34
BKOR Ⓢ1 Ⓢ2 Ⓓ n	n=1	36	16	16	16
BKORP Ⓢ1 Ⓢ2 Ⓓ n	n=96	74	32	32	32
WXOR Ⓢ Ⓓ WXORP Ⓢ Ⓓ	執行時	0.39	0.17	0.17	0.17
WXOR Ⓢ1 Ⓢ2 Ⓓ WXORP Ⓢ1 Ⓢ2 Ⓓ	執行時	0.47	0.20	0.20	0.20
DXOR Ⓢ Ⓓ DXORP Ⓢ Ⓓ	執行時	0.71	0.31	0.31	0.31
DXOR Ⓢ1 Ⓢ2 Ⓓ DXORP Ⓢ1 Ⓢ2 Ⓓ	執行時	0.79	0.34	0.34	0.34
BKXOR Ⓢ1 Ⓢ2 Ⓓ n	n=1	36	16	16	16
BKXORP Ⓢ1 Ⓢ2 Ⓓ n	n=96	74	32	32	32

指令	條件（軟元件）	處理時間（μs）			
		Qn	QnH	QnPH	QnPRH
WXNR Ⓢ Ⓓ WXNRP Ⓢ Ⓓ	執行時	0.40	0.17	0.17	0.17
WXNR Ⓢ1 Ⓢ2 Ⓓ WXNRP Ⓢ1 Ⓢ2 Ⓓ	執行時	0.47	0.20	0.20	0.20
DXNR Ⓢ Ⓓ DXNRP Ⓢ Ⓓ	執行時	0.71	0.31	0.31	0.31
DXNR Ⓢ1 Ⓢ2 Ⓓ DXNRP Ⓢ1 Ⓢ2 Ⓓ	執行時	0.79	0.34	0.34	0.34
BKXNR Ⓢ1 Ⓢ2 Ⓓ n	n=1	36	16	16	16
BKXNRP Ⓢ1 Ⓢ2 Ⓓ n	n=96	74	32	32	32
ROR Ⓓ n	n=1	2.0	0.85	0.85	0.85
RORP Ⓓ n	n=15	2.0	0.85	0.85	0.85
RCR Ⓓ n	n=1	1.6	0.68	0.68	0.68
RCRP Ⓓ n	n=15	1.6	0.68	0.68	0.68
ROL Ⓓ n	n=1	2.0	0.85	0.85	0.85
ROLP Ⓓ n	n=15	2.0	0.85	0.85	0.85
RCL Ⓓ n	n=1	1.6	0.68	0.68	0.68
RCLP Ⓓ n	n=15	1.6	0.68	0.68	0.68
DROR Ⓓ n	n=1	3.9	1.7	1.7	1.7
DRORP Ⓓ n	n=31	4.0	1.7	1.7	1.7
DRCR Ⓓ n	n=1	4.3	1.8	1.8	1.8
DRCRP Ⓓ n	n=31	4.3	1.9	1.9	1.9
DROL Ⓓ n	n=1	3.9	1.7	1.7	1.7
DROLP Ⓓ n	n=31	4.0	1.7	1.7	1.7
DRCL Ⓓ n	n=1	4.3	1.8	1.8	1.8
DRCLP Ⓓ n	n=31	4.3	1.9	1.9	1.9
SFR Ⓓ n	n=1	1.7	0.75	0.75	0.75
SFRP Ⓓ n	n=15	2.0	0.85	0.85	0.85
SFL Ⓓ n	n=1	1.7	0.75	0.75	0.75
SFLP Ⓓ n	n=15	2.0	0.85	0.85	0.85
BSFR Ⓓ n	n=1	20	8.6	8.6	8.6
BSFRP Ⓓ n	n=96	24	10	10	10
BSFL Ⓓ n	n=1	20	8.5	8.5	8.5
BSFLP Ⓓ n	n=96	23	10	10	10
DSFR Ⓓ n	n=1	1.3	0.58	0.58	0.58
DSFRP Ⓓ n	n=96	25	11	11	11
DSFL Ⓓ n	n=1	1.3	0.58	0.58	0.58
DSFLP Ⓓ n	n=96	26	11	11	11
BSET Ⓓ n	n=1	7.6	3.3	3.3	3.3
BSETP Ⓓ n	n=15	7.6	3.3	3.3	3.3
BRST Ⓓ n	n=1	7.6	3.3	3.3	3.3
BRSTP Ⓓ n	n=15	7.6	3.3	3.3	3.3
TEST Ⓢ1 Ⓢ2 Ⓓ TESTP Ⓢ1 Ⓢ2 Ⓓ	---	8.2	3.5	3.5	3.5

指令	條件（軟元件）		處理時間（μs）			
			Qn	QnH	QnPH	QnPRH
DTEST ⓢ1 ⓢ2 ⓓ DTESTP ⓢ1 ⓢ2 ⓓ	---		9.2	3.9	3.9	3.9
BKRST ⓓ n	n=1		18	7.8	7.8	7.8
BKRSTP ⓓ n	n=96		19	8.2	8.2	8.2
SER ⓢ1 ⓢ2 ⓓ n SERP ⓢ1 ⓢ2 ⓓ n	n=1	全部一致	22	9.6	9.6	9.6
		全部不一致	21	8.9	8.9	8.9
	n=96	全部一致	115	49	49	49
		全部不一致	133	57	57	57
DSER ⓢ1 ⓢ2 ⓓ n DSERP ⓢ1 ⓢ2 ⓓ n	n=1	全部一致	23	9.9	9.9	9.9
		全部不一致	23	9.7	9.7	9.7
	n=96	全部一致	142	61	61	61
		全部不一致	132	57	57	57
SUM SUMP	ⓢ=0 ⓢ=FFFF		3.9	1.7	1.7	1.7
DSUM DSUMP	ⓢ=0		4.7	2.0	2.0	2.0
	ⓢ=$FFFFFFFF_H$		12	5.0	5.0	5.0
DECO ⓢ ⓓ n	n=2		20	8.6	8.6	8.6
DECOP ⓢ ⓓ n	n=8		27	12	12	12
ENCO ⓢ ⓓ n ENCOP ⓢ ⓓ n	n=2	M1=ON	21	9.1	9.1	9.1
		M4=ON	21	9.1	9.1	9.1
	n=8	M1=ON	28	12	12	12
		M256=ON	26	11	11	11
SEG SEGP	---		1.3	0.54	0.54	0.54
DIS ⓢ ⓓ n	n=1		18	7.7	7.7	7.7
DISP ⓢ ⓓ n	n=4		19	8.3	8.3	8.3
UNI ⓢ ⓓ n	n=1		21	8.9	8.9	8.9
UNIP ⓢ ⓓ n	n=4		23	9.7	9.7	9.7
NDIS ⓢ1 ⓓ ⓢ2 NDISP ⓢ1 ⓓ ⓢ2	---		41	18	18	18
NUNI ⓢ1 ⓓ ⓢ2 NUNIP ⓢ1 ⓓ ⓢ2	---		42	18	18	18
WTOB ⓢ ⓓ n	n=1		47	20	20	20
WTOBP ⓢ ⓓ n	n=96		99	43	43	43
BTOW ⓢ ⓓ n	n=1		45	19	19	19
BTOWP ⓢ ⓓ n	n=96		89	38	38	38
MAX ⓢ ⓓ n	n=1		17	7.1	7.1	7.1
MAXP ⓢ ⓓ n	n=96		136	59	59	59
MIN ⓢ ⓓ n	n=1		17	7.1	7.1	7.1
MINP ⓢ ⓓ n	n=96		159	69	69	69
DMAX ⓢ ⓓ n	n=1		27	12	12	12
DMAXP ⓢ ⓓ n	n=96		181	78	78	78
DMIN ⓢ ⓓ n	n=1		27	12	12	12
DMINP ⓢ ⓓ n	n=96		112	48	48	48
SORT ⓢ1 n ⓢ2 ⓓ1 ⓓ2	n=1、ⓢ2 =1		16	7.1	7.1	7.1
	n=96、ⓢ2 =16		87.8	37.9	37.9	37.9
DSORT ⓢ1 n ⓢ2 ⓓ1 ⓓ2	n=1、ⓢ2 =1		17	7.1	7.1	7.1
	n=96、ⓢ2 =16		96.1	41.6	41.6	41.6

指令	條件（軟元件）	處理時間（μs）			
		Qn	QnH	QnPH	QnPRH
WSUM Ⓢ Ⓓ n	n=1	16.4	7.1	7.1	7.1
WSUMP Ⓢ Ⓓ n	n=96	68.4	29.5	29.5	29.5
DWSUM Ⓢ Ⓓ n	n=1	18.9	8.2	8.2	8.2
DWSUMP Ⓢ Ⓓ n	n=96	130.4	56.1	56.1	56.1
FOR n	n=0	2.3	1.0	1.0	1.0
NEXT	---	3.3	1.4	1.4	1.4
BREAK BREAKP	---	11	4.6	4.6	4.6
CALL Pn	文件內指標	2.1	0.88	0.88	0.88
CALLP Pn	公共指標	33	14	14	14
CALL Pn Ⓢ1 ～ Ⓢ5 CALLP Pn Ⓢ1 ～ Ⓢ5	---	135	58	58	58
RET	返回至本程式	2.9	1.3	1.3	1.3
	返回至其他程式	20	8.5	8.5	8.5
FCALL Pn	文件內指標	3.6	1.6	1.6	1.6
FCALLP Pn	公共指標	20	8.7	8.7	8.7
FCALL Pn Ⓢ1 ～ Ⓢ5 FCALLP Pn Ⓢ1 ～ Ⓢ5	---	134	57	57	57
ECALL * Pn ECALLP * Pn *: 程式名	---	77	33	33	33
ECALL * Pn Ⓢ1 ～ Ⓢ5 ECALLP * Pn Ⓢ1 ～ Ⓢ5 *: 程式名	---	162	70	70	70
EFCALL * Pn EFCALLP * Pn *: 程式名	---	78	34	34	34
EFCALL * Pn Ⓢ1 ～ Ⓢ5 EFCALLP * Pn Ⓢ1 ～ Ⓢ5 *: 程式名	---	200	86	86	86
COM	---	55	16	16	16
IX	---	12	5.2	5.2	5.2
IXEND	---	4.7	2.0	2.0	2.0
IXDEV+IXSET	觸點數 1	48	21	21	21
	觸點數 14	93	40	40	40
FIFW	資料數 0	11	4.5	4.5	4.5
FIFWP	資料數 96	11	4.5	4.5	4.5
FIFR	資料數 1	13	5.6	5.6	5.6
FIFRP	資料數 96	32	14	14	14
FPOP	資料數 1	16	7.0	7.0	7.0
FPOPP	資料數 96	16	7.0	7.0	7.0
FINS	資料數 0	20	8.4	8.4	8.4
FINSP	資料數 96	36	15	15	15
FDEL	資料數 1	19	7.5	7.5	7.5
FDELP	資料數 96	39	15	15	15
FROM n1 n2 Ⓓ n3 FROMP n1 n2 Ⓓ n3 *1	n3=1	---	---	---	---
		---	---	---	---
		47	22	22	22
	n3=1000	---	---	---	---
		---	---	---	---
		476	437	437	437

*1: 上欄表示使用 A38B/A1S38B 及擴展基板時的處理時間。
中欄表示使用 A38HB/A1S38HB 時的處理時間。
下欄表示對第 0 插槽的 QJ71C24 使用 Q312B 時的處理時間。
FROM/TO 指令的處理時間根據插槽數及安裝的模組而有所不同。
(QnCPU/QnHCPU 中根據擴展基板類型其指令的處理時間也有所不同。)

指令	條件（軟元件）		處理時間（μs）			
			Qn	QnH	QnPH	QnPRH
DFRO n1 n2 Ⓓ n3 DFROP n1 n2 Ⓓ n3 *1	n3=1		---	---	---	---
			---	---	---	---
			51	24	24	24
	n3=500		---	---	---	---
			---	---	---	---
			478	437	437	437
TO n1 n2 Ⓢ n3 TOP n1 n2 Ⓢ n3 *1	n3=1		---	---	---	---
			---	---	---	---
			48	20	20	20
	n3=1000		---	---	---	---
			---	---	---	---
			479	412	412	412
DTO n1 n2 Ⓢ n3 DTOP n1 n2 Ⓢ n3 *1	n3=1		---	---	---	---
			---	---	---	---
			50	23	23	23
	n3=500		---	---	---	---
			---	---	---	---
			457	416	416	416
PR	SM7010N	可變 1 字元	33	11	11	---
		可變 32 字元	48	18	18	---
	SM7010FF		21	7.8	7.8	---
PRC	---		181	16	16	---
LED	顯示時		---	---	---	---
	顯示結束		---	---	---	---
LEDC	顯示時		---	---	---	---
	顯示結束		---	---	---	---
LEDR	無顯示→無顯示		0.40	0.17	0.17	0.17
	LED 指令執行→無顯示		103	44	44	44
CHKST	---		5.8	2.5	2.5	2.5
CHK	無 1 觸點出錯		24	10	10	10
	無 150 觸點出錯		1676	721	721	721
	有 1 觸點出錯		88	38	38	38
CHKCIR	10 步		5.8	2.5	2.5	2.5
SLT	全內置軟元件		---	---	---	---
	文件寄存器 8k 點		---	---	---	---
	SLT 執行結束		---	---	---	---
SLTR	---		---	---	---	---
STRA	開始		---	---	---	---
	STRA 執行結束		---	---	---	---
STRAR	---		---	---	---	---
PTRA	---		---	---	---	---
PTRAR	---		---	---	---	---
PTRAEXE PTRAEXEP	運行中		---	---	---	---
	跟蹤中		---	---	---	---
BINDA BINDAP	Ⓢ=1		15	6.7	6.7	6.7
	Ⓢ=-32768		24	10	10	10
DBINDA DBINDAP	Ⓢ=1		43	18	18	18
	Ⓢ=-2147483648		86	37	37	37
BINHA BINHAP	Ⓢ=1		18	7.7	7.7	7.7
	Ⓢ=$FFFF_H$		19	8.2	8.2	8.2
DBINHA DBINHAP	Ⓢ=1		23	10	10	10
	Ⓢ=$FFFFFFFF_H$		24	10	10	10

*1: 上欄表示使用 A38B/A1S38B 以及擴展基板時的處理時間。
中欄表示使用 A38HB/A1S38HB 時的處理時間。
下欄表示使用 Q312B 時，對第 0 號插槽的 QJ71C24 執行了指令時的處理時間。
FROM/TO 指令的處理時間根據插槽數及安裝模組而有所不同。
（在 QnCPU/QnHCPU 中根據擴展基板的類型其處理時間也有所不同。）

指令	條件（軟元件）		處理時間（μs）			
			Qn	QnH	QnPH	QnPRH
BCDDA BCDDAP	Ⓢ=1		23	9.8	9.8	9.8
	Ⓢ=9999		21	8.9	8.9	8.9
DBCDDA DBCDDAP	Ⓢ=1		22	9.5	9.5	9.5
	Ⓢ=99999999		29	13	13	13
DABIN DABINP	Ⓢ=1		57	25	25	25
	Ⓢ=-32768		58	25	25	25
DDABIN DDABINP	Ⓢ=1		92	40	40	40
	Ⓢ=-2147483648		106	46	46	46
HABIN HABINP	Ⓢ=1		13	5.8	5.8	5.8
	Ⓢ=$FFFF_H$		15	6.4	6.4	6.4
DHABIN DHABINP	Ⓢ=1		22	9.5	9.5	9.5
	Ⓢ=$FFFFFFFF_H$		25	11	11	11
DABCD DABCDP	Ⓢ=1		16	6.9	6.9	6.9
	Ⓢ=9999		17	7.2	7.2	7.2
DDABCD DDABCDP	Ⓢ=1		25	11	11	11
	Ⓢ=99999999		29	13	13	13
COMRD COMRDP	---		40	17	17	17
LEN	1 字元		18	8.0	8.0	8.0
LENP	96 字元		86	37	37	37
STR STRP	---		53	23	23	23
DSTR DSTRP	---		123	53	53	53
VAL VALP	---		95	41	41	41
DVAL DVALP	---		166	72	72	72
ESTR ESTRP	---		564	243	243	243
EVAL EVALP	小數點形式全 2 位數指定		100	43	43	43
	指數形式全 6 位數指定		127	55	55	55
ASC Ⓢ Ⓓ n	n=1		64	28	28	28
ASCP Ⓢ Ⓓ n	n=96		289	125	125	125
HEX Ⓢ Ⓓ n	n=1		60	26	26	26
HEXP Ⓢ Ⓓ n	n=96		343	148	148	148
RIGHT Ⓢ Ⓓ n	n=1		49	21	21	21
RIGHTP Ⓢ Ⓓ n	n=96		131	56	56	56
LEFT Ⓢ Ⓓ n	n=1		50	21	21	21
LEFTP Ⓢ Ⓓ n	n=96		131	56	56	56
MIDR MIDRP	---		53	23	23	23
MIDW MIDWP	---		128	55	55	55
INSTR INSTRP	無一致		58	25	25	25
	有一致	起始	55	24	24	24
		最終	58	25	25	25
EMOD EMODP	---		527	227	227	227
EREXP EREXPP	---		1656	713	713	713

指令	條件（軟元件）		處理時間（μs）			
			Qn	QnH	QnPH	QnPRH
SIN	單精度		115	50	50	50
SINP	雙精度		1945	837	---	---
COS	單精度		122	53	53	53
COSP	雙精度		2618	1127	---	---
TAN	單精度		123	53	53	53
TANP	雙精度		2618	1127	---	---
ASIN	單精度		111	48	48	48
ASINP	雙精度		2491	1072	---	---
ACOS	單精度		115	49	49	49
ACOSP	雙精度		2367	1019	---	---
ATAN	單精度		157	68	68	68
ATANP	雙精度		3140	1352	---	---
RAD	單精度		17	7.2	7.2	7.2
RADP	雙精度		24	10	---	---
DEG	單精度		17	7.2	7.2	7.2
DEGP	雙精度		23	9.9	---	---
SQR	單精度		28	12	12	12
SQRP	雙精度		1812	780	---	---
EXP EXPP	單精度	Ⓢ=-10	129	56	56	56
		Ⓢ=1				
	雙精度	Ⓢ=-10	2386	1026	---	---
		Ⓢ=1				
LOG LOGP	單精度	Ⓢ=1	113	49	49	49
		Ⓢ=10				
	雙精度	Ⓢ=1	2146	924	---	---
		Ⓢ=10				
RND RNDP	---		3.9	1.7	1.7	1.7
SRND SRNDP	---		3.5	1.5	1.5	1.5
BSQR BSQRP	Ⓢ=0		6.2	2.7	2.7	2.7
	Ⓢ=9999		38	16	16	16
BDSQR BDSQRP	Ⓢ=0		6.2	2.7	2.7	2.7
	Ⓢ=99999999		38	16	16	16
BSIN BSINP	---		12	5.1	5.1	5.1
BCOS BCOSP	---		12	5.2	5.2	5.2
BTAN BTANP	---		12	5.2	5.2	5.2
BASIN BASINP	---		20	8.7	8.7	8.7
BACOS BACOSP	---		21	9.0	9.0	9.0
BATAN BATANP	---		22	9.6	9.6	9.6
LIMIT LIMITP	---		10	4.3	4.3	4.3
DLIMIT DLIMITP	---		11	4.7	4.7	4.7
BAND BANDP	---		9.8	4.2	4.2	4.2
DBAND DBANDP	---		11	4.9	4.9	4.9

指令	條件（軟元件）	處理時間（μs）			
		Qn	QnH	QnPH	QnPRH
ZONE ZONEP	---	9.1	3.9	3.9	3.9
DZONE DZONEP	---	11	4.6	4.6	4.6
RSET RSETP	---	6.8	2.9	2.9	2.9
QDRSET QDRSETP	---	205	88	88	88
QCDSET QCDSETP	---	147	63	63	63
DATERD DATERDP	---	13	5.5	5.5	5.5
DATEWR DATEWRP	---	15	6.4	6.4	6.4
DATE+	無進位	13	5.4	5.4	5.4
DATE+P	有進位	13	5.4	5.4	5.4
DATE-	無進位	12	5.2	5.2	5.2
DATE-P	有進位	12	5.2	5.2	5.2
SECOND SECONDP	---	10	4.5	4.5	4.5
HOUR HOURP	---	12	5.2	5.2	5.2
MSG	1 字元	3.0	1.3	1.3	1.3
	32 字元	3.0	1.3	1.3	1.3
PKEY	初次	20	8.6	8.6	8.6
	無受理	19	8.2	8.2	8.2
PSTOP PSTOPP	---	79	34	34	34
POFF POFFP	---	79	34	34	34
PSCAN PSCNAP	---	75	32	32	32
PLOW PLOWP	---	80	34	34	---
WDT WDTP	---	5.9	2.6	2.6	2.6
DUTY	---	9.3	4.0	4.0	4.0
ZRRDB ZRRDBP	---	7.9	3.4	3.4	3.4
ZRWRB ZRWRBP	---	9.4	4.0	4.0	4.0
ADRSET ADRSETP	---	4.9	2.1	2.1	2.1
KEY	---	17	7.3	7.3	---
ZPUSH ZPUSHP	---	11	4.7	4.7	4.7
ZPOP ZPOPP	---	5.1	2.2	2.2	2.2
EROMWR EROMWRP	---	---	---	---	---
ZCOM	---	691	289	289	289
READ	---	---	---	---	---
SREAD	---	---	---	---	---
WRITE	---	---	---	---	---
SWRITE	---	---	---	---	---
SEND	---	---	---	---	---
RECV	---	---	---	---	---
REQ	---	---	---	---	---
ZNFR	---	---	---	---	---
ZNTO	---	---	---	---	---

指令	條件（軟元件）	處理時間（μs）			
		Qn	QnH	QnPH	QnPRH
ZNRD	MELSECNET/10	---	---	---	---
	MELSECNET(II)	---	---	---	---
ZNWR	MELSECNET/10	---	---	---	---
	MELSECNET(II)	---	---	---	---
RFRP	---	---	---	---	---
RTOP	---	---	---	---	---

(a) 功能版本 A 中可使用的指令

指令	條件（軟元件）		處理時間（μs）			
			Qn	QnH	QnPH	QnPRH
UNIRD	---		79	34	34	34
TRACE	開始		176	76	76	76
	STRA 執行結束		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	使用標準 RAM	1 點	45.5	20	20	20
		1000 點	215	91	91	91
	使用 SRAM 卡	1 點	49.5	22	22	22
		1000 點	540	305	305	305

(b) 功能版本 B 中可使用的指令

指令	條件・處理點數			處理時間（μs）			
				Qn	QnH	QnPH	QnPRH
COM *1	有 CPU 共用記憶體的自動刷新	刷新範圍： 2k 字（各 CPU 0.5k 字的均等分配）		720	660	660	---
		刷新範圍： 4k 字（各 CPU 1k 字的均等分配）		860	730	730	---
	無 CPU 共用記憶體的自動刷新	---		43	20	20	20
FROM *1	其他機號 CPU 共用記憶體讀取	n3=1		59	29	29	---
		n3=1000		530	500	500	---
	智慧功能模組的緩衝記憶體讀取 *2	n3=1	主基板	51	24	24	---
			擴展基板	54	27	27	---
		n3=1000	主基板	540	480	480	---
			擴展基板	1100	1050	1050	---
S.TO	至本機 CPU 共用記憶體的寫入	n2=1		74	33	33	---
		n2=256		126	54	54	---
S (P). DATERD *3	擴展時鐘資料的讀取	---		25	11	11	11
S (P). DATE+ *3	擴展時鐘資料的加法運算	---		38	17	17	17
S (P). DATE- *3	擴展時鐘資料的減法運算	---		38	17	17	17

*1: 在多 CPU 系統中，與其他機號 CPU 的處理重疊時，將最多延遲下述時間。

僅主基板的系統時
（指令的延遲時間）= 0.54 ×（處理點數）×（其他機號 CPU 個數）(μs)
包括擴展基板時
（指令的延遲時間）= 1.30 ×（處理點數）×（其他機號 CPU 個數）(μs)

*2: 在多 CPU 系統中，本機 CPU 管理的智慧功能模組與及其它機號管理的智慧功能模組的指令處理時間相同。

*3: 以序列號的前 5 位數爲 “ 07032 ” 以後的模組爲物件。

(4) 冗餘系統用指令（冗餘 CPU 專用）

指令	條件（軟元件）	處理時間（μs）			
		Qn	QnH	QnPH	QnPRH
SP.CONTSW	---	---	---	---	9.6

(5) 使用文件寄存器、模組訪問軟元件、鏈結直接軟元件時的加法運算時間一覽表

軟元件名		資料	軟元件指定位置	處理時間（μs）			
				QnCPU	QnHCPU	QnPHCPU	QnPRHCPU
文件寄存器 (ZR)	使用標準 RAM 時	位	源	5.56	2.40	2.40	2.40
			目標	4.44	1.91	1.91	1.91
		字	源	2.60	1.12	1.12	1.12
			目標	3.76	1.62	1.62	1.62
		雙字	源	2.83	1.22	1.22	1.22
			目標	4.00	1.72	1.72	1.72
	使用 SRAM 卡 (Q2MEM-1MBS、Q2MEM-2MBS) 時	位	源	5.22	2.25	2.25	2.25
			目標	4.09	1.76	1.76	1.76
		字	源	2.25	0.97	0.97	0.97
			目標	3.42	1.47	1.47	1.47
		雙字	源	2.49	1.07	1.07	1.07
			目標	3.65	1.57	1.57	1.57
模組訪問軟元件 (Un\G□、U3En\G0 ～ G4095)		位	源	35.56	15.31	15.31	15.31
			目標	65.08	28.01	28.01	28.01
		字	源	32.76	14.10	14.10	14.10
			目標	28.84	12.41	12.41	12.41
		雙字	源	32.99	14.20	14.20	14.20
			目標	29.07	12.51	12.51	12.51
鏈結直接軟元件 (Jn\□)		位	源	75.67	32.57	32.57	32.57
			目標	138.65	59.67	59.67	59.67
		字	源	72.73	31.30	31.30	31.30
			目標	137.32	59.10	59.10	59.10
		雙字	源	72.96	31.40	31.40	31.40
			目標	137.55	59.20	59.20	59.20

附錄 1.4 通用型 QCPU 的運算處理時間

各指令的運算處理時間如本頁以後的表中所示。

由於運算處理時間根據源、目標的內容而有所不同，因此表中的值只應作爲處理時間的大致參考。

附錄 1.4.1 子集指令處理時間

子集處理指令的處理時間如下所示。

要 點

1. 指令中使用的軟元件滿足子集處理的軟元件條件之一時的子集指令的指令處理時間一覽表如下述 (1) 所示。
   (關於子集處理的軟元件條件， 請參閱 98 頁 3.5.1 項。)
2. 使用文件寄存器 (R、ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)、模組訪問軟元件 (U3En\G10000 ～) 時，請參閱 (2) 的加法運算時間，對各指令的處理時間進行加法運算。
3. 在 OUT/SET/RST 指令中使用 F、T(ST)、C 軟元件時，請參閱 (3) 中列出的加法運算時間，對各指令的處理時間進行加法運算。
4. 由於高速緩衝記憶體功能的影響，各指令的處理時間不恒定，因此對最小值和最大值進行了記載。

(1) 子集指令的指令處理時間一覽表

(a) 使用 Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 時

分類	指令	條件 (軟元件)		處理時間 (μs)							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
順控程式指令	LD LDI AND ANI OR ORI LDP LDF ANDP ANDF ORP ORF	執行時		0.120		0.080		0.060		0.040	
	LDPI LDFI	執行時		0.360		0.240		0.180		0.120	
	ANDPI ANDFI ORPI ORFI	執行時		0.480		0.320		0.240		0.160	
	OUT	無變化時		0.120		0.080		0.060		0.040	
		變化時									
	SET RST	未執行時		0.120		0.080		0.060		0.040	
		執行時	無變化時								
			變化時								
基本指令	LD=	導通時		0.360		0.240		0.180		0.120	
		非導通時									
	AND=	未執行時		0.360		0.240		0.180		0.120	
		執行時	導通時								
			非導通時								
	OR=	未執行時		0.360		0.240		0.180		0.120	
		執行時	導通時								
			非導通時								
	LD<>	導通時		0.360		0.240		0.180		0.120	
		非導通時									

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	AND<>	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	OR<>	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	LD>	導通時			0.360		0.240		0.180		0.120
		非導通時									
	AND>	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	OR>	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	LD<=	導通時			0.360		0.240		0.180		0.120
		非導通時									
	AND<=	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	OR<=	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	LD<	導通時			0.360		0.240		0.180		0.120
		非導通時									
	AND<	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	OR<	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	LD>=	導通時			0.360		0.240		0.180		0.120
		非導通時									
	AND>=	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	OR>=	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	LDD=	導通時			0.360		0.240		0.180		0.120
		非導通時									
	ANDD=	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	ORD=	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	LDD<>	導通時			0.360		0.240		0.180		0.120
		非導通時									
	ANDD<>	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	ORD<>	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	LDD>	導通時			0.360		0.240		0.180		0.120
		非導通時									
	ANDD>	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	ORD>	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	LDD<=	導通時			0.360		0.240		0.180		0.120
		非導通時									
	ANDD<=	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	ORD<=	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	LDD<	導通時			0.360		0.240		0.180		0.120
		非導通時									
	ANDD<	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	ORD<	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	LDD>=	導通時			0.360		0.240		0.180		0.120
		非導通時									
	ANDD>=	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	ORD>=	未執行時			0.360		0.240		0.180		0.120
		執行時	導通時								
			非導通時								
	+ Ⓢ Ⓓ	執行時			0.360		0.240		0.180		0.120
	+ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.480		0.320		0.240		0.160
	- Ⓢ Ⓓ	執行時			0.360		0.240		0.180		0.120
	- Ⓢ1 Ⓢ2 Ⓓ	執行時			0.480		0.320		0.240		0.160
	D+ Ⓢ Ⓓ	執行時			0.360		0.240		0.180		0.120
	D+ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.480		0.320		0.240		0.160
	D- Ⓢ Ⓓ	執行時			0.360		0.240		0.180		0.120
	D- Ⓢ1 Ⓢ2 Ⓓ	執行時			0.480		0.320		0.240		0.160
	* Ⓢ1 Ⓢ2 Ⓓ	執行時			0.420		0.300		0.240		0.180
	/ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.520		0.400		0.340		0.280
	D* Ⓢ1 Ⓢ2 Ⓓ	執行時			0.500		0.380		0.320		0.260
	D/ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.640		0.520		0.460		0.400
	B+ Ⓢ Ⓓ	執行時		3.100	12.300	3.100	12.300	3.100	12.300	3.300	8.300
	B+ Ⓢ1 Ⓢ2 Ⓓ	執行時		5.900	13.500	5.900	13.500	5.900	13.500	4.600	6.200
	B- Ⓢ Ⓓ	執行時		3.150	12.300	3.150	12.300	3.150	12.300	3.300	9.000
	B- Ⓢ1 Ⓢ2 Ⓓ	執行時		5.950	13.600	5.950	13.600	5.950	13.600	4.600	8.200
	B* Ⓢ1 Ⓢ2 Ⓓ	執行時		3.700	12.100	3.700	12.100	3.700	12.100	4.000	8.200
	B/ Ⓢ1 Ⓢ2 Ⓓ	執行時		4.000	14.000	4.000	14.000	4.000	14.000	4.200	12.400
	E+ Ⓢ Ⓓ	單精度	Ⓢ=0、Ⓓ=0		0.420		0.300		0.240		0.180
			Ⓢ=2^{127}、Ⓓ=2^{127}		0.420		0.300		0.240		0.180
	E+ Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0		0.540		0.380		0.300		0.220
			Ⓢ1=2^{127}、Ⓢ2=2^{127}		0.540		0.380		0.300		0.220
	E- Ⓢ Ⓓ	單精度	Ⓢ=0、Ⓓ=0		0.420		0.300		0.240		0.180
			Ⓢ=2^{127}、Ⓓ=2^{127}		0.420		0.300		0.240		0.180

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	E- ⓢ1 ⓢ2 Ⓓ	單精度	ⓢ1=0、ⓢ2=0		0.540		0.380		0.300		0.220
			ⓢ1=2^{127}、ⓢ2=2^{127}		0.540		0.380		0.300		0.220
	E* ⓢ1 ⓢ2 Ⓓ	單精度	ⓢ1=0、ⓢ2=0		0.420		0.300		0.240		0.180
			ⓢ1=2^{127}、ⓢ2=2^{127}		0.420		0.300		0.240		0.180
	E/ ⓢ1 ⓢ2 Ⓓ	單精度	ⓢ1=2^{127}、ⓢ2=2^{127}	4.900	18.900	4.900	18.900	4.900	18.900	5.100	14.100
	INC	執行時			0.240		0.160		0.120		0.080
	DINC	執行時			0.240		0.160		0.120		0.080
	DEC	執行時			0.240		0.160		0.120		0.080
	DDEC	執行時			0.240		0.160		0.120		0.080
	BCD	執行時			0.320		0.240		0.200		0.160
	DBCD	執行時			0.400		0.320		0.280		0.240
	BIN	執行時			0.260		0.180		0.140		0.100
	DBIN	執行時			0.260		0.180		0.140		0.100
	FLT	單精度	Ⓢ=0		0.300		0.220		0.180		0.140
			Ⓢ=7FFF$_H$		0.300		0.220		0.180		0.140
	DFLT	單精度	Ⓢ=0		0.300		0.220		0.180		0.140
			Ⓢ=7FFFFFFF$_H$		0.300		0.220		0.180		0.140
	INT	單精度	Ⓢ=0		0.300		0.220		0.180		0.140
			Ⓢ=32766.5		0.300		0.220		0.180		0.140
	DINT	單精度	Ⓢ=0		0.300		0.220		0.180		0.140
			Ⓢ=1234567890.3		0.300		0.220		0.180		0.140
	MOV	-			0.240		0.160		0.120		0.080
	DMOV	-			0.240		0.160		0.120		0.080
	EMOV	-			0.240		0.160		0.120		0.080
	CML	-			0.240		0.160		0.120		0.080
	DCML	-			0.240		0.160		0.120		0.080
	BMOV	SM237=ON	n=1	4.200	4.600	4.200	4.600	4.200	4.600	4.100	4.500
			n=96	4.850	5.150	4.850	5.150	4.850	5.150	4.700	5.100
		SM237=OFF	n=1	6.800	11.300	6.800	11.300	6.800	11.300	6.300	8.900
			n=96	7.450	11.900	7.450	11.900	7.450	11.900	5.900	9.500
	FMOV	SM237=ON	n=1	4.100	4.600	4.100	4.600	4.100	4.600	4.100	4.600
			n=96	4.800	5.200	4.800	5.200	4.800	5.200	4.800	5.200
		SM237=OFF	n=1	4.600	8.250	4.600	8.250	4.600	8.250	4.600	7.900
			n=96	6.150	10.600	6.150	10.600	6.150	10.600	5.300	8.500
	XCH	---		2.250	8.100	2.250	8.100	2.250	8.100	2.500	6.000
	DXCH	---		2.400	8.200	2.400	8.200	2.400	8.200	2.800	7.900
	DFMOV	SM237=ON	n=1	2.700	2.800	2.700	2.800	2.700	2.800	2.350	2.450
			n=96	6.500	6.800	6.500	6.800	6.500	6.800	5.950	6.000
		SM237=OFF	n=1	4.000	8.150	4.000	8.150	4.000	8.150	3.000	6.950
			n=96	8.000	12.200	8.000	12.200	8.000	12.200	6.600	10.600
	CJ	---		3.500	10.100	3.500	10.100	3.500	10.100	1.900	10.100
	SCJ	---		3.500	10.100	3.500	10.100	3.500	10.100	1.900	10.100
	JMP	---		3.500	10.100	3.500	10.100	3.500	10.100	1.900	10.100
應用指令	WAND Ⓢ Ⓓ	執行時			0.360		0.240		0.180		0.120
	WAND ⓢ1 ⓢ2 Ⓓ	執行時			0.480		0.320		0.240		0.160
	DAND Ⓢ Ⓓ	執行時			0.360		0.240		0.180		0.120
	DAND ⓢ1 ⓢ2 Ⓓ	執行時			0.480		0.320		0.240		0.160
	WOR Ⓢ Ⓓ	執行時			0.360		0.240		0.180		0.120
	WOR ⓢ1 ⓢ2 Ⓓ	執行時			0.480		0.320		0.240		0.160

分類	指令	條件（軟元件）	處理時間（μs）							
			Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	DOR Ⓢ Ⓓ	執行時		0.360		0.240		0.180		0.120
	DOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.480		0.320		0.240		0.160
	WXOR Ⓢ Ⓓ	執行時		0.360		0.240		0.180		0.120
	WXOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.480		0.320		0.240		0.160
	DXOR Ⓢ Ⓓ	執行時		0.360		0.240		0.180		0.120
	DXOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.480		0.320		0.240		0.160
	WXNR Ⓢ Ⓓ	執行時		0.360		0.240		0.180		0.120
	WXNR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.480		0.320		0.240		0.160
	DXNR Ⓢ Ⓓ	執行時		0.360		0.240		0.180		0.120
	DXNR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.480		0.320		0.240		0.160
	ROR Ⓓ n	n=1	2.250	10.800	2.250	10.800	2.250	10.800	2.300	7.800
		n=15	2.250	10.800	2.350	10.800	2.350	10.800	2.400	7.800
	RCR Ⓓ n	n=1	2.250	10.800	2.250	10.800	2.250	10.800	2.300	3.900
		n=15	2.250	10.800	2.250	10.800	2.250	10.800	2.400	4.100
	ROL Ⓓ n	n=1	2.250	10.800	2.350	10.800	2.350	10.800	2.500	4.600
		n=15	2.250	10.800	2.350	10.800	2.350	10.800	2.400	4.600
	RCL Ⓓ n	n=1	2.250	11.500	2.300	11.500	2.300	11.500	2.400	7.500
		n=15	2.250	11.500	2.300	11.500	2.300	11.500	2.500	7.500
	DROR Ⓓ n	n=1	2.350	11.500	2.350	11.500	2.350	11.500	2.400	10.300
		n=31	2.350	11.500	2.350	11.500	2.350	11.500	2.500	10.300
	DRCR Ⓓ n	n=1	2.350	13.300	2.350	13.300	2.350	13.300	2.500	12.700
		n=31	2.350	14.900	2.350	14.900	2.350	14.900	2.500	12.700
	DROL Ⓓ n	n=1	2.350	10.800	2.350	10.800	2.350	10.800	2.500	11.800
		n=31	2.350	10.800	2.350	10.800	2.350	10.800	2.500	11.800
	DRCL Ⓓ n	n=1	2.350	13.300	2.350	13.300	2.350	13.300	2.500	5.100
		n=31	2.350	13.300	2.350	13.300	2.350	13.300	2.500	5.100
	SFR Ⓓ n	n=1	2.350	9.900	2.350	9.900	2.350	9.900	2.400	6.100
		n=15	2.350	9.900	2.350	9.900	2.350	9.900	2.300	5.700
	SFL Ⓓ n	n=1	2.350	9.850	2.350	9.850	2.350	9.850	2.400	4.300
		n=15	2.350	9.850	2.350	9.850	2.350	9.850	2.400	4.300
	DSFR Ⓓ n	n=1	3.250	15.500	3.250	15.500	3.250	15.500	3.300	12.000
		n=96	32.600	45.000	32.600	45.000	32.600	45.000	32.600	42.200
	DSFL Ⓓ n	n=1	3.200	15.500	3.200	15.500	3.200	15.500	3.300	8.200
		n=96	32.600	45.100	32.600	45.100	32.600	45.100	32.600	37.700
	SUM	Ⓢ=0	3.100	8.950	3.100	8.950	3.100	8.950	3.400	6.700
		Ⓢ=$FFFF_H$	3.000	8.850	3.000	8.850	3.000	8.850	3.500	6.700
	SEG	執行時	2.100	7.700	2.100	7.700	2.100	7.700	2.100	5.900
	FOR	---	1.500	7.500	1.500	7.500	1.500	7.500	1.200	6.300
	CALL Pn	文件內指標	4.800	5.400	4.800	5.400	4.800	5.400	2.700	4.800
		公共指標	7.100	30.500	7.100	30.500	7.100	30.500	4.400	5.700
	CALL Pn Ⓢ1 ～ Ⓢ5	---	50.200	62.000	50.200	62.000	50.200	62.000	28.700	42.600

備註

對於表中未記述上升沿執行指令（□ P）的指令，其處理時間與 ON 時執行的指令的處理時間相同。

例 MOVP 指令、WANDP 指令等。

(b) 使用 Q03UD(E)CPU、Q04UD(E)HCPU、Q06UD(E)HCPU、Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU、Q50UDEHCPU、Q100UDEHCPU 時

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
順控程式指令	LD LDI AND ANI OR ORI LDP LDF ANDP ANDF ORP ORF	執行時		0.020		0.0095		0.0095		0.0095	
	LDPI LDFI	執行時		0.060		0.0285		0.0285		0.0285	
	ANDPI ANDFI ORPI ORFI	執行時		0.080		0.038		0.038		0.038	
	OUT	無變化時		0.020		0.0095		0.0095		0.0095	
		變化時									
	SET RST	未執行時		0.020		0.0095		0.0095		0.0095	
		執行時	無變化時								
			變化時								
基本指令	LD=	導通時		0.060		0.0285		0.0285		0.0285	
		非導通時									
	AND=	未執行時		0.060		0.0285		0.0285		0.0285	
		執行時	導通時								
			非導通時								
	OR=	未執行時		0.060		0.0285		0.0285		0.0285	
		執行時	導通時								
			非導通時								
	LD<>	導通時		0.060		0.0285		0.0285		0.0285	
		非導通時									
	AND<>	未執行時		0.060		0.0285		0.0285		0.0285	
		執行時	導通時								
			非導通時								
	OR<>	未執行時		0.060		0.0285		0.0285		0.0285	
		執行時	導通時								
			非導通時								
	LD>	導通時		0.060		0.0285		0.0285		0.0285	
		非導通時									
	AND>	未執行時		0.060		0.0285		0.0285		0.0285	
		執行時	導通時								
			非導通時								
	OR>	未執行時		0.060		0.0285		0.0285		0.0285	
		執行時	導通時								
			非導通時								
	LD<=	導通時		0.060		0.0285		0.0285		0.0285	
		非導通時									
	AND<=	未執行時		0.060		0.0285		0.0285		0.0285	
		執行時	導通時								
			非導通時								

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	OR<=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	LD<	導通時			0.060		0.0285		0.0285		0.0285
		非導通時									
	AND<	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	OR<	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	LD>=	導通時			0.060		0.0285		0.0285		0.0285
		非導通時									
	AND>=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	OR>=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	LDD=	導通時			0.060		0.0285		0.0285		0.0285
		非導通時									
	ANDD=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	ORD=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	LDD<>	導通時			0.060		0.0285		0.0285		0.0285
		非導通時									
	ANDD<>	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	ORD<>	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	LDD>	導通時			0.060		0.0285		0.0285		0.0285
		非導通時									
	ANDD>	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	ORD>	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	LDD<=	導通時			0.060		0.0285		0.0285		0.0285
		非導通時									
	ANDD<=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	ORD<=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	LDD<	導通時			0.060		0.0285		0.0285		0.0285
		非導通時									
	ANDD<	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	ORD<	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	LDD>=	導通時			0.060		0.0285		0.0285		0.0285
		非導通時									
	ANDD>=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	ORD>=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時								
			非導通時								
	+ Ⓢ Ⓓ	執行時			0.060		0.0285		0.0285		0.0285
	+ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.080		0.038		0.038		0.038
	- Ⓢ Ⓓ	執行時			0.060		0.0285		0.0285		0.0285
	- Ⓢ1 Ⓢ2 Ⓓ	執行時			0.080		0.038		0.038		0.038
	D+ Ⓢ Ⓓ	執行時			0.060		0.0285		0.0285		0.0285
	D+ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.080		0.038		0.038		0.038
	D- Ⓢ Ⓓ	執行時			0.060		0.0285		0.0285		0.0285
	D- Ⓢ1 Ⓢ2 Ⓓ	執行時			0.080		0.038		0.038		0.038
	* Ⓢ1 Ⓢ2 Ⓓ	執行時			0.120		0.057		0.057		0.057
	/ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.220		0.110		0.110		0.110
	D* Ⓢ1 Ⓢ2 Ⓓ	執行時			0.200		0.095		0.095		0.095
	D/ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.340		0.170		0.170		0.170
	B+ Ⓢ Ⓓ	執行時		3.300	5.500	3.000	4.100	3.000	4.100	3.000	4.100
	B+ Ⓢ1 Ⓢ2 Ⓓ	執行時		4.600	6.200	4.200	5.900	4.200	5.900	4.200	5.900
	B- Ⓢ Ⓓ	執行時		3.300	4.400	2.900	3.800	2.900	3.800	2.900	3.800
	B- Ⓢ1 Ⓢ2 Ⓓ	執行時		4.600	6.300	4.200	4.600	4.200	4.600	4.200	4.600
	B* Ⓢ1 Ⓢ2 Ⓓ	執行時		4.000	4.800	3.400	4.800	3.400	4.800	3.400	4.800
	B/ Ⓢ1 Ⓢ2 Ⓓ	執行時		4.200	5.700	3.700	5.200	3.700	5.200	3.700	5.200
	E+ Ⓢ Ⓓ	單精度	Ⓢ=0、Ⓓ=0		0.120		0.057		0.057		0.057
			Ⓢ=2^{127}、Ⓓ=2^{127}		0.120		0.057		0.057		0.057
	E+ Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0		0.140		0.0665		0.0665		0.0665
			Ⓢ1=2^{127}、Ⓢ2=2^{127}		0.140		0.0665		0.0665		0.0665
	E- Ⓢ Ⓓ	單精度	Ⓢ=0、Ⓓ=0		0.120		0.057		0.057		0.057
			Ⓢ=2^{127}、Ⓓ=2^{127}		0.120		0.057		0.057		0.057
	E- Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0		0.140		0.0665		0.0665		0.0665
			Ⓢ1=2^{127}、Ⓢ2=2^{127}		0.140		0.0665		0.0665		0.0665
	E* Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0		0.120		0.057		0.057		0.057
			Ⓢ1=2^{127}、Ⓢ2=2^{127}		0.120		0.057		0.057		0.057
	E/ Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=2^{127}、Ⓢ2=2^{127}	4.500	5.600	3.900	4.900		0.285		0.285

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	INC	執行時			0.040		0.019		0.019		0.019
	DINC	執行時			0.040		0.019		0.019		0.019
	DEC	執行時			0.040		0.019		0.019		0.019
	DDEC	執行時			0.040		0.019		0.019		0.019
	BCD	執行時			0.120		0.057		0.057		0.057
	DBCD	執行時			0.200		0.095		0.095		0.095
	BIN	執行時			0.060		0.0285		0.0285		0.0285
	DBIN	執行時			0.060		0.0285		0.0285		0.0285
	FLT	單精度	Ⓢ=0		0.100		0.0475		0.0475		0.0475
			Ⓢ=$7FFF_H$		0.100		0.0475		0.0475		0.0475
	DFLT	單精度	Ⓢ=0		0.100		0.0475		0.0475		0.0475
			Ⓢ=$7FFFFFFF_H$		0.100		0.0475		0.0475		0.0475
	INT	單精度	Ⓢ=0		0.100		0.0475		0.0475		0.0475
			Ⓢ=32766.5		0.100		0.0475		0.0475		0.0475
	DINT	單精度	Ⓢ=0		0.100		0.0475		0.0475		0.0475
			Ⓢ=1234567890.3		0.100		0.0475		0.0475		0.0475
	MOV	---			0.040		0.019		0.019		0.019
	DMOV	---			0.040		0.019		0.019		0.019
	EMOV	---			0.040		0.019		0.019		0.019
	CML	---			0.040		0.019		0.019		0.019
	DCML	---			0.040		0.019		0.019		0.019
	BMOV			6.300	8.200	5.400	7.000	5.400	7.000	5.400	7.000
		n=1	SM237=OFF*1	8.200	10.600	3.900	5.100	3.900	5.100	3.900	5.100
			SM237=ON*1	6.000	7.800	2.900	3.700	2.900	3.700	2.900	3.700
		n=96		7.100	8.800	5.900	7.600	5.900	7.600	5.900	7.600
			SM237=OFF*1	9.300	11.900	4.400	5.700	4.400	5.700	4.400	5.700
			SM237=ON*1	7.100	9.100	3.400	4.300	3.400	4.300	3.400	4.300
	FMOV			5.300	5.900	4.200	4.800	4.200	4.800	4.200	4.800
		n=1	SM237=OFF*1	7.000	8.000	3.400	3.800	3.400	3.800	3.400	3.800
			SM237=ON*1	5.900	6.800	2.800	3.200	2.800	3.200	2.800	3.200
		n=96		5.300	7.600	4.400	6.800	4.400	6.800	4.400	6.800
			SM237=OFF*1	7.400	12.200	3.600	5.800	3.600	5.800	3.600	5.800
			SM237=ON*1	6.300	11.000	3.000	5.200	3.000	5.200	3.000	5.200
	XCH	---		2.500	2.900	1.800	2.300	1.800	2.300	1.800	2.300
	DXCH	---		2.800	3.700	2.100	2.900	2.100	2.900	2.100	2.900
	DFMOV*2	n=1	SM237=OFF	2.600	3.750	2.250	3.150	2.250	3.150	2.250	3.150
			SM237=ON	2.050	2.250	1.750	1.750	1.750	1.750	1.750	1.750
		n=96	SM237=OFF	5.850	7.350	4.200	5.500	4.200	5.500	5.380	7.440
			SM237=ON	5.300	6.000	3.650	4.150	3.650	4.150	4.700	5.500
	CJ	---		1.800	2.800	1.400	2.400	1.400	2.400	1.400	2.400
	SCJ	---		1.800	2.800	1.400	2.400	1.400	2.400	1.400	2.400
	JMP	---		1.800	2.800	1.100	2.400	1.100	2.400	1.100	2.400

*1: 在序列號的前 5 位數爲 “ 10012 ” 以後的 Q03UDCPU、Q04UDHCPU、Q06UDHCPU 中可以使用。

*2: 在序列號的前 5 位數爲 “ 10102 ” 的 Q03UD(E)CPU、Q04UD(E)HCPU、Q06UD(E)HCPU、Q13UD(E)HCPU、Q26UD(E)HCPU 中可以使用。

分類	指令	條件（軟元件）	處理時間（μs）							
			Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	WAND Ⓢ Ⓓ	執行時		0.060		0.0285		0.0285		0.0285
	WAND Ⓢ1 Ⓢ2 Ⓓ	執行時		0.080		0.038		0.038		0.038
	DAND Ⓢ Ⓓ	執行時		0.060		0.0285		0.0285		0.0285
	DAND Ⓢ1 Ⓢ2 Ⓓ	執行時		0.080		0.038		0.038		0.038
	WOR Ⓢ Ⓓ	執行時		0.060		0.0285		0.0285		0.0285
	WOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.080		0.038		0.038		0.038
	DOR Ⓢ Ⓓ	執行時		0.060		0.0285		0.0285		0.0285
	DOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.080		0.038		0.038		0.038
	WXOR Ⓢ Ⓓ	執行時		0.060		0.0285		0.0285		0.0285
	WXOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.080		0.038		0.038		0.038
	DXOR Ⓢ Ⓓ	執行時		0.060		0.0285		0.0285		0.0285
	DXOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.080		0.038		0.038		0.038
	WXNR Ⓢ Ⓓ	執行時		0.060		0.0285		0.0285		0.0285
	WXNR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.080		0.038		0.038		0.038
	DXNR Ⓢ Ⓓ	執行時		0.060		0.0285		0.0285		0.0285
	DXNR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.080		0.038		0.038		0.038
	ROR Ⓓ n	n=1	2.300	3.100	1.700	2.500	1.700	2.500	1.700	2.500
		n=15	2.400	3.100	1.800	2.500	1.800	2.500	1.800	2.500
	RCR Ⓓ n	n=1	2.300	3.900	1.700	3.200	1.700	3.200	1.700	3.200
		n=15	2.400	4.100	1.700	3.200	1.700	3.200	1.700	3.200
	ROL Ⓓ n	n=1	2.400	3.300	1.800	3.200	1.800	3.200	1.800	3.200
		n=15	2.400	3.300	1.800	3.200	1.800	3.200	1.800	3.200
	RCL Ⓓ n	n=1	2.400	2.700	1.800	2.100	1.800	2.100	1.800	2.100
		n=15	2.400	2.800	1.800	2.200	1.800	2.200	1.800	2.200
	DROR Ⓓ n	n=1	2.400	3.400	1.900	2.700	1.900	2.700	1.900	2.700
		n=31	2.500	3.400	1.900	2.700	1.900	2.700	1.900	2.700
	DRCR Ⓓ n	n=1	2.500	4.800	1.900	4.200	1.900	4.200	1.900	4.200
		n=31	2.500	4.900	1.900	4.200	1.900	4.200	1.900	4.200
	DROL Ⓓ n	n=1	2.500	3.900	1.800	3.200	1.800	3.200	1.800	3.200
		n=31	2.500	3.900	1.800	3.300	1.800	3.300	1.800	3.300
	DRCL Ⓓ n	n=1	2.500	4.800	1.900	3.800	1.900	3.800	1.900	3.800
		n=31	2.500	4.600	1.900	3.800	1.900	3.800	1.900	3.800
	SFR Ⓓ n	n=1	2.400	3.900	1.700	2.600	1.700	2.600	1.700	2.600
		n=15	2.300	3.900	1.800	2.600	1.800	2.600	1.800	2.600
	SFL Ⓓ n	n=1	2.400	4.300	1.800	2.700	1.800	2.700	1.800	2.700
		n=15	2.400	4.300	1.800	2.700	1.800	2.700	1.800	2.700
	DSFR Ⓓ n	n=1	2.700	4.800	2.200	4.300	2.200	4.300	2.200	4.300
		n=96	32.600	35.900	23.900	26.100	23.900	26.100	23.900	26.100
	DSFL Ⓓ n	n=1	2.700	4.600	2.100	4.000	2.100	4.000	2.100	4.000
		n=96	32.600	35.300	23.700	25.800	23.700	25.800	23.700	25.800
	SUM	Ⓢ=0	3.400	4.300	2.900	3.600	2.900	3.600	2.900	3.600
		Ⓢ=$FFFF_H$	3.500	4.200	2.900	3.600	2.900	3.600	2.900	3.600
	SEG	執行時	2.100	2.800	1.500	2.100	1.500	2.100	1.500	2.100
	FOR	---	1.200	2.400	0.870	2.100	0.870	2.100	0.870	2.100
	CALL Pn	文件內指標	2.600	4.000	2.300	3.600	2.300	3.600	2.300	3.600
		公共指標	4.000	5.300	3.200	4.900	3.200	4.900	3.200	4.900
	CALL Pn Ⓢ1～Ⓢ5	---	28.700	33.400	26.100	29.300	26.100	29.300	26.100	29.300

備註

對於表中未記述上升沿執行指令 (□ P) 的指令，其處理時間與 ON 時執行的指令的處理時間相同。

例 MOVP 指令、WANDP 指令等。

(2) 使用了文件寄存器、擴展資料寄存器、擴展鏈結寄存器、模組訪問軟元件時的加法運算時間一覽表

(a) 使用 Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 時

軟元件名		資料	軟元件指定位置	加法運算時間 (μs)			
				Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU
文件寄存器 (R)	使用標準 RAM 時	位	源	0.100	0.100	0.100	0.100
			目標	0.220	0.220	0.220	0.220
		字	源	0.100	0.100	0.100	0.100
			目標	0.100	0.100	0.100	0.100
		雙字	源	0.200	0.200	0.200	0.200
			目標	0.200	0.200	0.200	0.200
	使用 SRAM 卡 (Q2MEM-1MBS、Q2MEM-2MBS) 時	位	源	---	---	---	0.220
			目標	---	---	---	0.420
		字	源	---	---	---	0.220
			目標	---	---	---	0.180
		雙字	源	---	---	---	0.440
			目標	---	---	---	0.380
	使用 SRAM 卡 (Q3MEM-4MBS、Q3MEM-8MBS) 時	位	源	---	---	---	0.160
			目標	---	---	---	0.320
		字	源	---	---	---	0.160
			目標	---	---	---	0.140
		雙字	源	---	---	---	0.320
			目標	---	---	---	0.300
文件寄存器 (ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	使用標準 RAM 時	位	源	0.220	0.180	0.160	0.140
			目標	0.360	0.320	0.300	0.280
		字	源	0.220	0.180	0.160	0.140
			目標	0.220	0.180	0.160	0.140
		雙字	源	0.320	0.280	0.260	0.240
			目標	0.320	0.280	0.260	0.240
	使用 SRAM 卡 (Q2MEM-1MBS、Q2MEM-2MBS) 時	位	源	---	---	---	0.260
			目標	---	---	---	0.480
		字	源	---	---	---	0.260
			目標	---	---	---	0.220
		雙字	源	---	---	---	0.480
			目標	---	---	---	0.420
	使用 SRAM 卡 (Q3MEM-4MBS、Q3MEM-8MBS) 時	位	源	---	---	---	0.200
			目標	---	---	---	0.380
		字	源	---	---	---	0.200
			目標	---	---	---	0.180
		雙字	源	---	---	---	0.360
			目標	---	---	---	0.340
模組訪問軟元件 (多 CPU 高速通信區) (U3En\G10000 ～)		位	源	---	---	---	---
			目標	---	---	---	---
		字	源	---	---	---	---
			目标	---	---	---	---
		雙字	源	---	---	---	---
			目標	---	---	---	---

(b) 使用 Q03UD(E)CPU、Q04UD(E)HC0U、Q06UD(E)HCPU、Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU、Q50UDEHCPU、Q100UDEHCPU 時

軟元件名		資料	軟元件指定位置	加法運算時間（μs）			
				Q03UD(E)CPU	Q04UD(E)HCPU、Q06UD(E)HCPU	Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU	Q50UDEHCPU、Q100UDEHCPU
文件寄存器 (R)	使用標準 RAM 時	位	源	0.100	0.048	0.048	0.048
			目標	0.100	0.038	0.038	0.038
		字	源	0.100	0.048	0.048	0.048
			目標	0.100	0.038	0.038	0.038
		雙字	源	0.200	0.095	0.095	0.095
			目標	0.200	0.086	0.086	0.086
	使用 SRAM 卡 (Q2MEM-1MBS、Q2MEM-2MBS) 時	位	源	0.220	0.200	0.200	0.200
			目標	0.180	0.162	0.162	0.162
		字	源	0.220	0.200	0.200	0.200
			目標	0.180	0.162	0.162	0.162
		雙字	源	0.440	0.399	0.399	0.399
			目標	0.380	0.361	0.361	0.361
	使用 SRAM 卡 (Q3MEM-4MBS、Q3MEM-8MBS) 時	位	源	0.160	0.152	0.152	0.152
			目標	0.140	0.133	0.133	0.133
		字	源	0.160	0.152	0.152	0.152
			目標	0.140	0.133	0.133	0.133
		雙字	源	0.320	0.304	0.304	0.304
			目標	0.300	0.295	0.295	0.295
文件寄存器 (ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	使用標準 RAM 時	位	源	0.120	0.057	0.057	0.057
			目標	0.120	0.048	0.048	0.048
		字	源	0.120	0.057	0.057	0.057
			目標	0.120	0.048	0.048	0.048
		雙字	源	0.220	0.105	0.105	0.105
			目標	0.220	0.095	0.095	0.095
	使用 SRAM 卡 (Q2MEM-1MBS、Q2MEM-2MBS) 時	位	源	0.240	0.209	0.209	0.209
			目標	0.200	0.171	0.171	0.171
		字	源	0.240	0.209	0.209	0.209
			目標	0.200	0.171	0.171	0.171
		雙字	源	0.460	0.409	0.409	0.409
			目標	0.400	0.371	0.371	0.371
	使用 SRAM 卡 (Q3MEM-4MBS、Q3MEM-8MBS) 時	位	源	0.180	0.162	0.162	0.162
			目標	0.160	0.143	0.143	0.143
		字	源	0.180	0.162	0.162	0.162
			目標	0.160	0.143	0.143	0.143
		雙字	源	0.340	0.314	0.314	0.314
			目標	0.320	0.304	0.304	0.304
模組訪問軟元件 (多 CPU 高速通信區) (U3En\G10000 ～)		位	源	0.220	0.181	0.181	0.181
			目標	0.140	0.105	0.105	0.105
		字	源	0.220	0.181	0.181	0.181
			目標	0.140	0.105	0.105	0.105
		雙字	源	0.500	0.437	0.437	0.437
			目標	0.340	0.285	0.285	0.285

(3) 在 OUT/SET/RST 指令的軟元件中，使用了 F、T(ST)、C 軟元件時的加法運算時間一覽表

(a) 使用 Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 時

指令名	軟元件名	條件		加法運算時間（μs）			
				Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU
OUT	F	未執行時		2.900	2.900	2.900	2.100
		執行時	顯示時	116.000	116.000	116.000	68.800
			顯示結束	116.000	116.000	116.000	61.600
	T(ST)、C	未執行時		0.360	0.240	0.180	0.120
		執行時	時間到後	0.360	0.240	0.180	0.120
			計數時	0.360	0.240	0.180	0.120
SET	F	未執行時		0.120	0.080	0.006	0.004
		執行時	顯示時	116.000	116.000	116.000	68.600
			顯示結束	116.000	116.000	116.000	65.700

指令名	軟元件名	條件		加法運算時間（μs）			
				Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU
RST	F	未執行時		0.120	0.080	0.006	0.004
		執行時	顯示時	55.800	55.800	55.800	26.500
			顯示結束	29.200	29.200	29.200	21.600
	T(ST)、C	未執行時		0.360	0.240	0.180	0.120
		執行時		0.360	0.240	0.180	0.120

(b) 使用 Q03UD(E)CPU、Q04UD(E)HCPU、Q06UD(E)HCPU、Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU、Q50UDEHCPU、Q100UDEHCPU 時

指令名	軟元件名	條件		加法運算時間（μs）			
				Q03UD(E)CPU	Q04UD(E)HCPU、Q06UD(E)HCPU	Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU	Q50UDEHCPU、Q100UDEHCPU
OUT	F	未執行時		1.940	1.570	1.570	1.570
		執行時	顯示時	39.930	38.090	38.090	38.090
			顯示結束	39.750	37.980	37.980	37.980
	T(ST)、C	未執行時		0.060	0.030	0.030	0.030
		執行時	時間到後	0.060	0.030	0.030	0.030
SET	F	未執行時		0.000	0.000	0.000	0.000
		執行時	顯示時	42.900	40.600	40.600	40.600
			顯示結束	39.270	37.900	37.900	37.900
RST	F	未執行時		0.000	0.000	0.000	0.000
		執行時	顯示時	45.260	36.600	36.600	36.600
			顯示結束	19.020	16.190	16.190	16.190
	T(ST)、C	未執行時		0.060	0.030	0.030	0.030
		執行時		0.060	0.030	0.030	0.030

附錄 1.4.2 子集指令以外的指令處理時間

子集處理指令以外的處理時間如下所示。

要 點

- 指令中使用的軟元件未滿足子集處理的軟元件條件時的子集指令以外的指令處理時間一覽表如下述 (1) 所示。（關於子集處理的軟元件條件，請參閱 98 頁 3.5.1 項。）
  關於下表中未記載的指令，請參閱 723 頁附錄 1.4.1(5) 的指令處理時間。
- 使用文件寄存器 (R、ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)、模組訪問軟元件 (Un\G □、U3En\G0 ～ G4095)、鏈結直接軟元件 (Jn\□) 時，請參閱 (2) 的加法運算時間，對各指令的處理時間進行加法運算。
- 由於高速緩衝記憶體功能的影響，各指令的處理時間不恒定，因此對最小值和最大值進行了記載。

(1) 子集指令以外的指令處理時間一覽表

(a) 使用 Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 時

分類	指令	條件（軟元件）	處理時間（μs）							
			Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
順控程式指令	ANB ORB MPS MRD MPP	---	0.120		0.080		0.060		0.040	
	INV	未執行時 執行時	0.120		0.080		0.060		0.040	
	MEP MEF	未執行時 執行時	0.120		0.080		0.060		0.040	
	EGP EGF	未執行時 執行時	0.120		0.080		0.060		0.040	

分類	指令	條件（軟元件）			處理時間（μs）							
					Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
					最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
順控程式指令	PLS	---			1.800	1.900	1.800	1.900	1.800	1.900	1.300	1.600
	PLF	---			1.800	1.900	1.800	1.900	1.800	1.900	1.600	1.700
	FF	未執行時				0.240		0.160		0.120		0.080
		執行時			1.700	1.800	1.700	1.800	1.700	1.800	1.200	1.500
	DELTA	未執行時				0.240		0.160		0.120		0.080
		執行時			4.000	14.700	4.000	14.700	4.000	14.700	2.800	3.600
	SFT	未執行時				0.240		0.160		0.120		0.800
		執行時			1.800	12.600	1.800	12.600	1.800	12.600	1.600	6.600
	MC	---				0.240		0.160		0.120		0.080
	MCR	---				0.120		0.080		0.060		0.040
	FEND	進行出錯檢查			250.000	250.000	250.000	250.000	250.000	250.000	175.000	252.000
	END	不進行出錯檢查			250.000	250.000	250.000	250.000	250.000	250.000	175.000	221.000
	STOP	---				---		---		---		---
	NOP NOPLF PAGE	---				0.120		0.080		0.060		0.040
基本指令	LDE=	單精度	導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.700	10.100
			非導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.700	10.100
	ANDE=	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.200	12.500
				非導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.400	11.900
	ORE=	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.600	10.800
				非導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.500	9.800
	LDE< >	單精度	導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.700	7.700
			非導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.600	8.200
	ANDE< >	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.300	14.200
				非導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.400	14.200
	ORE< >	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.600	6.700
				非導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.400	6.600
	LDE>	單精度	導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.700	13.700
			非導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.600	13.700
	ANDE>	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.300	8.100
				非導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.200	8.100
	ORE>	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.600	8.500
				非導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.400	8.100
	LDE<=	單精度	導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.700	11.100
			非導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.700	9.600
	ANDE<=	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.100	7.800
				非導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.400	8.200
	ORE<=	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.500	10.300
				非導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.400	9.800
	LDE<	單精度	導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.700	11.500
			非導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.700	10.900
	ANDE<	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.300	9.200
				非導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.400	9.400

分類	指令	條件（軟元件）			處理時間（μs）							
					Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
					最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	ORE<	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.600	10.400
				非導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.400	9.800
	LDE>=	單精度	導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.700	12.200
			非導通時		4.400	20.900	4.400	20.900	4.400	20.900	4.700	11.800
	ANDE>=	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.100	6.700
				非導通時	4.200	19.600	4.200	19.600	4.200	19.600	4.400	7.000
	ORE>=	單精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.600	14.000
				非導通時	4.200	17.400	4.200	17.400	4.200	17.400	4.500	14.300
	LDED=	雙精度	導通時		4.700	37.400	4.700	37.400	4.700	37.400	4.200	21.000
			非導通時		4.700	37.400	4.700	37.400	4.700	37.400	5.100	21.900
	ANDED=	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.500	34.700	4.500	34.700	4.500	34.700	3.800	17.800
				非導通時	4.500	34.700	4.500	34.700	4.500	34.700	4.100	18.100
	ORED=	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.700	33.200	4.700	33.200	4.700	33.200	4.100	23.800
				非導通時	4.700	33.200	4.700	33.200	4.700	33.200	4.900	25.500
	LDED< >	雙精度	導通時		4.700	37.400	4.700	37.400	4.700	37.400	5.100	23.500
			非導通時		4.700	37.400	4.700	37.400	4.700	37.400	4.200	22.600
	ANDED< >	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.500	34.700	4.500	34.700	4.500	34.700	4.000	18.800
				非導通時	4.500	34.700	4.500	34.700	4.500	34.700	4.000	18.700
	ORED< >	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.700	33.200	4.700	33.200	4.700	33.200	5.000	25.200
				非導通時	4.700	33.200	4.700	33.200	4.700	33.200	4.100	23.400
	LDED>	雙精度	導通時		4.700	37.400	4.700	37.400	4.700	37.400	5.100	25.100
			非導通時		4.700	37.400	4.700	37.400	4.700	37.400	4.200	23.400
	ANDED>	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.500	34.700	4.500	34.700	4.500	34.700	4.000	19.500
				非導通時	4.500	34.700	4.500	34.700	4.500	34.700	4.100	19.700
	ORED>	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.700	33.200	4.700	33.200	4.700	33.200	5.000	24.200
				非導通時	4.700	33.200	4.700	33.200	4.700	33.200	4.900	25.800
	LDED<=	雙精度	導通時		4.700	37.400	4.700	37.400	4.700	37.400	4.200	22.500
			非導通時		4.700	37.400	4.700	37.400	4.700	37.400	4.200	13.500
	ANDED<=	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.500	34.700	4.500	34.700	4.500	34.700	4.000	19.600
				非導通時	4.500	34.700	4.500	34.700	4.500	34.700	4.100	19.700
	ORED<=	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.700	33.200	4.700	33.200	4.700	33.200	5.000	26.300
				非導通時	4.700	33.200	4.700	33.200	4.700	33.200	5.000	25.200
	LDED<	雙精度	導通時		4.700	37.400	4.700	37.400	4.700	37.400	5.100	25.000
			非導通時		4.700	37.400	4.700	37.400	4.700	37.400	4.200	24.100
	ANDED<	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.500	34.700	4.500	34.700	4.500	34.700	4.000	19.400
				非導通時	4.500	34.700	4.500	34.700	4.500	34.700	4.100	19.700
	ORED<	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.700	33.200	4.700	33.200	4.700	33.200	5.000	25.100
				非導通時	4.700	33.200	4.700	33.200	4.700	33.200	5.000	25.100

分類	指令	條件（軟元件）			處理時間（μs）							
					Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
					最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	LDED>=	雙精度	導通時		4.700	37.400	4.700	37.400	4.700	37.400	4.200	13.100
			非導通時		4.700	37.400	4.700	37.400	4.700	37.400	4.300	13.100
	ANDED>=	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.500	34.700	4.500	34.700	4.500	34.700	3.900	19.500
				非導通時	4.500	34.700	4.500	34.700	4.500	34.700	4.100	19.800
	ORED>=	雙精度	未執行時			0.360		0.240		0.180		0.120
			執行時	導通時	4.700	33.200	4.700	33.200	4.700	33.200	5.000	25.100
				非導通時	4.700	33.200	4.700	33.200	4.700	33.200	4.200	18.500
	LD$=	導通時			8.300	38.500	8.300	38.500	8.300	38.500	5.500	14.900
		非導通時			8.300	38.500	8.300	38.500	8.300	38.500	5.500	15.600
	AND$=	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		7.200	37.300	7.200	37.300	7.200	37.300	5.200	13.800
			非導通時		7.200	37.300	7.200	37.300	7.200	37.300	5.300	14.500
	OR$=	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		7.500	36.600	7.500	36.600	7.500	36.600	5.500	14.900
			非導通時		7.500	36.600	7.500	36.600	7.500	36.600	5.300	14.600
	LD$< >	導通時			8.300	39.300	8.300	39.300	8.300	39.300	5.600	15.200
		非導通時			8.300	39.300	8.300	39.300	8.300	39.300	5.600	15.400
	AND$< >	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		8.000	38.200	8.000	38.200	8.000	38.200	4.300	21.500
			非導通時		8.000	38.200	8.000	38.200	8.000	38.200	4.500	23.400
	OR$< >	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		8.300	37.300	8.300	37.300	8.300	37.300	5.400	17.700
			非導通時		8.300	37.300	8.300	37.300	8.300	37.300	5.300	19.400
	LD$>	導通時			8.300	41.600	8.300	41.600	8.300	41.600	6.400	19.200
		非導通時			8.300	41.600	8.300	41.600	8.300	41.600	5.600	20.100
	AND$>	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		8.000	38.100	8.000	38.100	8.000	38.100	4.500	15.400
			非導通時		8.000	38.100	8.000	38.100	8.000	38.100	4.600	15.300
	OR$>	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		8.200	35.700	8.200	35.700	8.200	35.700	5.400	20.000
			非導通時		8.200	35.700	8.200	35.700	8.200	35.700	5.400	22.100
	LD$<=	導通時			8.300	39.200	8.300	39.200	8.300	39.200	5.800	12.800
		非導通時			8.300	39.200	8.300	39.200	8.300	39.200	6.300	13.900
	AND$<=	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		7.100	36.500	7.100	36.500	7.100	36.500	6.000	16.000
			非導通時		7.100	36.500	7.100	36.500	7.100	36.500	6.100	16.200
	OR$<=	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		7.400	35.600	7.400	35.600	7.400	35.600	4.700	14.600
			非導通時		7.400	35.600	7.400	35.600	7.400	35.600	4.600	14.400
	LD$<	導通時			7.400	40.000	7.400	40.000	7.400	40.000	4.800	17.000
		非導通時			7.400	40.000	7.400	40.000	7.400	40.000	5.500	18.000
	AND$<	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		8.000	37.300	8.000	37.300	8.000	37.300	5.900	13.400
			非導通時		8.000	37.300	8.000	37.300	8.000	37.300	6.200	14.500
	OR$<	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		8.300	35.600	8.300	35.600	8.300	35.600	6.200	18.700
			非導通時		8.300	35.600	8.300	35.600	8.300	35.600	5.400	19.700
	LD$>=	導通時			7.400	38.300	7.400	38.300	7.400	38.300	4.800	10.000
		非導通時			7.400	38.300	7.400	38.300	7.400	38.300	5.500	11.200
	AND$>=	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		7.200	37.300	7.200	37.300	7.200	37.300	4.400	21.600
			非導通時		7.200	37.300	7.200	37.300	7.200	37.300	4.500	21.800
	OR$>=	未執行時				0.360		0.240		0.180		0.120
		執行時	導通時		8.200	36.400	8.200	36.400	8.200	36.400	5.400	15.400
			非導通時		8.200	36.400	8.200	36.400	8.200	36.400	5.300	15.300

分類	指令	條件（軟元件）	處理時間（μs）							
			Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	BKCMP= (S1) (S2) (D) n	n=1	15.300	36.100	15.300	36.100	15.300	36.100	8.200	22.600
		n=96	64.500	85.500	64.500	85.500	64.500	85.500	57.400	72.500
	BKCMP< > (S1) (S2) (D) n	n=1	15.300	36.100	15.300	36.100	15.300	36.100	8.200	22.500
		n=96	66.600	87.500	66.600	87.500	66.600	87.500	59.500	74.500
	BKCMP> (S1) (S2) (D) n	n=1	15.300	36.100	15.300	36.100	15.300	36.100	8.200	23.100
		n=96	66.600	87.500	66.600	87.500	66.600	87.500	59.500	74.400
	BKCMP<= (S1) (S2) (D) n	n=1	15.300	36.100	15.300	36.100	15.300	36.100	8.200	22.500
		n=96	64.500	85.500	64.500	85.500	64.500	85.500	57.400	72.400
	BKCMP< (S1) (S2) (D) n	n=1	15.300	36.100	15.300	36.100	15.300	36.100	8.300	23.000
		n=96	66.600	87.500	66.600	87.500	66.600	87.500	59.500	74.500
	BKCMP>= (S1) (S2) (D) n	n=1	15.300	36.100	15.300	36.100	15.300	36.100	8.200	22.500
		n=96	64.500	85.500	64.500	85.500	64.500	85.500	57.400	72.400
	DBKCMP= (S1) (S2) (D) n	n=1	15.800	36.300	15.800	36.300	15.800	36.300	9.350	29.000
		n=96	64.900	85.700	64.900	85.700	64.900	85.700	60.700	78.400
	DBKCMP< > (S1) (S2) (D) n	n=1	15.700	36.300	15.700	36.300	15.700	36.300	9.350	28.900
		n=96	67.000	87.700	67.000	87.700	67.000	87.700	62.500	80.300
	DBKCMP> (S1) (S2) (D) n	n=1	15.800	36.300	15.800	36.300	15.800	36.300	9.350	29.000
		n=96	67.000	87.700	67.000	87.700	67.000	87.700	62.600	80.300
	DBKCMP<= (S1) (S2) (D) n	n=1	15.700	36.300	15.700	36.300	15.700	36.300	9.350	29.000
		n=96	64.800	85.700	64.800	85.700	64.800	85.700	60.800	78.400
	DBKCMP< (S1) (S2) (D) n	n=1	15.800	36.300	15.800	36.300	15.800	36.300	9.350	29.000
		n=96	67.000	87.700	67.000	87.700	67.000	87.700	62.700	80.400
	DBKCMP>= (S1) (S2) (D) n	n=1	15.700	36.300	15.700	36.300	15.700	36.300	9.300	29.000
		n=96	64.800	85.700	64.800	85.700	64.800	85.700	60.700	78.400
	DB+ (S) (D)	執行時	5.750	13.300	5.750	13.300	5.750	13.300	4.900	7.500
	DB+ (S1) (S2) (D)	執行時	5.650	13.200	5.650	13.200	5.650	13.200	5.200	11.000
	DB- (S) (D)	執行時	5.750	12.700	5.750	12.700	5.750	12.700	4.900	10.200
	DB- (S1) (S2) (D)	執行時	5.650	12.600	5.650	12.600	5.650	12.600	5.200	8.600
	DB* (S1) (S2) (D)	執行時	8.750	40.200	8.750	40.200	8.750	40.200	8.300	22.200
	DB/ (S1) (S2) (D)	執行時	5.750	21.500	5.750	21.500	5.750	21.500	6.100	19.200

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	ED+ Ⓢ Ⓓ	雙精度	Ⓢ=0、Ⓓ=0	4.500	26.700	4.500	26.700	4.500	26.700	4.800	16.800
			Ⓢ=2^{1023}、Ⓓ=2^{1023}	5.800	32.900	5.800	32.900	5.800	32.900	4.800	16.800
	ED+ Ⓢ1 Ⓢ2 Ⓓ	雙精度	Ⓢ1=0、Ⓢ2=0	5.450	35.400	5.450	35.400	5.450	35.400	7.100	20.100
			Ⓢ1=2^{1023}、Ⓢ2=2^{1023}	6.750	41.400	6.750	41.400	6.750	41.400	7.100	20.100
	ED- Ⓢ Ⓓ	雙精度	Ⓢ=0、Ⓓ=0	5.200	25.900	5.200	25.900	5.200	25.900	5.000	17.300
			Ⓢ=2^{1023}、Ⓓ=2^{1023}	6.000	27.700	6.000	27.700	6.000	27.700	5.000	17.300
	ED- Ⓢ1 Ⓢ2 Ⓓ	雙精度	Ⓢ1=0、Ⓢ2=0	5.550	32.900	5.550	32.900	5.550	32.900	6.000	16.300
			Ⓢ1=2^{1023}、Ⓢ2=2^{1023}	5.750	33.900	5.750	33.900	5.750	33.900	6.000	16.300
	ED* Ⓢ1 Ⓢ2 Ⓓ	雙精度	Ⓢ1=0、Ⓢ2=0	5.550	34.400	5.550	34.400	5.550	34.400	10.500	22.300
			Ⓢ1=2^{1023}、Ⓢ2=2^{1023}	5.950	39.100	5.950	39.100	5.950	39.100	10.500	22.300
	ED/ Ⓢ1 Ⓢ2 Ⓓ	雙精度	Ⓢ1=2^{1023}、Ⓢ2=2^{1023}	8.050	44.200	8.050	44.200	8.050	44.200	7.500	27.200
	BK+ Ⓢ1 Ⓢ2 Ⓓ n	n=1		13.500	28.500	13.500	28.500	13.500	28.500	12.100	19.700
		n=96		63.100	78.200	63.100	78.200	63.100	78.200	61.700	69.300
	BK- Ⓢ1 Ⓢ2 Ⓓ n	n=1		13.500	28.500	13.500	28.500	13.500	28.500	12.100	20.600
		n=96		63.100	78.200	63.100	78.200	63.100	78.200	61.700	70.200
	DBK+ Ⓢ1 Ⓢ2 Ⓓ n	n=1		10.100	24.200	10.100	24.200	10.100	24.200	7.050	19.200
		n=96		59.800	73.900	59.800	73.900	59.800	73.900	59.400	68.900
	DBK- Ⓢ1 Ⓢ2 Ⓓ n	n=1		10.100	24.200	10.100	24.200	10.100	24.200	7.050	19.900
		n=96		59.800	73.900	59.800	73.900	59.800	73.900	59.400	69.600
	$+ Ⓢ Ⓓ	---		15.400	64.300	15.400	64.300	15.400	64.300	14.400	34.000
	$+ Ⓢ1 Ⓢ2 Ⓓ	---		19.700	71.000	19.700	71.000	19.700	71.000	9.200	22.900
	FLTD	雙精度	Ⓢ=0	3.100	19.600	3.100	19.600	3.100	19.600	4.000	8.900
			Ⓢ=$7FFF_H$	3.350	19.900	3.350	19.900	3.350	19.900	3.400	9.000
	DFLTD	雙精度	Ⓢ=0	3.200	20.400	3.200	20.400	3.200	20.400	4.100	10.800
			Ⓢ=$7FFFFFFF_H$	3.450	20.500	3.450	20.500	3.450	20.500	3.600	10.800
	INTD	雙精度	Ⓢ=0	3.200	22.900	3.200	22.900	3.200	22.900	3.500	9.300
			Ⓢ=32766.5	4.100	34.300	4.100	34.300	4.100	34.300	5.100	19.500
	DINTD	雙精度	Ⓢ=0	3.200	23.000	3.200	23.000	3.200	23.000	2.600	6.800
			Ⓢ=1234567890.3	4.050	33.500	4.050	33.500	4.050	33.500	3.400	11.700

分類	指令	條件（軟元件）	處理時間（μs）							
			Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	DBL	執行時	3.300	5.900	3.300	5.900	3.300	5.900	2.700	3.800
	WORD	執行時	3.000	7.250	3.000	7.250	3.000	7.250	2.900	7.000
	GRY	執行時	3.350	7.500	3.350	7.500	3.350	7.500	2.700	6.100
	DGRY	執行時	3.000	7.200	3.000	7.200	3.000	7.200	2.900	4.600
	GBIN	執行時	4.600	9.700	4.600	9.700	4.600	9.700	4.000	8.200
	DGBIN	執行時	5.550	10.700	5.550	10.700	5.550	10.700	5.500	8.000
	NEG	執行時	3.300	6.850	3.300	6.850	3.300	6.850	2.400	4.100
	DNEG	執行時	3.050	5.700	3.050	5.700	3.050	5.700	2.500	4.300
	ENEG	浮點 =0	3.100	7.350	3.100	7.350	3.100	7.350	2.500	3.400
		浮點 =-1.0	3.350	11.700	3.350	11.700	3.350	11.700	2.700	4.500
	EDNEG	浮點 =0	3.000	21.200	3.000	21.200	3.000	21.200	2.200	3.500
		浮點 =-1.0	3.100	22.900	3.100	22.900	3.100	22.900	2.400	3.500
	BKBCD Ⓢ Ⓓ n	n=1	8.700	27.600	8.700	27.600	8.700	27.600	9.700	22.000
		n=96	84.200	104.000	84.200	104.000	84.200	104.000	74.200	86.500
	BKBIN Ⓢ Ⓓ n	n=1	8.450	28.100	8.450	28.100	8.450	28.100	8.900	16.300
		n=96	56.100	75.800	56.100	75.800	56.100	75.800	58.500	65.100
	ECON	---	3.100	21.300	3.100	21.300	3.100	21.300	4.300	6.800
	EDCON	---	5.050	24.000	5.050	24.000	5.050	24.000	2.800	5.400
	EDMOV	---	2.900	22.900	2.900	22.900	2.900	22.900	3.200	7.800
	$MOV	傳送字串 =0	6.250	30.100	6.250	30.100	6.250	30.100	4.500	13.900
		傳送字串 =32	15.500	39.300	15.500	39.300	15.500	39.300	15.400	17.500
	BXCH Ⓓ1 Ⓓ2 n	n=1	8.400	20.900	8.400	20.900	8.400	20.900	8.700	15.200
		n=96	67.100	79.900	67.100	79.900	67.100	79.900	67.200	74.000
	SWAP	---	3.300	3.550	3.300	3.550	3.300	3.550	2.400	2.700
	GOEND	---		0.550		0.550		0.550		0.500
	DI	---	2.800	8.400	2.800	8.400	2.800	8.400	1.800	2.200
	EI	---	4.300	12.300	4.300	12.300	4.300	12.300	3.100	3.800
	IMASK	---	12.900	40.600	12.900	40.600	12.900	40.600	9.800	25.000
	IRET	---		1.000		1.000		1.000		1.000
	RFS X n	n=1	7.500	26.500	7.500	26.500	7.500	26.500	4.300	16.100
		n=96	11.400	30.400	11.400	30.400	11.400	30.400	11.400	23.700
	RFS Y n	n=1	7.300	26.300	7.300	26.300	7.300	26.300	3.800	10.000
		n=96	10.900	29.900	10.900	29.900	10.900	29.900	8.500	15.200
	UDCNT1	---	1.500	7.100	1.500	7.100	1.500	7.100	1.000	2.000
	UDCNT2	---	1.500	6.300	1.500	6.300	1.500	6.300	1.000	4.000
	TTMR	---	5.300	20.900	5.300	20.900	5.300	20.900	3.900	6.100
	STMR	---	8.900	49.800	8.900	49.800	8.900	49.800	7.200	30.000
	ROTC	---	52.300	52.600	52.300	52.600	52.300	52.600	15.200	16.100
	RAMP	---	7.400	30.900	7.400	30.900	7.400	30.900	5.900	18.300
	SPD	---	1.500	6.300	1.500	6.300	1.500	6.300	1.000	2.800
	PLSY	---	6.400	7.100	6.400	7.100	6.400	7.100	3.500	4.700
	PWM	---	3.900	4.600	3.900	4.600	3.900	4.600	3.400	3.400
	MTR	---	10.100	61.400	10.100	61.400	10.100	61.400	20.500	28.400

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	BKAND ⓢ1 ⓢ2 Ⓓ n	n=1		13.600	28.500	13.600	28.500	13.600	28.500	12.100	20.100
		n=96		63.200	78.200	63.200	78.200	63.200	78.200	57.400	63.200
	BKOR ⓢ1 ⓢ2 Ⓓ n	n=1		13.500	28.500	13.500	28.500	13.500	28.500	7.700	13.200
		n=96		63.100	78.200	63.100	78.200	63.100	78.200	57.400	62.800
	BKXOR ⓢ1 ⓢ2 Ⓓ n	n=1		13.600	28.300	13.600	28.300	13.600	28.300	7.800	13.200
		n=96		63.100	78.000	63.100	78.000	63.100	78.000	57.300	62.800
	BKXNR ⓢ1 ⓢ2 Ⓓ n	n=1		13.500	28.300	13.500	28.300	13.500	28.300	7.800	14.100
		n=96		63.100	78.000	63.100	78.000	63.100	78.000	57.400	62.900
	BSFR Ⓓ n	n=1		5.050	21.100	5.050	21.100	5.050	21.100	3.700	6.300
		n=96		9.000	34.800	9.000	34.800	9.000	34.800	10.200	12.800
	BSFL Ⓓ n	n=1		4.800	19.100	4.800	19.100	4.800	19.100	4.500	8.900
		n=96		8.550	34.300	8.550	34.300	8.550	34.300	10.100	14.300
	SFTBR Ⓓ n1 n2	進行了移位的位的數=16/ 移位數 =1		10.300	46.500	10.300	46.500	10.300	46.500	8.800	43.400
		進行了移位的位的數=16/ 移位數 =15		10.300	46.400	10.300	46.400	10.300	46.400	8.750	43.400
	SFTBL Ⓓ n1 n2	進行了移位的位的數=16/ 移位數 =1		10.500	49.800	10.500	49.800	10.500	49.800	8.050	45.100
		進行了移位的位的數=16/ 移位數 =15		10.500	49.800	10.500	49.800	10.500	49.800	8.050	45.100
	SFTWR Ⓓ n1 n2	進行了移位的位的數=16/ 移位數 =1		7.950	24.000	7.950	24.000	7.950	24.000	6.500	22.800
		進行了移位的位的數=16/ 移位數 =15		7.950	24.100	7.950	24.100	7.950	24.100	6.500	22.800
	SFTWL Ⓓ n1 n2	進行了移位的位的數=16/ 移位數 =1		8.700	23.600	8.700	23.600	8.700	23.600	7.350	23.600
		進行了移位的位的數=16/ 移位數 =15		8.650	23.700	8.650	23.700	8.650	23.700	7.300	23.700
	BSET Ⓓ n	n=1		4.550	4.750	4.550	4.750	4.550	4.750	3.000	3.400
		n=15		4.550	4.750	4.550	4.750	4.550	4.750	3.000	3.500
	BRST Ⓓ n	n=1		4.600	4.750	4.600	4.750	4.600	4.750	3.000	3.400
		n=15		4.600	4.750	4.600	4.750	4.600	4.750	3.000	3.400
	TEST	執行時		7.250	13.200	7.250	13.200	7.250	13.200	4.400	6.900
	DTEST	執行時		6.950	12.900	6.950	12.900	6.950	12.900	4.500	7.000
	BKRST Ⓢ n	n=1		7.350	11.600	7.350	11.600	7.350	11.600	4.300	5.200
		n=96		10.100	22.600	10.100	22.600	10.100	22.600	6.500	13.200
	SER ⓢ1 ⓢ2 Ⓓ n	n=1	全部一致	6.650	6.800	6.650	6.800	6.650	6.800	5.000	5.300
			全部不一致	6.650	6.800	6.650	6.800	6.650	6.800	5.000	5.300
		n=96	全部一致	34.000	42.300	34.000	42.300	34.000	42.300	32.300	35.900
			全部不一致	34.000	42.300	34.000	42.300	34.000	42.300	32.400	35.900
	DSER ⓢ1 ⓢ2 Ⓓ n	n=1	全部一致	8.000	16.300	8.000	16.300	8.000	16.300	6.800	10.200
			全部不一致	8.000	16.300	8.000	16.300	8.000	16.300	6.800	10.200
		n=96	全部一致	54.100	62.600	54.100	62.600	54.100	62.600	52.800	56.300
			全部不一致	54.100	62.600	54.100	62.600	54.100	62.600	52.800	56.300
	DSUM Ⓢ Ⓓ	Ⓢ=0		4.100	4.200	4.100	4.200	4.100	4.200	3.700	4.100
		Ⓢ=FFFFFFFF$_H$		4.100	4.200	4.100	4.200	4.100	4.200	3.800	4.100
	DECO Ⓢ Ⓓ n	n=2		8.850	23.000	8.850	23.000	8.850	23.000	6.000	16.400
		n=8		13.600	36.600	13.600	36.600	13.600	36.600	8.100	15.200
	ENCO Ⓢ Ⓓ n	n=2	M1=ON	7.650	11.900	7.650	11.900	7.650	11.900	5.300	6.300
			M4=ON	7.500	11.700	7.500	11.700	7.500	11.700	5.200	6.200
		n=8	M1=ON	14.600	27.800	14.600	27.800	14.600	27.800	10.400	17.900
			M256=ON	10.600	23.700	10.600	23.700	10.600	23.700	5.700	13.300
	DIS Ⓢ Ⓓ n	n=1		6.500	14.800	6.500	14.800	6.500	14.800	5.000	10.900
		n=4		6.900	15.200	6.900	15.200	6.900	15.200	5.400	11.300
	UNI Ⓢ Ⓓ n	n=1		6.800	15.100	6.800	15.100	6.800	15.100	5.500	8.900
		n=4		7.500	15.900	7.500	15.900	7.500	15.900	6.200	9.600
	NDIS	執行時		4.750	18.700	4.750	18.700	4.750	18.700	11.000	16.300
	NUNI	執行時		4.750	18.700	4.750	18.700	4.750	18.700	10.600	16.000

分類	指令	條件（軟元件）	處理時間（μs）							
			Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	WTOB Ⓢ Ⓓ n	n=1	6.600	14.900	6.600	14.900	6.600	14.900	5.000	6.500
		n=96	37.700	46.100	37.700	46.100	37.700	46.100	36.000	38.400
	BTOW Ⓢ Ⓓ n	n=1	7.350	15.600	7.350	15.600	7.350	15.600	5.100	6.100
		n=96	32.100	40.500	32.100	40.500	32.100	40.500	29.900	32.000
	MAX Ⓢ Ⓓ n	n=1	8.250	24.900	8.250	24.900	8.250	24.900	4.300	6.900
		n=96	34.200	51.600	34.200	51.600	34.200	51.600	32.000	34.300
	MIN Ⓢ Ⓓ n	n=1	8.250	24.800	8.250	24.800	8.250	24.800	4.400	6.800
		n=96	34.200	51.600	34.200	51.600	34.200	51.600	30.300	34.800
	DMAX Ⓢ Ⓓ n	n=1	6.800	34.900	6.800	34.900	6.800	34.900	4.800	14.200
		n=96	60.300	89.200	60.300	89.200	60.300	89.200	56.400	68.000
	DMIN Ⓢ Ⓓ n	n=1	7.600	35.700	7.600	35.700	7.600	35.700	4.800	9.300
		n=96	59.400	90.000	59.400	90.000	59.400	90.000	55.400	62.800
	SORT Ⓢ1 n Ⓢ2 Ⓓ1 Ⓓ2	n=1、Ⓢ2 =1	9,400	28,900	9,400	28,900	9,400	28,900	6.200	24,900
		n=96、Ⓢ2 =16	31,500	74,000	31,500	74,000	31,500	74,000	27,500	70,100
	DSORT Ⓢ1 n Ⓢ2 Ⓓ1 Ⓓ2	n=1、Ⓢ2 =1	9,400	29,000	9,400	29,000	9,400	29,000	6.200	25,900
		n=96、Ⓢ2 =16	37,800	81,000	37,800	81,000	37,800	81,000	33,100	78,900
	WSUM Ⓢ Ⓓ n	n=1	6.700	15.000	6.700	15.000	6.700	15.000	4.800	6.200
		n=96	28.900	37.100	28.900	37.100	28.900	37.100	26.900	28.700
	DWSUM Ⓢ Ⓓ n	n=1	8.600	26.800	8.600	26.800	8.600	26.800	5.500	7.000
		n=96	56.200	74.700	56.200	74.700	56.200	74.700	53.000	56.300
	MEAN Ⓢ Ⓓ n	n=1	5.850	19.800	5.850	19.800	5.850	19.800	4.300	17.300
		n=96	17.300	38.200	17.300	38.200	17.300	38.200	16.000	35.500
	DMEAN Ⓢ Ⓓ n	n=1	6.900	23.300	6.900	23.300	6.900	23.300	5.750	21.900
		n=96	29.400	49.900	29.400	49.900	29.400	49.900	29.200	48.600
	NEXT	---	1.000	1.100	1.000	1.100	1.000	1.100	0.980	1.400
	BREAK	---	4.700	25.000	4.700	25.000	4.700	25.000	21.300	17.900
	RET	返回至本程式	4.100	19.500	4.100	19.500	4.100	19.500	2.000	3.000
		返回至其他程式	4.700	16.700	4.700	16.700	4.700	16.700	2.300	4.900
	FCALL Pn	文件內指標	5.400	5.400	5.400	5.400	5.400	5.400	3.300	5.300
		公共指標	7.600	30.500	7.600	30.500	7.600	30.500	4.900	6.600
	FCALL Pn Ⓢ1 ～ Ⓢ5	---	50.400	62.700	50.400	62.700	50.400	62.700	19.800	23.700
	ECALL * Pn *: 程式名	---	105.000	214.000	105.000	214.000	105.000	214.000	75.700	134.000
	ECALL * Pn Ⓢ1 ～ Ⓢ5 *: 程式名	---	164.000	271.000	164.000	271.000	164.000	271.000	109.000	173.000
	EFCALL * Pn *: 程式名	---	105.000	214.000	105.000	214.000	105.000	214.000	76.200	134.000
	EFCALL * Pn Ⓢ1 ～ Ⓢ5 *: 程式名	---	164.000	271.000	164.000	271.000	164.000	271.000	90.500	170.000
	XCALL	---	5.100	6.700	5.100	6.700	5.100	6.700	3.800	6.400
	FIFW	資料數 = 0	6.100	14.200	6.100	14.200	6.100	14.200	3.700	10.100
		資料數 = 96	6.100	14.200	6.100	14.200	6.100	14.200	3.800	5.200
	FIFR	資料數 = 1	7.500	15.600	7.500	15.600	7.500	15.600	4.400	5.800
		資料數 = 96	37.000	45.000	37.000	45.000	37.000	45.000	33.500	35.200
	FPOP	資料數 = 1	7.600	15.600	7.600	15.600	7.600	15.600	4.400	10.800
		資料數 = 96	7.600	15.600	7.600	15.600	7.600	15.600	4.400	10.800
	FINS	資料數 = 0	6.900	15.000	6.900	15.000	6.900	15.000	5.000	10.700
		資料數 = 96	36.600	44.700	36.600	44.700	36.600	44.700	4.400	10.900
	FDEL	資料數 = 1	8.000	16.100	8.000	16.100	8.000	16.100	4.900	11.300
		資料數 = 96	37.300	45.500	37.300	45.500	37.300	45.500	34.200	35.900
	FROM n1 n2 Ⓓ n3	n3=1	17.400	74.700	17.400	74.700	17.400	74.700	12.100	71.300
		n3=1000	406.000	498.500	406.000	498.500	406.000	498.500	402.600	495.100
	DFRO n1 n2 Ⓓ n3	n3=1	19.600	85.600	19.600	85.600	19.600	85.600	14.600	81.800
		n3=500	406.000	498.500	406.000	498.500	406.000	498.500	402.600	495.100

分類	指令	條件（軟元件）	處理時間（μs）							
			Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	TO n1 n2 Ⓢ n3	n3=1	16.400	69.600	16.400	69.600	16.400	69.600	11.700	63.400
		n3=1000	381.300	471.200	381.300	471.200	381.300	471.200	375.900	464.300
	DTO n1 n2 Ⓢ n3	n3=1	18.600	85.100	18.600	85.100	18.600	85.100	14.200	78.500
		n3=500	381.300	471.200	381.300	471.200	381.300	471.200	375.900	464.300
	CCOM COM	僅選擇 I/O 刷新時	18.100	89.100	18.100	89.100	18.100	89.100	12.800	79.000
		僅選擇 CC-Link 刷新時（主站側）	33.300	132.000	33.300	132.000	33.300	132.000	24.900	119.000
		僅選擇 CC-Link 刷新時（本地站側）	33.300	132.000	33.300	132.000	33.300	132.000	24.900	119.000
		・僅選擇 MELSECNET/H 刷新時（管理站側） ・僅選擇 CC-Link IE 控制網路刷新時（管理站側）	78.600	231.000	78.600	231.000	78.600	231.000	54.000	212.000
		・僅選擇 MELSECNET/H 刷新時（普通站側） ・僅選擇 CC-Link IE 控制網路刷新時（普通站側）	78.600	231.000	78.600	231.000	78.600	231.000	54.000	212.000
		僅選擇 CC-Link IE 現場網路刷新時（主站側）	32.000	127.000	32.000	127.000	32.000	127.000	22.000	118.000
		僅選擇 CC-Link IE 現場網路刷新時（本地站側）	32.000	127.000	32.000	127.000	32.000	127.000	22.000	118.000
		僅選擇智慧自動刷新時	18.100	89.000	18.100	89.000	18.100	89.000	12.800	79.000
		僅選擇組外輸入輸出時（僅輸入）	15.700	71.600	15.700	71.600	15.700	71.600	8.600	76.500
		僅選擇組外輸入輸出時（僅輸出）	40.200	152.000	40.200	152.000	40.200	152.000	26.300	135.000
		僅選擇組外輸入輸出時（輸入及輸出均選擇）	45.800	153.000	45.800	153.000	45.800	153.000	26.100	135.000
		僅選擇多 CPU 高速通信區刷新時	---	---	---	---	---	---	---	---
		僅選擇與週邊設備通信時	18.200	89.000	18.200	89.000	18.200	89.000	7.250	54.300
	LEDR	無顯示→無顯示	1.500	7.100	1.500	7.100	1.500	7.100	5.100	5.100
		LED 指令執行→無顯示	38.900	109.000	38.900	109.000	38.900	109.000	35.700	89.200
	BINDA Ⓢ Ⓓ	Ⓢ=1	5.600	13.900	5.600	13.900	5.600	13.900	4.900	6.500
		Ⓢ=-32768	7.800	16.200	7.800	16.200	7.800	16.200	7.200	8.700
	DBINDA Ⓢ Ⓓ	Ⓢ=1	6.200	14.500	6.200	14.500	6.200	14.500	5.700	7.100
		Ⓢ=-2147483648	11.000	19.200	11.000	19.200	11.000	19.200	10.400	12.200
	BINHA Ⓢ Ⓓ	Ⓢ=1	5.050	13.400	5.050	13.400	5.050	13.400	4.400	5.900
		Ⓢ=$FFFF_H$	5.050	13.400	5.050	13.400	5.050	13.400	4.400	5.800

分類	指令	條件（軟元件）	處理時間（μs）							
			Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	DBINHA Ⓢ Ⓓ	Ⓢ=1	5.600	13.900	5.600	13.900	5.600	13.900	5.200	6.700
		Ⓢ=FFFFFFFF$_H$	5.600	13.900	5.600	13.900	5.600	13.900	5.100	6.500
	BCDDA Ⓢ Ⓓ	Ⓢ=1	4.850	13.200	4.850	13.200	4.850	13.200	4.300	5.800
		Ⓢ=9999	5.300	13.600	5.300	13.600	5.300	13.600	4.700	6.100
	DBCDDA Ⓢ Ⓓ	Ⓢ=1	5.300	13.600	5.300	13.600	5.300	13.600	4.800	6.300
		Ⓢ=99999999	6.200	14.500	6.200	14.500	6.200	14.500	5.600	7.100
	DABIN Ⓢ Ⓓ	Ⓢ=1	7.000	18.500	7.000	18.500	7.000	18.500	6.500	9.000
		Ⓢ=-32768	6.950	18.500	6.950	18.500	6.950	18.500	6.300	8.900
	DDABIN Ⓢ Ⓓ	Ⓢ=1	9.450	21.000	9.450	21.000	9.450	21.000	9.400	12.000
		Ⓢ=-2147483648	9.450	21.000	9.450	21.000	9.450	21.000	9.100	11.600
	HABIN Ⓢ Ⓓ	Ⓢ=1	5.650	17.100	5.650	17.100	5.650	17.100	4.900	7.500
		Ⓢ=FFFF$_H$	5.750	17.300	5.750	17.300	5.750	17.300	5.100	8.100
	DHABIN Ⓢ Ⓓ	Ⓢ=1	6.800	18.200	6.800	18.200	6.800	18.200	6.000	8.500
		Ⓢ=FFFFFFFF$_H$	7.100	18.600	7.100	18.600	7.100	18.600	6.300	8.900
	DABCD Ⓢ Ⓓ	Ⓢ=1	5.650	17.200	5.650	17.200	5.650	17.200	5.000	7.500
		Ⓢ=9999	5.700	17.200	5.700	17.200	5.700	17.200	5.000	7.500
	DDABCD Ⓢ Ⓓ	Ⓢ=1	6.850	18.300	6.850	18.300	6.850	18.300	6.200	8.800
		Ⓢ=99999999	6.850	18.300	6.850	18.300	6.850	18.300	6.200	8.800
	COMRD	---	185.000	188.000	185.000	188.000	185.000	188.000	97.300	97.400
	LEN	1 字元	4.700	16.200	4.700	16.200	4.700	16.200	4.100	6.600
		96 字元	20.600	32.900	20.600	32.900	20.600	32.900	19.800	22.400
	STR	---	9.800	36.500	9.800	36.500	9.800	36.500	6.900	14.400
	DSTR	---	12.100	40.400	12.100	40.400	12.100	40.400	10.200	20.800
	VAL	---	12.200	40.900	12.200	40.900	12.200	40.900	9.800	23.900
	DVAL	---	19.400	45.600	19.400	45.600	19.400	45.600	14.000	33.100
	ESTR	---	29.700	87.800	29.700	87.800	29.700	87.800	22.100	52.400
	EVAL	小數點形式全 2 位數	23.900	70.400	23.900	70.400	23.900	70.400	23.300	36.500
		指數形式全 6 位數指定	23.700	70.300	23.700	70.300	23.700	70.300	23.300	36.400
	ASC Ⓢ Ⓓ n	n=1	10.200	41.800	10.200	41.800	10.200	41.800	5.600	19.700
		n=96	31.900	66.600	31.900	66.600	31.900	66.600	30.200	44.700
	HEX Ⓢ Ⓓ n	n=1	8.600	43.400	8.600	43.400	8.600	43.400	7.500	23.100
		n=96	77.100	115.000	77.100	115.000	77.100	115.000	37.500	53.300
	RIGHT Ⓢ Ⓓ n	n=1	10.900	29.600	10.900	29.600	10.900	29.600	7.600	11.400
		n=96	41.400	60.300	41.400	60.300	41.400	60.300	36.300	46.000
	LEFT Ⓢ Ⓓ n	n=1	10.600	29.300	10.600	29.300	10.600	29.300	6.500	16.100
		n=96	41.300	60.200	41.300	60.200	41.300	60.200	36.200	46.200
	MIDR	---	11.700	30.600	11.700	30.600	11.700	30.600	9.500	19.100
	MIDW	---	12.400	24.000	12.400	24.000	12.400	24.000	10.300	18.200
	INSTR	無一致	22.000	38.200	22.000	38.200	22.000	38.200	19.300	29.000
		有一致 起始	13.300	29.600	13.300	29.600	13.300	29.600	10.300	20.000
		有一致 最終	21.900	38.100	21.900	38.100	21.900	38.100	51.100	60.800
	EMOD	---	11.600	24.000	11.600	24.000	11.600	24.000	10.300	15.300
	EREXP	---	19.700	28.000	19.700	28.000	19.700	28.000	19.300	22.300
	STRINS Ⓢ Ⓓ n	Ⓢ=128/Ⓓ=40/n=1	47.000	102.000	47.000	102.000	47.000	102.000	44.300	96.700
		Ⓢ=128/Ⓓ=40/n=48	70.100	134.000	70.100	134.000	70.100	134.000	58.800	112.000
	STRDEL Ⓢ Ⓓ n	Ⓢ=128/Ⓓ=40/n=1	46.400	93.600	46.400	93.600	46.400	93.600	39.000	78.100
		Ⓢ=128/Ⓓ=40/n=48	44.500	70.600	44.500	70.600	44.500	70.600	36.000	69.200
	SIN	單精度	6.400	13.900	6.400	13.900	6.400	13.900	4.500	9.900
	COS	單精度	6.100	13.500	6.100	13.500	6.100	13.500	4.300	8.200
	TAN	單精度	8.300	15.000	8.300	15.000	8.300	15.000	5.100	7.200
	ASIN	單精度	7.300	15.600	7.300	15.600	7.300	15.600	6.100	13.700
	ACOS	單精度	8.100	16.500	8.100	16.500	8.100	16.500	6.800	11.100
	ATAN	單精度	5.350	12.000	5.350	12.000	5.350	12.000	4.000	6.900

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	SIND	雙精度		13.400	51.300	13.400	51.300	13.400	51.300	9.600	26.000
	COSD	雙精度		14.700	51.700	14.700	51.700	14.700	51.700	10.000	26.900
	TAND	雙精度		17.400	54.400	17.400	54.400	17.400	54.400	11.400	25.300
	ASIND	雙精度		22.600	60.300	22.600	60.300	22.600	60.300	12.100	30.800
	ACOSD	雙精度		19.700	60.000	19.700	60.000	19.700	60.000	11.700	28.000
	ATAND	雙精度		15.000	51.800	15.000	51.800	15.000	51.800	9.700	22.000
	RAD	單精度		3.200	10.300	3.200	10.300	3.200	10.300	2.500	4.800
	RADD	雙精度		5.200	43.100	5.200	43.100	5.200	43.100	4.100	16.400
	DEG	單精度		3.200	11.500	3.200	11.500	3.200	11.500	2.500	4.700
	DEGD	雙精度		5.150	43.800	5.150	43.800	5.150	43.800	5.000	18.100
	SQR	單精度		3.900	12.300	3.900	12.300	3.900	12.300	3.500	9.300
	SQRD	雙精度		7.000	45.700	7.000	45.700	7.000	45.700	5.700	25.400
	EXP Ⓢ Ⓓ	單精度	Ⓢ=-10	6.350	13.800	6.350	13.800	6.350	13.800	4.000	13.000
			Ⓢ=1	6.350	13.800	6.350	13.800	6.350	13.800	4.000	13.000
	EXPD Ⓢ Ⓓ	雙精度	Ⓢ=-10	15.800	52.700	15.800	52.700	15.800	52.700	8.800	27.600
			Ⓢ=1	15.400	52.500	15.400	52.500	15.400	52.500	8.500	27.300
	LOG Ⓢ Ⓓ	單精度	Ⓢ=1	5.800	14.900	5.800	14.900	5.800	14.900	4.100	8.100
			Ⓢ=10	7.450	16.500	7.450	16.500	7.450	16.500	6.200	10.300
	LOGD Ⓢ Ⓓ	雙精度	Ⓢ=1	11.000	48.900	11.000	48.900	11.000	48.900	9.500	28.300
			Ⓢ=10	12.600	51.300	12.600	51.300	12.600	51.300	11.100	29.900
	SCL Ⓢ1 Ⓢ2 Ⓓ	SM750=ON	點 No.1<Ⓢ1<點 No.2	14.900	50.100	14.900	50.100	14.900	50.100	14.700	48.000
			點 No.9<Ⓢ1<點 No.10	15.800	50.900	15.800	50.900	15.800	50.900	19.600	50.400
		SM750=OFF	點 No.1<Ⓢ1<點 No.2	13.900	53.100	13.900	53.100	13.900	53.100	13.700	51.000
			點 No.9<Ⓢ1<點 No.10	16.600	56.600	16.600	56.600	16.600	56.600	20.400	56.200
	DSCL Ⓢ1 Ⓢ2 Ⓓ	SM750=ON	點 No.1<Ⓢ1<點 No.2	13.400	52.400	13.400	52.400	13.400	52.400	12.800	50.300
			點 No.9<Ⓢ1<點 No.10	14.200	54.100	14.200	54.100	14.200	54.100	17.300	53.500
		SM750=OFF	點 No.1<Ⓢ1<點 No.2	12.300	53.200	12.300	53.200	12.300	53.200	11.500	51.100
			點 No.9<Ⓢ1<點 No.10	15.000	57.600	15.000	57.600	15.000	57.600	18.100	57.100

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	SCL2 ⓢ1 ⓢ2 ⓓ	SM750=ON	點 No.1 <ⓢ1< 點 No.2	14.200	53.300	14.200	53.300	14.200	53.300	13.200	51.200
			點 No.9 <ⓢ1< 點 No.10	14.900	55.000	14.900	55.000	14.900	55.000	18.000	54.500
		SM750=OFF	點 No.1 <ⓢ1< 點 No.2	15.000	53.500	15.000	53.500	15.000	53.500	14.000	51.300
			點 No.9 <ⓢ1< 點 No.10	16.300	56.400	16.300	56.400	16.300	56.400	19.300	55.800
	DSCL2 ⓢ1 ⓢ2 ⓓ	SM750=ON	點 No.1 <ⓢ1< 點 No.2	13.400	52.700	13.400	52.700	13.400	52.700	13.100	50.500
			點 No.9 <ⓢ1< 點 No.10	14.200	54.300	14.200	54.300	14.200	54.300	18.100	53.700
		SM750=OFF	點 No.1 <ⓢ1< 點 No.2	12.300	53.200	12.300	53.200	12.300	53.200	12.100	51.000
			點 No.9 <ⓢ1< 點 No.10	15.000	57.600	15.000	57.600	15.000	57.600	18.900	57.100
	RSET	標準 RAM		6.800	26.900	6.800	26.900	6.800	26.900	3.000	16.400
		SRAM		---	---	---	---	---	---	3.000	16.400
	QDRSET	SRAM →標準 RAM		---	---	---	---	---	---	230.000	327.000
		標準 RAM → SRAM		---	---	---	---	---	---	997.000	1066.000
	QCDSET	SRAM →標準 ROM		---	---	---	---	---	---	525.000	690.000
		標準 ROM → SRAM		---	---	---	---	---	---	490.000	655.000
	DATERD	---		5.600	27.800	5.600	27.800	5.600	27.800	5.100	14.700
	DATEWR	---		7.800	42.100	7.800	42.100	7.800	42.100	7.100	23.000
	DATE +	無進位		14.200	41.200	14.200	41.200	14.200	41.200	6.500	13.100
		有進位		14.200	41.200	14.200	41.200	14.200	41.200	5.700	21.200
	DATE-	無進位		15.100	41.200	15.100	41.200	15.100	41.200	6.500	11.500
		有進位		15.100	41.200	15.100	41.200	15.100	41.200	5.700	17.200
	SECOND	---		5.800	20.500	5.800	20.500	5.800	20.500	2.600	5.900
	HOUR	---		6.200	22.500	6.200	22.500	6.200	22.500	3.000	5.300
	RND	---		1.950	5.450	1.950	5.450	1.950	5.450	1.200	2.300
	SRND	---		2.750	4.550	2.750	4.550	2.750	4.550	1.400	2.400
	BSQR ⓢ ⓓ	ⓢ=0		2.500	6.800	2.500	6.800	2.500	6.800	1.800	3.300
		ⓢ=9999		6.400	15.500	6.400	15.500	6.400	15.500	5.100	8.800
	BDSQR ⓢ ⓓ	ⓢ=0		2.600	6.050	2.600	6.050	2.600	6.050	1.900	3.700
		ⓢ=99999999		8.450	17.600	8.450	17.600	8.450	17.600	7.500	10.900
	BSIN	---		11.500	32.800	11.500	32.800	11.500	32.800	8.700	20.200
	BCOS	---		10.400	32.500	10.400	32.500	10.400	32.500	7.800	14.400
	BTAN	---		12.100	33.700	12.100	33.700	12.100	33.700	9.000	17.000
	BASIN	---		13.300	32.800	13.300	32.800	13.300	32.800	12.200	15.100
	BACOS	---		13.400	33.700	13.400	33.700	13.400	33.700	13.100	14.900
	BATAN	---		12.600	31.400	12.600	31.400	12.600	31.400	11.400	15.700
	POW ⓢ1 ⓢ2 ⓓ	單精度	ⓢ1=12.3E+5 ⓢ2=3.45E+0	12.200	22.100	12.200	22.100	12.200	22.100	8.950	19.500
	POWD ⓢ1 ⓢ2 ⓓ	雙精度	ⓢ1=12.3E+5 ⓢ2=3.45E+0	27.300	61.000	27.300	61.000	27.300	61.000	19.400	55.200

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	LOG10	單精度		8.200	16.500	8.200	16.500	8.200	16.500	5.950	14.800
	LOG10D	雙精度		15.100	48.000	15.100	48.000	15.100	48.000	12.400	46.500
	LIMIT	---		5.350	5.500	5.350	5.500	5.350	5.500	5.200	5.400
	DLIMIT	---		6.000	6.150	6.000	6.150	6.000	6.150	5.700	5.900
	BAND	---		5.450	12.400	5.450	12.400	5.450	12.400	5.400	6.300
	DBAND	---		6.050	11.900	6.050	11.900	6.050	11.900	5.800	6.900
	ZONE	---		6.250	10.700	6.250	10.700	6.250	10.700	5.200	11.100
	DZONE	---		6.000	11.900	6.000	11.900	6.000	11.900	5.700	10.800
	LDDT<=	與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ANDDT<=	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.200	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.200	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ORDT<=	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.300
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	LDDT<	與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ANDDT<	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.200	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.200	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ORDT<	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.300
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	LDDT>=	與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ANDDT>=	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.200	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.200	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ORDT>=	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.300
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	LDTM=	與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
	LDDT=	與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	8.200	25.500
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
应用指令	ANDDT=	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.200	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.200	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ORDT=	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.300
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	LDDT<>	與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ANDDT<>	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.200	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.200	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ORDT<>	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.300
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	LDDT>	與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.400	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ANDDT>	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	8.200	25.500	7.200	23.400
			非導通時	8.200	25.500	8.200	25.500	8.200	25.500	7.200	23.400
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.200
	ORDT>	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	8.200	25.500	7.400	23.300
			非導通時	8.200	25.500	8.200	25.500	8.200	25.500	7.400	23.300
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	LDTM<=	與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
	ANDTM<=	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
	ORTM<=	未執行時			0.480		0.320		0.240		0.160
		與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.200
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.200
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	LDTM<	與指定的日期比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
		與當前的日期比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
应用指令	ANDTM<	未執行時			0.480		0.320		0.240		0.160
		與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
	ORTM<	未執行時			0.480		0.320		0.240		0.160
		與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.200
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.200
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	LDTM>=	與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
	ANDTM>=	未執行時			0.480		0.320		0.240		0.160
		與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
	ORTM>=	未執行時			0.480		0.320		0.240		0.160
		與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
	ANDTM=	未執行時			0.480		0.320		0.240		0.160
		與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
	ORTM=	未執行時			0.480		0.320		0.240		0.160
		與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.200
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.200
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	LDTM<>	與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
	ANDTM<>	未執行時			0.480		0.320		0.240		0.160
		與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
	ORTM<>	未執行時			0.480		0.320		0.240		0.160
		與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.200
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.200
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	LDTM>	與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.300
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.100
	ANDTM>	未執行時			0.480		0.320		0.240		0.160
		與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.000	23.000
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.600	21.900

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	ORTM>	未執行時			0.480		0.320		0.240		0.160
		與指定的時間比較	導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.200
			非導通時	8.200	25.500	8.200	25.500	6.500	25.500	7.300	23.200
		與當前的時間比較	導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
			非導通時	6.500	23.100	6.500	23.100	6.500	23.100	5.900	22.000
	S.DATERD	---		9.250	51.000	9.250	51.000	9.250	51.000	7.500	23.400
	S.DATE+	無進位		16.800	75.400	16.800	75.400	16.800	75.400	9.100	23.400
		有進位		16.800	75.400	16.800	75.400	16.800	75.400	8.900	22.200
	S.DATE-	無進位		17.600	75.300	17.600	75.300	17.600	75.300	9.000	22.200
		有進位		16.900	75.300	16.900	75.300	16.900	75.300	9.800	22.100
	PSTOP	---		82.200	199.000	82.200	199.000	82.200	199.000	61.400	84.500
	POFF	---		82.600	198.000	82.600	198.000	82.600	198.000	121.000	246.000
	PSCAN	---		83.600	200.000	83.600	200.000	83.600	200.000	126.000	232.000
	WDT	---		2.900	12.000	2.900	12.000	2.900	12.000	1.300	3.000
	DUTY	---		7.700	27.500	7.700	27.500	7.700	27.500	4.900	24.300
	TIMCHK	---		5.350	24.500	5.350	24.500	5.350	24.500	7.400	23.300
	ZRRDB	標準 RAM 的文件寄存器		4.100	4.200	4.100	4.200	4.100	4.200	2.400	2.600
		SRAM 卡的文件寄存器		---	---	---	---	---	---	2.500	2.800
	ZRWRB	標準 RAM 的文件寄存器		5.400	5.500	5.400	5.500	5.400	5.500	3.100	3.300
		SRAM 卡的文件寄存器		---	---	---	---	---	---	3.300	3.600
	ADRSET	---		2.400	6.650	2.400	6.650	2.400	6.650	4.200	4.900
	ZPUSH	---		9.200	20.500	9.200	20.500	9.200	20.500	6.900	14.000
	ZPOP	---		9.000	15.500	9.000	15.500	9.000	15.500	7.500	12.500
	S.ZCOM	安裝 CC-Link 模組時（主站側）		29.400	91.700	29.400	91.700	29.400	91.700	20.600	55.000
		安裝 CC-Link 模組時（本地站側）		29.500	91.600	29.500	91.600	29.500	91.600	20.600	66.100
		・僅選擇 MELSECNET/H 刷新時（管理站側） ・僅選擇 CCLink IE 控制網路刷新時（管理站）		79.900	214.000	79.900	214.000	79.900	214.000	102.000	180.000
		・僅選擇 MELSECNET/H 刷新時（普通站側） ・僅選擇 MELSECNET/H 刷新時（普通站側）（普通站）		79.900	214.000	79.900	214.000	79.900	214.000	55.600	168.100
		僅選擇 CCLink IE 現場網路刷新時（主站）		60.000	167.000	60.000	167.000	60.000	167.000	51.000	154.000
		僅選擇 CCLink IE 現場網路刷新時（本地站）		60.000	167.000	60.000	167.000	60.000	167.000	51.000	154.000
	S.RTREAD	---		12.600	65.000	12.600	65.000	12.600	65.000	8.700	60.500
	S.RTWRITE	---		13.300	67.100	13.300	67.100	13.300	67.100	9.300	65.000
	UNIRD n1 Ⓓ n2	n2=1		6.000	33.100	6.000	33.100	6.000	33.100	4.000	29.100
		n2=16		16.500	43.600	16.500	43.600	16.500	43.600	12.500	37.600
	TYPERD			48.50	141.30	43.50	139.90	43.40	139.80	32.40	134.20
	TRACE	開始		174.000	174.000	174.000	174.000	174.000	174.000	96.600	103.000

分類	指令	條件（軟元件）		處理時間（μs）							
				Q00UJCPU		Q00UCPU		Q01UCPU		Q02UCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	TRACER	---		5.100	15.500	5.100	15.500	5.100	15.500	3.800	13.600
	RBMOV Ⓢ Ⓓ n	使用標準 RAM	1 點	---	---	12.200	34.900	12.200	34.900	9.400	31.300
			1000 點	---	---	121.500	145.100	121.500	145.100	118.500	141.300
		使用 SRAM 卡	1 點	---	---	---	---	---	---	9.400	31.400
			1000 點	---	---	---	---	---	---	178.500	201.300
	SP.FWRITE	---		---	---	---	---	---	---	87.000	144.000
	SP.FREAD	---		---	---	---	---	---	---	127.000	140.000
	SP.DEVST	---		125.000	125.000	125.000	125.000	125.000	125.000	8.100	98.000
	S.DEVLD	---		18.300	36.700	18.300	36.700	18.300	36.700	13.000	43.000
多 CPU 專用指令	S.TO n1 n2 n3 n4 Ⓓ	至本機 CPU 共用記憶體的寫入	n4=1	64.600	78.100	64.600	78.100	64.600	78.100	64.600	78.100
			n4=320	115.000	126.000	115.000	126.000	115.000	126.000	154.000	126.000
	TO n1 n2 Ⓢ n3	至本機 CPU 共用記憶體的寫入	n3=1	12.700	62.200	12.700	62.200	12.700	62.200	8.300	58.200
			n3=320	63.500	112.300	63.500	112.300	63.500	112.300	56.200	107.800
	DTO n1 n2 Ⓢ n3	至本機 CPU 共用記憶體的寫入	n3=1	13.500	62.300	13.500	62.300	13.500	62.300	8.600	58.300
			n3=320	112.900	160.800	112.900	160.800	112.900	160.800	106.800	157.300
	FROM n1 n2 Ⓓ n3	本機 CPU 共用記憶體讀取	n3=1	12.100	58.700	12.100	58.700	12.100	58.700	8.400	52.600
			n3=320	56.000	101.700	56.000	101.700	56.000	101.700	51.700	96.600
		其他機號 CPU 共用記憶體讀取	n3=1	24.400	82.900	24.400	82.900	24.400	82.900	16.600	37.000
			n3=320	152.000	243.000	152.000	243.000	152.000	243.000	153.000	185.000
			n3=1000	418.000	518.000	418.000	518.000	418.000	518.000	432.000	485.000
	DFRO n1 n2 Ⓓ n3	本機 CPU 共用記憶體讀取	n3=1	12.100	58.700	12.100	58.700	12.100	58.700	8.800	53.400
			n3=320	97.400	143.700	97.400	143.700	97.400	143.700	94.900	139.600
		其他機號 CPU 共用記憶體讀取	n3=1	24.800	94.200	24.800	94.200	24.800	94.200	16.600	47.300
			n3=320	276.000	367.000	276.000	367.000	276.000	367.000	278.000	339.000
			n3=1000	799.000	892.000	799.000	892.000	799.000	892.000	841.000	892.000

備註

對於表中未記述的上升沿執行指令（□ P）的指令，其處理時間與 ON 時執行的指令的處理時間相同。

例 WORDP 指令、TOP 指令等。

(b) 使用 Q03UD(E)CPU、Q04UD(E)HCOU、Q06UD(E)HCPU、Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU、Q50UDEHCPU、Q100UDEHCPU 時

分類	指令	條件（軟元件）			處理時間（μs）							
					Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
					最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
順控程式指令	ANB ORB MPS MRD MPP	---				0.020		0.0095		0.0095		0.0095
	INV	未執行時				0.020		0.0095		0.0095		0.0095
		執行時										
	MEP MEF	未執行時				0.020		0.0095		0.0095		0.0095
		執行時										
	EGP EGF	未執行時				0.020		0.0095		0.0095		0.0095
		執行時										
	PLS	---			1.300	1.600	0.890	1.100	0.890	1.100	0.890	1.100
	FF	未執行時				0.040		0.0185		0.0185		0.0185
		執行時			1.200	1.500	0.790	0.910	0.790	0.910	0.790	0.910
	DELTA	未執行時				0.040		0.0185		0.0185		0.0185
		執行時			2.800	3.600	2.400	3.200	2.400	3.200	2.400	3.200
	SFT	未執行時				0.040		0.0185		0.0185		0.0185
		執行時			1.600	3.300	1.100	2.700	1.100	2.700	1.100	2.700
	MC	---				0.040		0.0185		0.0185		0.0185
	MCR	---				0.040		0.0185		0.0185		0.0185
	FEND	進行出錯檢查			108.000	130.000	75.800	89.300	75.800	89.300	75.800	89.300
	END	不進行出錯檢查			107.000	124.000	75.800	89.800	75.800	89.800	75.800	89.800
	NOP NOPLF PAGE	---				0.020		0.0095		0.0095		0.0095
基本指令	LDE=	單精度	導通時		3.700	4.700	3.300	4.300		0.0285		0.0285
			非導通時		3.800	5.000	3.400	4.500				
	ANDE=	單精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.300	5.800	3.000	5.100				
				非導通時	3.500	5.600	3.000	5.200				
	ORE=	單精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.600	4.500	3.200	4.200				
				非導通時	3.500	4.800	3.200	4.300				
	LDE< >	單精度	導通時		4.000	4.700	3.600	4.200		0.0285		0.0285
			非導通時		3.900	4.500	3.500	4.000				
	ANDE< >	單精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.300	5.100	3.000	4.800				
				非導通時	3.500	5.000	3.100	4.600				
	ORE< >	單精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.600	6.000	3.300	5.500				
				非導通時	3.500	5.800	3.100	5.300				
	LDE>	單精度	導通時		3.800	5.000	3.300	4.600		0.0285		0.0285
			非導通時		3.700	4.900	3.300	4.400				
	ANDE>	單精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.500	4.700	3.100	4.200				
				非導通時	3.600	4.500	3.100	4.000				
	ORE>	單精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.600	5.100	3.300	4.600				
				非導通時	3.500	4.800	3.200	4.500				

分類	指令	條件（軟元件）			處理時間（μs）							
					Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
					最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	LDE<=	單精度	導通時		3.800	5.600	3.400	5.200				
			非導通時		3.800	5.600	3.400	5.100		0.0285		0.0285
	ANDE<=	單精度	未執行時			0.060		0.0285				
			執行時	導通時	3.200	4.600	2.800	4.200		0.0285		0.0285
				非導通時	3.500	5.000	3.100	4.500				
	ORE<=	單精度	未執行時			0.060		0.0285				
			執行時	導通時	3.700	5.800	3.400	5.400		0.0285		0.0285
				非導通時	3.800	5.700	3.300	5.300				
	LDE<	單精度	導通時		4.000	5.400	3.500	4.900				
			非導通時		4.000	5.200	3.500	4.900		0.0285		0.0285
	ANDE<	單精度	未執行時			0.060		0.0285				
			執行時	導通時	3.400	4.600	3.000	4.200		0.0285		0.0285
				非導通時	3.500	4.900	3.100	4.400				
	ORE<	單精度	未執行時			0.060		0.0285				
			執行時	導通時	3.600	5.200	3.300	4.900		0.0285		0.0285
				非導通時	3.400	4.900	3.200	4.500				
	LDE>=	單精度	導通時		3.800	6.000	3.300	5.500				
			非導通時		3.800	5.900	3.400	5.400		0.0285		0.0285
	ANDE>=	單精度	未執行時			0.060		0.0285				
			執行時	導通時	3.200	4.800	2.900	4.600		0.0285		0.0285
				非導通時	3.500	5.400	3.100	5.100				
	ORE>=	單精度	未執行時			0.060		0.0285				
			執行時	導通時	3.600	5.200	3.300	4.700		0.0285		0.0285
				非導通時	3.500	5.200	3.200	4.700				
	LDED=	雙精度	導通時		4.100	7.700	3.500	7.200	3.500	7.200	3.500	7.200
			非導通時		4.300	8.100	3.800	7.400	3.800	7.400	3.800	7.400
	ANDED=	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.600	7.600	3.200	7.000	3.200	7.000	3.200	7.000
				非導通時	3.900	7.700	3.400	7.400	3.400	7.400	3.400	7.400
	ORED=	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.800	8.800	3.400	8.300	3.400	8.300	3.400	8.300
				非導通時	4.000	9.300	3.700	8.800	3.700	8.800	3.700	8.800
	LDED< >	雙精度	導通時		4.400	8.200	3.900	7.700	3.900	7.700	3.900	7.700
			非導通時		4.100	7.900	3.500	7.500	3.500	7.500	3.500	7.500
	ANDED< >	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.800	7.600	3.300	7.200	3.300	7.200	3.300	7.200
				非導通時	3.800	7.700	3.400	7.300	3.400	7.300	3.400	7.300
	ORED< >	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	4.100	9.300	3.700	8.900	3.700	8.900	3.700	8.900
				非導通時	3.800	8.900	3.400	8.400	3.400	8.400	3.400	8.400
	LDED>	雙精度	導通時		4.300	8.100	3.800	7.500	3.800	7.500	3.800	7.500
			非導通時		4.100	7.800	3.500	7.200	3.500	7.200	3.500	7.200
	ANDED>	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.800	7.700	3.300	7.300	3.300	7.300	3.300	7.300
				非導通時	4.000	7.900	3.500	7.500	3.500	7.500	3.500	7.500
	ORED>	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	4.100	9.300	3.700	8.800	3.700	8.800	3.700	8.800
				非導通時	4.100	9.300	3.700	8.800	3.700	8.800	3.700	8.800

分類	指令	條件（軟元件）			處理時間（μs）							
					Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
					最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	LDED<=	雙精度	導通時		4.000	8.000	3.500	7.400	3.500	7.400	3.500	7.400
			非導通時		4.100	9.400	3.600	8.800	3.600	8.800	3.600	8.800
	ANDED<=	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.800	7.700	3.300	7.200	3.300	7.200	3.300	7.200
				非導通時	3.900	7.700	3.500	7.400	3.500	7.400	3.500	7.400
	ORED<=	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	4.100	9.600	3.700	9.200	3.700	9.200	3.700	9.200
				非導通時	4.100	9.600	3.700	9.200	3.700	9.200	3.700	9.200
	LDED<	雙精度	導通時		4.300	8.300	3.800	7.600	3.800	7.600	3.800	7.600
			非導通時		3.700	7.900	3.500	7.400	3.500	7.400	3.500	7.400
	ANDED<	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.800	7.800	3.300	7.300	3.300	7.300	3.300	7.300
				非導通時	3.900	7.900	3.400	3.900	3.400	3.900	3.400	3.900
	ORED<	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	4.100	9.600	3.700	9.200	3.700	9.200	3.700	9.200
				非導通時	4.000	9.600	3.700	9.200	3.700	9.200	3.700	9.200
	LDED>=	雙精度	導通時		4.100	9.600	3.600	9.000	3.600	9.000	3.600	9.000
			非導通時		4.100	9.600	3.600	8.900	3.600	8.900	3.600	8.900
	ANDED>=	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	3.800	7.900	3.400	7.400	3.400	7.400	3.400	7.400
				非導通時	3.900	8.100	3.400	7.500	3.400	7.500	3.400	7.500
	ORED>=	雙精度	未執行時			0.060		0.0285		0.0285		0.0285
			執行時	導通時	4.100	9.600	3.700	9.200	3.700	9.200	3.700	9.200
				非導通時	4.000	7.200	3.600	6.600	3.600	6.600	3.600	6.600
	LD$=	導通時			5.300	8.900	4.700	8.100	4.700	8.100	4.700	8.100
		非導通時			4.700	9.000	4.200	8.200	4.200	8.200	4.200	8.200
	AND$=	未執行時				0.060		0.0285		0.0285		0.0285
		執行時	導通時		4.400	6.800	3.900	6.400	3.900	6.400	3.900	6.400
			非導通時		4.500	6.700	4.000	6.300	4.000	6.300	4.000	6.300
	OR$=	未執行時				0.060		0.0285		0.0285		0.0285
		執行時	導通時		5.100	8.200	4.200	7.600	4.200	7.600	4.200	7.600
			非導通時		5.000	8.100	4.000	7.200	4.000	7.200	4.000	7.200
	LD$< >	導通時			4.800	8.100	4.300	7.500	4.300	7.500	4.300	7.500
		非導通時			4.700	8.400	4.200	7.800	4.200	7.800	4.200	7.800
	AND$< >	未執行時				0.060		0.0285		0.0285		0.0285
		執行時	導通時		4.300	5.500	4.100	5.100	4.100	5.100	4.100	5.100
			非導通時		4.500	5.900	4.400	5.400	4.400	5.400	4.400	5.400
	OR$< >	未執行時				0.060		0.0285		0.0285		0.0285
		執行時	導通時		5.200	7.300	4.100	6.700	4.100	6.700	4.100	6.700
			非導通時		5.100	7.200	4.100	6.700	4.100	6.700	4.100	6.700
	LD$>	導通時			4.800	7.200	4.300	6.700	4.300	6.700	4.300	6.700
		非導通時			4.800	7.700	4.200	7.100	4.200	7.100	4.200	7.100
	AND$>	未執行時				0.060		0.0285		0.0285		0.0285
		執行時	導通時		4.500	7.100	4.000	6.700	4.000	6.700	4.000	6.700
			非導通時		4.600	7.600	4.300	7.000	4.300	7.000	4.300	7.000
	OR$>	未執行時				0.060		0.0285		0.0285		0.0285
		執行時	導通時		5.100	6.800	4.300	6.200	4.300	6.200	4.300	6.200
			非導通時		5.200	7.200	4.300	6.600	4.300	6.600	4.300	6.600
	LD$<=	導通時			5.000	6.300	4.400	5.700	4.400	5.700	4.400	5.700
		非導通時			4.800	6.400	4.200	5.800	4.200	5.800	4.200	5.800

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	AND$<=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時	4.600	7.600	4.100	7.200	4.100	7.200	4.100	7.200
			非導通時	4.700	7.700	4.200	7.300	4.200	7.300	4.200	7.300
	OR$<=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時	4.700	7.700	4.400	7.200	4.400	7.200	4.400	7.200
			非導通時	4.600	7.600	4.400	7.100	4.400	7.100	4.400	7.100
	LD$<	導通時		4.800	8.100	4.500	7.500	4.500	7.500	4.500	7.500
		非導通時		5.000	8.300	4.500	7.900	4.500	7.900	4.500	7.900
	AND$<	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時	4.500	7.100	4.000	6.600	4.000	6.600	4.000	6.600
			非導通時	4.900	7.500	4.400	7.100	4.400	7.100	4.400	7.100
	OR$<	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時	5.100	7.800	4.100	7.200	4.100	7.200	4.100	7.200
			非導通時	5.000	8.100	4.100	7.600	4.100	7.600	4.100	7.600
	LD$>=	導通時		4.800	6.700	4.500	6.200	4.500	6.200	4.500	6.200
		非導通時		5.000	6.700	4.400	6.300	4.400	6.300	4.400	6.300
	AND$>=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時	4.400	6.800	4.100	6.300	4.100	6.300	4.100	6.300
			非導通時	4.500	7.000	4.200	6.600	4.200	6.600	4.200	6.600
	OR$>=	未執行時			0.060		0.0285		0.0285		0.0285
		執行時	導通時	5.400	6.600	4.100	5.800	4.100	5.800	4.100	5.800
			非導通時	5.300	6.300	4.100	5.700	4.100	5.700	4.100	5.700
	BKCMP= ⓢ1 ⓢ2 Ⓓ n	n=1		8.200	10.700	7.500	10.000	7.500	10.000	7.500	10.000
		n=96		57.400	61.800	46.400	48.700	46.400	48.700	46.400	48.700
	BKCMP< > ⓢ1 ⓢ2 Ⓓ n	n=1		8.200	10.700	7.500	10.000	7.500	10.000	7.500	10.000
		n=96		59.500	63.300	45.600	50.400	45.600	50.400	45.600	50.400
	BKCMP> ⓢ1 ⓢ2 Ⓓ n	n=1		8.200	10.800	7.500	10.100	7.500	10.100	7.500	10.100
		n=96		59.500	63.400	47.700	50.500	47.700	50.500	47.700	50.500
	BKCMP<= ⓢ1 ⓢ2 Ⓓ n	n=1		8.200	10.600	7.500	10.000	7.500	10.000	7.500	10.000
		n=96		57.400	61.700	46.400	49.000	46.400	49.000	46.400	49.000
	BKCMP< ⓢ1 ⓢ2 Ⓓ n	n=1		8.300	10.600	7.500	10.000	7.500	10.000	7.500	10.000
		n=96		59.500	63.600	47.600	50.500	47.600	50.500	47.600	50.500
	BKCMP>= ⓢ1 ⓢ2 Ⓓ n	n=1		8.200	10.900	7.500	10.000	7.500	10.000	7.500	10.000
		n=96		57.400	62.000	46.400	48.900	46.400	48.900	46.400	48.900
	DB* ⓢ1 ⓢ2 Ⓓ	執行時		8.300	12.100	8.100	11.600	8.100	11.600	8.100	11.600
	DB/ ⓢ1 ⓢ2 Ⓓ	執行時		6.100	9.100	5.800	8.800	5.800	8.800	5.800	8.800
	DBKCMP= ⓢ1 ⓢ2 Ⓓ n	n=1		9.250	14.000	8.600	13.000	8.600	13.000	8.600	13.000
		n=96		60.700	67.500	47.900	52.800	47.900	52.800	47.900	52.800
	DBKCMP< > ⓢ1 ⓢ2 Ⓓ n	n=1		9.250	14.000	8.600	13.000	8.600	13.000	8.600	13.000
		n=96		60.700	67.500	47.900	52.800	47.900	52.800	47.900	52.800
	DBKCMP> ⓢ1 ⓢ2 Ⓓ n	n=1		9.250	14.000	8.600	13.000	8.600	13.000	8.600	13.000
		n=96		60.700	67.500	47.900	52.800	47.900	52.800	47.900	52.800
	DBKCMP<= ⓢ1 ⓢ2 Ⓓ n	n=1		9.250	14.000	8.600	13.000	8.600	13.000	8.600	13.000
		n=96		60.700	67.500	47.900	52.800	47.900	52.800	47.900	52.800
	DBKCMP< ⓢ1 ⓢ2 Ⓓ n	n=1		9.250	14.000	8.600	13.000	8.600	13.000	8.600	13.000
		n=96		60.700	67.500	47.900	52.800	47.900	52.800	47.900	52.800
	DBKCMP>= ⓢ1 ⓢ2 Ⓓ n	n=1		9.250	14.000	8.600	13.000	8.600	13.000	8.600	13.000
		n=96		60.700	67.500	47.900	52.800	47.900	52.800	47.900	52.800
	DB+ Ⓢ Ⓓ	執行時		4.900	7.000	4.600	6.400	4.600	6.400	4.600	6.400
	DB+ ⓢ1 ⓢ2 Ⓓ	執行時		5.200	7.300	4.800	6.700	4.800	6.700	4.800	6.700
	DB- Ⓢ Ⓓ	執行時		4.900	6.600	4.700	6.000	4.700	6.000	4.700	6.000
	DB- ⓢ1 ⓢ2 Ⓓ	執行時		5.200	7.500	4.800	6.600	4.800	6.600	4.800	6.600

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	ED+ Ⓢ Ⓓ	雙精度	Ⓢ=0、Ⓓ=0	4.800	8.000	4.300	7.200	4.300	7.200	4.300	7.200
			Ⓢ=2^{1023}、Ⓓ=2^{1023}	4.800	8.000	4.300	7.200	4.300	7.200	4.300	7.200
	ED+ Ⓢ1 Ⓢ2 Ⓓ	雙精度	Ⓢ1=0、Ⓢ2=0	5.500	9.800	4.800	9.200	4.800	9.200	4.800	9.200
			Ⓢ1=2^{1023}、Ⓢ2=2^{1023}	5.500	9.800	4.800	9.200	4.800	9.200	4.800	9.200
	ED- Ⓢ Ⓓ	雙精度	Ⓢ=0、Ⓓ=0	5.000	8.200	4.400	7.500	4.400	7.500	4.400	7.500
			Ⓢ=2^{1023}、Ⓓ=2^{1023}	5.000	8.200	4.400	7.500	4.400	7.500	4.400	7.500
	ED- Ⓢ1 Ⓢ2 Ⓓ	雙精度	Ⓢ1=0、Ⓢ2=0	4.400	8.100	3.800	7.500	3.800	7.500	3.800	7.500
			Ⓢ1=2^{1023}、Ⓢ2=2^{1023}	4.400	8.100	3.800	7.500	3.800	7.500	3.800	7.500
	ED* Ⓢ1 Ⓢ2 Ⓓ	雙精度	Ⓢ1=0、Ⓢ2=0	5.800	9.500	5.100	8.800	5.100	8.800	5.100	8.800
			Ⓢ1=2^{1023}、Ⓢ2=2^{1023}	5.800	9.500	5.100	8.800	5.100	8.800	5.100	8.800
	ED/ Ⓢ1 Ⓢ2 Ⓓ	雙精度	Ⓢ1=2^{1023}、Ⓢ2=2^{1023}	6.600	10.600	5.900	10.000	5.900	10.000	5.900	10.000
	BK+ Ⓢ1 Ⓢ2 Ⓓ n	n=1		9.100	11.200	8.500	10.600	8.500	10.600	8.500	10.600
		n=96		60.700	62.900	44.600	47.000	44.600	47.000	44.600	47.000
	BK- Ⓢ1 Ⓢ2 Ⓓ n	n=1		9.700	12.000	8.900	11.300	8.900	11.300	8.900	11.300
		n=96		61.300	63.600	45.600	47.900	45.600	47.900	45.600	47.900
	DBK+ Ⓢ1 Ⓢ2 Ⓓ n	n=1		7.000	10.700	6.450	9.950	6.450	9.950	6.450	9.950
		n=96		59.400	63.100	43.700	47.500	43.700	47.500	43.700	47.500
	DBK- Ⓢ1 Ⓢ2 Ⓓ n	n=1		7.000	10.700	6.450	9.950	6.450	9.950	6.450	9.950
		n=96		59.400	63.100	43.700	47.500	43.700	47.500	43.700	47.500
	$+ Ⓢ Ⓓ	---		8.800	14.600	8.100	13.900	8.100	13.900	8.100	13.900
	$+ Ⓢ1 Ⓢ2 Ⓓ	---		7.300	11.100	6.500	10.300	6.500	10.300	6.500	10.300
	FLTD	雙精度	Ⓢ=0	2.300	5.000	1.800	4.700	1.800	4.700	1.800	4.700
			Ⓢ=$7FFF_H$	2.500	5.200	2.200	4.800	2.200	4.800	2.200	4.800
	DFLTD	雙精度	Ⓢ=0	2.400	5.200	2.000	4.900	2.000	4.900	2.000	4.900
			Ⓢ=$7FFFFFFF_H$	2.700	5.400	2.300	5.100	2.300	5.100	2.300	5.100
	INTD	雙精度	Ⓢ=0	2.700	4.100	2.200	4.100	2.200	4.100	2.200	4.100
			Ⓢ=32766.5	3.700	5.900	3.200	5.600	3.200	5.600	3.200	5.600
	DINTD	雙精度	Ⓢ=0	2.600	3.900	2.200	3.400	2.200	3.400	2.200	3.400
			Ⓢ=1234567890.3	3.400	5.600	3.000	5.100	3.000	5.100	3.000	5.100
	DBL	執行時		2.700	3.400	2.300	2.700	2.300	2.700	2.300	2.700
	WORD	執行時		2.900	4.300	2.600	3.600	2.600	3.600	2.600	3.600
	GRY	執行時		2.700	3.900	2.300	3.400	2.300	3.400	2.300	3.400
	DGRY	執行時		2.900	3.500	2.500	3.000	2.500	3.000	2.500	3.000
	GBIN	執行時		4.000	4.800	3.800	4.300	3.800	4.300	3.800	4.300
	DGBIN	執行時		5.500	6.100	5.000	5.900	5.000	5.900	5.000	5.900
	NEG	執行時		2.400	3.900	2.000	3.300	2.000	3.300	2.000	3.300
	DNEG	執行時		2.500	3.700	2.500	3.300	2.500	3.300	2.500	3.300
	ENEG	浮點=0		2.500	3.300	2.300	2.800	2.300	2.800	2.300	2.800
		浮點=-1.0		2.700	4.500	2.500	3.900	2.500	3.900	2.500	3.900
	EDNEG	浮點=0		2.200	3.500	1.800	3.100	1.800	3.100	1.800	3.100
		浮點=-1.0		2.400	3.500	1.900	3.000	1.900	3.000	1.900	3.000
	BKBCD Ⓢ Ⓓ n	n=1		6.600	8.900	5.900	8.200	5.900	8.200	5.900	8.200
		n=96		71.300	74.100	61.000	63.400	61.000	63.400	61.000	63.400

分類	指令	條件（軟元件）	處理時間（μs）							
			Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
基本指令	BKBIN Ⓢ Ⓓ n	n=1	6.500	9.800	5.600	9.300	5.600	9.300	5.600	9.300
		n=96	56.300	59.500	49.200	52.500	49.200	52.500	49.200	52.500
	ECON	---	2.600	5.400	2.100	4.500	2.100	4.500	2.100	4.500
	EDCON	---	2.800	5.400	2.500	5.400	2.500	5.400	2.500	5.400
	EDMOV	---	2.300	5.500	1.700	5.000	1.700	5.000	1.700	5.000
	$MOV	傳送字串 =0	4.000	6.300	3.400	5.600	3.400	5.600	3.400	5.600
		傳送字串 =32	14.600	16.500	11.400	13.300	11.400	13.300	11.400	13.300
	BXCH Ⓓ1 Ⓓ2 n	n=1	6.200	7.900	5.500	7.300	5.500	7.300	5.500	7.300
		n=96	67.000	68.800	47.300	49.300	47.300	49.300	47.300	49.300
	SWAP	---	2.400	2.700	1.900	2.200	1.900	2.200	1.900	2.200
	GOEND	---		0.500		0.500		0.500		0.500
	DI	---	1.800	2.200	1.500	1.800	1.500	1.800	1.500	1.800
	EI	---	3.100	3.800	3.000	3.300	3.000	3.300	3.000	3.300
	IMASK	---	9.800	13.300	7.200	10.500	7.200	10.500	7.200	10.500
	IRET	---		1.000		1.000		1.000		1.000
	RSF X n	n=1	4.200	5.900	3.700	5.600	3.700	5.600	3.700	5.600
		n=96	11.400	13.800	10.700	12.400	10.700	12.400	10.700	12.400
	RSF Y n	n=1	3.800	4.800	3.400	4.800	3.400	4.800	3.400	4.800
		n=96	8.500	9.500	8.100	8.900	8.100	8.900	8.100	8.900
	UDCNT1	---	0.900	1.500	0.500	0.983	0.500	0.983	0.500	0.983
	UDCNT2	---	0.900	1.700	0.600	1.300	0.600	1.300	0.600	1.300
	TTMR	---	3.900	6.100	3.400	5.400	3.400	5.400	3.400	5.400
	STMR	---	6.800	13.500	5.800	12.500	5.800	12.500	5.800	12.500
	ROTC	---	9.000	10.500	8.000	9.400	8.000	9.400	8.000	9.400
	RAMP	---	5.900	8.800	5.200	8.400	5.200	8.400	5.200	8.400
	SPD	---	0.900	1.900	0.500	1.400	0.500	1.400	0.500	1.400
	PLSY	---	1.900	2.200	1.500	1.800	1.500	1.800	1.500	1.800
	PWM	---	1.200	1.600	0.900	1.200	0.900	1.200	0.900	1.200
	MTR	---	10.400	19.800	9.400	10.000	9.400	10.000	9.400	10.000
應用指令	BKAND Ⓢ1 Ⓢ2 Ⓓ n	n=1	9.000	11.700	8.300	11.000	8.300	11.000	8.300	11.000
		n=96	57.400	63.100	43.800	47.300	43.800	47.300	43.800	47.300
	BKOR Ⓢ1 Ⓢ2 Ⓓ n	n=1	7.700	10.000	7.700	9.500	7.700	9.500	7.700	9.500
		n=96	57.400	61.900	44.300	45.800	44.300	45.800	44.300	45.800
	BKXOR Ⓢ1 Ⓢ2 Ⓓ n	n=1	7.800	10.100	7.300	9.200	7.300	9.200	7.300	9.200
		n=96	57.300	61.500	43.800	45.800	43.800	45.800	43.800	45.800
	BKXNR Ⓢ1 Ⓢ2 Ⓓ n	n=1	7.800	9.600	7.600	8.900	7.600	8.900	7.600	8.900
		n=96	57.400	61.400	43.900	45.300	43.900	45.300	43.900	45.300
	BSFR Ⓓ n	n=1	3.700	5.400	3.200	4.800	3.200	4.800	3.200	4.800
		n=96	6.900	9.000	5.800	7.700	5.800	7.700	5.800	7.700
	BSFL Ⓓ n	n=1	4.100	5.900	3.400	5.100	3.400	5.100	3.400	5.100
		n=96	7.100	9.100	6.000	7.900	6.000	7.900	6.000	7.900
	SFTBR Ⓓ n1 n2	進行了移位的位的數 =16/ 移位數 =1	7.950	17.500	7.600	16.900	7.600	16.900	7.600	16.900
		進行了移位的位的數 =16/ 移位數 =15	7.950	17.500	7.550	16.900	7.550	16.900	7.550	16.900
	SFTBL Ⓓ n1 n2	進行了移位的位的數 =16/ 移位數 =1	7.950	17.900	7.500	17.400	7.500	17.400	7.500	17.400
		進行了移位的位的數 =16/ 移位數 =15	7.900	17.800	7.500	17.300	7.500	17.300	7.500	17.300
	SFTWR Ⓓ n1 n2	進行了移位的位的數 =16/ 移位數 =1	5.950	10.600	4.600	8.700	4.600	8.700	4.600	8.700
		進行了移位的位的數 =16/ 移位數 =15	5.900	10.600	4.600	8.700	4.600	8.700	4.600	8.700

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	SFTWL Ⓓ n1 n2	進行了移位的位的數=16/ 移位數 =1		5.950	10.700	4.550	8.700	4.550	8.700	4.550	8.700
		進行了移位的位的數=16/ 移位數 =15		5.950	10.700	4.600	8.800	4.600	8.800	4.600	8.800
	BSET Ⓓ n	n=1		3.000	3.400	2.500	2.800	2.500	2.800	2.500	2.800
		n=15		3.000	3.500	2.500	2.800	2.500	2.800	2.500	2.800
	BRST Ⓓ n	n=1		3.000	3.400	2.600	2.800	2.600	2.800	2.600	2.800
		n=15		3.000	3.400	2.500	2.800	2.500	2.800	2.500	2.800
	TEST	執行時		4.400	5.300	3.700	4.700	3.700	4.700	3.700	4.700
	DTEST	執行時		4.500	5.400	3.900	4.800	3.900	4.800	3.900	4.800
	BKRST Ⓢ n	n=1		4.300	4.600	3.700	4.100	3.700	4.100	3.700	4.100
		n=96		6.000	6.800	5.100	6.000	5.100	6.000	5.100	6.000
	SER Ⓢ1 Ⓢ2 Ⓓ n	n=1	全部一致	4.900	5.300	4.200	4.600	4.200	4.600	4.200	4.600
			全部不一致	5.000	5.300	4.200	4.600	4.200	4.600	4.200	4.600
		n=96	全部一致	32.300	32.900	25.900	26.300	25.900	26.300	25.900	26.300
			全部不一致	32.400	32.900	25.900	26.300	25.900	26.300	25.900	26.300
	DSER Ⓢ1 Ⓢ2 Ⓓ n	n=1	全部一致	6.100	6.500	5.400	5.700	5.400	5.700	5.400	5.700
			全部不一致	6.200	6.600	5.500	5.900	5.500	5.900	5.500	5.900
		n=96	全部一致	52.800	54.200	41.200	41.800	41.200	41.800	41.200	41.800
			全部不一致	52.800	54.200	41.200	41.800	41.200	41.800	41.200	41.800
	DSUM Ⓢ Ⓓ	Ⓢ=0		3.700	4.100	3.300	3.600	3.300	3.600	3.300	3.600
		Ⓢ=FFFFFFFF$_H$		3.800	4.100	3.200	3.700	3.200	3.700	3.200	3.700
	DECO Ⓢ Ⓓ n	n=2		6.000	7.500	5.300	6.900	5.300	6.900	5.300	6.900
		n=8		8.100	9.300	6.800	7.800	6.800	7.800	6.800	7.800
	ENCO Ⓢ Ⓓ n	n=2	M1=ON	5.300	5.700	4.700	5.100	4.700	5.100	4.700	5.100
			M4=ON	5.200	5.700	4.600	5.000	4.600	5.000	4.600	5.000
		n=8	M1=ON	10.400	11.400	9.000	10.000	9.000	10.000	9.000	10.000
			M256=ON	5.700	6.800	5.100	6.100	5.100	6.100	5.100	6.100
	DIS Ⓢ Ⓓ n	n=1		4.400	5.300	3.800	4.600	3.800	4.600	3.800	4.600
		n=4		4.800	5.700	4.000	5.000	4.000	5.000	4.000	5.000
	UNI Ⓢ Ⓓ n	n=1		5.000	5.300	3.500	4.800	3.500	4.800	3.500	4.800
		n=4		5.600	6.000	4.000	5.100	4.000	5.100	4.000	5.100
	NDIS	執行時		11.000	13.100	11.000	13.200	11.000	13.200	11.000	13.200
	NUNI	執行時		10.600	12.700	7.300	13.200	7.300	13.200	7.300	13.200
	WTOB Ⓢ Ⓓ n	n=1		5.000	6.500	4.400	5.800	4.400	5.800	4.400	5.800
		n=96		36.000	38.400	28.200	29.300	28.200	29.300	28.200	29.300
	BTOW Ⓢ Ⓓ n	n=1		5.100	6.100	4.600	5.500	4.600	5.500	4.600	5.500
		n=96		29.900	32.000	22.800	23.800	22.800	23.800	22.800	23.800
	MAX Ⓢ Ⓓ n	n=1		4.300	6.900	4.000	6.100	4.000	6.100	4.000	6.100
		n=96		31.200	33.500	24.700	27.000	24.700	27.000	24.700	27.000
	MIN Ⓢ Ⓓ n	n=1		4.400	6.800	4.000	6.000	4.000	6.000	4.000	6.000
		n=96		30.300	34.800	26.500	28.300	26.500	28.300	26.500	28.300
	DMAX Ⓢ Ⓓ n	n=1		4.800	9.100	4.800	8.100	4.800	8.100	4.800	8.100
		n=96		56.400	62.200	47.100	49.600	47.100	49.600	47.100	49.600
	DMIN Ⓢ Ⓓ n	n=1		4.800	6.800	4.300	5.900	4.300	5.900	4.300	5.900
		n=96		55.400	60.200	45.400	47.400	45.400	47.400	45.400	47.400
	SORT Ⓢ1 n Ⓢ2 Ⓓ1 Ⓓ2	n=1、 Ⓢ2 =1		6,200	9,300	5,600	8,800	5,600	8,800	5,600	8,800
		n=96、 Ⓢ2 =16		28,200	38,500	22,200	32,200	22,200	32,200	22,200	32,200
	DSORT Ⓢ1 n Ⓢ2 Ⓓ1 Ⓓ2	n=1、 Ⓢ2 =1		6,200	11,600	5,600	10,900	5,600	10,900	5,600	10,900
		n=96、 Ⓢ2 =16		34,700	45,300	26,700	36,900	26,700	36,900	26,700	36,900
	WSUM Ⓢ Ⓓ n	n=1		4.800	6.200	4.200	5.500	4.200	5.500	4.200	5.500
		n=96		26.900	28.700	21.300	22.300	21.300	22.300	21.300	22.300
	DWSUM Ⓢ Ⓓ n	n=1		5.500	7.000	4.800	6.100	4.800	6.100	4.800	6.100
		n=96		53.000	56.300	42.700	44.000	42.700	44.000	42.700	44.000

分類	指令	條件（軟元件）	處理時間（μs）							
			Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	MEAN Ⓢ Ⓓ n	n=1	4.300	8.650	3.900	7.800	3.900	7.800	3.900	7.800
		n=96	16.000	21.400	12.900	18.000	12.900	18.000	12.900	18.000
	DMEAN Ⓢ Ⓓ n	n=1	5.700	10.600	5.300	9.950	5.300	9.950	5.300	9.950
		n=96	29.200	35.200	23.000	28.800	23.000	28.800	23.000	28.800
	NEXT	---	0.940	1.400	0.770	1.200	0.770	1.200	0.770	1.200
	BREAK	---	10.400	5.500	9.100	5.000	9.100	5.000	9.100	5.000
	RET	返回至本程式	2.000	3.000	1.600	2.600	1.600	2.600	1.600	2.600
		返回至其他程式	2.300	3.700	2.000	3.100	2.000	3.100	2.000	3.100
	FCALL Pn	文件內指標	3.100	4.400	2.700	3.600	2.700	3.600	2.700	3.600
		公共指標	4.000	5.700	3.600	5.100	3.600	5.100	3.600	5.100
	FCALL Pn Ⓢ1 ～ Ⓢ5	---	19.300	21.500	16.500	18.600	16.500	18.600	16.500	18.600
	ECALL * Pn *：程式名	---	70.300	82.300	65.900	77.600	65.900	77.600	65.900	77.600
	ECALL * Pn Ⓢ1 ～ Ⓢ5 *：程式名	---	101.000	114.000	91.800	105.000	91.800	105.000	91.800	105.000
	EFCALL * Pn *：程式名	---	70.700	82.800	66.200	78.100	66.200	78.100	66.200	78.100
	EFCALL * Pn Ⓢ1 ～ Ⓢ5 *：程式名	---	86.500	107.000	78.800	91.600	78.800	91.600	78.800	91.600
	XCALL	---	3.800	5.700	3.700	5.200	3.700	5.200	3.700	5.200
	COM CCOM	僅選擇 I/O 刷新時	12.800	29.100	12.400	28.600	12.400	28.600	12.400	28.600
		僅選擇 CC-Link 刷新時（主站側）	16.000	39.500	15.500	39.100	15.500	39.100	15.500	39.100
		僅選擇 CC-Link 刷新時（本地站側）	16.100	39.500	15.500	39.100	15.500	39.100	15.500	39.100
		・僅選擇 MELSECNET/H 刷新時（管理站側） ・僅選擇 CC-Link IE 控制網路刷新時（管理站側）	34.700	70.400	34.400	69.800	34.400	69.800	34.400	69.800
		・僅選擇 MELSECNET/H 刷新時（普通站側） ・僅選擇 CC-Link IE 控制網路刷新時（普通站側）	34.700	70.400	34.400	69.800	34.400	69.800	34.400	69.800
		僅選擇 CC-Link IE 現場網路刷新時（主站側）	17.000	38.800	16.600	38.000	16.600	38.000	16.600	38.000
		僅選擇 CC-Link IE 現場網路刷新時（本地站側）	17.000	38.800	16.600	38.000	16.600	38.000	16.600	38.000
		僅選擇智慧自動刷新時	12.800	33.200	12.800	33.200	12.800	33.200	12.800	33.200
		僅選擇組外輸入輸出時（僅輸入）	7.900	21.100	7.700	20.700	7.700	20.700	7.700	20.700

分類	指令	條件（軟元件）	處理時間（μs）							
			Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	COM CCOM	僅選擇組外輸入輸出時（僅輸出）	16.900	44.800	16.500	44.200	16.500	44.200	16.500	44.200
		僅選擇組外輸入輸出時（輸入輸出兩方）	22.600	52.600	22.400	52.600	22.400	52.600	22.400	52.600
		僅選擇多 CPU 高速通信區域刷新時	13.000	33.800	12.700	33.200	12.700	33.200	12.700	33.200
		僅選擇與週邊設備的通信時	7.250	18.800	7.100	18.500	7.100	18.500	7.100	18.500
	FIFW	資料數 = 0	3.700	5.300	3.200	4.600	3.200	4.600	3.200	4.600
		資料數 = 96	3.800	4.400	3.300	3.800	3.300	3.800	3.300	3.800
	FIFR	資料數 = 1	4.300	5.000	3.800	4.400	3.800	4.400	3.800	4.400
		資料數 = 96	33.500	35.500	24.800	25.700	24.800	25.700	24.800	25.700
	FPOP	資料數 = 1	4.300	5.900	3.800	5.300	3.800	5.300	3.800	5.300
		資料數 = 96	4.300	5.900	3.700	5.400	3.700	5.400	3.700	5.400
	FINS	資料數 = 0	4.800	5.900	3.700	5.300	3.700	5.300	3.700	5.300
		資料數 = 96	4.300	5.900	3.700	5.300	3.700	5.300	3.700	5.300
	FDEL	資料數 = 1	4.900	6.500	4.200	5.800	4.200	5.800	4.200	5.800
		資料數 = 96	34.200	35.900	25.400	25.900	25.400	25.900	25.400	25.900
	FROM n1 n2 Ⓓ n3	n3=1	10.800	24.100	10.700	23.600	10.700	23.600	10.700	23.600
		n3=1000	392.600	413.300	390.900	410.200	390.900	410.200	390.900	410.200
	DFRO n1 n2 Ⓓ n3	n3=1	13.600	27.700	12.600	26.700	12.600	26.700	12.600	26.700
		n3=500	392.600	413.300	390.900	410.200	390.900	410.200	390.900	410.200
	TO n1 n2 Ⓢ n3	n3=1	10.200	21.900	9.600	21.300	9.600	21.300	9.600	21.300
		n3=1000	373.700	394.100	372.500	390.800	372.500	390.800	372.500	390.800
	DTO n1 n2 Ⓢ n3	n3=1	13.000	26.700	12.000	25.700	12.000	25.700	12.000	25.700
		n3=500	373.700	394.100	372.500	390.800	372.500	390.800	372.500	390.800
	LEDR	無顯示→無顯示	2.400	2.600	1.900	2.000	1.900	2.000	1.900	2.000
		LED 指令執行→無顯示	28.100	39.400	24.400	35.800	24.400	35.800	24.400	35.800
	BINDA Ⓢ1 Ⓓ	Ⓢ=1	4.900	6.500	4.300	5.600	4.300	5.600	4.300	5.600
		Ⓢ=-32768	7.200	8.700	6.500	8.000	6.500	8.000	6.500	8.000
	DBINDA Ⓢ Ⓓ	Ⓢ=1	5.700	7.100	4.900	6.300	4.900	6.300	4.900	6.300
		Ⓢ=-2147483648	10.400	12.000	9.600	11.000	9.600	11.000	9.600	11.000
	BINHA Ⓢ Ⓓ	Ⓢ=1	4.400	5.900	3.800	5.200	3.800	5.200	3.800	5.200
		Ⓢ=$FFFF_H$	4.400	5.800	3.700	5.200	3.700	5.200	3.700	5.200
	DBINHA Ⓢ Ⓓ	Ⓢ=1	5.200	6.700	4.600	6.000	4.600	6.000	4.600	6.000
		Ⓢ=$FFFFFFFF_H$	5.100	6.500	4.600	6.000	4.600	6.000	4.600	6.000
	BCDDA Ⓢ Ⓓ	Ⓢ=1	4.300	5.800	3.600	5.000	3.600	5.000	3.600	5.000
		Ⓢ=9999	4.700	6.100	4.100	5.400	4.100	5.400	4.100	5.400
	DBCDDA Ⓢ Ⓓ	Ⓢ=1	4.800	6.300	4.000	5.500	4.000	5.500	4.000	5.500
		Ⓢ=99999999	5.600	7.100	4.900	6.300	4.900	6.300	4.900	6.300
	DABIN Ⓢ Ⓓ	Ⓢ=1	6.500	8.500	5.800	7.800	5.800	7.800	5.800	7.800
		Ⓢ=-32768	6.300	8.300	5.600	7.700	5.600	7.700	5.600	7.700
	DDABIN Ⓢ Ⓓ	Ⓢ=1	9.400	11.500	8.500	10.500	8.500	10.500	8.500	10.500
		Ⓢ=-2147483648	9.100	11.200	8.100	10.200	8.100	10.200	8.100	10.200
	HABIN Ⓢ Ⓓ	Ⓢ=1	4.900	7.100	4.400	6.400	4.400	6.400	4.400	6.400
		Ⓢ=$FFFF_H$	5.100	7.300	4.600	6.500	4.600	6.500	4.600	6.500
	DHABIN Ⓢ Ⓓ	Ⓢ=1	6.000	8.100	5.300	7.300	5.300	7.300	5.300	7.300
		Ⓢ=$FFFFFFFF_H$	6.300	8.500	5.600	7.700	5.600	7.700	5.600	7.700

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	DABCD Ⓢ Ⓓ	Ⓢ=1		5.000	7.100	4.400	6.300	4.400	6.300	4.400	6.300
		Ⓢ=9999		5.000	7.100	4.300	6.300	4.300	6.300	4.300	6.300
	DDABCD Ⓢ Ⓓ	Ⓢ=1		6.200	8.300	5.500	7.400	5.500	7.400	5.500	7.400
		Ⓢ=99999999		6.200	8.300	5.500	7.500	5.500	7.500	5.500	7.500
	COMRD	---		51.600	52.400	50.900	51.200	50.900	51.200	50.900	51.200
	LEN	1 字元		4.100	6.200	3.600	5.500	3.600	5.500	3.600	5.500
		96 字元		19.800	22.200	16.800	18.700	16.800	18.700	16.800	18.700
	STR	---		6.900	11.100	6.600	10.400	6.600	10.400	6.600	10.400
	DSTR	---		10.200	12.500	9.600	11.500	9.600	11.500	9.600	11.500
	VAL	---		9.800	14.200	8.900	13.000	8.900	13.000	8.900	13.000
	DVAL	---		14.000	18.700	12.700	16.800	12.700	16.800	12.700	16.800
	ESTR	---		18.700	24.100	17.900	23.100	17.900	23.100	17.900	23.100
	LDTM=	與指定的時間比較	導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的時間比較	導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
			非導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
	ANDTM=	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
			非導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
		與當前的時間比較	導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
			非導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
	ORTM=	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
			非導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
		與當前的時間比較	導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
			非導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
	LDTM<>	與指定的時間比較	導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的時間比較	導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
			非導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
	ANDTM<>	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
			非導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
		與當前的時間比較	導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
			非導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
	ORTM<>	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
			非導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
		與當前的時間比較	導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
			非導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
	LDTM>	與指定的時間比較	導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的時間比較	導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
			非導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
	ANDTM>	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
			非導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
		與當前的時間比較	導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
			非導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
	ORTM>	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
			非導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
		與當前的時間比較	導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
			非導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	LDTM<=	與指定的時間比較	導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的時間比較	導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
			非導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
	ANDTM<=	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
			非導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
		與當前的時間比較	導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
			非導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
	ORTM<=	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
			非導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
		與當前的時間比較	導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
			非導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
	LDTM<	與指定的時間比較	導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的時間比較	導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
			非導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
	ANDTM<	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
			非導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
		與當前的時間比較	導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
			非導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
	ORTM<	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
			非導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
		與當前的時間比較	導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
			非導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
	LDTM>=	與指定的時間比較	導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.300	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的時間比較	導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
			非導通時	5.800	9.900	5.400	9.500	5.400	9.500	5.400	9.500
	ANDTM>=	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
			非導通時	7.000	11.500	6.300	10.800	6.300	10.800	6.300	10.800
		與當前的時間比較	導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
			非導通時	5.500	9.900	5.100	9.500	5.100	9.500	5.100	9.500
	ORTM>=	未執行時			0.008		0.038		0.038		0.038
		與指定的時間比較	導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
			非導通時	7.300	11.500	6.600	10.800	6.600	10.800	6.600	10.800
		與當前的時間比較	導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
			非導通時	5.900	9.900	5.300	9.500	5.300	9.500	5.300	9.500
	LDDT=	與指定的日期比較	導通時	7.400	11.400	6.800	10.900	6.800	10.900	6.800	10.900
			非導通時	7.400	11.600	6.800	10.900	6.800	10.900	6.800	10.900
		與當前的日期比較	導通時	5.900	10.000	5.500	9.700	5.500	9.700	5.500	9.700
			非導通時	5.900	10.100	5.500	9.700	5.500	9.700	5.500	9.700
	ANDDT=	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
			非導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
		與當前的日期比較	導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
			非導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
	ORDT=	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的日期比較	導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600
			非導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	LDDT<>	與指定的日期比較	導通時	7.400	11.400	6.800	10.900	6.800	10.900	6.800	10.900
			非導通時	7.400	11.600	6.800	10.900	6.800	10.900	6.800	10.900
		與當前的日期比較	導通時	5.900	10.000	5.500	9.700	5.500	9.700	5.500	9.700
			非導通時	5.900	10.100	5.500	9.700	5.500	9.700	5.500	9.700
	ANDDT<>	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
			非導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
		與當前的日期比較	導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
			非導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
	ORDT<>	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的日期比較	導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600
			非導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600
	LDDT>	與指定的日期比較	導通時	7.400	11.400	6.800	10.900	6.800	10.900	6.800	10.900
			非導通時	7.400	11.600	6.800	10.900	6.800	10.900	6.800	10.900
		與當前的日期比較	導通時	5.900	10.000	5.500	9.700	5.500	9.700	5.500	9.700
			非導通時	5.900	10.100	5.500	9.700	5.500	9.700	5.500	9.700
	ANDDT>	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
			非導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
		與當前的日期比較	導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
			非導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
	ORDT>	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的日期比較	導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600
			非導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600
	LDDT<=	與指定的日期比較	導通時	7.400	11.400	6.800	10.900	6.800	10.900	6.800	10.900
			非導通時	7.400	11.600	6.800	10.900	6.800	10.900	6.800	10.900
		與當前的日期比較	導通時	5.900	10.000	5.500	9.700	5.500	9.700	5.500	9.700
			非導通時	5.900	10.100	5.500	9.700	5.500	9.700	5.500	9.700
	ANDDT<=	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
			非導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
		與當前的日期比較	導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
			非導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
	ORDT<=	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的日期比較	導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600
			非導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600
	LDDT<	與指定的日期比較	導通時	7.400	11.400	6.800	10.900	6.800	10.900	6.800	10.900
			非導通時	7.400	11.600	6.800	10.900	6.800	10.900	6.800	10.900
		與當前的日期比較	導通時	5.900	10.000	5.500	9.700	5.500	9.700	5.500	9.700
			非導通時	5.900	10.100	5.500	9.700	5.500	9.700	5.500	9.700
	ANDDT<	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
			非導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
		與當前的日期比較	導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
			非導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
	ORDT<	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的日期比較	導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600
			非導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	LDDT>=	與指定的日期比較	導通時	7.400	11.400	6.800	10.900	6.800	10.900	6.800	10.900
			非導通時	7.400	11.600	6.800	10.900	6.800	10.900	6.800	10.900
		與當前的日期比較	導通時	5.900	10.000	5.500	9.700	5.500	9.700	5.500	9.700
			非導通時	5.900	10.100	5.500	9.700	5.500	9.700	5.500	9.700
	ANDDT>=	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
			非導通時	7.200	11.400	6.500	10.700	6.500	10.700	6.500	10.700
		與當前的日期比較	導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
			非導通時	5.700	9.900	5.300	9.300	5.300	9.300	5.300	9.300
	ORDT>=	未執行時			0.008		0.038		0.038		0.038
		與指定的日期比較	導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
			非導通時	7.400	11.500	6.700	10.800	6.700	10.800	6.700	10.800
		與當前的日期比較	導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600
			非導通時	5.900	10.000	5.400	9.600	5.400	9.600	5.400	9.600
	EVAL	小數點形式全 2 位數指定		23.300	30.400	22.800	29.000	22.800	29.000	22.800	29.000
		指數形式全 6 位數指定		23.300	30.500	22.500	29.000	22.500	29.000	22.500	29.000
	ASC Ⓢ Ⓓ n	n=1		5.600	9.000	5.400	8.300	5.400	8.300	5.400	8.300
		n=96		28.700	32.100	25.200	28.400	25.200	28.400	25.200	28.400
	HEX Ⓢ Ⓓ n	n=1		6.000	9.700	5.400	9.000	5.400	9.000	5.400	9.000
		n=96		35.600	39.800	31.300	35.000	31.300	35.000	31.300	35.000
	RIGHT Ⓢ Ⓓ n	n=1		7.600	9.400	6.600	7.300	6.600	7.300	6.600	7.300
		n=96		36.300	40.000	29.200	31.600	29.200	31.600	29.200	31.600
	LEFT Ⓢ Ⓓ n	n=1		6.500	8.900	5.900	8.200	5.900	8.200	5.900	8.200
		n=96		36.200	39.700	29.200	31.500	29.200	31.500	29.200	31.500
	MIDR	---		9.500	12.100	8.100	10.300	8.100	10.300	8.100	10.300
	MIDW	---		10.300	12.000	8.800	10.200	8.800	10.200	8.800	10.200
	INSTR	無一致		19.300	21.800	16.600	18.400	16.600	18.400	16.600	18.400
		有一致	起始	10.300	12.800	9.100	10.900	9.100	10.900	9.100	10.900
			最終	51.100	54.200	42.700	44.900	42.700	44.900	42.700	44.900
	EMOD	---		10.300	11.800	9.600	11.000	9.600	11.000	9.600	11.000
	EREXP	---		19.300	21.000	18.800	20.100	18.800	20.100	18.800	20.100
	STRINS Ⓢ Ⓓ n	Ⓢ=128/Ⓓ=40/n=1		41.100	54.200	35.300	47.600	35.300	47.600	35.300	47.600
		Ⓢ=128/Ⓓ=40/n=48		56.700	81.400	48.600	61.700	48.600	61.700	48.600	61.700
	STRDEL Ⓢ Ⓓ n	Ⓢ=128/Ⓓ=40/n=1		39.000	49.500	34.800	44.600	34.800	44.600	34.800	44.600
		Ⓢ=128/Ⓓ=40/n=48		36.000	45.200	29.200	38.100	29.200	38.100	29.200	38.100
	SIN	單精度		4.500	6.200	4.100	5.700	4.100	5.700	4.100	5.700
	COS	單精度		4.300	6.000	4.000	5.600	4.000	5.600	4.000	5.600
	TAN	單精度		5.100	7.200	5.100	6.700	5.100	6.700	5.100	6.700
	ASIN	單精度		6.100	8.900	5.900	8.500	5.900	8.500	5.900	8.500
	ACOS	單精度		6.800	9.300	6.700	8.900	6.700	8.900	6.700	8.900
	ATAN	單精度		4.000	6.500	3.900	6.000	3.900	6.000	3.900	6.000
	SIND	雙精度		8.800	14.300	8.500	13.800	8.500	13.800	8.500	13.800
	COSD	雙精度		9.300	15.100	8.800	14.600	8.800	14.600	8.800	14.600
	TAND	雙精度		11.200	16.900	10.800	16.500	10.800	16.500	10.800	16.500
	ASIND	雙精度		12.000	17.100	11.600	16.600	11.600	16.600	11.600	16.600
	ACOSD	雙精度		11.700	16.500	11.200	16.200	11.200	16.200	11.200	16.200
	ATAND	雙精度		9.500	14.200	9.100	13.800	9.100	13.800	9.100	13.800
	RAD	單精度		2.500	4.800	2.100	4.300	2.100	4.300	2.100	4.300
	RADD	雙精度		4.000	9.600	3.600	9.200	3.600	9.200	3.600	9.200
	DEG	單精度		2.500	4.700	2.200	4.400	2.200	4.400	2.200	4.400
	DEGD	雙精度		4.300	9.000	3.800	9.000	3.800	9.000	3.800	9.000
	SQR	單精度		3.000	4.600	2.600	4.300	2.600	4.300	2.600	4.300
	SQRD	雙精度		5.600	11.500	5.200	11.000	5.200	11.000	5.200	11.000

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	EXP Ⓢ Ⓓ	單精度	Ⓢ=-10	4.000	6.100	3.800	5.500	3.800	5.500	3.800	5.500
			Ⓢ=1	4.000	6.100	3.800	5.600	3.800	5.600	3.800	5.600
	EXPD Ⓢ Ⓓ	雙精度	Ⓢ=-10	8.700	13.900	8.200	13.500	8.200	13.500	8.200	13.500
			Ⓢ=1	8.400	13.600	8.000	13.200	8.000	13.200	8.000	13.200
	LOG Ⓢ Ⓓ	單精度	Ⓢ=1	4.100	6.900	3.800	6.400	3.800	6.400	3.800	6.400
			Ⓢ=10	5.600	8.200	5.200	7.700	5.200	7.700	5.200	7.700
	LOGD Ⓢ Ⓓ	雙精度	Ⓢ=1	8.100	13.000	7.700	12.500	7.700	12.500	7.700	12.500
			Ⓢ=10	9.700	14.800	9.200	14.300	9.200	14.300	9.200	14.300
	RND	---		1.200	2.300	0.800	1.800	0.800	1.800	0.800	1.800
	SRND	---		1.400	2.400	1.100	2.000	1.100	2.000	1.100	2.000
	BSQR Ⓢ Ⓓ	Ⓢ=0		1.800	3.300	1.600	2.800	1.600	2.800	1.600	2.800
		Ⓢ=9999		5.100	8.800	5.100	8.000	5.100	8.000	5.100	8.000
	BDSQR Ⓢ Ⓓ	Ⓢ=0		1.900	3.400	1.500	3.000	1.500	3.000	1.500	3.000
		Ⓢ=99999999		7.500	10.200	7.500	9.900	7.500	9.900	7.500	9.900
	BSIN	---		8.600	15.100	8.100	14.500	8.100	14.500	8.100	14.500
	BCOS	---		7.800	14.400	7.800	13.700	7.800	13.700	7.800	13.700
	BTAN	---		9.000	13.800	9.000	13.300	9.000	13.300	9.000	13.300
	BASIN	---		10.600	13.400	10.100	12.800	10.100	12.800	10.100	12.800
	BACOS	---		11.600	14.400	11.100	14.100	11.100	14.100	11.100	14.100
	BATAN	---		9.800	11.700	9.100	10.900	9.100	10.900	9.100	10.900
	POW Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=12.3E+5 Ⓢ2=3.45E+0	8.750	11.400	8.400	10.900	8.400	10.900	8.400	10.900
	POWD Ⓢ1 Ⓢ2 Ⓓ	雙精度	Ⓢ1=12.3E+5 Ⓢ2=3.45E+0	18.600	27.200	18.200	26.500	18.200	26.500	18.200	26.500
	LOG10	---		5.900	8.550	5.700	8.050	5.700	8.050	5.700	8.050
	LOG10D	---		11.500	19.400	11.100	18.600	11.100	18.600	11.100	18.600
	LIMIT	---		2.800	3.100	2.400	2.700	2.400	2.700	2.400	2.700
	DLIMIT	---		3.200	3.500	2.800	3.000	2.800	3.000	2.800	3.000
	BAND	---		3.000	4.300	2.700	3.800	2.700	3.800	2.700	3.800
	DBAND	---		3.600	5.100	3.300	4.600	3.300	4.600	3.300	4.600
	ZONE	---		3.000	4.700	2.600	4.300	2.600	4.300	2.600	4.300
	DZONE	---		3.400	5.000	3.000	4.600	3.000	4.600	3.000	4.600
	SCL Ⓢ1 Ⓢ2 Ⓓ	SM750=ON	點No.1<Ⓢ1<點No.2	13.200	23.600	12.300	22.500	12.300	22.500	12.300	22.500
			點No.9<Ⓢ1<點No.10	13.300	23.600	12.600	22.700	12.600	22.700	12.600	22.700
		SM750=OFF	點No.1<Ⓢ1<點No.2	12.000	23.100	11.400	22.200	11.400	22.200	11.400	22.200
			點No.9<Ⓢ1<點No.10	14.100	25.300	12.800	23.900	12.800	23.900	12.800	23.900
	DSCL Ⓢ1 Ⓢ2 Ⓓ	SM750=ON	點No.1<Ⓢ1<點No.2	12.800	23.800	11.900	23.000	11.900	23.000	11.900	23.000
			點No.9<Ⓢ1<點No.10	12.900	23.900	12.100	23.000	12.100	23.000	12.100	23.000
		SM750=OFF	點No.1<Ⓢ1<點No.2	11.500	22.400	10.900	21.500	10.900	21.500	10.900	21.500
			點No.9<Ⓢ1<點No.10	13.800	24.900	12.700	23.600	12.700	23.600	12.700	23.600

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	SCL2 ⓢ1 ⓢ2 ⓓ	SM750=ON	點 No.1<ⓢ1 <點 No.2	12.700	24.200	11.900	23.300	11.900	23.300	11.900	23.300
			點 No.9<ⓢ1 <點 No.10	12.900	24.600	12.100	23.300	12.100	23.300	12.100	23.300
		SM750=OFF	點 No.1<ⓢ1 <點 No.2	12.300	23.400	11.500	22.600	11.500	22.600	11.500	22.600
			點 No.9<ⓢ1 <點 No.10	13.700	25.000	12.600	23.900	12.600	23.900	12.600	23.900
	DSCL2 ⓢ1 ⓢ2 ⓓ	SM750=ON	點 No.1<ⓢ1 <點 No.2	12.600	23.800	11.800	22.900	11.800	22.900	11.800	22.900
			點 No.9<ⓢ1 <點 No.10	13.000	23.900	12.200	22.800	12.200	22.800	12.200	22.800
		SM750=OFF	點 No.1<ⓢ1 <點 No.2	11.500	22.400	11.000	21.400	11.000	21.400	11.000	21.400
			點 No.9<ⓢ1 <點 No.10	13.900	24.900	12.800	23.600	12.800	23.600	12.800	23.600
	RSET	標準 RAM		3.000	6.300	2.700	5.900	2.700	5.900	2.700	5.900
		SRAM 卡		3.000	6.400	2.600	5.800	2.600	5.800	2.600	5.800
	DATE-	無進位		5.800	8.500	4.600	7.000	4.600	7.000	4.600	7.000
		有進位		5.700	7.400	4.600	6.500	4.600	6.500	4.600	6.500
	SECOND	---		2.600	3.900	2.200	3.400	2.200	3.400	2.200	3.400
	HOUR	---		2.900	4.800	2.400	4.300	2.400	4.300	2.400	4.300
	QDRSET	SRAM 卡→標準 RAM		120.000	134.000	115.000	134.000	115.000	134.000	115.000	134.000
		標準 RAM → SRAM 卡		533.000	560.000	520.000	553.000	520.000	553.000	520.000	553.000
	QCDSET	SRAM 卡→標準 ROM		306.000	346.000	305.000	346.000	305.000	346.000	305.000	346.000
		標準 ROM → SRAM 卡		311.000	342.000	300.000	334.000	300.000	334.000	300.000	334.000
	DATERD	---		3.200	5.000	2.500	4.200	2.500	4.200	2.500	4.200
	DATEWR	---		4.900	9.700	4.100	8.900	4.100	8.900	4.100	8.900
	DATE +	無進位		5.100	8.000	4.700	6.600	4.700	6.600	4.700	6.600
		有進位		5.700	8.000	4.600	6.500	4.600	6.500	4.600	6.500
	S.DATERD	---		5.600	8.000	4.800	7.100	4.800	7.100	4.800	7.100
	S.DATE+	無進位		9.100	11.700	7.400	10.000	7.400	11.000	7.400	11.000
		有進位		8.900	11.800	7.400	10.000	7.400	11.000	7.400	11.000
	S.DATE-	無進位		9.000	12.000	7.400	10.300	7.400	10.300	7.400	10.300
		有進位		9.000	12.200	7.500	10.200	7.500	10.200	7.500	10.200
	PSTOP	---		61.400	84.500	56.600	79.800	56.600	79.800	56.600	79.800
	POFF	---		61.800	84.500	57.200	79.800	57.200	79.800	57.200	79.800
	PSCAN	---		64.900	84.700	60.100	79.900	60.100	79.900	60.100	79.900
	WDT	---		1.300	2.700	1.100	2.400	1.100	2.400	1.100	2.400
	DUTY	---		4.900	10.100	4.800	9.600	4.800	9.600	4.800	9.600
	TIMCHK	---		3.800	5.300	3.500	4.700	3.500	4.700	3.500	4.700
	ZRRDB	標準 RAM 的文件寄存器		2.400	2.600	1.800	2.100	1.800	2.100	1.800	2.100
		SRAM 卡的文件寄存器		2.500	2.800	2.000	2.300	2.000	2.300	2.000	2.300
	ZRWRB	標準 RAM 的文件寄存器		3.000	3.300	2.400	2.700	2.400	2.700	2.400	2.700
		SRAM 卡的文件寄存器		3.200	3.600	2.600	3.000	2.600	3.000	2.600	3.000
	ADRSET	---		2.600	3.100	2.100	2.600	2.100	2.600	2.100	2.600
	ZPUSH	---		6.900	9.200	5.800	7.500	5.800	7.500	5.800	7.500
	ZPOP	---		7.500	8.800	5.800	6.400	5.800	6.400	5.800	6.400

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
應用指令	S.ZCOM	安裝 CC-Link 模組時（主站側）		19.600	26.500	19.300	26.000	19.300	26.000	19.300	26.000
		安裝 CC-Link 模組時（本地站側）		19.600	26.500	19.100	26.200	19.100	26.200	19.100	26.200
		·僅選擇 MELSECNET/H 刷新時 ·（管理站側） ·僅選擇 CC-Link IE 控制網路刷新時（管理站側）		53.500	73.500	53.000	72.700	53.000	72.700	53.000	72.700
		·僅選擇 MELSECNET/H 刷新時 ·（普通站側） ·僅選擇 CC-Link IE 控制網路刷新時（普通站側）		29.800	41.200	29.800	40.600	29.800	40.600	29.800	40.600
		僅選擇 CC-Link IE 現場網路刷新時（主站側）		31.500	60.000	31.000	58.000	31.000	58.000	31.000	58.000
		僅選擇 CC-Link IE 現場網路刷新時（本地站側）		31.500	60.000	31.000	58.000	31.000	58.000	31.000	58.000
	S.RTREAD	---		8.200	20.500	7.400	19.000	7.400	19.000	7.400	19.000
	S.RTWRITE	---		8.700	21.500	8.300	19.800	8.300	19.800	8.300	19.800
	UNIRD n1 Ⓓ n2	n2=1		4.000	8.400	3.700	8.000	3.700	8.000	3.700	8.000
		n2=16		12.500	17.000	12.200	16.600	12.200	16.600	12.200	16.600
	TYPERD	---		29.800	53.000	29.500	52.300	29.500	52.300	29.500	52.300
	TRACE	開始		46.600	48.300	43.800	44.700	43.800	44.700	43.800	44.700
	TRACER	---		3.300	6.800	2.600	6.000	2.600	6.000	2.600	6.000
	RBMOV Ⓢ Ⓓ n	使用標準 RAM	1 點	11.300	16.800	9.200	15.100	9.200	15.100	9.200	15.100
			1000 點	120.700	127.100	61.000	68.600	61.000	68.600	61.000	68.600
		使用 SRAM	1 點	11.200	16.700	9.400	15.600	9.400	15.600	9.400	15.600
			1000 點	180.700	187.100	165.000	172.600	165.000	172.600	165.000	172.600
	SP.FWRITE	---		4.100	55.600	3.700	53.700	3.700	53.700	3.700	53.700
	SP.FREAD	---		4.600	54.600	4.000	53.000	4.000	53.000	4.000	53.000
	SP.DEVST	---		4.500	36.500	4.000	34.500	4.000	34.500	4.000	34.500
	S.DEVLD	---		11.000	17.800	10.000	17.000	10.000	17.000	10.000	17.000

分類	指令	條件（軟元件）		處理時間（μs）							
				Q03UD(E)CPU		Q04UD(E)HCPU、Q06UD(E)HCPU		Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU		Q50UDEHCPU、Q100UDEHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
多 CPU 專用指令	S.TO n1 n2 n3 n4 Ⓓ	至本機 CPU 共用記憶體的寫入	n4=1	34.700	34.900	33.500	34.400	33.500	34.400	33.500	34.400
			n4=320	85.900	87.600	75.200	75.500	75.200	75.500	75.200	75.500
	TO n1 n2 Ⓢ n3	至本機 CPU 共用記憶體的寫入	n3=1	4.700	23.800	4.500	23.300	4.500	23.300	4.500	23.300
			n3=320	57.500	76.200	47.100	64.500	47.100	64.500	47.100	64.500
	DTO n1 n2 Ⓢ n3	至本機 CPU 共用記憶體的寫入	n3=1	5.300	23.800	5.800	23.300	5.800	23.300	5.800	23.300
			n3=320	111.300	128.400	91.500	108.500	91.500	108.500	91.500	108.500
	FROM n1 n2 Ⓓ n3	本機 CPU 共用記憶體讀取	n3=1	5.000	23.800	4.300	23.300	4.300	23.300	4.300	23.300
			n3=320	51.400	65.600	44.400	60.700	44.400	60.700	44.400	60.700
		其他機號 CPU 共用記憶體讀取	n3=1	11.600	17.700	10.600	13.900	10.600	13.900	10.600	13.900
			n3=320	142.000	160.000	142.000	149.000	142.000	149.000	142.000	149.000
			n3=1000	431.000	463.000	422.000	448.000	422.000	448.000	422.000	448.000
	DFRO n1 n2 Ⓓ n3	本機 CPU 共用記憶體讀取	n3=1	6.600	23.800	5.600	23.300	5.600	23.300	5.600	23.300
			n3=320	96.400	113.200	83.600	100.800	83.600	100.800	83.600	100.800
		其他機號 CPU 共用記憶體讀取	n3=1	12.900	20.800	12.200	17.100	12.200	17.100	12.200	17.100
			n3=320	277.000	299.000	274.000	291.000	274.000	291.000	274.000	291.000
			n3=1000	838.000	860.000	835.000	857.000	835.000	857.000	835.000	857.000
多 CPU 高速通信專用指令	D.DDWR n Ⓢ1 Ⓢ2 Ⓓ1 Ⓓ2	至其他機號 CPU 的軟元件寫入	值 =1	82.000	133.000	80.000	130.000	80.000	130.000	80.000	130.000
			值 =16	91.000	142.000	89.000	139.000	89.000	139.000	89.000	139.000
			n=96*1	136.000	189.000	132.000	182.000	132.000	182.000	132.000	182.000
	DP.DDWR n Ⓢ1 Ⓢ2 Ⓓ1 Ⓓ2		n=1	82.000	133.000	80.000	130.000	80.000	130.000	80.000	130.000
			n=16	91.000	142.000	89.000	139.000	89.000	139.000	89.000	139.000
			n=96*1	136.000	189.000	132.000	182.000	132.000	182.000	132.000	182.000
	D.DDRD n Ⓢ1 Ⓢ2 Ⓓ1 Ⓓ2	至其他機號 CPU 的軟元件讀取	n=1	82.000	133.000	80.000	130.000	80.000	130.000	80.000	130.000
			n=16	82.000	133.000	80.000	130.000	80.000	130.000	80.000	130.000
			n=96*1	82.000	133.000	80.000	130.000	80.000	130.000	80.000	130.000
	DP.DDRD n Ⓢ1 Ⓢ2 Ⓓ1 Ⓓ2		n=1	82.000	133.000	80.000	130.000	80.000	130.000	80.000	130.000
			n=16	82.000	133.000	80.000	130.000	80.000	130.000	80.000	130.000
			n=96*1	82.000	133.000	80.000	130.000	80.000	130.000	80.000	130.000

*1: 序列號的前 5 位數爲 “ 10012 ” 以後的 Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU、Q13UDHCPU、Q26UDHCPU 可以使用。

備 註

對於表中未記述上升沿執行指令 (□ P) 的指令，其處理時間與 ON 時執行的指令的處理時間相同。

例 WORDP 指令、TOP 指令等。

(2) 使用了文件寄存器、擴展資料寄存器、擴展鏈結寄存器、模組訪問軟元件、鏈結直接軟元件時的加法運算時間一覽表

(a) 使用 Q00UJCPU、Q00UCPU、Q01UCPU、Q02UCPU 時

軟元件名		資料	軟元件指定位置	加法運算時間（μs）			
				Q00UJCPU	Q00UCPU	Q01UCPU	Q02UCPU
文件寄存器 (R)	使用標準 RAM 時	位	源	0.100	0.100	0.100	0.100
			目標	0.100	0.100	0.100	0.100
		字	源	0.100	0.100	0.100	0.100
			目標	0.100	0.100	0.100	0.100
		雙字	源	0.100	0.100	0.100	0.200
			目標	0.100	0.100	0.100	0.200
	使用 SRAM 卡 (Q2MEM-1MBS、Q2MEM-2MBS) 時	位	源	---	---	---	0.220
			目標	---	---	---	0.180
		字	源	---	---	---	0.220
			目標	---	---	---	0.180
		雙字	源	---	---	---	0.440
			目標	---	---	---	0.380
	使用 SRAM 卡 (Q3MEM-4MBS、Q3MEM-8MBS) 時	位	源	---	---	---	0.160
			目標	---	---	---	0.140
		字	源	---	---	---	0.160
			目標	---	---	---	0.140
		雙字	源	---	---	---	0.320
			目標	---	---	---	0.300
文件寄存器 (ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	使用標準 RAM 時	位	源	0.120	0.120	0.120	0.120
			目標	0.120	0.120	0.120	0.120
		字	源	0.120	0.120	0.120	0.120
			目標	0.120	0.120	0.120	0.120
		雙字	源	0.120	0.120	0.120	0.220
			目標	0.120	0.120	0.120	0.220
	使用 SRAM 卡 (Q2MEM-1MBS、Q2MEM-2MBS) 時	位	源	---	---	---	0.240
			目標	---	---	---	0.200
		字	源	---	---	---	0.240
			目標	---	---	---	0.200
		雙字	源	---	---	---	0.460
			目標	---	---	---	0.400
	使用 SRAM 卡 (Q3MEM-4MBS、Q3MEM-8MBS) 時	位	源	---	---	---	0.180
			目標	---	---	---	0.160
		字	源	---	---	---	0.180
			目標	---	---	---	0.160
		雙字	源	---	---	---	0.340
			目標	---	---	---	0.320
模組訪問軟元件 (Un\G □、U3En\G0 ～ G4095)		位	源	---	---	---	12.000
			目標	---	---	---	17.300
		字	源	---	---	---	9.700
			目標	---	---	---	33.000
		雙字	源	---	---	---	24.200
			目標	---	---	---	34.800
鏈結直接軟元件 (Jn\ □)		位	源	70.900	70.900	70.900	46.200
			目標	120.100	120.100	120.100	75.000
		字	源	68.400	68.400	68.400	44.800
			目標	53.700	53.700	53.700	33.600
		雙字	源	75.600	75.600	75.600	60.300
			目標	58.900	58.900	58.900	41.900

(b) 使用 Q03UD(E)CPU、Q04UD(E)HCPU、Q06UD(E)HCPU、Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU、Q50UDEHCPU、Q100UDEHCPU 時

軟元件名		資料	軟元件指定位置	加法運算時間（μs）			
				Q03UD(E)CPU	Q04UD(E)HCPU、Q06UD(E)HCPU	Q10UD(E)HCPU、Q13UD(E)HCPU、Q20UD(E)HCPU、Q26UD(E)HCPU	Q50UDEHCPU、Q100UDEHCPU
文件寄存器 (R)	使用標準 RAM 時	位	源	0.100	0.048	0.048	0.048
			目標	0.100	0.038	0.038	0.038
		字	源	0.100	0.048	0.048	0.048
			目標	0.100	0.038	0.038	0.038
		雙字	源	0.200	0.095	0.095	0.095
			目標	0.200	0.086	0.086	0.086
	使用 SRAM 卡 (Q2MEM-1MBS、Q2MEM-2MBS) 時	位	源	0.220	0.200	0.200	0.200
			目標	0.180	0.162	0.162	0.162
		字	源	0.220	0.200	0.200	0.200
			目標	0.180	0.162	0.162	0.162
		雙字	源	0.440	0.399	0.399	0.399
			目標	0.380	0.361	0.361	0.361
	使用 SRAM 卡 (Q3MEM-4MBS、Q3MEM-8MBS) 時	位	源	0.160	0.152	0.152	0.152
			目標	0.140	0.133	0.133	0.133
		字	源	0.160	0.152	0.152	0.152
			目標	0.140	0.133	0.133	0.133
		雙字	源	0.320	0.304	0.304	0.304
			目標	0.300	0.295	0.295	0.295
文件寄存器 (ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	使用標準 RAM 時	位	源	0.120	0.057	0.057	0.057
			目標	0.120	0.048	0.048	0.048
		字	源	0.120	0.057	0.057	0.057
			目標	0.120	0.048	0.048	0.048
		雙字	源	0.220	0.105	0.105	0.105
			目標	0.220	0.095	0.095	0.095
	使用 SRAM 卡 (Q2MEM-1MBS、Q2MEM-2MBS) 時	位	源	0.240	0.209	0.209	0.209
			目標	0.200	0.171	0.171	0.171
		字	源	0.240	0.209	0.209	0.209
			目標	0.200	0.171	0.171	0.171
		雙字	源	0.460	0.409	0.409	0.409
			目標	0.400	0.371	0.371	0.371
	使用 SRAM 卡 (Q3MEM-4MBS、Q3MEM-8MBS) 時	位	源	0.180	0.162	0.162	0.162
			目標	0.160	0.143	0.143	0.143
		字	源	0.180	0.162	0.162	0.162
			目標	0.160	0.143	0.143	0.143
		雙字	源	0.340	0.314	0.314	0.314
			目標	0.320	0.304	0.304	0.304
模組訪問軟元件 (Un\G □、U3En\G0 ～ G4095)		位	源	11.700	11.200	11.200	11.200
			目標	15.400	15.300	15.300	15.300
		字	源	9.460	9.410	9.410	9.410
			目標	19.000	19.000	19.000	19.000
		雙字	源	11.000	10.900	10.900	10.900
			目標	18.800	18.700	18.700	18.700
鏈結直接軟元件 (Jn\ □)		位	源	32.700	31.300	31.300	31.300
			目標	52.300	51.800	51.800	51.800
		字	源	30.600	30.100	30.100	30.100
			目標	28.900	28.400	28.400	28.400
		雙字	源	38.900	38.400	38.400	38.400
			目標	34.800	34.300	34.300	34.300

附錄 1.5 LCPU 的運算處理時間

各指令的運算處理時間如本頁以後的表中所示。

由於運算處理時間根據源、目標的內容而有所不同，因此表中的值只應作爲處理時間的大致參考。

附錄 1.5.1 子集指令處理時間

子集處理指令的處理時間如下所示。

要點

1. 指令中使用的軟元件滿足子集處理的軟元件條件之一時的子集指令的指令處理時間一覽表如下述 (1) 所示。
   (關於子集處理的軟元件條件，請參閱 98 頁 3.5.1 項。)
2. 使用文件寄存器 (R、ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W) 時，請參閱 (2) 的加法運算時間，對各指令的處理時間進行加法運算。
3. 在 OUT/SET/RST 指令中使用 F、T(ST)、C 軟元件時，請參閱 (3) 中列出的加法運算時間，對各指令的處理時間進行加法運算。
4. 由於高速緩衝記憶體功能的影響，各指令的處理時間不恒定，因此對最小值和最大值進行了記載。

(1) 子集指令的指令處理時間一覽表

(a) 使用 L02CPU、L26CPU-BT、L02CPU-P、L26CPU-PBT 時

分類	指令	條件 (軟元件)		處理時間 (μs)			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
順控程式指令	LD LDI AND ANI OR ORI LDP LDF ANDP ANDF ORP ORF	執行時		0.040		0.0095	
	LDPI LDFI	執行時		0.120		0.0285	
	ANDPI ANDFI ORPI ORFI	執行時		0.160		0.038	
	OUT	無變化時		0.040		0.0095	
		變化時					
	OUTH	無變化時		0.040		0.0095	
		變化時					
	SET RST	未執行時		0.040		0.0095	
		執行時	無變化時				
			變化時				
基本指令	LD=	導通時		0.120		0.0285	
		非導通時					
	AND=	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	OR=	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	LD<>	導通時		0.120		0.0285	
		非導通時					
	AND<>	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
基本指令	OR<>	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	LD>	導通時		0.120		0.0285	
		非導通時					
	AND>	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	OR>	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	LD<=	導通時		0.120		0.0285	
		非導通時					
	AND<=	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	OR<=	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	LD<	導通時		0.120		0.0285	
		非導通時					
	AND<	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	OR<	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	LD>=	導通時		0.120		0.0285	
		非導通時					
	AND>=	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	OR>=	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	LDD=	導通時		0.120		0.0285	
		非導通時					
	ANDD=	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	ORD=	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	LDD<>	導通時		0.120		0.0285	
		非導通時					
	ANDD<>	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	ORD<>	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	LDD>	導通時		0.120		0.0285	
		非導通時					
	ANDD>	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	ORD>	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				
	LDD<=	導通時		0.120		0.0285	
		非導通時					
	ANDD<=	未執行時		0.120		0.0285	
		執行時	導通時				
			非導通時				

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
基本指令	ORD<=	未執行時					
		執行時	導通時		0.120		0.0285
			非導通時				
	LDD<	導通時			0.120		0.0285
		非導通時					
	ANDD<	未執行時					
		執行時	導通時		0.120		0.0285
			非導通時				
	ORD<	未執行時					
		執行時	導通時		0.120		0.0285
			非導通時				
	LDD>=	導通時			0.120		0.0285
		非導通時					
	ANDD>=	未執行時					
		執行時	導通時		0.120		0.0285
			非導通時				
	ORD>=	未執行時					
		執行時	導通時		0.120		0.0285
			非導通時				
	+ Ⓢ Ⓓ	執行時			0.120		0.0285
	+ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.160		0.038
	- Ⓢ Ⓓ	執行時			0.120		0.0285
	- Ⓢ1 Ⓢ2 Ⓓ	執行時			0.160		0.038
	D+ Ⓢ Ⓓ	執行時			0.120		0.0285
	D+ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.160		0.038
	D- Ⓢ Ⓓ	執行時			0.120		0.0285
	D- Ⓢ1 Ⓢ2 Ⓓ	執行時			0.160		0.038
	* Ⓢ1 Ⓢ2 Ⓓ	執行時			0.180		0.057
	/ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.280		0.105
	D* Ⓢ1 Ⓢ2 Ⓓ	執行時			0.260		0.095
	D/ Ⓢ1 Ⓢ2 Ⓓ	執行時			0.400		0.162
	B+ Ⓢ Ⓓ	執行時		3.100	6.800	2.900	4.100
	B+ Ⓢ1 Ⓢ2 Ⓓ	執行時		4.800	8.900	4.200	5.900
	B- Ⓢ Ⓓ	執行時		3.100	6.800	2.900	4.100
	B- Ⓢ1 Ⓢ2 Ⓓ	執行時		4.800	8.900	4.200	4.600
	B* Ⓢ1 Ⓢ2 Ⓓ	執行時		3.900	7.400	3.400	4.800
	B/ Ⓢ1 Ⓢ2 Ⓓ	執行時		3.900	8.500	3.700	5.200
	E+ Ⓢ Ⓓ	單精度	Ⓢ=0、Ⓓ=0		0.180		0.057
			Ⓢ=2^{127}、Ⓓ=2^{127}		0.180		0.057
	E+ Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0		0.220		0.0665
			Ⓢ1=2^{127}、Ⓢ2=2^{127}		0.220		0.0665
	E- Ⓢ Ⓓ	單精度	Ⓢ=0、Ⓓ=0		0.180		0.057
			Ⓢ=2^{127}、Ⓓ=2^{127}		0.180		0.057
	E- Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0		0.220		0.0665
			Ⓢ1=2^{127}、Ⓢ2=2^{127}		0.220		0.0665
	E* Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=0、Ⓢ2=0		0.180		0.057
			Ⓢ1=2^{127}、Ⓢ2=2^{127}		0.180		0.057
	E/ Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=2^{127}、Ⓢ2=2^{127}	3.900	8.500		0.285

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
基本指令	INC	執行時			0.080		0.019
	DINC	執行時			0.080		0.019
	DEC	執行時			0.080		0.019
	DDEC	執行時			0.080		0.019
	BCD	執行時			0.160		0.057
	DBCD	執行時			0.240		0.095
	BIN	執行時			0.100		0.0285
	DBIN	執行時			0.100		0.0285
	FLT	單精度	Ⓢ=0		0.100		0.0475
			Ⓢ=$7FFF_H$		0.140		0.0475
	DFLT	單精度	Ⓢ=0		0.140		0.0475
			Ⓢ=$7FFFFFFF_H$		0.140		0.0475
	INT	單精度	Ⓢ=0		0.140		0.0475
			Ⓢ=32766.5		0.140		0.0475
	DINT	單精度	Ⓢ=0		0.140		0.0475
			Ⓢ=1234567890.3		0.140		0.0475
	MOV	---			0.080		0.019
	DMOV	---			0.080		0.019
	EMOV	---			0.080		0.019
	CML	---			0.080		0.019
	DCML	---			0.080		0.019
	BMOV	SM237=ON	n=1	3.600	4.100	2.900	3.200
			n=96	4.500	4.700	3.400	3.700
		SM237=OFF	n=1	5.000	7.400	4.200	5.500
			n=96	6.000	7.900	4.700	6.000
	FMOV	SM237=ON	n=1	5.900	6.800	2.800	3.200
			n=96	6.300	11.000	3.000	5.200
		SM237=OFF	n=1	7.000	8.000	3.400	3.800
			n=96	5.200	6.900	3.600	5.800
	XCH	---		2.100	4.100	1.800	2.300
	DXCH	---		2.200	4.200	2.100	2.900
	DFMOV	SM237=ON	n=1	2.000	3.200	1.750	1.750
			n=96	5.600	6.100	3.650	4.150
		SM237=OFF	n=1	2.900	4.600	2.250	3.150
			n=96	6.100	8.200	4.200	5.500
	CJ	---		2.100	2.900	1.100	2.400
	SCJ	---		2.100	2.900	1.100	2.400
	JMP	---		2.100	2.900	1.100	2.400

分類	指令	條件（軟元件）	處理時間（μs）			
			L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
			最小值	最大值	最小值	最大值
應用指令	WAND Ⓢ Ⓓ	執行時		0.120		0.0285
	WAND Ⓢ1 Ⓢ2 Ⓓ	執行時		0.160		0.038
	DAND Ⓢ Ⓓ	執行時		0.120		0.0285
	DAND Ⓢ1 Ⓢ2 Ⓓ	執行時		0.160		0.038
	WOR Ⓢ Ⓓ	執行時		0.120		0.0285
	WOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.160		0.038
	DOR Ⓢ Ⓓ	執行時		0.120		0.0285
	DOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.160		0.038
	WXOR Ⓢ Ⓓ	執行時		0.120		0.0285
	WXOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.160		0.038
	DXOR Ⓢ Ⓓ	執行時		0.120		0.0285
	DXOR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.160		0.038
	WXNR Ⓢ Ⓓ	執行時		0.120		0.0285
	WXNR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.160		0.038
	DXNR Ⓢ Ⓓ	執行時		0.120		0.0285
	DXNR Ⓢ1 Ⓢ2 Ⓓ	執行時		0.160		0.038
	ROR Ⓓ n	n=1	2.200	4.900	1.700	2.500
		n=15	2.200	4.900	1.700	2.500
	RCR Ⓓ n	n=1	2.100	4.800	1.700	3.200
		n=15	2.100	4.800	1.700	3.200
	ROL Ⓓ n	n=1	2.100	4.800	1.800	3.200
		n=15	2.100	4.800	1.800	3.200
	RCL Ⓓ n	n=1	2.100	5.200	1.800	2.200
		n=15	2.100	5.200	1.800	2.200
	DROR Ⓓ n	n=1	2.200	5.200	1.900	2.700
		n=31	2.200	5.200	1.900	2.700
	DRCR Ⓓ n	n=1	2.200	5.900	1.900	4.200
		n=31	2.200	5.900	1.900	4.200
	DROL Ⓓ n	n=1	2.200	4.900	1.800	3.300
		n=31	2.200	4.900	1.800	3.300
	DRCL Ⓓ n	n=1	2.200	5.900	1.900	3.800
		n=31	2.200	5.900	1.900	3.800
	SFR Ⓓ n	n=1	2.200	4.600	1.700	2.600
		n=15	2.200	4.600	1.700	2.600
	SFL Ⓓ n	n=1	2.200	4.600	1.800	2.700
		n=15	2.200	4.600	1.800	2.700
	DSFR Ⓓ n	n=1	2.200	6.100	2.200	4.300
		n=96	33.400	38.100	23.900	26.100
	DSFL Ⓓ n	n=1	2.200	6.100	2.100	4.000
		n=96	33.500	38.000	23.700	25.800
	SUM	Ⓢ=0	3.000	4.800	2.900	3.600
		Ⓢ=FFFF$_H$	3.000	4.900	2.900	3.600
	SEG	執行時	1.700	3.600	1.500	2.100
	FOR	---	1.300	3.200	0.870	2.100
	CALL Pn	文件內指標	2.600	4.000	2.300	3.600
		公共指標	4.600	13.500	3.200	4.900
	CALL Pn Ⓢ1 ～ Ⓢ5	---	31.200	36.000	26.100	29.300

備註

對於表中未記述的上升沿執行指令 (□ P) 的指令，其處理時間與 ON 時執行的指令的處理時間相同。

例 MOVP 指令、WANDP 指令等。

(2) 使用了文件寄存器、擴展資料寄存器、擴展鏈結寄存器時的加法運算時間一覽表

(a) 使用 L02CPU、L26CPU-BT、L02CPU-P、L26CPU-PBT 時

軟元件名		資料	軟元件指定位置	加法運算時間 (μs)	
				L02CPU、L02CPU-P	L26CPU-BT、L26CPU-BT
文件寄存器 (R)	使用標準 RAM 時	位	源	0.100	0.048
			目標	0.220	0.038
		字	源	0.100	0.048
			目標	0.100	0.038
		雙字	源	0.200	0.095
			目標	0.200	0.086
文件寄存器 (ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)	使用標準 RAM 時	位	源	0.140	0.057
			目標	0.280	0.048
		字	源	0.140	0.057
			目標	0.140	0.048
		雙字	源	0.240	0.105
			目標	0.240	0.095

(3) 在 OUT/SET/RST 指令的軟元件中，使用了 F、T(ST)、C 軟元件時的加法運算時間一覽表

(a) 使用 L02CPU、L26CPU-BT、L02CPU-P、L26CPU-PBT 時

指令名	軟元件名	條件		加法運算時間 (μs)	
				L02CPU、L02CPU-P	L26CPU-BT、L26CPU-BT
OUT	F	未執行時		2.000	1.570
		執行時	顯示時	53.100	38.090
			顯示結束	53.000	37.980
	T(ST)、C	未執行時		0.120	0.030
		執行時	時間到後	0.120	0.030
			計數時	0.120	0.030
SET	F	未執行時		0.040	0.010
		執行時	顯示時	52.000	40.600
			顯示結束	43.600	37.900
RST	F	未執行時		0.040	0.010
		執行時	顯示時	45.700	36.600
			顯示結束	19.000	16.190
	T(ST)、C	未執行時		0.120	0.030
		執行時		0.120	0.030

附錄 1.5.2 子集指令以外的指令處理時間一覽表

子集處理指令以外的處理時間如下所示。

要點

- 指令中使用的軟元件未滿足子集處理的軟元件條件時的子集指令以外的指令處理時間一覽表如下述 (1) 所示。(關於子集處理的軟元件條件，請參閱 98 頁 3.5.1 項。)
  關於下表中未記載的指令，請參閱 772 頁附錄 1.5.1(2) 的指令處理時間。
- 使用文件寄存器 (R、ZR)、擴展資料寄存器 (D)、擴展鏈結寄存器 (W)、模組訪問軟元件 (Un\G □)、鏈結直接軟元件 (Jn\ □) 時，請參閱 (2) 的加法運算時間，對各指令的處理時間進行加法運算。
- 由於高速緩衝記憶體功能的影響，各指令的處理時間不恒定，因此對最小值和最大值進行了記載。

(1) 子集指令以外的指令處理時間一覽表

(a) 使用 L02CPU、L26CPU-BT、L02CPU-P、L26CPU-PBT 時

分類	指令	條件（軟元件）			處理時間（μs）			
					L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
					最小值	最大值	最小值	最大值
順控程式指令	ANB ORB MPS MRD MPP	---				0.040		0.0095
	INV	未執行時				0.040		0.0095
		執行時						
	MEP MEF	未執行時				0.040		0.0095
		執行時						
	EGP EGF	未執行時				0.040		0.0095
		執行時						
	PLS	---			1.600	1.700	0.890	1.200
	PLF	---			1.600	1.700	0.890	1.200
	FF	未執行時				0.080		0.0185
		執行時			1.500	1.500	0.790	0.910
	DELTA	未執行時				0.080		0.0185
		執行時			2.700	6.800	2.400	3.200
	SFT	未執行時				0.080		0.0185
		執行時			1.700	4.300	1.100	2.700
	MC	---				0.080		0.0185
	MCR	---				0.040		0.0185
	FEND	進行出錯檢查			170.000	210.000	130.000	170.000
	END	不進行出錯檢查			170.000	210.000	130.000	170.000
	STOP	---				---		---
	NOP NOPLF PAGE	---				0.040		0.0095
基本指令	LDE=	單精度	導通時		3.900	10.000		0.0285
			非導通時		3.900	10.000		0.0285
	ANDE=	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.400	9.300		0.0285
				非導通時	3.400	9.300		0.0285
	ORE=	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.500	8.500		0.0285
				非導通時	3.500	8.500		0.0285
	LDE< >	單精度	導通時		3.900	10.000		0.0285
			非導通時		3.900	10.000		0.0285
	ANDE< >	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.400	9.300		0.0285
				非導通時	3.400	9.300		0.0285
	ORE< >	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.500	8.500		0.0285
				非導通時	3.500	8.500		0.0285
	LDE>	單精度	導通時		3.900	10.000		0.0285
			非導通時		3.900	10.000		0.0285
	ANDE>	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.400	9.300		0.0285
				非導通時	3.400	9.300		0.0285
	ORE>	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.500	8.500		0.0285
				非導通時	3.500	8.500		0.0285
	LDE<=	單精度	導通時		3.900	10.000		0.0285
			非導通時		3.900	10.000		0.0285

分類	指令	條件（軟元件）			處理時間（μs）			
					L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
					最小值	最大值	最小值	最大值
基本指令	ANDE<=	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.400	9.300		0.0285
				非導通時	3.400	9.300		0.0285
	ORE<=	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.500	8.500		0.0285
				非導通時	3.500	8.500		0.0285
	LDE<	單精度	導通時		3.900	10.000		0.0285
			非導通時		3.900	10.000		0.0285
	ANDE<	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.400	9.300		0.0285
				非導通時	3.400	9.300		0.0285
	ORE<	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.500	8.500		0.0285
				非導通時	3.500	8.500		0.0285
	LDE>=	單精度	導通時		3.900	10.000		0.0285
			非導通時		3.900	10.000		0.0285
	ANDE>=	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.400	9.300		0.0285
				非導通時	3.400	9.300		0.0285
	ORE>=	單精度	未執行時			0.120		0.0285
			執行時	導通時	3.500	8.500		0.0285
				非導通時	3.500	8.500		0.0285
	LDED=	雙精度	導通時		4.800	16.000	3.500	9.000
			非導通時		4.800	16.000	3.500	9.000
	ANDED=	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.400	15.100	3.200	7.500
				非導通時	4.400	15.100	3.200	7.500
	ORED=	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.500	14.900	3.400	9.200
				非導通時	4.500	14.900	3.400	9.200
	LDED< >	雙精度	導通時		4.800	16.000	3.500	9.000
			非導通時		4.800	16.000	3.500	9.000
	ANDED< >	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.400	15.100	3.200	7.500
				非導通時	4.400	15.100	3.200	7.500
	ORED< >	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.500	14.900	3.400	9.200
				非導通時	4.500	14.900	3.400	9.200
	LDED>	雙精度	導通時		4.800	16.000	3.500	9.000
			非導通時		4.800	16.000	3.500	9.000
	ANDED>	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.400	15.100	3.200	7.500
				非導通時	4.400	15.100	3.200	7.500
	ORED>	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.500	14.900	3.400	9.200
				非導通時	4.500	14.900	3.400	9.200
	LDED<=	雙精度	導通時		4.800	16.000	3.500	9.000
			非導通時		4.800	16.000	3.500	9.000
	ANDED<=	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.400	15.100	3.200	7.500
				非導通時	4.400	15.100	3.200	7.500
	ORED<=	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.500	14.900	3.400	9.200
				非導通時	4.500	14.900	3.400	9.200
	LDED<	雙精度	導通時		4.800	16.000	3.500	9.000
			非導通時		4.800	16.000	3.500	9.000
	ANDED<	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.400	15.100	3.200	7.500
				非導通時	4.400	15.100	3.200	7.500

分類	指令	條件（軟元件）			處理時間（μs）			
					L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
					最小值	最大值	最小值	最大值
基本指令	ORED<	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.500	14.900	3.400	9.200
				非導通時	4.500	14.900	3.400	9.200
	LDED>=	雙精度	導通時		4.800	16.000	3.500	9.000
			非導通時		4.800	16.000	3.500	9.000
	ANDED>=	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.400	15.100	3.200	7.500
				非導通時	4.400	15.100	3.200	7.500
	ORED>=	雙精度	未執行時			0.120		0.0285
			執行時	導通時	4.500	14.900	3.400	9.200
				非導通時	4.500	14.900	3.400	9.200
	LD$=	導通時			5.600	17.100	4.200	8.200
		非導通時			5.600	17.100	4.200	8.200
	AND$=	未執行時				0.120		0.0285
		執行時		導通時	5.300	16.400	3.900	7.300
				非導通時	5.300	16.400	3.900	7.300
	OR$=	未執行時				0.120		0.0285
		執行時		導通時	5.200	15.700	4.000	7.600
				非導通時	5.200	15.700	4.000	7.600
	LD$< >	導通時			5.600	17.100	4.200	8.200
		非導通時			5.600	17.100	4.200	8.200
	AND$< >	未執行時				0.120		0.0285
		執行時		導通時	5.300	16.400	3.900	7.300
				非導通時	5.300	16.400	3.900	7.300
	OR$< >	未執行時				0.120		0.0285
		執行時		導通時	5.200	15.700	4.000	7.600
				非導通時	5.200	15.700	4.000	7.600
	LD$>	導通時			5.600	17.100	4.200	8.200
		非導通時			5.600	17.100	4.200	8.200
	AND$>	未執行時				0.120		0.0285
		執行時		導通時	5.300	16.400	3.900	7.300
				非導通時	5.300	16.400	3.900	7.300
	OR$>	未執行時				0.120		0.0285
		執行時		導通時	5.200	15.700	4.000	7.600
				非導通時	5.200	15.700	4.000	7.600
	LD$<=	導通時			5.600	17.100	4.200	8.200
		非導通時			5.600	17.100	4.200	8.200
	AND$<=	未執行時				0.120		0.0285
		執行時		導通時	5.300	16.400	3.900	7.300
				非導通時	5.300	16.400	3.900	7.300
	OR$<=	未執行時				0.120		0.0285
		執行時		導通時	5.200	15.700	4.000	7.600
				非導通時	5.200	15.700	4.000	7.600
	LD$<	導通時			5.600	17.100	4.200	8.200
		非導通時			5.600	17.100	4.200	8.200
	AND$<	未執行時				0.120		0.0285
		執行時		導通時	5.300	16.400	3.900	7.300
				非導通時	5.300	16.400	3.900	7.300
	OR$<	未執行時				0.120		0.0285
		執行時		導通時	5.200	15.700	4.000	7.600
				非導通時	5.200	15.700	4.000	7.600
	LD$>=	導通時			5.600	17.100	4.200	8.200
		非導通時			5.600	17.100	4.200	8.200
	AND$>=	未執行時				0.120		0.0285
		執行時		導通時	5.300	16.400	3.900	7.300
				非導通時	5.300	16.400	3.900	7.300
	OR$>=	未執行時				0.120		0.0285
		執行時		導通時	5.200	15.700	4.000	7.600
				非導通時	5.200	15.700	4.000	7.600

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
基本指令	BKCMP= (S1) (S2) (D) n	n=1		9.200	15.600	7.500	10.100
		n=96		60.700	69.100	45.600	50.500
	BKCMP< > (S1) (S2) (D) n	n=1		9.200	15.600	7.500	10.100
		n=96		60.700	69.100	45.600	50.500
	BKCMP> (S1) (S2) (D) n	n=1		9.200	15.600	7.500	10.100
		n=96		60.700	69.100	45.600	50.500
	BKCMP<= (S1) (S2) (D) n	n=1		9.200	15.600	7.500	10.100
		n=96		60.700	69.100	45.600	50.500
	BKCMP< (S1) (S2) (D) n	n=1		9.200	15.600	7.500	10.100
		n=96		60.700	69.100	45.600	50.500
	BKCMP>= (S1) (S2) (D) n	n=1		9.200	15.600	7.500	10.100
		n=96		60.700	69.100	45.600	50.500
	DBKCMP= (S1) (S2) (D) n	n=1		9.700	16.400	8.600	13.000
		n=96		61.200	69.900	47.900	52.800
	DBKCMP< > (S1) (S2) (D) n	n=1		9.700	16.400	8.600	13.000
		n=96		61.200	69.900	47.900	52.800
	DBKCMP> (S1) (S2) (D) n	n=1		9.700	16.400	8.600	13.000
		n=96		61.200	69.900	47.900	52.800
	DBKCMP<= (S1) (S2) (D) n	n=1		9.700	16.400	8.600	13.000
		n=96		61.200	69.900	47.900	52.800
	DBKCMP< (S1) (S2) (D) n	n=1		9.700	16.400	8.600	13.000
		n=96		61.200	69.900	47.900	52.800
	DBKCMP>= (S1) (S2) (D) n	n=1		9.700	16.400	8.600	13.000
		n=96		61.200	69.900	47.900	52.800
	DB+ (S) (D)	執行時		4.800	8.400	4.600	6.400
	DB+ (S1) (S2) (D)	執行時		5.100	8.700	4.800	6.700
	DB- (S) (D)	執行時		4.800	8.400	4.600	6.400
	DB- (S1) (S2) (D)	執行時		5.100	8.700	4.800	6.700
	DB* (S1) (S2) (D)	執行時		8.700	18.900	8.100	11.600
	DB/ (S1) (S2) (D)	執行時		6.100	9.100	5.800	8.800
	ED+ (S) (D)	雙精度	(S)=0、(D)=0	4.800	8.000	4.300	7.200
			(S)=2^{1023}、(D)=2^{1023}	5.400	14.900	4.300	7.200
	ED+ (S1) (S2) (D)	雙精度	(S1)=0、(S2)=0	5.500	9.800	4.800	9.200
			(S1)=2^{1023}、(S2)=2^{1023}	6.100	17.800	4.800	9.200
	ED- (S) (D)	雙精度	(S)=0、(D)=0	4.400	10.800	4.400	7.500
			(S)=2^{1023}、(D)=2^{1023}	5.400	15.500	4.400	7.500
	ED- (S1) (S2) (D)	雙精度	(S1)=0、(S2)=0	4.700	13.900	3.800	7.500
			(S1)=2^{1023}、(S2)=2^{1023}	5.700	17.200	3.800	7.500
	ED* (S1) (S2) (D)	雙精度	(S1)=0、(S2)=0	5.800	9.500	5.100	8.800
			(S1)=2^{1023}、(S2)=2^{1023}	5.900	17.600	5.100	8.800
	ED/ (S1) (S2) (D)	雙精度	(S1)=2^{1023}、(S2)=2^{1023}	7.300	18.700	5.900	10.000
	BK+ (S1) (S2) (D) n	n=1		9.100	11.200	8.500	10.600
		n=96		60.500	66.200	44.600	47.900
	BK- (S1) (S2) (D) n	n=1		9.700	12.000	8.900	11.300
		n=96		60.500	66.200	44.600	47.900
	DBK+ (S1) (S2) (D) n	n=1		7.500	12.400	6.450	9.950
		n=96		59.900	65.200	43.700	47.500
	DBK- (S1) (S2) (D) n	n=1		7.500	12.400	6.450	9.950
		n=96		59.900	65.200	43.700	47.500
	$+ (S) (D)	---		11.200	24.700	8.100	13.900
	$+ (S1) (S2) (D)	---		7.900	16.600	6.500	10.300
	FLTD	雙精度	(S)=0	2.800	9.400	1.800	4.700
			(S)=$7FFF_H$	3.300	9.600	2.200	4.800

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
基本指令	DFLTD	雙精度	Ⓢ=0	2.900	9.100	2.000	4.900
			Ⓢ=7FFFFFFF$_H$	3.400	9.300	2.300	5.100
	INTD	雙精度	Ⓢ=0	3.500	8.700	2.200	4.100
			Ⓢ=32766.5	4.100	12.900	3.200	5.600
	DINTD	雙精度	Ⓢ=0	3.200	9.500	2.200	3.400
			Ⓢ=1234567890.3	4.100	13.400	3.000	5.100
	DBL	執行時		2.500	4.400	2.300	2.700
	WORD	執行時		2.800	3.900	2.600	3.600
	GRY	執行時		2.700	4.300	2.300	3.000
	DGRY	執行時		2.700	4.300	2.300	3.000
	GBIN	執行時		4.000	6.400	3.800	4.300
	DGBIN	執行時		5.000	6.900	5.000	5.900
	NEG	執行時		2.100	4.400	2.000	3.300
	DNEG	執行時		2.500	3.700	2.500	3.300
	ENEG	浮點 =0		2.500	3.300	2.300	2.800
		浮點 =-1.0		2.800	5.600	2.500	3.900
	EDNEG	浮點 =0		3.000	8.800	1.800	3.100
		浮點 =-1.0		2.700	9.400	1.900	3.000
	BKBCD Ⓢ Ⓓ n	n=1		6.000	13.400	5.900	8.200
		n=96		83.300	91.400	61.000	63.400
	BKBIN Ⓢ Ⓓ n	n=1		6.500	9.800	5.600	9.300
		n=96		55.400	62.900	49.200	52.500
	ECON	---		3.000	9.800	2.100	4.500
	EDCON	---		3.300	10.300	2.500	5.400
	EDMOV	---		2.700	8.500	1.700	5.000
	$MOV	傳送字串 =0		4.400	12.300	3.400	5.600
		傳送字串 =32		14.000	21.900	11.400	13.300
	BXCH Ⓓ1 Ⓓ2 n	n=1		6.200	7.900	5.500	7.300
		n=96		67.300	71.400	47.300	49.300
	SWAP	---		2.400	2.700	1.900	2.200
	GOEND	---			0.700		0.500
	DI	---		2.100	4.000	1.500	1.800
	EI	---		3.600	6.300	3.000	3.300
	IMASK	---		11.800	20.500	7.200	10.500
	IRET	---			1.400		1.000
	RFS X n	n=1		5.900	12.500	3.700	5.600
		n=96		12.900	19.300	10.700	12.400
	RFS Y n	n=1		5.100	11.500	3.400	4.800
		n=96		8.600	15.300	8.100	8.900
	UDCNT1	---		6.200	16.400	5.100	12.300
	UDCNT2	---		6.300	16.800	5.400	12.500
	TTMR	---		4.500	9.500	3.400	5.400
	STMR	---		7.800	21.400	5.800	12.500
	ROTC	---		20.900	21.500	8.000	9.400
	RAMP	---		6.700	14.600	5.200	8.400
	SPD	---		5.400	14.800	4.900	11.200
	PLSY	---		10.500	10.500	7.900	7.900
	PWM	---		10.100	10.100	7.500	7.500
	MTR	---		14.700	25.100	9.400	10.000
應用指令	BKAND Ⓢ1 Ⓢ2 Ⓓ n	n=1		9.000	11.700	8.300	11.000
		n=96		60.600	66.400	43.800	47.300
	BKOR Ⓢ1 Ⓢ2 Ⓓ n	n=1		7.900	14.000	7.700	9.500
		n=96		60.700	66.500	44.300	45.800
	BKXOR Ⓢ1 Ⓢ2 Ⓓ n	n=1		8.800	13.800	7.300	9.200
		n=96		61.300	66.300	43.800	45.800
	BKXNR Ⓢ1 Ⓢ2 Ⓓ n	n=1		8.400	13.900	7.600	8.900
		n=96		60.900	66.700	43.900	45.300
	BSFR Ⓓ n	n=1		3.600	9.500	3.200	4.800
		n=96		6.500	15.900	5.800	7.700
	BSFL Ⓓ n	n=1		3.600	9.300	3.400	5.100
		n=96		6.300	15.800	6.000	7.900

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
應用指令	SFTBR Ⓓ n1 n2	進行了移位的位的數=16/ 移位數 =1		8.100	21.000	7.500	17.400
		進行了移位的位的數=16/ 移位數 =15		8.100	22.100	7.500	17.300
	SFTBL Ⓓ n1 n2	進行了移位的位的數=16/ 移位數 =1		8.100	21.000	7.500	17.400
		進行了移位的位的數=16/ 移位數 =15		8.100	22.100	7.500	17.300
	SFTWR Ⓓ n1 n2	進行了移位的位的數=16/ 移位數 =1		6.200	13.100	4.500	8.700
		進行了移位的位的數=16/ 移位數 =15		6.100	13.100	4.600	8.800
	SFTWL Ⓓ n1 n2	進行了移位的位的數=16/ 移位數 =1		6.200	13.100	4.500	8.700
		進行了移位的位的數=16/ 移位數 =15		6.100	13.100	4.600	8.800
	BSET Ⓓ n	n=1		2.800	3.100	2.500	2.800
		n=15		2.800	3.100	2.500	2.800
	BRST Ⓓ n	n=1		2.800	3.100	2.500	2.800
		n=15		2.800	3.100	2.500	2.800
	TEST	執行時		4.700	6.100	3.700	4.800
	DTEST	執行時		4.700	6.100	3.700	4.800
	BKRST Ⓢ n	n=1		4.300	5.700	3.700	4.100
		n=96		6.200	10.000	5.100	6.000
	SER Ⓢ1 Ⓢ2 Ⓓ n	n=1	全部一致	4.800	5.300	4.200	4.600
			全部不一致	4.700	5.300	4.200	4.600
		n=96	全部一致	33.200	35.900	25.900	26.300
			全部不一致	33.200	35.900	25.900	26.300
	DSER Ⓢ1 Ⓢ2 Ⓓ n	n=1	全部一致	6.500	9.000	5.400	5.700
			全部不一致	6.500	9.000	5.500	5.900
		n=96	全部一致	54.800	57.500	41.200	41.800
			全部不一致	54.700	57.500	41.200	41.800
	DSUM Ⓢ Ⓓ	Ⓢ=0		3.400	3.700	3.200	3.700
		Ⓢ=FFFFFFFF$_H$		3.400	3.700	3.200	3.700
	DECO Ⓢ Ⓓ n	n=2		6.000	10.700	5.300	6.900
		n=8		9.500	16.700	6.800	7.800
	ENCO Ⓢ Ⓓ n	n=2	M1=ON	5.400	6.900	4.700	5.100
			M4=ON	5.300	6.600	4.600	5.000
		n=8	M1=ON	10.700	14.000	9.000	10.000
			M256=ON	7.000	11.100	5.100	6.100
	DIS Ⓢ Ⓓ n	n=1		4.600	7.000	3.800	4.600
		n=4		4.900	7.300	4.000	5.000
	UNI Ⓢ Ⓓ n	n=1		5.000	7.300	3.500	4.800
		n=4		5.700	8.300	4.000	5.100
	NDIS	執行時		11.200	15.200	11.000	13.200
	NUNI	執行時		10.600	12.700	7.300	13.200
	WTOB Ⓢ Ⓓ n	n=1		5.400	8.100	4.400	5.800
		n=96		38.400	40.900	28.200	29.300
	BTOW Ⓢ Ⓓ n	n=1		5.300	8.200	4.600	5.500
		n=96		31.700	34.200	22.800	23.800
	MAX Ⓢ Ⓓ n	n=1		5.400	11.900	4.000	6.100
		n=96		34.200	41.100	24.700	27.000
	MIN Ⓢ Ⓓ n	n=1		6.100	12.000	4.000	6.000
		n=96		32.900	39.300	26.500	28.300
	DMAX Ⓢ Ⓓ n	n=1		6.000	14.800	4.800	8.100
		n=96		61.100	69.500	47.100	49.600
	DMIN Ⓢ Ⓓ n	n=1		6.000	14.800	4.300	5.900
		n=96		57.000	67.000	45.400	47.400
	SORT Ⓢ1 n Ⓢ2 Ⓓ1 Ⓓ2	n=1、Ⓢ2 =1		6.800	13.700	5.600	8.800
		n=96、Ⓢ2 =16		31,300	46,800	24,300	34,400
	DSORT Ⓢ1 n Ⓢ2 Ⓓ1 Ⓓ2	n=1、Ⓢ2 =1		6.800	14.300	5.600	10,900
		n=96、Ⓢ2 =16		34,900	49,700	26,200	36,700

分類	指令	條件（軟元件）	處理時間（μs）			
			L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
			最小值	最大值	最小值	最大值
應用指令	WSUM Ⓢ Ⓓ n	n=1	5.000	7.300	4.200	5.500
		n=96	28.100	30.700	21.300	22.300
	DWSUM Ⓢ Ⓓ n	n=1	6.100	11.300	4.800	6.100
		n=96	56.200	62.100	42.700	44.000
	MEAN Ⓢ Ⓓ n	n=1	4.400	10.400	3.900	7.800
		n=96	16.100	24.500	12.900	18.000
	DMEAN Ⓢ Ⓓ n	n=1	6.000	12.500	5.300	9.950
		n=96	34.000	42.000	23.000	28.800
	NEXT	---	0.940	1.400	0.770	1.200
	BREAK	---	3.500	10.200	3.100	7.600
	RET	返回至本程式	2.900	8.800	1.600	2.600
		返回至其他程式	3.200	10.500	2.000	3.100
	FCALL Pn	文件內指標	3.600	3.800	2.700	3.600
		公共指標	5.300	13.500	3.600	5.100
	FCALL Pn Ⓢ1 ～ Ⓢ5	---	20.900	30.300	16.500	18.600
	ECALL * Pn *: 程式名	---	72.700	109.000	65.900	77.600
	ECALL * Pn Ⓢ1 ～ Ⓢ5 *: 程式名	---	101.400	141.400	91.800	105.000
	EFCALL * Pn *: 程式名	---	72.800	109.600	66.200	78.100
	EFCALL * Pn Ⓢ1 ～ Ⓢ5 *: 程式名	---	101.900	141.500	78.800	91.600
	XCALL	---	5.200	14.600	3.700	5.200
	CCOM COM	僅選擇 I/O 刷新時	8.400	14.600	12.600	17.200
		僅選擇 CC-Link 刷新時（主站側）	10.500	29.400	10.100	22.000
		僅選擇 CC-Link 刷新時（本地站側）	10.500	29.400	10.100	22.000
		僅選擇 CC-Link IE 現場網路刷新時（主站側）	17.000	49.500	16.600	38.000
		僅選擇 CC-Link IE 現場網路刷新時（本地站側）	17.000	49.500	16.600	38.000
		僅選擇 MELSECNET/H（管理站側）	---	---	---	---
		僅選擇 MELSECNET/H（普通站側）	---	---	---	---
		僅選擇智慧自動刷新時	7.900	14.400	7.400	11.900
		僅選擇組外輸入輸出時（僅輸入）	---	---	---	---
		僅選擇組外輸入輸出時（僅輸出）	---	---	---	---
		僅選擇組外輸入輸出時（輸入及輸出均選擇）	---	---	---	---
		僅選擇多 CPU 高速通信區刷新時	---	---	---	---
		僅選擇與模組通信時	29.700	79.900	26.800	60.700
		僅選擇與週邊設備通信時	9.500	32.800	9.200	25.200
	FIFW	資料數 = 0	4.200	6.700	3.200	4.600
		資料數 = 96	4.400	6.800	3.300	3.800

分類	指令	條件（軟元件）	處理時間（μs）			
			L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
			最小值	最大值	最小值	最大值
應用指令	FIFR	資料數 = 1	5.100	7.400	3.800	4.400
		資料數 = 96	36.100	38.800	24.800	25.700
	FPOP	資料數 = 1	4.900	7.500	3.800	5.300
		資料數 = 96	5.000	7.500	3.700	5.400
	FINS	資料數 = 0	5.400	7.500	3.700	5.300
		資料數 = 96	5.000	7.400	3.700	5.300
	FDEL	資料數 = 1	5.700	8.300	4.200	5.800
		資料數 = 96	36.900	39.300	25.400	25.900
	FROM n1 n2 Ⓓ n3	n3=1	11.600	31.000	10.700	23.600
		n3=1000	403.900	432.900	390.900	410.200
	DFRO n1 n2 Ⓓ n3	n3=1	13.300	35.400	12.600	26.700
		n3=500	405.000	434.600	390.900	410.200
	TO n1 n2 Ⓢ n3	n3=1	11.200	28.400	9.600	21.300
		n3=1000	381.500	410.900	372.500	390.800
	DTO n1 n2 Ⓢ n3	n3=1	12.500	33.900	12.000	25.700
		n3=500	379.800	410.400	372.500	390.800
	LEDR	無顯示→無顯示	2.400	2.600	1.900	2.000
		LED 指令執行→無顯示	32.700	50.600	24.400	35.800
	BINDA Ⓢ Ⓓ	Ⓢ=1	5.000	7.300	4.300	5.600
		Ⓢ=-32768	7.400	9.800	6.500	8.000
	DBINDA Ⓢ Ⓓ	Ⓢ=1	5.600	8.300	4.900	6.300
		Ⓢ=-2147483648	10.500	12.900	9.600	11.000
	BINHA Ⓢ Ⓓ	Ⓢ=1	4.500	6.900	3.700	5.200
		Ⓢ=$FFFF_H$	4.500	6.900	3.700	5.200
	DBINHA Ⓢ Ⓓ	Ⓢ=1	5.000	7.600	4.600	6.000
		Ⓢ=$FFFFFFFF_H$	5.000	7.600	4.600	6.000
	BCDDA Ⓢ Ⓓ	Ⓢ=1	4.300	6.700	3.600	5.000
		Ⓢ=9999	4.800	7.100	4.100	5.400
	DBCDDA Ⓢ Ⓓ	Ⓢ=1	4.900	7.200	4.000	5.500
		Ⓢ=99999999	5.700	8.300	4.900	6.300
	DABIN Ⓢ Ⓓ	Ⓢ=1	5.800	10.100	5.600	7.800
		Ⓢ=-32768	5.800	10.100	5.600	7.800
	DDABIN Ⓢ Ⓓ	Ⓢ=1	8.300	12.600	8.100	10.500
		Ⓢ=-2147483648	8.300	12.600	8.100	10.500
	HABIN Ⓢ Ⓓ	Ⓢ=1	4.500	8.800	4.400	6.500
		Ⓢ=$FFFF_H$	4.500	8.800	4.400	6.500
	DHABIN Ⓢ Ⓓ	Ⓢ=1	5.500	10.000	5.300	7.700
		Ⓢ=$FFFFFFFF_H$	5.500	10.000	5.300	7.700
	DABCD Ⓢ Ⓓ	Ⓢ=1	4.500	8.700	4.300	6.300
		Ⓢ=9999	4.500	8.700	4.300	6.300
	DDABCD Ⓢ Ⓓ	Ⓢ=1	5.500	9.800	5.500	7.500
		Ⓢ=99999999	5.500	9.800	5.500	7.500
	COMRD	---	65.700	65.700	50.900	51.200
	LEN	1 字元	3.900	7.800	3.600	5.500
		96 字元	19.700	23.900	16.800	18.700
	STR	---	7.500	16.700	6.600	10.400
	DSTR	---	10.200	19.700	9.600	11.500
	VAL	---	9.800	19.900	8.900	13.000
	DVAL	---	12.700	23.900	12.700	16.800
	ESTR	---	21.200	43.400	17.900	23.100
	EVAL	小數點格式全 2 位數指定	28.300	41.000	22.500	29.00
		指數格式全 6 位數指定	28.300	41.000	22.500	29.00
	ASC Ⓢ Ⓓ n	n=1	6.200	17.100	5.400	8.300
		n=96	30.300	42.100	25.200	28.400

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
應用指令	HEX Ⓢ Ⓓ n	n=1		5.400	16.000	5.400	9.000
		n=96		42.400	54.900	31.300	35.000
	RIGHT Ⓢ Ⓓ n	n=1		7.400	13.900	6.600	7.300
		n=96		39.300	45.800	29.200	31.600
	LEFT Ⓢ Ⓓ n	n=1		6.900	13.400	5.900	8.200
		n=96		39.300	45.800	29.200	31.500
	MIDR	---		10.200	16.500	8.100	10.300
	MIDW	---		10.700	14.900	8.800	10.200
	INSTR	無一致		20.000	25.600	16.600	18.400
		有一致	起始	11.000	16.500	9.100	10.900
			最終	53.900	60.000	42.700	44.900
	EMOD	---		11.200	15.100	9.600	11.000
	EREXP	---		20.400	22.900	18.800	20.100
	STRINS Ⓢ Ⓓ n	Ⓢ=128/Ⓓ=40/n=1		45.300	63.400	35.300	47.600
		Ⓢ=128/Ⓓ=40/n=48		63.200	81.900	48.600	61.700
	STRDEL Ⓢ Ⓓ n	Ⓢ=128/Ⓓ=40/n=1		39.000	53.500	34.800	44.600
		Ⓢ=128/Ⓓ=40/n=48		40.800	50.400	29.200	38.100
	SIN	單精度		5.000	8.400	4.100	5.700
	COS	單精度		5.200	8.000	4.000	5.600
	TAN	單精度		6.100	9.200	5.100	6.700
	ASIN	單精度		6.900	10.900	5.900	8.500
	ACOS	單精度		7.800	11.000	6.700	8.900
	ATAN	單精度		4.700	7.300	3.900	6.000
	SIND	雙精度		9.400	22.300	8.500	13.800
	COSD	雙精度		10.000	22.300	8.800	14.600
	TAND	雙精度		12.200	24.900	10.800	16.500
	ASIND	雙精度		12.800	25.900	11.600	16.600
	ACOSD	雙精度		12.600	25.900	11.200	16.200
	ATAND	雙精度		10.500	22.900	9.100	13.800
	RAD	單精度		3.000	6.400	2.100	4.300
	RADD	雙精度		5.200	16.900	3.600	9.200
	DEG	單精度		2.900	6.600	2.200	4.400
	DEGD	雙精度		5.200	16.800	3.800	9.000
	SQR	單精度		3.600	7.200	2.600	4.300
	SQRD	雙精度		6.200	19.100	5.200	11.000
	EXP Ⓢ Ⓓ	單精度	Ⓢ=-10	4.700	7.500	3.800	5.600
			Ⓢ=1	4.700	7.500	3.800	5.600
	EXPD Ⓢ Ⓓ	雙精度	Ⓢ=-10	9.300	22.100	8.000	13.500
			Ⓢ=1	9.300	22.100	8.000	13.500
	LOG Ⓢ Ⓓ	單精度	Ⓢ=1	4.700	8.800	3.800	6.400
			Ⓢ=10	6.300	10.400	5.200	7.700
	LOGD Ⓢ Ⓓ	雙精度	Ⓢ=1	8.600	21.100	7.700	12.500
			Ⓢ=10	10.200	23.000	9.200	14.300
	RND	---		1.500	2.500	0.800	1.800
	SRND	---		1.800	2.900	1.100	2.000
	BSQR Ⓢ Ⓓ	Ⓢ=0		2.700	4.400	1.500	3.000
		Ⓢ=9999		6.100	12.500	5.100	8.000
	BDSQR Ⓢ Ⓓ	Ⓢ=0		2.700	4.400	1.500	3.000
		Ⓢ=99999999		8.500	15.200	7.500	9.900
	BSIN	---		9.500	21.500	8.100	14.500
	BCOS	---		9.500	21.400	7.800	13.700
	BTAN	---		10.400	22.600	9.000	13.300
	BASIN	---		11.800	23.600	10.100	12.800
	BACOS	---		13.100	23.700	11.100	14.100
	BATAN	---		11.100	21.500	9.100	10.900
	POW Ⓢ1 Ⓢ2 Ⓓ	單精度	Ⓢ1=12.3E+5 Ⓢ2=3.45E+0	9.600	13.300	8.400	10.900

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
應用指令	POWD (S1) (S2) (D)	雙精度	(S1)=12.3E+5 (S2)=3.45E+0	18.900	30.600	18.200	26.500
	LOG10	單精度		6.000	9.600	5.700	8.050
	LOG10D	雙精度		11.900	22.900	11.100	18.600
	LIMIT	---		4.000	4.000	2.400	2.700
	DLIMIT	---		4.400	4.400	2.800	3.000
	BAND	---		4.500	6.600	2.700	3.800
	DBAND	---		4.800	6.900	3.300	4.600
	ZONE	---		4.200	6.100	2.600	4.300
	DZONE	---		4.700	6.900	3.000	4.600
	LDDT=	與指定的日期比較	導通時	7.700	14.200	6.800	10.900
			非導通時	7.700	14.200	6.800	10.900
		與當前的日期比較	導通時	6.400	12.800	5.500	9.700
			非導通時	6.400	12.800	5.500	9.700
	ANDDT=	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.300	14.000	6.500	10.700
			非導通時	7.300	14.000	6.500	10.700
		與當前的日期比較	導通時	6.100	12.700	5.300	9.300
			非導通時	6.100	12.700	5.300	9.300
	ORDT=	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.400	14.400	6.700	10.800
			非導通時	7.400	14.400	6.700	10.800
		與當前的日期比較	導通時	6.000	12.800	5.400	9.600
			非導通時	6.000	12.800	5.400	9.600
	LDDT<>	與指定的日期比較	導通時	7.700	14.200	6.800	10.900
			非導通時	7.700	14.200	6.800	10.900
		與當前的日期比較	導通時	6.400	12.800	5.500	9.700
			非導通時	6.400	12.800	5.500	9.700
	ANDDT<>	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.300	14.000	6.500	10.700
			非導通時	7.300	14.000	6.500	10.700
		與當前的日期比較	導通時	6.100	12.700	5.300	9.300
			非導通時	6.100	12.700	5.300	9.300
	ORDT<>	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.400	14.400	6.700	10.800
			非導通時	7.400	14.400	6.700	10.800
		與當前的日期比較	導通時	6.000	12.800	5.400	9.600
			非導通時	6.000	12.800	5.400	9.600
	LDDT>	與指定的日期比較	導通時	7.700	14.200	6.800	10.900
			非導通時	7.700	14.200	6.800	10.900
		與當前的日期比較	導通時	6.400	12.800	5.500	9.700
			非導通時	6.400	12.800	5.500	9.700
	ANDDT>	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.300	14.000	6.500	10.700
			非導通時	7.300	14.000	6.500	10.700
		與當前的日期比較	導通時	6.100	12.700	5.300	9.300
			非導通時	6.100	12.700	5.300	9.300
	ORDT>	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.400	14.400	6.700	10.800
			非導通時	7.400	14.400	6.700	10.800
		與當前的日期比較	導通時	6.000	12.800	5.400	9.600
			非導通時	6.000	12.800	5.400	9.600
	LDDT<=	與指定的日期比較	導通時	7.700	14.200	6.800	10.900
			非導通時	7.700	14.200	6.800	10.900
		與當前的日期比較	導通時	6.400	12.800	5.500	9.700
			非導通時	6.400	12.800	5.500	9.700
	ANDDT<=	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.300	14.000	6.500	10.700
			非導通時	7.300	14.000	6.500	10.700
		與當前的日期比較	導通時	6.100	12.700	5.300	9.300
			非導通時	6.100	12.700	5.300	9.300

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
應用指令	ORDT<=	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.400	14.400	6.700	10.800
			非導通時	7.400	14.400	6.700	10.800
		與當前的日期比較	導通時	6.000	12.800	5.400	9.600
			非導通時	6.000	12.800	5.400	9.600
	LDDT<	與指定的日期比較	導通時	7.700	14.200	6.800	10.900
			非導通時	7.700	14.200	6.800	10.900
		與當前的日期比較	導通時	6.400	12.800	5.500	9.700
			非導通時	6.400	12.800	5.500	9.700
	ANDDT<	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.300	14.000	6.500	10.700
			非導通時	7.300	14.000	6.500	10.700
		與當前的日期比較	導通時	6.100	12.700	5.300	9.300
			非導通時	6.100	12.700	5.300	9.300
	ORDT<	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.400	14.400	6.700	10.800
			非導通時	7.400	14.400	6.700	10.800
		與當前的日期比較	導通時	6.000	12.800	5.400	9.600
			非導通時	6.000	12.800	5.400	9.600
	LDDT>=	與指定的日期比較	導通時	7.700	14.200	6.800	10.900
			非導通時	7.700	14.200	6.800	10.900
		與當前的日期比較	導通時	6.400	12.800	5.500	9.700
			非導通時	6.400	12.800	5.500	9.700
	ANDDT>=	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.300	14.000	6.500	10.700
			非導通時	7.300	14.000	6.500	10.700
		與當前的日期比較	導通時	6.100	12.700	5.300	9.300
			非導通時	6.100	12.700	5.300	9.300
	ORDT>=	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.400	14.400	6.700	10.800
			非導通時	7.400	14.400	6.700	10.800
		與當前的日期比較	導通時	6.000	12.800	5.400	9.600
			非導通時	6.000	12.800	5.400	9.600
	LDTM=	與指定的日期比較	導通時	7.600	14.000	6.700	10.800
			非導通時	7.600	14.000	6.700	10.800
		與當前的日期比較	導通時	6.200	12.700	5.400	9.500
			非導通時	6.200	12.700	5.400	9.500
	ANDTM=	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.200	13.900	6.300	10.800
			非導通時	7.200	13.900	6.300	10.800
		與當前的日期比較	導通時	5.900	12.500	5.100	9.500
			非導通時	5.900	12.500	5.100	9.500
	ORTM=	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.300	14.100	6.600	10.800
			非導通時	7.300	14.100	6.600	10.800
		與當前的日期比較	導通時	6.000	12.700	5.300	9.500
			非導通時	6.000	12.700	5.300	9.500
	LDTM<>	與指定的日期比較	導通時	7.600	14.000	6.700	10.800
			非導通時	7.600	14.000	6.700	10.800
		與當前的日期比較	導通時	6.200	12.700	5.400	9.500
			非導通時	6.200	12.700	5.400	9.500
	ANDTM<>	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.200	13.900	6.300	10.800
			非導通時	7.200	13.900	6.300	10.800
		與當前的日期比較	導通時	5.900	12.500	5.100	9.500
			非導通時	5.900	12.500	5.100	9.500
	ORTM<>	未執行時			0.160		0.038
		與指定的日期比較	導通時	7.300	14.100	6.600	10.800
			非導通時	7.300	14.100	6.600	10.800
		與當前的日期比較	導通時	6.000	12.700	5.300	9.500
			非導通時	6.000	12.700	5.300	9.500
	LDTM>	與指定的日期比較	導通時	7.600	14.000	6.700	10.800
			非導通時	7.600	14.000	6.700	10.800
		與當前的日期比較	導通時	6.200	12.700	5.400	9.500
			非導通時	6.200	12.700	5.400	9.500

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
應用指令	ANDTM>	未執行時		0.160		0.038	
		與指定的日期比較	導通時	7.200	13.900	6.300	10.800
			非導通時	7.200	13.900	6.300	10.800
		與當前的日期比較	導通時	5.900	12.500	5.100	9.500
			非導通時	5.900	12.500	5.100	9.500
	ORTM>	未執行時		0.160		0.038	
		與指定的日期比較	導通時	7.300	14.100	6.600	10.800
			非導通時	7.300	14.100	6.600	10.800
		與當前的日期比較	導通時	6.000	12.700	5.300	9.500
			非導通時	6.000	12.700	5.300	9.500
	LDTM<=	與指定的日期比較	導通時	7.600	14.000	6.700	10.800
			非導通時	7.600	14.000	6.700	10.800
		與當前的日期比較	導通時	6.200	12.700	5.400	9.500
			非導通時	6.200	12.700	5.400	9.500
	ANDTM<=	未執行時		0.160		0.038	
		與指定的日期比較	導通時	7.200	13.900	6.300	10.800
			非導通時	7.200	13.900	6.300	10.800
		與當前的日期比較	導通時	5.900	12.500	5.100	9.500
			非導通時	5.900	12.500	5.100	9.500
	ORTM<=	未執行時		0.160		0.038	
		與指定的日期比較	導通時	7.300	14.100	6.600	10.800
			非導通時	7.300	14.100	6.600	10.800
		與當前的日期比較	導通時	6.000	12.700	5.300	9.500
			非導通時	6.000	12.700	5.300	9.500
	LDTM<	與指定的日期比較	導通時	7.600	14.000	6.700	10.800
			非導通時	7.600	14.000	6.700	10.800
		與當前的日期比較	導通時	6.200	12.700	5.400	9.500
			非導通時	6.200	12.700	5.400	9.500
	ANDTM<	未執行時		0.160		0.038	
		與指定的日期比較	導通時	7.200	13.900	6.300	10.800
			非導通時	7.200	13.900	6.300	10.800
		與當前的日期比較	導通時	5.900	12.500	5.100	9.500
			非導通時	5.900	12.500	5.100	9.500
	ORTM<	未執行時		0.160		0.038	
		與指定的日期比較	導通時	7.300	14.100	6.600	10.800
			非導通時	7.300	14.100	6.600	10.800
		與當前的日期比較	導通時	6.000	12.700	5.300	9.500
			非導通時	6.000	12.700	5.300	9.500
	LDTM>=	與指定的日期比較	導通時	7.600	14.000	6.700	10.800
			非導通時	7.600	14.000	6.700	10.800
		與當前的日期比較	導通時	6.200	12.700	5.400	9.500
			非導通時	6.200	12.700	5.400	9.500
	ANDTM>=	未執行時		0.160		0.038	
		與指定的日期比較	導通時	7.200	13.900	6.300	10.800
			非導通時	7.200	13.900	6.300	10.800
		與當前的日期比較	導通時	5.900	12.500	5.100	9.500
			非導通時	5.900	12.500	5.100	9.500
	ORTM>=	未執行時		0.160		0.038	
		與指定的日期比較	導通時	7.300	14.100	6.600	10.800
			非導通時	7.300	14.100	6.600	10.800
		與當前的日期比較	導通時	6.000	12.700	5.300	9.500
			非導通時	6.000	12.700	5.300	9.500

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
應用指令	SCL ⓢ1 ⓢ2 ⓓ	SM750=ON	點 No.1<ⓢ1 <點 No.2	12.500	29.200	11.900	23.000
			點 No.9<ⓢ1 <點 No.10	13.200	29.100	12.100	23.000
		SM750=OFF	點 No.1<ⓢ1 <點 No.2	12.100	28.900	10.900	22.200
			點 No.9<ⓢ1 <點 No.10	13.900	30.900	12.700	23.900
	DSCL ⓢ1 ⓢ2 ⓓ	SM750=ON	點 No.1<ⓢ1 <點 No.2	12.500	29.200	11.900	23.000
			點 No.9<ⓢ1 <點 No.10	13.200	29.100	12.100	23.000
		SM750=OFF	點 No.1<ⓢ1 <點 No.2	12.100	28.900	10.900	22.200
			點 No.9<ⓢ1 <點 No.10	13.900	30.900	12.700	23.900
	SCL2 ⓢ1 ⓢ2 ⓓ	SM750=ON	點 No.1<ⓢ1 <點 No.2	13.400	29.700	11.800	23.300
			點 No.9<ⓢ1 <點 No.10	12.900	29.500	12.100	23.300
		SM750=OFF	點 No.1<ⓢ1 <點 No.2	12.200	29.100	11.000	22.600
			點 No.9<ⓢ1 <點 No.10	13.900	30.700	12.600	23.900
	DSCL2 ⓢ1 ⓢ2 ⓓ	SM750=ON	點 No.1<ⓢ1 <點 No.2	13.400	29.700	11.800	23.300
			點 No.9<ⓢ1 <點 No.10	12.900	29.500	12.100	23.300
		SM750=OFF	點 No.1<ⓢ1 <點 No.2	12.200	29.100	11.000	22.600
			點 No.9<ⓢ1 <點 No.10	13.900	30.700	12.600	23.900
	RSET	標準 RAM		3.500	11.100	2.700	5.900
		SRAM 卡		---	---	---	---
	DATE-	無進位		9.000	17.900	4.600	7.000
		有進位		10.000	19.200	4.600	6.500
	SECOND	---		4.600	9.800	2.200	3.400
	HOUR	---		4.600	10.300	2.400	4.300
	QDRSET	SRAM 卡→標準 RAM		---	---	---	---
		標準 RAM → SRAM 卡		---	---	---	---
	QCDSET	SD 存儲卡→ 標準 ROM		690.800	736.470	1146.900	1179.500
		標準 ROM → SD 存儲卡		6981.400	7232.070	5613.900	5653.500
	DATERD	---		4.600	11.200	2.500	4.200
	DATEWR	---		6.500	19.300	4.100	8.900
	DATE +	無進位		10.000	19.400	4.700	6.600
		有進位		9.900	19.700	4.600	6.500
	S.DATERD	---		7.800	22.500	4.800	7.100
	S.DATE+	無進位		15.100	34.100	7.400	10.000
		有進位		15.000	34.100	7.400	10.000
	S.DATE-	無進位		13.700	33.600	7.400	10.300
		有進位		13.700	33.600	7.500	10.200
	PSTOP	---		67.600	104.100	56.600	79.800
	POFF	---		66.800	103.600	57.200	79.800
	PSCAN	---		67.900	104.800	60.100	79.900
	WDT	---		1.600	4.800	1.100	2.400
	DUTY	---		4.900	10.100	4.800	9.600
	TIMCHK	---		4.100	9.100	3.500	4.700
	ZRRDB	標準 RAM 的文件寄存器		2.900	3.300	1.800	2.100
		SRAM 卡的文件寄存器		---	---	---	---

分類	指令	條件（軟元件）		處理時間（μs）			
				L02CPU、L02CPU-P		L26CPU-BT、L26CPU-PBT	
				最小值	最大值	最小值	最大值
應用指令	ZRWRB	標準 RAM 的文件寄存器		3.600	3.800	2.400	2.700
		SRAM 卡的文件寄存器		---	---	---	---
	ADRSET	---		2.200	4.800	2.100	2.600
	ZPUSH	---		8.000	12.000	5.800	7.500
	ZPOP	---		8.200	10.900	5.800	6.400
	S.ZCOM	安裝 CC-Link 模組時（主站側）		23.700	48.500	19.300	26.000
		安裝 CC-Link 模組時（本地站側）		23.700	48.500	19.100	26.200
		僅選擇 CCLink IE 現場網路刷新時（主站側）		31.500	72.000	31.000	58.000
		僅選擇 CCLink IE 現場網路刷新時（本地站側）		31.500	72.000	31.000	58.000
		安裝 MELSECNET/H、CCLink IE 控制網路模組時（管理站側）		---	---	---	---
		安裝 MELSECNET/H、CCLink IE 控制網路模組時（普通站側）		---	---	---	---
	S.RTREAD	---		8.500	27.000	7.400	19.000
	S.RTWRITE	---		9.000	28.000	8.300	19.800
	UNIRD n1 Ⓓ n2	n2=1		5.000	14.100	3.700	8.000
		n2=16		13.600	22.600	12.200	16.600
	TYPERD	---		32.100	67.600	29.500	52.500
	TRACE	開始		58.100	58.100	43.800	44.700
	TRACER	---		6.100	6.100	4.500	4.500
	UMSG	顯示字元數 =1		7.300	17.000	7.000	13.500
		顯示字元數 =32		16.500	26.300	14.300	21.300
	RBMOV Ⓢ Ⓓ n	使用標準 RAM	1 點	---	---	---	---
			1000 點	---	---	---	---
		使用 SRAM 卡	1 點	---	---	---	---
			1000 點	---	---	---	---
	SP.FWRITE	---		81.000	81.800	63.500	64.100
	SP.FREAD	---		81.100	81.700	61.600	62.500
	SP.DEVST	---		50.100	50.100	39.400	39.400
	S.DEVLD	---		12.000	27.600	10.000	17.000
多 CPU 專用指令	S.TO n1 n2 n3 n4 Ⓓ	至本機 CPU 共用記憶體的寫入	n4=1	---	---	---	---
			n4=320	---	---	---	---
	TO n1 n2 Ⓢ n3	至本機 CPU 共用記憶體的寫入	n3=1	---	---	---	---
			n3=320	---	---	---	---
	DTO n1 n2 Ⓢ n3	至本機 CPU 共用記憶體的寫入	n3=1	---	---	---	---
			n3=320	---	---	---	---
	FROM n1 n2 Ⓓ n3	本機 CPU 共用記憶體讀取	n3=1	---	---	---	---
			n3=320	---	---	---	---
		其他機號 CPU 共用記憶體讀取	n3=1	---	---	---	---
			n3=320	---	---	---	---
			n3=1000	---	---	---	---
	DFRO n1 n2 Ⓓ n3	本機 CPU 共用記憶體讀取	n3=1	---	---	---	---
			n3=320	---	---	---	---
		其他機號 CPU 共用記憶體讀取	n3=1	---	---	---	---
			n3=320	---	---	---	---
			n3=1000	---	---	---	---
多 CPU 高速通信專用指令	D.DDWR n Ⓢ1 Ⓢ2 Ⓓ1 Ⓓ2	至其他機號的軟元件寫入	值 =1	---	---	---	---
			值 =16	---	---	---	---
			n=96	---	---	---	---
	DP.DDWR n Ⓢ1 Ⓢ2 Ⓓ1 Ⓓ2		n=1	---	---	---	---
			n=16	---	---	---	---
			n=96	---	---	---	---
	D.DDRD n Ⓢ1 Ⓢ2 Ⓓ1 Ⓓ2	至其他機號的軟元件讀取	n=1	---	---	---	---
			n=16	---	---	---	---
			n=96	---	---	---	---
	DP.DDRD n Ⓢ1 Ⓢ2 Ⓓ1 Ⓓ2		n=1	---	---	---	---
			n=16	---	---	---	---
			n=96	---	---	---	---

備 註

對於表中未記述的上升沿執行指令 (□ P) 的指令，其處理時間與 ON 時執行的指令的處理時間相同。

例 WORDP 指令、TOP 指令等。

(2) 使用了文件寄存器、擴展資料寄存器、擴展鏈結寄存器、模組訪問軟元件時的加法運算時間一覽表

(a) 使用 L02CPU、L26CPU-BT、L02CPU-P、L26CPU-PBT 時

軟元件名		資料	軟元件指定位置	加法運算時間 (μs)	
				L02CPU、L02CPU-P	L26CPU-BT、L26CPU-BT
文件寄存器 (R)	使用標準 RAM 時	位	源	0.100	0.048
			目標	0.220	0.038
		字	源	0.100	0.048
			目標	0.100	0.038
		雙字	源	0.200	0.095
			目標	0.200	0.086
文件寄存器 (ZR)、 擴展資料寄存器 (D)、 擴展鏈結寄存器 (W)	使用標準 RAM 時	位	源	0.140	0.057
			目標	0.280	0.048
		字	源	0.140	0.057
			目標	0.140	0.048
		雙字	源	0.240	0.105
			目標	0.240	0.095
模組訪問軟元件 (Un\G □)		位	源	11.700	11.200
			目標	15.400	15.300
		字	源	9.460	9.410
			目標	19.000	19.000
		雙字	源	11.000	10.900
			目標	18.800	18.700
鏈結直接軟元件 (Jn\ □)		位	源	41.600	37.900
			目標	63.200	58.100
		字	源	40.700	37.500
			目標	31.700	30.800
		雙字	源	49.400	43.400
			目標	39.600	37.300

附錄 2 CPU 的性能比較

附錄 2.1 QCPU/LCPU 與 AnNCPU/AnACPU/AnUCPU 的比較

附錄 2.1.1 可用的軟元件

軟元件名		QCPU			LCPU	AnUCPU	AnACPU	AnNCPU
輸入輸出點數 *9		Q00J：256 點 Q00：1024 點 Q01：1024 點	Q00UJ：256 點 Q00U：1024 點 Q01U：1024 點	Q02、Q02H、Q06H、Q12H、Q25H、Q02PH、Q06PH、Q12PH、Q25PH、Q12PRH、Q25PRH、Q03UD(E)、Q04UD(E)H、Q06UD(E)H、Q10UD(E)H、Q13UD(E)H、Q20UD(E)H、Q26UD(E)H、Q50UDEHCPU、Q100UDEHCPU：4096 點 Q02UCPU：2048 點	L02CPU、L02CPU-P：1024 點 L26CPU-BT、L26CPU-PBT：4096 點	A2U：512 點 A2U-S1：1024 點 A3U：2048 點 A4U：4096 點	A2A：512 點 A2A-S1：1024 點 A3A：2048 點	A1N：256 點 A2N：512 點 A2N-S1：1024 點 A3N：2048 點
輸入輸出軟元件點數 *8		2048 點 *1	8192 點 *1		8192 點 *1	8192 點	與各 CPU 的輸入輸出相同	
內部繼電器		8192 點 *1			8192 點 *1	合計 8192 點		合計 2048 點
鎖存繼電器		2048 點 *1	8192 點 *1		8192 點 *1			
步進繼電器	順控程式用	--			--			--
	SFC 用	2048 點 *6	8192 點		8192 點	--		
報警器		1024 點 *1	2048 點 *1		2048 點 *1	2048 點		256 點
變址繼電器		1024 點 *1	2048 點 *1		2048 點 *1	--		
鏈結繼電器		2048 點 *1	8192 點 *1		8192 點 *1	8192 點	4096 點	1024 點
鏈結用特殊繼電器		1024 點	2048 點		2048 點	56 点點		
計時器		512 點 *1	2048 點 *1		2048 點 *1	合計 2048 點		合計 256 點
累計計時器		0 點 *1			0 點 *1			
計數器		512 點 *1	1024 點 *1		1024 點 *1	1024 點		256 點
資料寄存器		11136 點 *1	12288 點 *1		12288 點 *1	8192 點	6144 點	1024 點
鏈結寄存器		2048 點 *1	8192 點 *1		8192 點 *1	8192 點	4096 點	1024 點
鏈結用特殊寄存器		1024 點	2048 點		2048 點	56 點		
功能輸入		16 點 (FX0 ～ FXF) *7			16 點 (FX0 ～ FXF) *7	--		
功能輸出		16 點 (FY0 ～ FYF) *7			16 點 (FY0 ～ FYF) *7			
特殊繼電器		1000 点	2048 點		2048 點	256 點		
功能寄存器		5 點 (FD0 ～ FD4)			5 點 (FD0 ～ FD4)	--		
特殊寄存器		1000 點	2048 點		2048 點	256 點		
鏈結直接軟元件		通過 J □ \ □指定			--	--		
智慧功能模組軟元件		通過 U □ \G □指定			通過 U □ \G □指定			
變址寄存器	Z	10 點 (Z0 ～ Z9)	通用型 QCPU 以外：16 點 (Z0 ～ Z15) 通用型 QCPU：20 點 (Z0 ～ Z19)		20 點 (Z0 ～ Z19)	7 點 (Z、Z1 ～ Z6)		1 點 (Z)
	V *2	--			--	7 點 (V、V1 ～ V6)		1 點 (V)
文件寄存器		32768 點 / 塊 *5 (R0 ～ R32767)	32768 點 / 塊 (R0 ～ R32767) *10		32768 點 / 塊 (R0 ～ R32767)	8192 點 / 塊 (R0 ～ R8191)		
累加器 *3		--			--	2 點		
嵌套		15 點			15 點	8 點		
指標		300 點	512 點	4096 點	4096 點	256 點		
中斷指標		128 點	128 點	256 點	256 點	32 點		
SFC 塊		126 *6	320 點		320 點	--		
SFC 移位軟元件		--	512 點		512 點			
10 進制常數		K-2147483648 ～ K2147483647						
16 進制常數		H0 ～ HFFFFFFFF						
實數常數 *6		E ± 1.17550-38 ～ E ± 3.40282+38				--		
字串		“ QnACPU ”、“ ABCD ” *4						

*1: 可以對參數中使用的點數進行變更。
*2: 將 V 作爲變址繼電器使用。
*3: 在 AnNCPU/AnACPU/AnUCPU 中使用了累加器的指令，在 QCPU 中指令的格式將改變。
*4: 在 Q00JCPU、Q00CPU、Q01CPU 中只能通過 $MOV 指令使用。
*5: Q00JCPU 中沒有文件寄存器。
*6: 序列號的前 5 位數爲 “ 04122 ” 以後的 Q00JCPU、Q00CPU、Q01CPU 可以使用。
*7: 在程式中，只能使用 FX0 ～ FX4、FY0 ～ FY4 的各 5 點。
*8: 可用於程式的點數。
*9: 實際輸入輸出模組的可訪問點數。
*10: Q00UJCPU 中沒有文件寄存器。

附錄 2.1.2 I/O 控制方式

I/O 控制方式			QCPU	LCPU	AnUCPU	AnACPU	AnNCPU
刷新方式			○	○	○	○	○ *2
	直接輸入輸出方式	部分刷新指令	○	○	○	○	○
		專用指令 *1	---	---	○	○	---
		直接訪問輸入	○	○	---	---	---
		直接訪問輸出	○	○	---	---	---
直接方式			---	---	---	---	○ *2

表中的符號…○： 可以使用； --: 不能使用

*1: 直接輸出專用指令中，有 DOUT、DSET、SRST 指令。
沒有直接輸入專用指令。
*2: 刷新方式與直接方式的切換是通過 AnNCPU 的插杆開關進行。

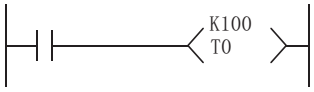
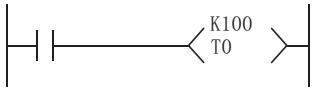
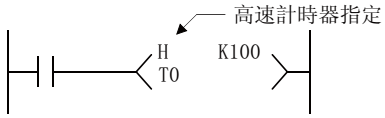
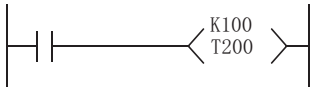
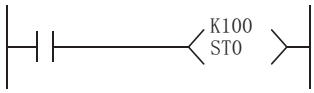
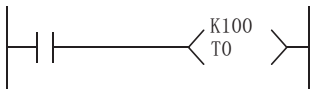
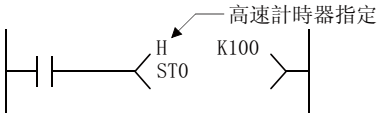
附錄 2.1.3 可用於指令的資料

設置資料		QCPU	LCPU	AnUCPU	AnACPU	AnNCPU
位元資料	位元軟元件	○	○	○	○	○
	字軟元件	○ （需要位元指定）	○ （需要位元指定）	---	---	---
字資料	位元軟元件	○ （需要位元數指定）	○ （需要位元數指定）	○ （需要位元數指定）	○ （需要位元數指定）	○ （需要位元數指定）
	字軟元件	○	○	○	○	○
雙字資料	位元軟元件	○ （需要位元數指定）	○ （需要位元數指定）	○ （需要位元數指定	○ （需要位元數指定	○ （需要位元數指定
	字軟元件	○	○	○	○	○
實數資料		○ *2	○	○	○	△ *1
字串資料		○ *3	○	---	---	---

表中的符號…○： 可以使用；△： 有條件使用； --: 不能使用

*1: SW0SRXV-FUN2 型軟元件包的浮點實數類型用微機套裝軟體登錄時可以使用。
*2: 序列號的前 5 位數爲 “ 04122 ” 以後的 Q00JCPU、Q00CPU、Q01CPU 可以使用。
*3: 在 Q00JCPU、Q00CPU、Q01CPU 中只能通過 $MOV 指令使用。

附錄 2.1.4　計時器的比較

功能		QCPU/LCPU	AnUCPU	AnACPU	AnNCPU
低速計時器	計量單位	100ms(預設值) 在參數中可對計量單位進行變更。 QCPU/LCPU: 1 ～ 1000ms(1ms 單位)	固定爲 100ms		
	指定方法				
高速計時器	計量單位	10ms(預設值) 通過參數可對計量單位進行變更。 QnUCPU/LCPU: 0.01 ～ 100ms(0.01ms 單位) QCPU(除 QnUCPU/LCPU 以外) : 0.1 ～ 100ms(0.1ms 單位)	固定爲 10ms		
	指定方法	 ※高速計時器設置： 通過順控程式進行。	 ※高速計時器設置： 通過參數進行。		
累計計時器	計量單位	與低速計時器的計量單位相同。	固定爲 100ms		
	指定方法				
高速累計計時器	計量單位	與高速計時器的計量單位相同。	無		
	指定方法	 ※高速計時器設置： 通過順控程式進行。			
設置值的設置範圍		1 ～ 32767	1 ～ 32767		
設置值 0 的處理		暫態 ON	無限大 (無時間到)		
變址修飾	觸點	可以 (僅 Z0、Z1 可以使用)	可以		不能
	線圈	可以 (僅 Z0、Z1 可以使用)	不能		不能
	設置值	可以 (Z0 ～ Z15 可以使用)*1	不能		不能
	當前值	可以 (Z0 ～ Z15 可以使用)*1	可以		可以
當前值的更新處理		執行 OUT Tn 指令時	END 處理時		
觸點的 ON/OFF 處理					

*1:　Q00JCPU、Q00CPU、Q01CPU 可以使用 Z0 ～ Z9。
　　通用型 QCPU、LCPU 可以使用 Z0 ～ Z19。

(1) 使用計時器時的注意事項

在 QCPU、LCPU 中，執行 OUT T □指令時進行當前值的更新及觸點的 ON/OFF 操作。

因此計時器的線圈變爲 ON 時，如果 (當前值) ≥ (設置值)，則該計時器的觸點將 ON。

創建通過計時器的觸點進行其他計時器的計量的程式時，應按從後計量的計時器開始的順序進行編程。

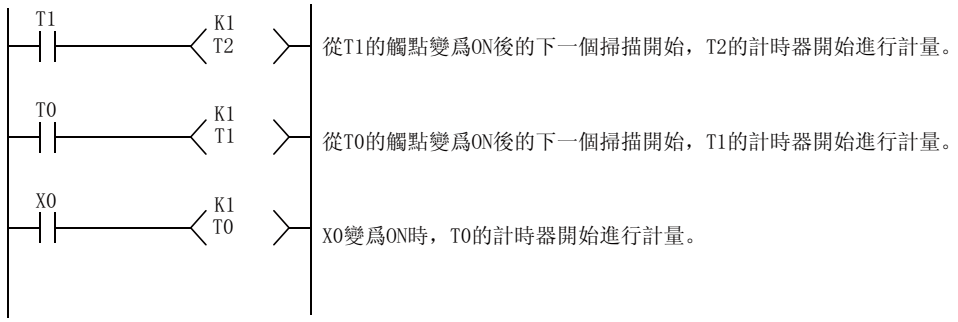
在下述情況下，如果按計量順序進行編程，所有的計時器將在同一個掃描中變爲 ON。

· 高速計時器中設置值短于掃描時間時

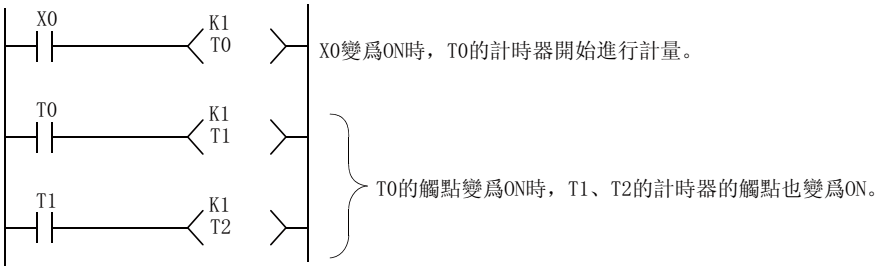
· 低速計時器中設置值爲 “ 1 ” 時

例

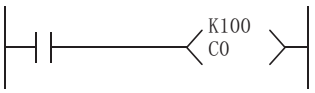
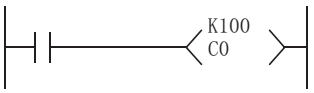
・將 T0 ～ T2 的計時器按照從後計量的計時器開始的順序進行編程時



・將 T0 ～ T2 的計時器按照計量順序進行編程時



附錄 2.1.5　計數器的比較

功能		QCPU/LCPU	AnUCPU	AnACPU	AnNCPU
指定方法					
變址修飾	觸點	・可以 (僅 Z0、Z1 可以使用)	・可以		・不能
	線圈	・可以 (僅 Z0、Z1 可以使用)	・不能		・不能
	設置值	・不能	・不能		・不能
	當前值	・可以 (Z0 ～ Z15 可以使用)*1	・可以		・可以
當前值的更新處理 觸點的 ON/OFF 處理		・執行 OUT Cn 指令時	・END 處理時		

*1:　Q00JCPU、Q00CPU、Q01CPU 可以使用 Z0 ～ Z9。
通用型 QCPU/LCPU 可以使用 Z0 ～ Z19。

附錄 2.1.6　顯示指令的比較

指令	QCPU/LCPU	AnUCPU	AnACPU	AnNCPU
PR*1	・SM701 變爲 OFF 時：輸出直到 00_H 爲止 ・SM701 變爲 ON 時 ： 輸出 16 字元	・M9049 變爲 OFF 時：輸出直到 00_H 爲止 ・M9049 變爲 ON 時 ： 輸出 16 字元		
PRC*1	・SM701 變爲 OFF 時：輸出 32 字元的注釋 ・SM701 變爲 ON 時 ： 輸出高 16 位元字元	輸出 16 字元的注釋		

*1:　在 Q00JCPU/Q00CPU/Q01CPU 中不能使用。

附錄 2.1.7　指定格式被變更的指令 (AnACPU/AnUCPU 的專用指令除外)

由於在 QCPU、LCPU 中沒有累加器 (A0、A1)，因此對於 AnUCPU、AnACPU、AnNCPU 中使用了累加器的指令，其格式將被變更。

功能	QCPU/LCPU		AnUCPU/AnACPU/AnNCPU	
	指令的格式	備註	指令的格式	備註
16 位右旋轉	ROR D n	· D：旋轉資料	ROR n	· 旋轉資料被設置到 A0 中。
	RCR D n	· D：旋轉資料 · 進位元標誌使用 SM700。	RCR n	· 旋轉資料被設置到 A0 中。 · 進位元標誌使用 M9012。
16 位左旋轉	ROL D n	· D：旋轉資料	ROL n	· 旋轉資料被設置到 A0 中。
	RCL D n	· D：旋轉資料 · 進位元標誌使用 SM700。	RCL n	· 旋轉資料被設置到 A0 中。 · 進位元標誌使用 M9012。
32 位右旋轉	DROR D n	· D：旋轉資料	DROR n	· 旋轉資料被設置到 A0、A1 中。
	DRCR D n	· D：旋轉資料 · 進位元標誌使用 SM700。	DRCR n	· 旋轉資料被設置到 A0、A1 中。 · 進位元標誌使用 M9012。
32 位左旋轉	DROL D n	· D：旋轉資料	DROL n	· 旋轉資料被設置到 A0、A1 中。
	DRCL D n	· D：旋轉資料 · 進位元標誌使用 SM700。	DRCL n	· 旋轉資料被設置到 A0、A1 中。 · 進位元標誌使用 M9012。
16 位元資料搜索	SER S1 S2 D n	· 搜索結果被存儲到 D、D+1 的軟元件中。	SER S1 S2 n	· 搜索結果被存儲到 A0、A1 中。
32 位元資料搜索	DSER S1 S2 D n	· 搜索結果被存儲到 D、D+1 的軟元件中。	DSER S1 S2 n	· 搜索結果被存儲到 A0、A1 中。
16 位元資料搜索	SUM S D	· 檢查結果被存儲到 D 的軟元件中。	SUM S	· 檢查結果被存儲到 A0 中。
32 位元資料搜索	DSUM S D	· 檢查結果被存儲到 D 的軟元件中。	DSUM S	· 檢查結果被存儲到 A0 中。
部分刷新	RFS D n	· 添加專用指令。	SEG D n	· 僅在 M9052 爲 ON 時。
8 字元的 ASCII 碼轉換	$MOV (字串) D	---	ASC (字串) D	---
進位元標誌的設置	SET SM700	· 無專用指令。	STC	---
進位元標誌的重定	RST SM700	· 無專用指令。	CLC	---
至 END 指令的跳轉	GOEND	· 添加專用指令。	CJ P255	· P255：END 指令指定
CHK 指令 *1	CHKST CHK	· 添加 CHKST 指令	CJ Pn P254 CHK	---

*1:　在 Q00J/Q00/Q01CPU/ 通用型 QCPU/LCPU 中不能使用。

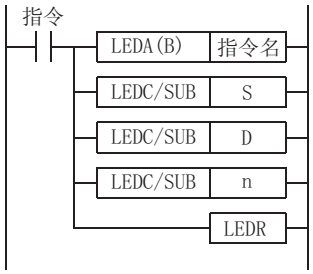
附錄 2.1.8 AnACPU/AnUCPU 的專用指令

(1) 專用指令的表示方法

對於 QCPU/LCPU，將 AnACPU/AnUCPU 中 LEDA、LEDB、LEDC、SUB、LEDR 指令等專用指令變更爲與基本指令 / 應用指令相同的格式。

對於 QCPU/LCPU 中由於無相應指令而無法轉換的指令，將被轉換爲 OUT SM1255/OUT SM999(Q00J/Q00/Q01CPU 時)。

應將被轉換爲 OUT SM1255/OUT SM999 的指令用其他指令替換或者將其刪除。

QCPU/LCPU	AnUCPU/AnACPU
	 S、D、n 表示指令中使用的資料。

(2) 指令名被變更的專用指令

對於在 AnACPU/AnUCPU 的專用指令的指令名中，與 QCPU 的基本指令 / 應用指令具有相同名稱的指令，在 QCPU/LCPU 中其名稱將被變更。

功能	QCPU/LCPU	AnUCPU/AnACPU
浮點數加法運算	E+	ADD
浮點數減法運算	E-	SUB
浮點數乘法運算	E*	MUL
浮點數除法運算	E/	DIV
資料的分離	NDIS	DIS
資料的合併	NUNI	UNI
檢查模式的變更	CHKCIR*1、CHKEND*1	CHK、CHKEND

*1: 在 Q00J/Q00/Q01CPU/ 通用型 QCPU/LCPU 中不能使用。

附錄 3　應用程式示例

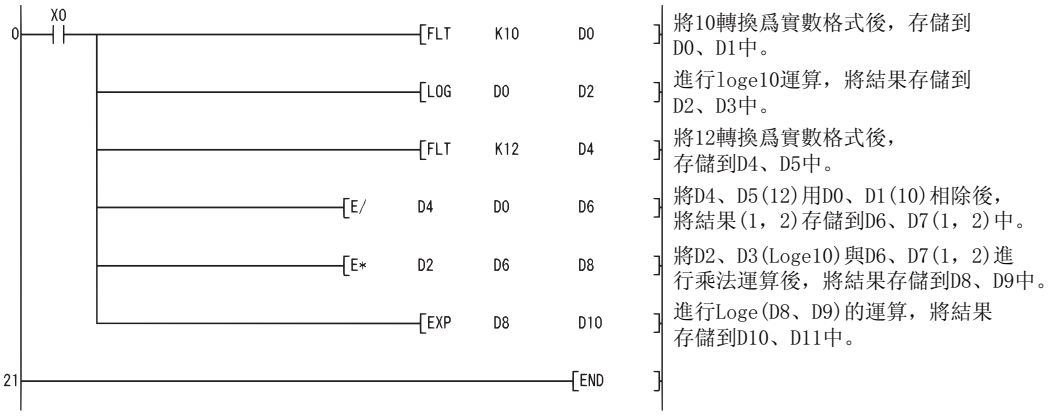
附錄 3.1　執行 X^n、$\sqrt[n]{X}$ 運算的程式的思路

通過 LOG 指令與 EXP 指令的組合可以執行 X^n、$\sqrt[n]{X}$ 運算。

(1) 執行 X^n 運算的程式的思路

X^n 可以通過 $e^{(n\log eX)}$ 進行運算。

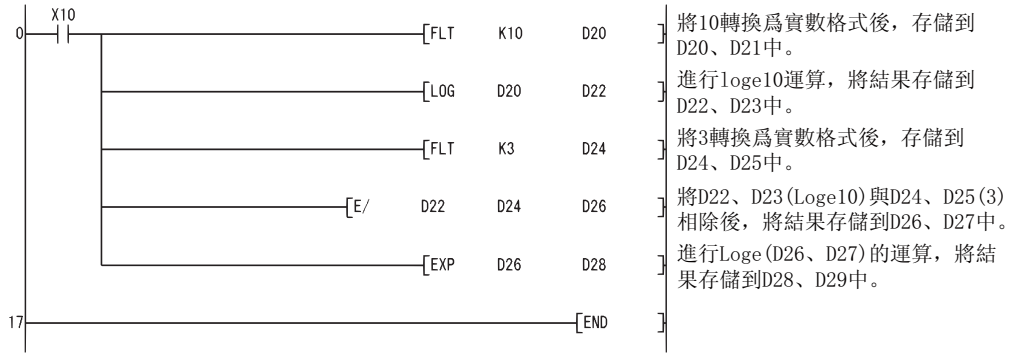
例如，$10^{1.2}$ 的運算等同於 $e^{(1.2 \times \log e10)}$，通過順控程式進行運算時的情況如下所示。



(2) 執行$\sqrt[n]{X}$ 運算的程式的思路

$\sqrt[n]{X}$ 可以通過 $e^{(\frac{1}{n}\log eX)}$ 進行運算。

例如，$\sqrt[3]{10}$ 的運算等同於 $e^{(\frac{1}{3} \times \log e10)}$，通過順控程式進行運算時的情況如下所示。



術語索引

0 ～ 9

10 進制 ASCII 碼→ BCD4 位轉換 444
10 進制 ASCII 碼→ BCD8 位轉換 444
10 進制 ASCII 碼→ BIN16 位轉換 440
10 進制 ASCII 碼→ BIN32 位轉換 440
16 進制 ASCII 碼→ BIN16 位轉換 443
16 進制 ASCII 碼→ BIN32 位轉換 443
16 進制 BIN → ASCII 碼轉換 466
16 位元 BIN 資料遞減 219
16 位元 BIN 資料遞增 219
16 位變址修飾 88
16 位元資料傳送 246
16 位元資料的 4 位元分離 352
16 位元資料的 4 位元合併 353
16 位元資料的 n 位元右移 328
16 位元資料的 n 位元左移 328
16 位元資料的位元檢查 347
16 位元資料的右旋 319
16 位元資料的左旋 321
16 位元資料否定傳送 251
16 位元資料否定排他邏輯和 313
16 位元資料合計值計算 367
16 位元資料邏輯和 301
16 位元資料邏輯積 295
16 位元資料排他邏輯和 307
16 位元資料排序 364
16 位元資料平均值計算 369
16 位元資料搜索 343
16 位元資料置換 261
16 位元資料最大值查找 361
16 位元資料最小值查找 362
256 → 8 位元編碼 348
32 位元 BIN 資料遞減 221
32 位元 BIN 資料遞增 221
32 位變址修飾 89
32 位元資料傳送 246
32 位元資料的位元檢查 345
32 位元資料的右旋 324
32 位元資料的左旋 326
32 位元資料否定傳送 251
32 位元資料否定排他邏輯和 313
32 位元資料合計值計算 368
32 位元資料邏輯和 301
32 位元資料邏輯積 295
32 位元資料排他邏輯和 307
32 位元資料排序 364
32 位元資料平均值計算 369
32 位元資料搜索 343
32 位元資料置換 261
32 位元資料最大值查找 361
32 位元資料最小值查找 362
7 段解碼 350
8 → 256 位解碼 347

A

A5□B 24
A6□B 24
AnACPU/AnUCPU 的專用指令 798
ASCII 碼→ BIN16 進制資料轉換 468
ASCII 碼列印 418

B

BCD4 位→ 10 進制 ASCII 碼轉換 438
BCD4 位元→ BIN 資料轉換 224
BCD4 位乘法和除法運算 196
BCD4 位加法和減法運算 190
BCD4 位平方根 522
BCD8 位→ 10 進制 ASCII 碼轉換 438
BCD8 位元→ BIN 資料轉換 224
BCD8 位乘法和除法運算 198
BCD8 位加法和減法運算 193
BCD8 位平方根 522
BCD 格式資料→浮點數據 482
BCD 型 COS^{-1} 運算 532
BCD 型 COS 運算 526
BCD 型 SIN^{-1} 運算 530
BCD 型 SIN 運算 524
BCD 型 TAN^{-1} 運算 534
BCD 型 TAN 運算 528
BIN16 位乘法和除法運算 187
BIN16 位→ 10 進制 ASCII 碼轉換 433
BIN16 位→ 16 進制 ASCII 碼轉換 435
BIN16 位元→字串轉換 450
BIN16 位加法和減法運算 181
BIN16 位元塊資料比較 175
BIN16 位元區域控制 541
BIN16 位元上下限控制 536
BIN16 位元資料→ 32 位元資料 233
BIN16 位元資料→浮點數據轉換（單精確度） . . . 226
BIN16 位元資料→浮點數據轉換（雙精度） 228
BIN16 位元資料→格雷瑪轉換 235
BIN16 位元資料 2 進制補數（符號取反） 237
BIN16 位元資料比較 166
BIN16 位資料塊加法和減法運算 212
BIN16 位元死區控制 538
BIN32 位→ 10 進制 ASCII 碼轉換 433
BIN32 位→ 16 進制 ASCII 碼轉換 435
BIN32 位元→字串轉換 450
BIN32 位乘法和除法運算 188
BIN32 位加法和減法運算 184
BIN32 位元塊資料比較 178
BIN32 位元塊資料加法和減法運算 214
BIN32 位元區域控制 541
BIN32 位元上下限控制 536
BIN32 位元資料→ BIN16 位元資料轉換 234
BIN32 位元資料→浮點數據轉換（單精確度） . . . 226
BIN32 位元資料→浮點數據轉換（雙精度） 228
BIN32 位元資料→格雷瑪轉換 235
BIN32 位元資料 2 進制補數（符號取反） 237
BIN32 位元資料比較 167
BIN32 位元死區控制 538
BIN 資料→ BCD4 位元轉換 223
BIN 資料→ BCD8 位元轉換 223
報警器的重定 145

報警器的設置 145
報警器輸出 140
比較運算指令 166
比較運算指令一覽表 32
編程工具 24
變址繼電器運算結果脈衝化 132
變址寄存器的批量保存 596
變址寄存器的批量恢復 596
變址修飾 88
標度 (X/Y 座標資料) 546
標度 (點座標資料) 543
並行連接 120
步數的思路 106

C

CC-Link 24
CPU 的性能比較
  計時器 795
  計數器 796
  可用軟元件 793
  顯示指令 796
  指令中可以使用的資料 794
CPU 模組 23
程式待機 579
程式低速執行登錄 582
程式分支指令 265
程式分支指令一覽表 46
程式控制用指令一覽表 73
程式掃描執行登錄 581
程式輸出 OFF 待機 580
程式文件之間副程式調用 382
程式文件之間副程式輸出 OFF 調用 387
程式執行控制指令 269
程式執行控制指令一覽表 47
程式執行狀態檢查 583
出錯顯示或報警器重定 423
觸點指令 120
觸點指令一覽表 28
串列連接 120
從標準 ROM 中讀取資料 629
從程式記憶體中卸載程式 633
從其他站 CPU 共用記憶體讀取 660
從其他站 CPU 共用記憶體讀取指令一覽表 77
從其他站軟元件讀取 679
從資料表中讀取最舊資料 407
從資料表中讀取最新資料 409
從指定文件中讀取資料 617
從智慧功能模組的 1 字資料讀取 413
從智慧功能模組的 2 字資料讀取 413
從中斷程式的恢復 274
從副程式返回 377
從字串的右側提取資料 470
從字串的左側提取資料 470

D

單精確度→雙精度轉換 244
單相輸入加法 / 減法計數器 277
低速計時器 136
低速累計計時器 136
定時脈衝發生 586
多 CPU 高速通信專用指令一覽表 78

F

FOR ～ NEXT 371
FOR ～ NEXT 強制結束 372
浮點數據→ BCD 的分解 480
浮點數據→字符串的转换 458
浮點數據 COS^{-1} 運算 (單精確度) 496
浮點數據 COS^{-1} 運算 (雙精度) 498
浮點數據 COS 運算 (單精確度) 487
浮點數據 COS 運算 (雙精度) 488
浮點數據 SIN^{-1} 運算 (單精確度) 493
浮點數據 SIN^{-1} 運算 (雙精度) 495
浮點數據 SIN 運算 (單精確度) 450
浮點數據 SIN 運算 (雙精度) 485
浮點數據 TAN^{-1} 運算 (單精確度) 499
浮點數據 TAN^{-1} 運算 (雙精度) 501
浮點數據 TAN 運算 (單精確度) 490
浮點數據 TAN 運算 (雙精度) 492
浮點數據常用對數運算 (單精確度) 519
浮點數據常用對數運算 (雙精度) 520
浮點數據符號取反 (單精確度) 239
浮點數據符號取反 (雙精度) 240
浮點數據弧度→角度轉換 (單精確度) 505
浮點數據弧度→角度轉換 (雙精度) 506
浮點數據弧度→角度轉換 (單精確度) 502
浮點數據弧度→角度轉換 (雙精度) 503
浮點數據冪運算 (單精確度) 508
浮點數據冪運算 (雙精度) 509
浮點數據平方根 (單精確度) 510
浮點數據平方根 (雙精度) 512
浮點數據資料→ BIN16 位 (單精確度) 229
浮點數據資料→ BIN16 位 (雙精度) 231
浮點數據資料→ BIN32 位 (單精確度) 229
浮點數據資料→ BIN32 位 (雙精度) 231
浮點數據資料比較 (單精確度) 169
浮點數據資料比較 (雙精度) 171
浮點數據資料乘法和除法運算 (單精確度) 208
浮點數據資料乘法和除法運算 (雙精度) 210
浮點數據資料傳送 (單精確度) 247
浮點數據資料傳送 (雙精度) 248
浮點數據資料加法和減法運算 (單精確度) 200
浮點數據資料加法和減法運算 (雙精度) 204
浮點數據指數運算 (單精確度) 513
浮點數據指數運算 (雙精度) 515
浮點數據自然對數運算 (單精確度) 516
浮點數據自然對數運算 (雙精度) 518

G

GX Developer 24
GX Works2 24
改變檢查指令的檢查格式 429
高速計時器 136
高速累計計時器 136
高性能型 QCPU 23
高位元組和低位元組的交換 264
格雷瑪→ BIN16 位元資料轉換 236
格雷瑪→ BIN32 位元資料轉換 236

索

跟蹤重定 606
跟蹤設置 606
過程 CPU 23

H

恒定周期脈衝輸出 289
緩衝記憶體訪問指令一覽表 59

I

I/O 刷新 275
I/O 刷新指令 275
I/O 刷新指令一覽表 47

J

基本型 QCPU 23
計數器 139
間接地址讀取 591
間接指定 96
鍵盤的數位鍵輸入 592
教學計時器 281
結構化指令一覽表 57
結束指令 157
結束指令一覽表 31
矩陣輸入 292

K

塊 16 位元資料傳送 254
塊 16 位元資料交換 262
塊 BCD4 位元資料→塊 BIN16 位元資料的轉換 . . . 242
塊 BIN16 位元資料→塊 BCD4 位元資料的轉換 . . . 241
塊否定排他邏輯和 317
塊邏輯和 305
塊邏輯積 329
塊排他邏輯和 311
塊切換方式 116
擴展時鐘資料的讀取 571
擴展時鐘資料的加法運算 573
擴展時鐘資料的減法運算 575
擴展時鐘指令一覽表 72

L

LCPU 23
L 系列 23
連號訪問方式 176
連接指令 126
連接指令一覽表 29
兩相輸入加法 / 減法計數器 279
路由資訊的登錄 649
路由資訊的讀取 648
路由資訊的讀取／登錄指令 76
邏輯運算指令一覽表 49

M

MELSECNET(II、/B) 24
MELSECNET/10 24
MELSECNET/H 24
脈衝並行連接 122
脈衝串列連接 122
脈衝否定並行連接 124
脈衝否定串列連接 124
脈衝否定運算開始 124
脈衝寬度調製 290
脈衝密度的測定 288
脈衝運算開始 122
模組資訊讀取 598
模組型號讀取 602
目標 (D) 79

N

n 位元資料的 1 位元右移 330
n 位元資料的 1 位元左移 330
n 位元資料的 n 位元右移 332
n 位元資料的 n 位元左移 332
n 字資料的 1 字右移 334
n 字資料的 1 字左移 334
n 字資料的 n 字右移 336
n 字資料的 n 字左移 336

O

OUT(計時器、計數器、報警器除外) 134

Q

Q3 □ DB 24
Q5 □ B 24
Q6 □ B 24
Q6 □ WRB 24
QA1S6 □ B 24
QA6 □ B 24
QA6ADP 24
QA6ADP+A5 □ B/A6 □ B 24
QnCPU 24
QnHCPU 24
QnPHCPU 23
QnPRHCPU 23
QnU(D)(H)CPU 23
QnUCPU 23
QnUD(H)CPU 23
QnUDE(H)CPU 23
Q 系列 23
其他使用方便的指令 277
其他使用方便的指令一覽表 47
其他指令 161
其他指令一覽表 . . . . . 31、73
切換指令一覽表 69

R

任意資料的位元分離 354
任意資料的位元連接 354
日期比較 564
冗餘 CPU 23
冗餘系統用指令 (冗餘 CPU 用) 一覽表 78
軟元件的重定 (報警器除外) 143

軟元件的設置（報警器除外） 143
軟元件的注釋資料讀取 447
軟元件範圍檢查 100
軟元件數 / 傳送數 (n) 79

S

上升沿輸出 146
時間比較 567
時間檢查 587
時間資料的轉換（秒→時、分、秒） 563
時間資料的轉換（時、分、秒→秒） 561
時鐘資料的讀取 555
時鐘資料的加法運算 558
時鐘資料的減法運算 560
時鐘資料的寫入 556
時鐘用指令一覽表 69
輸出指令一覽表 30
輸入輸出控制方式 794
資料表操作指令一覽表 59
資料表的資料插入 411
資料表的資料刪除 411
資料表的防寫 406
資料處理指令一覽表 54
資料傳送指令 273
資料傳送指令一覽表 44
資料的指定方法 80
資料控制指令一覽表 67
資料轉換指令 223
資料轉換指令一覽表 42
刷新 395
雙精度→單精確度轉換 245
順控程式結束 159
順控程式停止 161
算術運算指令 181
算術運算指令一覽表 38
亂數的產生 521

T

特定格式故障檢查 426
特殊功能計時器 282
特殊函數指令 64
梯形圖塊並行連接 126
梯形圖塊串列連接 126
跳轉至 END 267
調試．故障診斷指令一覽表 60
通過存儲卡的程式裝載 631
通用型 QCPU 23
通用運算寄存器 (Z) 99
同一 16 位資料塊傳送 257
同一 32 位資料塊傳送 259

W

WDT 重定 585
網路刷新指令一覽表 76
位測試 339
位元處理指令一覽表 53
位軟元件的批量重定 341
位軟元件輸出取反 148
位軟元件移位 151
文件寄存器的塊號切換 549
文件寄存器高速塊傳送 636
文件寄存器用文件的設置 550
文件寄存器直接 1 位元組讀取 588
文件寄存器直接 1 位元組寫入 589
無處理 162

X

系列變更 521
系統切換 683
下降沿輸出 146
顯示指令一覽表 60
向標準 ROM 中寫入資料 628
斜坡信號 286
寫入資料到指定的文件 607
旋轉台的就近控制 284
旋轉指令一覽表 51
選擇刷新 . . . . . 397、400

Y

移位元指令 151
移位元指令一覽表 . . . . . 30、52
乙太網 24
用戶資訊 640
源 (S) 79
運算處理時間
LCPU 772
高性能型 QCPU/ 過程 CPU/ 冗餘 CPU 701
基本型 QCPU 687
通用型 QCPU 723
運算結果讀取 127
運算結果脈衝化 131
運算結果取反 130
運算結果入棧 127
運算結果退棧 127
運算開始 120

Z

整個梯形圖的變址修飾 400
整個梯形圖的變址修飾中修飾值的指定 403
直接輸出的脈衝化 149
指定格式更改的指令 797
指定模組刷新 644
指令的分類 25
指令的構成 79
指令的執行條件 105
指令一覽表的閱讀方法 26
指標分支 265
至其他機號的軟元件寫入 675
至智慧功能模組的 1 字資料寫入 415
至智慧功能模組的 2 字資料寫入 415
至自站 CPU 共用記憶體的寫入 . . . . . 652、655
至自站 CPU 共用記憶體的寫入指令一覽表 77
智慧功能模組軟元件 24
中斷程式遮罩 269
中斷禁止 269
中斷允許 269
主程序結束 157
主控的重定 153

索

主控的設置 153
主控指令 153
主控指令一覽表 31
注釋的列印 421
注釋用文件的設置 552
裝載＋卸載 634
副程式的輸出 OFF 調用 378
副程式調用 . . . . . 374、390
子集處理 98
字串→ BIN16 位元資料的轉換 454
字串→ BIN32 位元資料的轉換 454
字串→浮點數據的轉換 463
字串插入 477
字串處理指令一覽表 61
字串傳送 249
字串的長度檢測 449
字串的連接 217
字串刪除 479
字串資料比較 173
字串搜索 475
字串中的任意提取 472
字串中的任意替換 472
位元組單位資料分離 358
位元組單位資料連接 358
字軟元件的位重定 338
字軟元件的位設置 338

指令索引

Symbols

< 166
<= 166
<> 166
$+(P) 217
$MOV(P) 249
$= 173
$< 173
$<= 173
$< > 173
$> 173
$>= 173
-(P) 181
*(P) 187
+(P) 181
/(P) 187
= 166
> 166
>= 166

A

ACOS(P) 496
ACOSD(P) 498
ADRSET(P) 591
ANB 126
AND 120
ANDF 122
ANDFI 124
ANDP 122
ANDPI 124
ANI 120
ASC(P) 466
ASIN(P) 493
ASIND(P) 495
ATAN(P) 499
ATAND(P) 501

B

B-(P) 190
B*(P) 196
B+(P) 190
B/(P) 196
BACOS(P) 532
BAND(P) 538
BASIN(P) 530
BATAN(P) 534
BCD(P) 223
BCDDA(P) 438
BCOS(P) 526
BDSQR(P) 522
BIN(P) 224
BINDA(P) 433
BINHA(P) 435
BK-(P) 212
BK+(P) 212
BKAND(P) 299
BKBCD(P) 241
BKBIN(P) 242
BKCMP □ 175
BKCMP □ P 175
BKOR(P) 305
BKRST(P) 341
BKXNR(P) 317
BKXOR(P) 311
BMOV(P) 254
BREAK(P) 372
BRST(P) 338
BSET(P) 338
BSFL(P) 330
BSFR(P) 330
BSIN(P) 524
BSQR(P) 522
BTAN(P) 528
BTOW(P) 358
BXCH(P) 262

C

CALL(P) 374
CCOM(P) 400
CHK 428
CHKCIR 429
CHKEND 429
CHKST 428
CJ 265
CML(P) 251
COM . . . . . 395、397
COMRD(P) 447
COS(P) 487
COSD(P) 488

D

D-(P) 184
D(P).DDRD 679
D(P).DDWR 675
D*(P) 188
D+(P) 184
D/(P) 188
DABCD(P) 444
DABIN(P) 440
DAND(P) 295
DATE-(P) 560
DATE+(P) 558
DATERD(P) 555
DATEWR(P) 556
DB-(P) 193
DB*(P) 198
DB+(P) 193
DB/(P) 198
DBAND(P) 538
DBCD(P) 223
DBCDDA(P) 438
DBIN(P) 224
DBINDA(P) 433
DBINHA(P) 435
DBK-(P) 214
DBK+(P) 214
DBKCMP □ 178

索

DBKCMP □ P 178
DBL(P) 233
DCML(P) 251
DDABCD(P) 444
DDABIN(P) 440
DDEC(P) 221
DEC(P) 219
DECO(P) 348
DEG(P) 505
DEGD(P) 506
DELTA(P) 149
DFLT(P) 226
DFLTD(P) 228
DFMOV(P) 259
DFRO(P) . . . . . 413、660
DGBIN(P) 236
DGRY(P) 235
DHABIN(P) 443
DI 269
DINC(P) 221
DINT(P) 229
DINTD(P) 231
DIS(P) 352
DLIMIT(P) 536
DMAX(P) 361
DMEAN(P) 369
DMIN(P) 362
DMOV(P) 246
DNEG(P) 237
DOR(P) 301
DRCL(P) 326
DRCR(P) 324
DROL(P) 326
DROR(P) 324
DSCL(P) 543
DSCL2(P) 546
DSER(P) 343
DSFL(P) 334
DSFR(P) 334
DSORT 364
DSTR(P) 450
DSUM(P) 345
DT< 564
DT<= 564
DT<> 564
DT= 564
DT> 564
DT>= 564
DTEST(P) 339
DTO(P) . . . . . 415、655
DUTY 586
DVAL(P) 454
DWSUM(P) 368
DXCH(P) 261
DXNR(P) 313
DXOR(P) 307
DZONE(P) 541
D= 167
D< 167
D<= 167
D< > 167
D> 167
D>= 167

E

E-(P) 200
E*(P) 208
E+(P) 200
E/(P) 208
ECALL(P) 382
ECON(P) 244
ED-(P) 204
ED*(P) 210
ED+(P) 204
ED/(P) 210
EDCON(P) 245
EDMOV(P) 248
EDNEG(P) 240
ED=. 171
ED<. 171
ED<= 171
ED< > 171
ED>. 171
ED>= 171
EFCALL(P) 387
EGF. 132
EGP. 132
EI 269
EMOD(P) 480
EMOV(P) 247
ENCO(P) 348
END. 159
ENEG(P) 239
EREXP(P) 482
ESTR(P) 458
EVAL(P) 463
EXP(P) 513
EXPD(P) 515
E= 169
E< 169
E<=. 169
E< > 169
E> 169
E>=. 169

F

FCALL(P) 378
FDEL(P) 411
FEND 157
FF 148
FIFR(P) 407
FIFW(P) 406
FINS(P) 411
FLT(P) 226
FLTD(P) 228
FMOV(P) 257
FOR. 371
FPOP(P) 409
FROM(P) . . . . . 413、660

G

GBIN(P) 236
GOEND 267
GRY(P) 235

H

HABIN(P) 443
HEX(P) 468
HOUR(P) 563

I

IMASK 269
INC(P) 219
INSTR(P) 475
INT(P) 229
INTD(P) 231
INV 130
IRET 274
IX 400
IXDEV 403
IXEND 400
IXSET 403

J

JMP 265

K

KEY 592

L

LD 120
LDF 122
LDFI 124
LDI 120
LDP 122
LDPI 124
LEDR 423
LEFT(P) 470
LEN(P) 449
LIMIT(P) 536
LOG(P) 516
LOG10(P) 519
LOG10D(P) 520
LOGD(P) 518

M

MAX(P) 361
MC 153
MCR 153
MEAN(P) 369
MEF 131
MEP 131
MIDR(P) 472
MIDW(P) 472
MIN(P) 362
MOV(P) 246
MPP 127
MPS 127
MRD 127
MTR 292

N

NDIS(P) 354
NEG(P) 237
NEXT 371
NOP 162
NOPLF 162
NUNI(P) 354

O

OR 120
ORB 126
ORF 122
ORFI 124
ORI 120
ORP 122
ORPI 124
OUT 134
OUT C 139
OUT F 140
OUT T 136
OUTH T 136

P

PAGE n 162
PCHK 583
PLF 146
PLOADP 631
PLOW(P) 582
PLS 146
PLSY 289
POFF(P) 580
POW(P) 508
POWD(P) 509
PR 418
PRC 421
PSCAN(P) 581
PSTOP(P) 579
PSWAPP 634
PUNLOADP 633
PWM 290

Q

QCDSET(P) 552
QDRSET(P) 550

R

RAD(P) 502
RADD(P) 503
RAMP 286
RBMOV(P) 636
RCL(P) 321
RCR(P) 319
RET 377
RFS(P) 275
RIGHT(P) 470
RND(P) 521
ROL(P) 321

索

ROR(P) 319
ROTC 284
RSET(P) 549
RST 143
RST F 145

S

S(P).DATE- 575
S(P).DATE+ 573
S(P).DATERD 571
S(P).DEVLD 629
S(P).RTREAD 648
S(P).RTWRITE 649
S(P).TO 652
S(P).ZCOM 644
SCJ 265
SCL(P) 543
SCL2(P) 546
SECOND(P) 561
SEG(P) 350
SER(P) 343
SET 142
SET F 145
SFL(P) 328
SFR(P) 328
SFT(P) 151
SFTBL(P) 332
SFTBR(P) 332
SFTWL(P) 336
SFTWR(P) 336
SIN(P) 484
SIND(P) 485
SORT 364
SP.CONTSW 683
SP.DEVST 628
SP.FREAD 617
SP.FWRITE 607
SPD 288
SQR(P) 510
SQRD(P) 512
SRND(P) 521
STMR 282
STOP 161
STR(P) 450
STRDEL(P) 479
STRINS(P) 477
SUM(P) 345
SWAP(P) 264

T

TAN(P) 490
TAND(P) 491
TEST(P) 339
TIMCHK 587
TM< 567
TM<= 567
TM<> 567
TM= 567
TM> 567
TM>= 567
TO(P) . . . . . 415、655
TRACE 606
TRACER 606
TTMR 281
TYPERD(P) 602

U

UDCNT1 277
UDCNT2 279
UMSG 640
UNI(P) 353
UNIRD(P) 598

V

VAL(P) 454

W

WAND(P) 295
WDT(P) 585
WOR(P) 301
WORD(P) 234
WSUM(P) 367
WTOB(P) 358
WXNR(P) 313
WXOR(P) 307

X

XCALL 390
XCH(P) 261

Z

ZONE(P) 541
ZPOP(P) 596
ZPUSH(P) 596
ZRRDB(P) 588
ZRWRB(P) 589

質保

使用之前請確認以下產品質保的詳細說明。

1. 免費質保期限和免費質保範圍

在免費質保期內使用本產品時如果出現任何屬於三菱電機責任的故障或缺陷（以下稱“故障”），則經銷商或三菱電機服務公司將負責免費維修。
但是如果需要在國內現場或海外維修時，則要收取派遣工程師的費用。對於涉及到更換故障模組後的任何再試運轉、維護或現場測試，三菱電機將不負任何責任。
[免費質保期限]
免費質保期限爲自購買日或交貨的一年內。
注意產品從三菱電機生產並出貨之後，最長分銷時間爲 6 個月，生產後最長的免費質保期爲 18 個月。維修零部件的免費質保期不得超過修理前的免費質保期。
[免費質保範圍]

(1) 範圍局限於按照使用手冊、用戶手冊及產品上的警示標簽規定的使用狀態、使用方法和使用環境正常使用的情況下。
(2) 以下情況下，即使在免費質保期內，也要收取維修費用。
    1. 因不適當存儲或搬運、用戶過失或疏忽而引起的故障。因用戶的硬體或軟體設計而導致的故障。
    2. 因用戶未經批准對產品進行改造而導致的故障等。
    3. . 對於裝有三菱電機產品的用戶設備，如果根據現有的法定安全措施或工業標準要求配備必需的功能或結構後本可以避免的故障。
    4. 如果正確維護或更換了使用手冊中指定的耗材（電池、背光燈、保險絲等）後本可以避免的故障。
    5. 因火災或異常電壓等外部因素以及因地震、雷電、大風和水災等不可抗力而導致的故障。
    6. 根據從三菱出貨時的科技標準還無法預知的原因而導致的故障。
    7. 任何非三菱電機或用戶責任而導致的故障。

2. 產品停產後的有償維修期限

(1) 三菱電機在本產品停產後的 7 年內受理該產品的有償維修。
    停產的消息將以三菱電機技術公告等方式予以通告。
(2) 產品停產後，將不再提供產品（包括維修零件）。

3. 海外服務

在海外，維修由三菱電機在當地的海外 FA 中心受理。注意各個 FA 中心的維修條件可能會不同。

4. 意外損失和間接損失不在質保責任範圍內

無論是否在免費質保期內，對於任何非三菱電機責任的原因而導致的損失、機會損失、因三菱電機產品故障而引起的用戶利潤損失、無論能否預測的特殊損失和間接損失、事故賠償、除三菱電機以外產品的損失賠償、用戶更換設備、現場機械設備的再調試、運行測試及其它作業等，三菱電機將不承擔責任。

5. 產品規格的改變

目錄、手冊或技術文檔中的規格如有改變，恕不另行通知。

Microsoft、Windows、Windows NT、Windows Vista 是美國 Microsoft Corporation 在美國及其它國家的注冊商標。
Penium 是 Intel Corporation 在美國及其它國家的商標和注冊商標。
Ethernet 是美國 Xerox Corporation 的注冊商標。
本手冊中使用的其他公司名稱和產品名稱是各自公司的商標或注冊商標。

MELSEC-Q/L
編程手冊
公共指令篇

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

書號	SH(NA)-081065CHT-A(1204)MEACH
印號	MEACH-Q/LCPU-CI-PM(1204)

Specifications subject to change without notice.