

三菱電機 **通用** 可程式控制器

MELSEC iQ-R
series

MELSEC iQ-R 程式手冊
(CPU模組用指令/通用FUN/通用FB篇)

安全注意事項

(使用之前應務必閱讀)

使用MELSEC iQ-R系列可程式控制器前，請仔細閱讀各產品手冊及各產品手冊所介紹的關聯手冊，同時在充分注意安全的前提下正確地操作。

請妥善保管本手冊以備需要時查閱，並應將本手冊交給最終用戶。

關於產品的應用

(1) 使用三菱電機可程式控制器時，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失效安全防護功能。

(2) 三菱電機可程式控制器是以一般工業等用途為對象，設計和製造的泛用產品。

因此，三菱電機可程式控制器不適用於以下設備、系統的特殊用途上。如果用於以下特殊用途時，對於三菱電機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、違法行為責任、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，對公眾有較大影響的用途。
- 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、財產有較大影響的用途。

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電機可程式控制器，詳細情形請洽詢當地三菱電機代表窗口。

(3) 由於阻斷服務攻擊（DoS攻擊）、非法訪問、電腦病毒及其他網路攻擊而發生的系統方面的各種問題，三菱電機概不負責。

• 使用SIL2過程CPU時

(1) 儘管本產品已經自我聲明符合國際安全標準IEC61508和IEC61511，但這並不保證本產品不發生任何故障。本產品的用戶應遵守所有現行的安全標準、規則或法律，並應對本產品所安裝或使用的系統採取適當的安全措施，除了本產品之外還應當同時採取其它的安全措施。對於如果遵守了現行的安全標準、規則或法律而可以預防的損害，三菱電機公司（簡稱三菱電機）不負任何責任。

(2) 三菱電機禁止將本產品用於可能涉及人員生命健康安全和重大財產安全的用途，如果違反了三菱電機的指示將其用於以下用途，對於由此引起的一切責任（包括但不僅限於債務未履行責任、瑕疵擔保責任、品質保證責任、違法行為責任、製造物責任），三菱電機將不負責。

- 1) 火力/水力/核能發電廠
- 2) 火車/鐵路系統、飛機、航空管理、其它交通系統
- 3) 醫院、醫療及與生命維持相關設備的應用
- 4) 娛樂設備
- 5) 焚燒和燃料裝置
- 6) 核物質、有害物質及化學物質的處理設備
- 7) 採礦、挖掘
- 8) 其它上述1)~7)中未包含的涉及人員生命、健康或重大財產安全的用途

(3) 由於阻斷服務攻擊（DoS攻擊）、非法訪問、電腦病毒及其他網路攻擊而發生的可程式控制器及系統方面的各種問題，三菱電機概不負責。

• 使用安全CPU時

- (1) 儘管安全控制器已經取得了德國TUV Rheinland的國際安全標準IEC61508和ISO13849-1的產品可靠性認證，但這並不保證本產品不發生任何故障。本產品的用戶應遵守所有現行的安全標準、規則或法律，並應對本產品所安裝或使用的系統採取適當的安全措施，除了本產品之外還應當同時採取其它的安全措施。對於如果遵守了現行的安全標準、規則或法律而可以預防的損害，三菱電機公司(簡稱三菱電機)不負任何責任。
- (2) 三菱電機禁止將本產品用於可能涉及人員生命健康安全和重大財產安全的用途，如果違反了三菱電機的指示將其用於以下用途，對於由此引起的一切責任(包括但不僅限於債務未履行責任、瑕疵擔保責任、質量保證責任、違法行為責任、製造物責任)，三菱電機將不負責。
 - 1) 火力/水力/核能發電廠
 - 2) 火車/鐵路系統、飛機、航空管理、其它交通系統
 - 3) 醫院、醫療及與生命維持相關設備的應用
 - 4) 娛樂設備
 - 5) 焚燒和燃料裝置
 - 6) 核物質、有害物質及化學物質的處理設備
 - 7) 採礦、挖掘
 - 8) 其它上述1)~7)中未包含的涉及人員生命、健康或重大財產安全的用途
- (3) 由於阻斷服務攻擊(DoS攻擊)、非法訪問、電腦病毒及其他網路攻擊而發生的系統方面的各種問題，三菱電機概不負責。

前言

非常感謝貴方購買了三菱電機可程式控制器MELSEC iQ-R系列產品。

本手冊是用於讓用戶了解進程式時必要的指令、通用函數和通用FB相關內容的手冊。

在使用之前應熟讀本手冊及關聯手冊，在充分了解MELSEC iQ-R系列可程式控制器的功能及性能的基礎上正確使用本產品。

此外，將本手冊中介绍的程式示例應用於實際系統的情況下，應充分驗證對象系統中不存在控制方面的問題。

應將本手冊交給最終用戶。

目錄

安全注意事項	1
關於產品的應用	1
前言	3
關聯手冊	21
術語	22
總稱/簡稱	23
手冊閱讀方法	25

第1部分 概要

第1章 指令概要	30
1.1 指令的構成	30
1.2 資料的指定方法	32
位元資料	36
16位元資料(字元資料)	38
32位元資料(雙字元資料)	41
實數資料(浮點資料)	44
字元串資料	47
1.3 執行條件	49
1.4 指令處理時間的高速化	50
子集處理	50
1.5 程式時的注意事項	51
指令通用的出錯	51
執行指令時的元件、標籤的範圍檢查	51
使用超長定時器、超長累計定時器元件時的動作	54
使用同一元件的OUT指令、SET/RST指令、PLS/PLF指令時的動作	56
使用檔案寄存器時的限制事項	62

第2部分 指令/FUN/FB一覽

第2章 CPU模組用指令	64
2.1 順控程式指令	64
2.2 基本指令	68
2.3 應用指令	90
程式控制	90
資料處理	92
偵錯、故障診斷	100
字元串處理	100
實數處理	102
隨機數	109
元件操作	110
定時器、計數器	111
就近控制	112
斜坡信號	112
矩陣輸入	112

CPU模組內建資料庫存取功能	113
時鐘	114
模組存取	117
參數設定操作	119
CPU模組資料記錄功能	119
記錄功能	119
內建乙太網路功能用指令	120
PID運算指令	122
PID控制指令	122
過程控制指令	124
多CPU之間專用指令	125
SFC程式用指令	126
二重化系統用指令	128
安全系統用指令	129
第3章 模組專用指令	131
第4章 通用函數/通用FB	133
4.1 通用函數	133
4.2 通用FB	144
第3部分 順控程式指令	
第5章 順控程式指令	148
5.1 觸點指令	148
運算開始、串聯連接、並聯連接	148
脈衝運算開始、脈衝串聯連接、脈衝並聯連接	150
脈衝否定運算開始、脈衝否定串聯連接、脈衝否定並聯連接	153
5.2 合併指令	156
梯形圖塊串聯連接、並聯連接	156
運算結果推入、讀取、彈出	157
運算結果反轉	159
運算結果脈衝化	160
變址繼電器運算結果脈衝化	161
5.3 輸出指令	163
OUT(定時器、計數器、報警器除外)	163
定時器	165
超長定時器	168
計數器	171
超長計數器	173
報警器	175
元件的設定(報警器除外)	176
元件的重設(報警器除外)	178
報警器的設定	180
報警器的重設	182
上升沿輸出	184
下降沿輸出	186
位元件輸出反轉	188
直接輸出的脈衝化	190

5.4	移位指令	192
	位元元件移位	192
5.5	主控制指令	194
	主控制的設定、重設	194
5.6	階段處理指令	198
	概要	198
	階段處理開始	199
	執行階段的移轉	201
	執行階段的結束	203
5.7	結束指令	204
	主程式結束	204
	順控程式結束	205
5.8	停止指令	207
	順控程式停止	207
5.9	無處理指令	208
	無處理 (NOP)	208
	無處理 (NOPLF)	209

第4部分 基本指令

第6章 基本指令 212

6.1	比較運算指令	212
	BIN16位元資料比較	212
	BIN32位元資料比較	214
	BIN16位元資料比較輸出	216
	BIN32位元資料比較輸出	218
	BIN16位元資料區域比較輸出	220
	BIN32位元資料區域比較輸出	222
	BIN16位元塊資料比較	224
	BIN32位元塊資料比較	226
6.2	算術運算指令	229
	BIN16位元加法運算	229
	BIN16位元減法運算	233
	BIN32位元加法運算	237
	BIN32位元減法運算	241
	BIN16位元乘法運算	245
	BIN16位元除法運算	247
	BIN32位元乘法運算	249
	BIN32位元除法運算	251
	BCD4位加法運算	253
	BCD4位減法運算	256
	BCD8位加法運算	259
	BCD8位減法運算	262
	BCD4位乘法運算	265
	BCD4位除法運算	267
	BCD8位乘法運算	269
	BCD8位除法運算	271
	BIN16位元塊資料加法運算	273
	BIN16位元塊資料減法運算	275
	BIN32位元塊資料加法運算	277

	BIN32位元塊資料減法運算	280
	16位元BIN資料遞增	283
	16位元BIN資料遞減	285
	32位元BIN資料遞增	287
	32位元BIN資料遞減	289
6.3	邏輯運算指令	291
	16位元資料邏輯且	291
	32位元資料邏輯且	295
	16位元塊資料邏輯且	299
	16位元資料邏輯或	301
	32位元資料邏輯或	305
	16位元塊資料邏輯或	309
	16位元資料排他邏輯或	311
	32位元資料排他邏輯或	315
	16位元塊資料排他邏輯或	319
	16位元資料否定排他邏輯或	321
	32位元資料否定排他邏輯或	325
	16位元塊資料否定排他邏輯或	329
6.4	位元處理指令	331
	字元元件的位設定	331
	字元元件的位元重設	333
	16位元測試	335
	32位元測試	337
	位元元件的批量重設	339
6.5	資料移位指令	341
	BIN16位元資料的n位元右移	341
	BIN16位元資料的n位元左移	343
	n位元資料的1位元右移	345
	n位元資料的1位元左移	347
	n字元資料的1字元右移	349
	n字元資料的1字元左移	351
	n雙字元資料的1雙字元右移	353
	n雙字元資料的1雙字元左移	355
	單精度實數n點的1點右移	357
	單精度實數n點的1點左移	359
	雙精度實數n點的1點右移	361
	雙精度實數n點的1點左移	363
	n位元資料的n位元右移	365
	n位資料的n位元左移	369
	n字資料的n字元右移	373
	n字元資料的n字元左移	377
	n雙字元資料的n雙字元右移	381
	n雙字元資料的n雙字元左移	385
	單精度實數n點的n點右移	389
	單精度實數n點的n點左移	393
	雙精度實數n點的n點右移	397
	雙精度實數n點的n點左移	401
6.6	資料轉換指令	405
	BIN資料→BCD4位轉換	405
	BIN資料→BCD8位轉換	407
	BCD4位→BIN16位元資料轉換	409

BCD8位→BIN32位元資料轉換	411
單精度實數→帶符號BIN16位元資料	413
單精度實數→無符號BIN16位元資料	415
單精度實數→帶符號BIN32位元資料	417
單精度實數→無符號BIN32位元資料	419
雙精度實數→帶符號BIN16位元資料	421
雙精度實數→無符號BIN16位元資料	423
雙精度實數→帶符號BIN32位元資料	425
雙精度實數→無符號BIN32位元資料	427
帶符號BIN16位元資料→無符號BIN16位元資料轉換	429
帶符號BIN16位元資料→帶符號BIN32位元資料轉換	431
帶符號BIN16位元資料→無符號BIN32位元資料轉換	433
無符號BIN16位元資料→帶符號BIN16位元資料轉換	435
無符號BIN16位元資料→帶符號BIN32位元資料轉換	437
無符號BIN16位元資料→無符號BIN32位元資料轉換	439
帶符號BIN32位元資料→帶符號BIN16位元資料轉換	441
帶符號BIN32位元資料→無符號BIN16位元資料轉換	443
帶符號BIN32位元資料→無符號BIN32位元資料轉換	445
無符號BIN32位元資料→帶符號BIN16位元資料轉換	447
無符號BIN32位元資料→無符號BIN16位元資料轉換	449
無符號BIN32位元資料→帶符號BIN32位元資料轉換	451
BIN16位元資料→BIN格雷碼資料轉換	453
BIN32位元資料→格雷碼資料轉換	455
BIN16位元格雷碼資料→BIN16位元資料轉換	457
BIN32位元格雷碼資料→BIN32位元資料轉換	459
BIN16位元資料塊→BCD4位資料轉換塊	461
塊BCD4位資料→塊BIN16位元轉換資料	463
10進制ASCII資料→BIN16位元資料轉換	465
10進制ASCII資料→BIN32位元資料轉換	468
16進制ASCII資料→BIN16位元資料轉換	471
16進制ASCII資料→BIN32位元資料轉換	473
10進制ASCII資料→BCD4位資料轉換	476
10進制ASCII資料→BCD8位資料轉換	479
10進制字元串→BIN16位元資料轉換	482
10進制字元串→BIN32位元資料轉換	485
16進制ASCII→16進制資料BIN轉換	488
單精度實數→BCD分解	490
BIN16位元資料2的補數(符號反轉)	492
BIN32位元資料2的補數(符號反轉)	493
8→256位元解碼	494
256→8位編碼	496
7段解碼	498
BIN16位元資料的4位元分離	500
BIN16位元資料的4位合併	502
任意資料的位分離	504
任意資料的位元合併	506
字節單位資料分離	508
字節單位資料合併	510
6.7 資料傳送指令	512
BIN16位元資料傳送	512
BIN32位元資料傳送	514

BIN16位元資料否定傳送	516
BIN32位元資料否定傳送	518
位移動	520
1位元資料反轉傳送	523
BIN16位元資料塊傳送(16位元)	525
BIN16位元資料塊傳送(32位元)	527
同一BIN16位元資料塊傳送(16位元)	529
同一BIN16位元資料塊傳送(32位元)	531
同一BIN32位元資料塊傳送(16位元)	533
同一BIN32位元資料塊傳送(32位元)	535
BIN16位元資料交換	537
BIN32位元資料交換	539
BIN16位元資料塊交換	541
BIN16位元資料上下字節交換	543
BIN32位元資料上下字節交換	544
1位元資料傳送	546
n位資料傳送	548

第5部分 應用指令

第7章 程式控制 553

7.1 程式分支指令	553
指標分支	553
跳轉至END	556
7.2 程式執行控制指令	557
中斷禁止、中斷允許	557
指定優先度及其以下的中斷禁止	559
中斷程式屏蔽	564
指定中斷指標的禁止/允許	566
從中斷程式返回	568
WDT重設	569
7.3 結構化指令	570
FOR~NEXT	570
FOR~NEXT強制結束	572
子程式調用	574
從子程式的返回	578
子程式的輸出OFF調用	579
程式檔案之間子程式調用	583
程式檔案之間子程式輸出OFF調用	588
子程式調用	592
7.4 程式控制用指令	597
程式待機	597
程式輸出OFF待機	599
程式掃描執行登錄	601

第8章 資料處理 603

8.1 旋轉指令	603
BIN16位元資料的右旋轉	603
BIN16位元資料的左旋轉	606
BIN32位元資料的右旋轉	609

	BIN32位元資料的左旋轉	611
8.2	資料表操作指令	613
	從資料表的先輸入資料的讀取	613
	從資料表的後輸入資料的讀取	615
	至資料表的資料寫入	617
	資料表的資料插入	619
	資料表的資料刪除	621
8.3	資料讀取/寫入指令	623
	從資料記憶體的資料讀取	624
	至資料記憶體的資料寫入	626
8.4	檔案操作指令	628
	從指定檔案的資料讀取	628
	至指定檔案的資料寫入	645
	刪除指定檔案	661
	複製指定檔案	664
	移動指定檔案	669
	更改指定檔案名稱	674
	取得指定檔案的狀態	677
	在檔案操作指令中發生的出錯代碼	680
8.5	資料控制指令	681
	BIN16位元資料上下限限位控制	681
	BIN32位元資料上下限限位控制	683
	BIN16位元資料死區控制	685
	BIN32位元資料死區控制	687
	BIN16位元資料區域控制	689
	BIN32位元資料區域控制	691
	BIN16位元單位定標(各點座標資料)	693
	BIN32位元單位定標(各點座標資料)	696
	BIN16位元單位定標(各X/Y座標資料)	699
	BIN32位元單位定標(各X/Y座標資料)	701
8.6	資料處理指令	703
	BIN16位元資料搜尋	703
	BIN32位元資料搜尋	705
	BIN16位元資料搜尋(最小・相同・最大)	707
	BIN32位元資料搜尋(最小・相同・最大)	709
	BIN16位元資料位元檢查	711
	BIN32位元資料位元檢查	712
	BIN16位元資料的位元判定	713
	BIN32位元資料的位元判定	715
	BIN16位元資料最大值搜尋	717
	BIN32位元資料最大值搜尋	719
	BIN16位元資料最小值搜尋	721
	BIN32位元資料最小值搜尋	723
	BIN16位元資料排序	725
	BIN32位元資料排序	727
	BIN16位元資料表排序	729
	BIN16位元資料表排序2	733
	BIN32位元資料表排序2	737
	BIN16位元資料合計值計算	741
	BIN32位元資料合計值計算	743
	BIN16位元資料平均值計算	745

BIN32位元資料平均值計算	747
計算BIN16位元平方根	749
計算BIN32位元平方根	750
CRC運算	751
8.7 檢查代碼指令	753
檢查代碼	753
第9章 偵錯、故障診斷	756
9.1 偵錯、故障診斷指令	756
出錯顯示或報警器重設	756
運轉繼續異常指令	758
運轉停止異常指令	760
第10章 字元串處理	762
10.1 字元串處理指令	762
字元串比較	762
字元串的合併	766
字元串傳送	770
Unicode對應字元串傳送	772
BIN16位元資料→10進制ASCII轉換	774
BIN32位元資料→10進制ASCII轉換	778
BIN16位元資料→16進制ASCII轉換	783
BIN32位元資料→16進制ASCII轉換	787
BIN16位元資料→字元串轉換	791
BIN32位元資料→字元串轉換	794
BCD4位資料→10進制ASCII碼轉換	797
BCD8位資料→10進制ASCII碼轉換	801
單精度實數→字元串轉換	806
16進制BIN資料→16進制ASCII碼轉換	810
Unicode字元串→移位JIS字元串轉換	812
移位JIS字元串→Unicode字元串轉換(無字節順序標記)	814
移位JIS字元串→Unicode轉換(有字節順序標記)	816
字元串的長度檢測	818
從字元串的右側開始提取	820
從字元串的左側開始提取	822
字元串中的任意提取	824
字元串中的任意替換	826
字元串搜尋	829
字元串插入	831
字元串刪除	833
第11章 實數處理	835
11.1 浮點指令	835
單精度實數比較	835
雙精度實數比較	838
單精度實數比較輸出	841
雙精度實數比較輸出	843
單精度實數區域比較輸出	845
雙精度實數區域比較輸出	847
單精度實數加法運算	849

單精度實數減法運算	853
雙精度實數加法運算	857
雙精度實數減法運算	861
單精度實數乘法運算	865
單精度實數除法運算	867
雙精度實數乘法運算	869
雙精度實數除法運算	871
帶符號BIN16位元資料→單精度實數轉換	873
無符號BIN16位元資料→單精度實數轉換	874
帶符號BIN32位元資料→單精度實數轉換	875
無符號BIN32位元資料→單精度實數轉換	877
雙精度實數→單精度實數轉換	879
帶符號BIN16位元資料→雙精度實數轉換	881
無符號BIN16位元資料→雙精度實數轉換	882
帶符號BIN32位元資料→雙精度實數轉換	883
無符號BIN32位元資料→雙精度實數轉換	884
單精度實數→雙精度實數轉換	885
字元串→單精度實數轉換	887
BCD格式資料→單精度實數資料轉換	891
單精度實數符號反轉	893
雙精度實數符號反轉	894
單精度實數資料傳送	895
雙精度實數資料傳送	896
單精度實數SIN運算	897
單精度實數COS運算	899
單精度實數TAN運算	901
單精度實數 SIN^{-1} 運算	903
單精度實數 COS^{-1} 運算	905
單精度實數 TAN^{-1} 運算	907
雙精度實數SIN運算	909
雙精度實數COS運算	911
雙精度實數TAN運算	913
雙精度實數 SIN^{-1} 運算	915
雙精度實數 COS^{-1} 運算	917
雙精度實數 TAN^{-1} 運算	919
BCD型SIN運算	921
BCD型COS運算	923
BCD型TAN運算	925
BCD型 SIN^{-1} 運算	927
BCD型 COS^{-1} 運算	929
BCD型 TAN^{-1} 運算	931
單精度實數角度→弧度轉換	933
單精度實數弧度→角度轉換	935
雙精度實數角度→弧度轉換	937
雙精度實數弧度→角度轉換	939
單精度實數平方根	941
雙精度實數平方根	943
單精度實數指數運算	945
雙精度實數指數運算	947
單精度實數自然對數運算	949
雙精度實數自然對數運算	951

BCD4位平方根	953
BCD8位平方根	955
單精度實數冪運算	957
雙精度實數冪運算	959
單精度實數常用對數運算	961
雙精度實數常用對數運算	963
單精度實數最大值搜尋	965
雙精度實數最大值搜尋	967
單精度實數最小值搜尋	969
雙精度實數最小值搜尋	971
第12章 隨機數	973
12.1 隨機數指令	973
隨機數發生	973
系列更改	974
第13章 元件操作	975
13.1 變址寄存器指令	975
變址寄存器的批量儲存	975
變址寄存器的批量恢復	977
變址寄存器/超長變址寄存器選擇儲存	978
變址寄存器/超長變址寄存器選擇恢復	981
13.2 檔案寄存器操作指令	983
檔案寄存器的塊No. 切換	983
檔案寄存器用檔案的設定	985
13.3 檔案寄存器的1字節單位的讀取/寫入指令	987
檔案寄存器的1字節資料讀取	987
檔案寄存器的1字節資料寫入	989
13.4 間接地址讀取指令	991
間接地址讀取	991
第14章 定時器、計數器	993
14.1 特殊計數器指令	993
單相輸入升值/降值計數器	993
2相輸入升值/降值計數器	996
14.2 特殊定時器指令	998
示教定時器	998
特殊功能定時器	1000
14.3 脈衝系統指令	1002
脈衝密度的測定	1002
恆定週期脈衝輸出	1004
脈衝寬度調制	1006
第15章 就近控制	1008
15.1 就近控制指令	1008
旋轉檯的就近控制	1008
第16章 斜坡信號	1011
16.1 斜坡信號指令	1011
斜坡信號	1011

第17章 矩陣輸入	1014
17.1 矩陣輸入指令	1014
矩陣輸入	1014
第18章 CPU模組內建資料庫存取功能	1018
18.1 資料庫存取指令	1018
資料庫導入	1018
資料庫導出	1021
資料庫連接	1023
資料庫斷開	1025
資料庫記錄添加	1027
資料庫記錄更新	1034
資料庫記錄搜尋	1040
資料庫記錄刪除	1048
資料庫事物處理開始	1052
資料庫提交	1054
資料庫回滾	1056
資料庫存取指令中發生的出錯代碼	1058
第19章 時鐘	1062
19.1 時鐘用指令	1062
時鐘資料的讀取	1062
時鐘資料的寫入	1064
時鐘資料的加法運算	1066
時鐘資料的減法運算	1068
時間資料的轉換(時分秒→秒)	1070
時間資料的轉換(秒→時分秒)	1072
日期時間資料的轉換(日期時間→秒)	1074
日期時間資料的轉換(秒→日期時間)	1076
日期比較	1078
時間比較	1082
時間比較輸出	1085
時間區域比較輸出	1087
擴展時鐘資料的讀取	1089
擴展時鐘資料的加法運算	1091
擴展時鐘資料的減法運算	1093
19.2 時機計測指令	1095
時機脈衝發生	1095
指定資料的時間計測	1097
計時器	1099
第20章 模組存取	1103
20.1 模組存取指令	1103
I/O更新	1103
選擇更新	1105
模組更新	1107
從模組中的1字/2字資料讀取(16位元指定)	1109
至模組的1字元/2字元資料寫入(16位元指定)	1113
從模組中的1字元/2字元資料讀取(32位元指定)	1118
至模組的1字元/2字元資料寫入(32位元指定)	1122

模組型號讀取	1127
模組特有資訊讀取	1131
第21章 參數設定操作	1136
21.1 路由資訊指令	1136
路由資訊的讀取	1136
路由資訊的登錄	1138
第22章 CPU模組資料記錄功能	1140
22.1 記錄用指令	1140
觸發記錄設定	1140
觸發記錄重設	1142
第23章 記錄功能	1143
23.1 資料採集指令	1143
資料採集觸發設定	1143
第24章 內建乙太網路功能用指令	1145
24.1 打開/關閉處理指令	1145
連接的建立	1145
連接的斷開	1148
24.2 套接字通訊用指令	1150
接收資料的END處理時讀取	1150
執行接收資料的指令時讀取	1153
資料發送	1155
連接資訊的讀取	1158
連接的通訊目標更改(UDP/IP)	1160
連接的接收模式更改	1162
套接字通訊接收資料讀取	1165
24.3 通訊協定支援功能指令	1167
登錄協定執行	1167
24.4 SLMP幀發送指令	1174
SLMP幀發送	1174
24.5 檔案傳送功能用指令	1181
發送FTP客戶端檔案	1181
獲取FTP客戶端檔案	1186
第25章 PID運算指令	1192
25.1 概要	1192
控制用資料	1194
自動調諧	1201
25.2 PID運算指令	1203
第26章 PID控制指令	1207
26.1 概要	1207
運算方式	1207
PID控制步驟	1208
其它功能	1212
26.2 PID控制指令(不完全微分)	1214
PID控制用資料的設定	1216

PID運算	1219
指定環路No. 的運算停止	1222
指定環路No. 的運算開始	1223
指定環路No. 的參數更改	1224
26.3 PID控制指令(完全微分)	1226
PID控制用資料的設定	1228
PID運算	1230
指定環路No. 的運算停止	1233
指定環路No. 的運算開始	1234
指定環路No. 的參數更改	1235

第27章 多CPU之間專用指令 1237

27.1 其它機號CPU模組存取指令	1237
從其它機號CPU模組的元件讀取	1241
至其它機號CPU模組的元件寫入	1244

第28章 SFC程式用指令 1247

28.1 SFC控制指令	1247
步活性檢查	1247
塊激活檢查	1249
激活步批量讀取	1251
塊啟動	1259
塊結束	1261
塊停止	1263
塊重啟	1265
步啟動	1267
步結束	1269
切換對象塊	1271
28.2 SFC專用指令	1273
移轉條件虛擬輸出	1273

第29章 二重化系統用指令 1274

29.1 系統切換	1274
29.2 系統切換禁止/允許	1278
29.3 由待機系統寫入資料至控制系統	1280

第30章 安全系統用指令 1284

30.1 安全資料一致性檢查資訊讀取指令	1284
在安全系統用指令中發生的出錯代碼	1286

第6部分 模組專用指令

第31章 模組專用指令 1288

第7部分 通用FUN

第32章 類型轉換函數 1292

32.1 BOOL型→WORD型轉換	1292
32.2 BOOL型→DWORD型轉換	1293

32.3	BOOL型→INT型轉換	1294
32.4	BOOL型→DINT型轉換	1295
32.5	BOOL型→TIME型轉換	1296
32.6	BOOL型→STRING型轉換	1297
32.7	WORD型→BOOL型轉換	1298
32.8	WORD型→DWORD型轉換	1299
32.9	WORD型→INT型轉換	1300
32.10	WORD型→DINT型轉換	1301
32.11	WORD型→TIME型轉換	1303
32.12	WORD型→STRING型轉換	1304
32.13	DWORD型→BOOL型轉換	1305
32.14	DWORD型→WORD型轉換	1306
32.15	DWORD型→INT型轉換	1308
32.16	DWORD型→DINT型轉換	1310
32.17	DWORD型→TIME型轉換	1311
32.18	DWORD型→STRING型轉換	1312
32.19	INT型→BOOL型轉換	1313
32.20	INT型→WORD型轉換	1314
32.21	INT型→DWORD型轉換	1315
32.22	INT型→DINT型轉換	1317
32.23	INT型→BCD型轉換	1318
32.24	INT型→REAL型轉換	1320
32.25	INT型→LREAL型轉換	1321
32.26	INT型→TIME型轉換	1322
32.27	INT型→STRING型轉換	1323
32.28	DINT型→BOOL型轉換	1325
32.29	DINT型→WORD型轉換	1326
32.30	DINT型→DWORD型轉換	1328
32.31	DINT型→INT型轉換	1329
32.32	DINT型→BCD型轉換	1330
32.33	DINT型→REAL型轉換	1332
32.34	DINT型→LREAL型轉換	1333
32.35	DINT型→TIME型轉換	1334
32.36	DINT型→STRING型轉換	1335
32.37	BCD型→INT型轉換	1337
32.38	BCD型→DINT型轉換	1339
32.39	BCD型→STRING型轉換	1342
32.40	REAL型→INT型轉換	1344
32.41	REAL型→DINT型轉換	1346
32.42	REAL型→LREAL型轉換	1348
32.43	REAL型→STRING型轉換	1350
32.44	LREAL型→INT型轉換	1353
32.45	LREAL型→DINT型轉換	1354
32.46	LREAL型→REAL型轉換	1355
32.47	TIME型→BOOL型轉換	1356
32.48	TIME型→WORD型轉換	1357
32.49	TIME型→DWORD型轉換	1358
32.50	TIME型→INT型轉換	1359
32.51	TIME型→DINT型轉換	1360
32.52	TIME型→STRING型轉換	1361
32.53	STRING型→BOOL型轉換	1363

32.54	STRING型→WORD型轉換	1364
32.55	STRING型→DWORD型轉換	1365
32.56	STRING型→INT型轉換	1366
32.57	STRING型→DINT型轉換	1368
32.58	STRING型→BCD型轉換	1370
32.59	STRING型→REAL型轉換	1372
32.60	STRING型→TIME型轉換	1375
32.61	位元數組→INT型轉換	1376
32.62	位元數組→DINT型轉換	1377
32.63	INT型→位元數組轉換	1378
32.64	DINT型→位元數組轉換	1379
32.65	位元數組的複製	1380
32.66	字元標籤的指定位元讀取	1381
32.67	字元標籤的指定位元寫入	1383
32.68	字元標籤的指定位元複製	1385
32.69	起始資料的獲取	1387

第33章 單數值變數函數 1388

33.1	絕對值	1388
33.2	平方根	1390
33.3	自然對數運算	1391
33.4	常用對數運算	1392
33.5	指數運算	1394
33.6	SIN運算	1395
33.7	COS運算	1396
33.8	TAN運算	1397
33.9	SIN^{-1} 運算	1398
33.10	COS^{-1} 運算	1399
33.11	TAN^{-1} 運算	1400

第34章 算術運算函數 1401

34.1	加法運算	1401
34.2	乘法運算	1404
34.3	減法運算	1406
34.4	除法運算	1409
34.5	餘數	1411
34.6	冪	1413
34.7	代入	1414

第35章 位移函數 1416

35.1	n位元左移	1416
35.2	n位元右移	1418
35.3	n位元左旋轉	1420
35.4	n位元右旋轉	1422

第36章 位型布爾函數 1424

36.1	邏輯且、邏輯或、排他邏輯或	1424
36.2	邏輯否定	1427

第37章 選擇函數	1428
37.1 選擇值	1428
37.2 最大值、最小值選擇	1430
37.3 上下限位控制	1432
37.4 多路復用器	1435
第38章 比較函數	1437
38.1 比較	1437
38.2 比較	1439
第39章 字元串函數	1441
39.1 字元串的長度檢測	1441
39.2 從字元串的左側、右側提取	1443
39.3 字元串的提取	1445
39.4 字元串的合併	1447
39.5 字元串的插入	1449
39.6 字元串的刪除	1451
39.7 字元串的替換	1453
39.8 字元串的搜尋	1456
第40章 時間資料類型函數	1458
40.1 加法運算	1458
40.2 減法運算	1460
40.3 乘法運算	1462
40.4 除法運算	1464
第8部分 通用FB	
第41章 雙穩態FB	1468
41.1 雙穩態FB(設定優先)	1468
41.2 雙穩態FB(重設優先)	1470
第42章 邊緣檢測FB	1472
42.1 上升沿邊緣檢測	1472
42.2 下降沿邊緣檢測	1474
第43章 計數器/定時器FB	1476
43.1 升值計數器	1476
43.2 降值計數器	1478
43.3 升值降值計數器	1480
43.4 計數器的FB	1483
43.5 脈衝定時器	1485
43.6 ON延遲定時器	1487
43.7 OFF延遲定時器	1489
43.8 定時器FB	1491
附錄	1495
附1 指令處理時間	1495
指令處理時間的加法運算時間	1524

附2	基本步數及子集處理的可否	1525
附3	PID的3常數的求出方法	1550
附4	PID運算程式示例	1552
	自動調諧(步響應法)+PID控制的程式示例	1553
	只有自動調諧(步響應法)的程式示例	1555
附5	PID控制程式示例	1557
	透過自動模式進行的PID控制程式示例	1557
	模式切換時的程式示例	1564
附6	從其他格式工程的替換	1571
	從GX Works2格式工程的替換	1571
	從PX Developer格式工程的替換	1588

索引 **1590**

指令索引 **1592**

修訂記錄	1604
保固	1605
商標	1606

關聯手冊

要取得最新的e-Manual以及手冊PDF，請向當地三菱電機代理店諮詢。

手冊名稱[手冊編號]	內容	提供形式
MELSEC iQ-R 程式手冊 (CPU模組用指令/通用FUN/通用FB篇) [SH-081323CHT] (本手冊)	記載CPU模組的指令、通用函數/通用FB有關內容。	e-Manual PDF
MELSEC iQ-R 程式手冊 (模組專用指令篇) [SH-081978CHT]	記載智能功能模組的專用指令有關內容。	e-Manual PDF
MELSEC iQ-R 程式手冊 (過程控制FB/指令篇) [SH-081751CHT]	記載過程控制中特有的通用過程FB、標籤存取FB、標籤FB及過程控制指令有關內容。	e-Manual PDF
MELSEC iQ-R 程式手冊 (程式設計篇) [SH-081320CHT]	記載梯形圖、ST、FBD/LD、SFC的程式規格有關內容。	e-Manual PDF
GX Works3 操作手冊 [SH-081272CHT]	對GX Works3的系統配置、參數設定、在線功能的操作方法等有關內容進行說明。	e-Manual PDF

要點

e-Manual是指可透過使用專用工具瀏覽的三菱電機FA電子書籍手冊。

e-Manual有如下所示的特點。

- 可以從多本手冊同時搜尋需要的資訊(手冊交叉搜尋)
- 可以從手冊內的連結參閱其他手冊
- 可以從產品插圖的各部分瀏覽想要了解的硬體規格
- 可以將頻繁瀏覽的資訊登錄到收藏夾
- 可以將樣本程式複製到工程工具中

術語

本手冊中，除特別標記的情況，將使用下述術語進行說明。

術語	說明
A系統	是為了判別透過跟蹤電纜連接的兩個系統而在A系統中所設定的系統。 同時啟動兩個系統的情況下，變為控制系統。即使切換系統，也保持A系統不變。
B系統	是為了判別透過跟蹤電纜連接的兩個系統而在B系統中所設定的系統。 同時啟動兩個系統的情況下，變為待機系統。即使切換系統，也保持B系統不變。
SIL2功能模組	是與SIL2過程CPU組合使用、進行安全控制的模組。不能與SIL2過程CPU以外的CPU模組組合。型號為R6PSFM。
SIL2過程CPU	是與SIL2功能模組組合使用，進行常規控制及安全控制的CPU模組。此外，與二重化功能模組組合使用，將系統二重化。型號為R08PSFCPU、R16PSFCPU、R32PSFCPU、R120PSFCPU。
安全CPU	安全CPU是與安全功能模組組合使用、進行常規控制及安全控制的CPU模組。型號為R08SFCPU、R16SFCPU、R32SFCPU、R120SFCPU。
安全功能模組	安全功能模組是與安全CPU組合使用、進行安全控制的模組。不能與安全CPU以外的CPU模組組合。型號為R6SFM。
智能功能模組	是類比模組等具有輸入輸出以外的功能的模組。
工程工具	是執行可程式控制器的設定、程式設計、偵錯、維護的工具。
響應報文	對於請求報文，是支援SLMP的設備發送至外部設備的處理結果的報文。
管理CPU	是控制各輸入輸出模組、智能功能模組的CPU模組。 在多CPU系統中，可以設定對各模組進行控制的CPU模組。
控制系統	是在使用二重化系統時進行控制和網路的通訊的系統。
分離模式	是在二重化後的系統中，在運轉中不停止控制的情況下進行系統維護的模式。
待機系統	是在使用二重化系統時備份用的系統。
通訊協定支援功能	是GX Works3(通訊協定支援功能)中可使用的功能。 功能概要如下所示。 <ul style="list-style-type: none"> 與對象設備相符合的協定的設定 協定設定資料的讀取/寫入
二重化系統	是對CPU模組、電源模組、網路模組等的基本系統進行二重化，在一方的系統中發生異常時也可透過另一方系統繼續控制的系統。
備份模式	是在二重化後的系統中，控制系統發生異常的情況下將控制切換至待機系統繼續運行的模式。
緩衝記憶體	是用於儲存設定值、監視值等的資料的智能功能模組的記憶體。 是指在CPU模組的情況下，用於儲存在乙太網路功能的設定值、監視值等的資料及多CPU功能的資料通訊中所使用的資料的記憶體。
過程CPU(二重化模式)	表示在二重化模式下動作的過程CPU。 可以配置二重化系統。也可在二重化模式下執行過程控制FB或線上模組更換等。
過程CPU(過程模式)	表示在過程模式下動作的過程CPU。 可執行過程控制FB或線上模組更換等。
主站	是控制網路整體的站。可以與全部的站進行循環傳送及瞬時傳送。
請求報文	是透過外部設備對支援SLMP的設備發送處理請求的報文。
標籤	是以任意字元串表示元件的標籤。
本地站	是透過主站及其他本地站進行循環傳送與瞬時傳送的站。

此外，使用SIL2過程CPU及安全CPU的情況下，也使用下述術語進行說明。

術語	內容
安全週期處理	是安全輸入輸出及安全程式的執行處理。
安全控制	執行安全程式及安全通訊實施機械控制。發生異常時，安全地停止機械。
安全通訊	是進行安全通訊協定中定義的安全層的發送接收處理的通訊服務。
安全元件	是安全程式中可使用的元件。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
安全程式	是用於執行安全控制的程式。
安全標籤	安全全局標籤、安全局部元件、常規/安全共享標籤的總稱。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
常規通訊	是安全通訊以外的通訊(CC-Link IE現場網路循環傳送及瞬時傳送等)。
常規元件	是CPU模組內部具有的安全元件以外的元件(X、Y、M、D等)。只能在常規程式中使用。(用於與安全元件區分。)
常規程式	是用於執行順控程式控制的安全程式以外的程式。(用於與安全程式區分。)

總稱/簡稱

本手冊中，除特別標記的情況，將使用下述總稱/簡稱進行說明。

總稱/簡稱	說明
A/D轉換模組	是表示MELSEC iQ-R系列類比-數位轉換模組、通道間絕緣類比-數位轉換模組和高速類比-數位轉換模組。
CC-Link IE TSN主站・本地站模組	是表示RJ71GN11-T2。
CC-Link IE控制網路搭載模組	是表示RJ71GP21-SX型CC-Link IE控制網路模組、RJ71GP21S-SX型CC-Link IE控制網路模組及CC-Link IE控制網路功能使用時的下述模組。 <ul style="list-style-type: none"> • RJ71EN71 • RnENCPU
CC-Link IE內建乙太網路介面模組	表示RJ71EN71。
CC-Link IE現場網路主站・本地站搭載模組	是表示RJ71GF11-T2型CC-Link IE現場網路主站・本站站及CC-Link IE現場網路功能使用時的下述模組。 <ul style="list-style-type: none"> • RJ71EN71 • RnENCPU
D/A轉換模組	是表示MELSEC iQ-R系列數位-類比轉換模組、通道間絕緣數位-類比轉換模組、高速數位-類比轉換模組。
FBD/LD	是表示FB圖/梯形圖語言。
MELSECNET/10	是表示MELSECNET/10網路系統。
MELSECNET/H	是表示MELSECNET/H網路系統。
RnCPU	是表示R00CPU、R01CPU、R02CPU、R04CPU、R08CPU、R16CPU、R32CPU、R120CPU。
RnPCPU	是表示R08PCPU、R16PCPU、R32PCPU、R120PCPU。
RnENCPU	是表示R04ENCPU、R08ENCPU、R16ENCPU、R32ENCPU、R120ENCPU。
RnENCPU(網路部分)	表示RnENCPU的右側(網路部分)。(□MELSEC iQ-R 乙太網路/CC-Link IE用戶手冊(入門篇))
SFC	是表示順序功能圖。
SLMP	是表示Seamless Message Protocol。 是為存取來自外部設備的支援SLMP的設備以及連接支援SLMP的設備的可程式控制器的協定。
支援SLMP的設備	是表示可發送接收SLMP的報文的三菱電機產品設備。(乙太網路適配器模組、乙太網路搭載模組)
ST語言	是表示結構化文字語言。
對象設備	是表示用於進行資料通訊而透過乙太網路連接的電腦、其它乙太網路搭載模組等。
類比模組	是表示A/D轉換模組、D/A轉換模組、溫度輸入模組。
定位模組	是表示MELSEC iQ-R系列的定位模組。
操作數	是表示各自指令及函數內部構成中使用的源資料(s)、目標資料(d)、元件數(n)等的元件部分。
溫度輸入模組	是表示MELSEC iQ-R系列通道間絕緣熱電偶輸入模組及通道間絕緣測溫電阻體輸入模組。
外部設備	是表示對支援SLMP的設備發送SLMP的請求報文的設備。(電腦、顯示器等)
可程式控制器CPU	是表示R00CPU、R01CPU、R02CPU、R04CPU、R04ENCPU、R08CPU、R08ENCPU、R16CPU、R16ENCPU、R32CPU、R32ENCPU、R120CPU、R120ENCPU。
輸入輸出模組	是表示輸入模組、輸出模組、輸入輸出混合模組、中斷模組。
網路模組	表示下述的模組。 <ul style="list-style-type: none"> • 乙太網路介面模組 • CC-Link IE TSN主站・本地站模組 • CC-Link IE控制網路模組 • CC-Link IE現場網路主站・本地站模組 • MELSECNET/H模組 • MELSECNET/10模組 • RnENCPU(網路部分)
過程CPU	是表示R08PCPU、R16PCPU、R32PCPU、R120PCPU。
遠程起始模組	是表示RJ72GF15-T2型CC-Link IE現場網路遠程起始模組。

此外，使用SIL2過程CPU及安全CPU的情況下，也使用下述術語進行說明。

術語	內容
安全標籤	是表示安全全局標籤、安全局部元件、常規/安全共享標籤。(□MELSEC iQ-R CPU模組用戶手冊(應用篇))
常規CPU	是表示執行常規控制的MELSEC iQ-R系列的各CPU模組。(與執行安全控制的CPU模組進行區分時使用。)

指令符號的總稱

本手冊中記載的指令符號，除了特別指定的情況以外，皆使用下述總稱。

分類	指令符號	總稱
PID控制指令	S(P).PIDINIT、PIDINIT(P)	PIDINIT
	S(P).PIDCONT、PIDCONT(P)	PIDCONT
	S(P).PIDPRMW、PIDPRMW(P)	PIDPRMW
多CPU之間專用指令	D(P).DDR、M(P).DDR	DDR
	D(P).DDWR、M(P).DDWR	DDWR

手冊閱讀方法

以下對本手冊的頁數配置及符號有關內容進行說明。

第3部分～第5部分的閱讀方法

下述是關於手冊閱讀方法的說明，與實際記載內容有所不同。

1 → 資料庫記錄添加

2 → DBINSERT (P)

3 → 對設定的識別編號所對應的資料庫的表格添加記錄。

4 → 梯形圖

5 → ENO= DBINSERT (EN, s1, s2, s3, s4, d1, d2);
ENO= DBINSERT (EN, s1, s2, s3, s4, d1, d2);

6 → FBD/LD

7 → EN ENO
s1 d1
s2 d2
s3
s4

8 → 執行條件

9 → 設置數據

10 → 內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	資料庫識別編號	1~4	帶符號BIN16位	ANY16
(s2)	存儲資料庫的表名的起始元件*	—	Unicode字元串	ANSISTRING, DOUBLE
(s3)	存儲資料庫的欄位名的起始元件*	—	字	字

11 → 可使用的元件

操作數	位	字	雙字	間接指定	常數	其他
(s1)	X, Y, M, L, SM, F, B, SB, FX, FY	J□□□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□V□□, J□□□□, USE□\□□□□	Z, LT, LST, LC, LZ	K, H, E, \$
(s2)	—	○	—	—	—	—
(s3)	—	○	—	—	—	—
(s4)	—	○	—	—	—	—

12 → 資料庫的欄位名

操作數: (s3)	項目	內容	設定範圍	設定方
+0	欄位數	欄位數設定希望添加值的欄位的個數。應設定為小於或等於所指定欄位數的對象。*	0~128 ²	用戶

13 → 功能

- 對(s1)中設定的識別編號所對應的資料庫的(s2)中設定的表格添加記錄。
- 在(s3)中指定添加的記錄的欄位個數、欄位名。對於添加的欄位名，不需要指定配置表的全部欄位。在不指定的欄位中存儲NULL。
- 在(s4)中指定添加的記錄的個數、每個記錄的存儲狀態。

14 → 注意事項

下述情況下，完成元件(d1)+1的異常完成信號將變為ON、(d2)的完成狀態中將存儲出錯代碼。

- 資料庫存取指令執行中執行了DBINSERT (P)指令的情況下
- (s1)中指定了超出範圍的識別編號的情況下
- (s1)中指定了未打開的識別編號的情況下
- 韌體版本為“28”以前的可程式控制器CPU中，指定了(s1)中韌體版本為“28”以後的可程式控制器CPU中建構或操作的資料庫的情況下。
- (s2)中指定的表名不存在的情況下
- (s2)中指定的表名的字元數超過了32字元的情況下
- (s3)中添加的欄位數設定了超出範圍的值的的情況下
- (s3)+1~(s3)+□中指定的欄位名中，設定了超出範圍的欄位名的情況下
- (s4)中添加的記錄數設定了超出範圍的值的的情況下
- 資料庫的INSERT處理失敗的情況下
- (s4)+2中設定的1個記錄的資料範圍與(s4)+1中指定的容量不一致的情況下

15 → 出錯

出錯代碼(SID0)	內容
2820H	(s2)、(s3)、(s4)、(d1)中指定的範圍超出了相應元件/標籤的情況下。

關於操作數的完成狀態中存儲的出錯代碼，請參閱下述內容。

①表示指令符號。

- 在指令符號中帶括弧的指令表示多個指令。例如，“GRY(P)(U)”的情況下，相當於GRY指令、GRYP指令、GRY_U指令、GRYP_U指令4個指令。

指令符號	符號含義
指令符號中附帶“(P)”	是僅OFF→ON的上升沿時執行的指令。
指令符號中附帶“(U)”	是處理BIN16位元、BIN32位元無符號資料的指令。

- 指令符號中附帶“□”的指令表示多個指令。例如，“LDDT□”的情況下，相當於LDDT=指令、LDDT<>指令、LDDT>指令、LDDT<=指令、LDDT<指令、LDDT>=指令的6個指令。

②表示各CPU模組指令的使用可否。(帶有×的CPU模組中不能使用指令。)

③表示梯形圖語言、ST語言、FBD/LD語言中的記述形式。

在梯形圖、FBD/LD記述內的方框圍住處分別放入相應的指令符號。

在ST、FBD/LD記述內的EN(執行條件)中輸入執行指令的控制條件。ENO(執行結果)中輸出指令的執行結果。

④表示執行條件。(☞49頁 執行條件)

⑤表示各操作數的內容、可設定範圍、資料類型、標籤中的資料類型。

- 關於資料類型，請參閱下述內容。

☞32頁 資料的指定方法

⑥表示各操作數中可使用的元件。使用區分如下所示。

操作數	位元		字元		雙字元		間接指定	常數			其他*5	
	X、Y、M、L、SM、F、B、SB、S、FX、FY	J□\□*4	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□, J□\□*4, U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
可使用的元件*1	X、Y、M、L、SM、F、B、SB、S、FX*2、FY*2	J□\X J□\Y J□\B J□\SB	T*3、ST*3、C*3、D、W、SD、SW、FD*2、R、ZR、RD	U□\G□ U3E□\G□ U3E□\HG□ J□\W J□\SW	Z	LT*3 LST*3 LC*3	LZ	@□ @□.□	K、H	E	\$	P、I、J、U、DX、DY、N、V、BL、BL□\S□

在SIL2過程CPU及安全CPU的安全程式中可透過各操作數使用的安全元件如下所示。使用區分如下所示。

操作數	位元	字元	常數
		SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD
可使用的元件*1	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T*3、SA\ST*3、SA\C*3、SA\D、SA\W、SA\SD	K、H

*1 關於各元件的說明有關內容，請參閱下述手冊。

☞MELSEC iQ-R CPU模組用戶手冊(應用篇)

*2 FX、FY僅位元資料，FD僅字元資料時可以使用。

*3 在下述指令以外使用T、ST、C、LT、LST、LC的情況下，僅字元資料中可以使用。在位元資料中不能使用。

[位元資料中可使用的指令]

LD、LDI、AND、ANI、OR、ORI、LDP、LDF、ANDP、ANDF、ORP、ORF、LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI、OUT、RST、BKRST、MOVB(P)、CMLB(P)

將SA\T、SA\ST、SA\C用於下述指令以外的情況下，只能在字元資料中使用。在位元資料中不能使用。

[位元資料中可使用的指令]

LD、LDI、AND、ANI、OR、ORI、LDP、LDF、ANDP、ANDF、ORP、ORF、LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI、OUT、RST、MOVB(P)

*4 在使用網路No.的網路模組中可以使用。

*5 “其他”欄記載可設定的元件。

⑦根據指令，存在用於設定指令的動作的控制資料，設定方為“用戶”的情況下，需要按照設定範圍指定值。

⑧表示指令的詳細功能。此外，沒特別記載的情況下，“中斷程式”相當於下述內容。

- 透過中斷指針(I)進行的中斷程式
- 恆定週期執行類型程式
- 將透過中斷指針(I)的中斷發生作為觸發的事件執行類型程式

⑨表示注意的內容。

⑩表示在有指令中特有的運算出錯的情況下，執行時發生的出錯代碼及出錯內容。

- 出錯代碼欄中，記載了儲存出錯代碼的元件。出錯代碼被儲存到SDO中的情況下，出錯標誌(SMO)將變為ON。(從CPU模組的模組標籤也可以確認。)

- 關於各指令中未記載的出錯有關內容，請參閱下述手冊。

☞MELSEC iQ-R CPU模組用戶手冊(應用篇)

第7部分、第8部分的閱讀方法

下述是關於手冊閱讀方法的說明，與實際記載內容有所不同。

1 表示函數符號。
2 表示各CPU模組的通用函數/FB的使用可否。(附帶×的CPU模組中不能使用通用函數/FB。)
3 表示梯形圖語言、ST語言、FBD/LD語言中的記述形式。
4 表示各引數的內容、類型、資料類型、標籤中的資料類型。
5 表示通用函數/FB的功能有關內容。
6 表示有通用函數/FB中特有運算出錯的情況下，執行時發生的出錯代碼及出錯內容有關內容。

將BOOL型資料轉換為WORD型資料。

梯形圖、FBD/LD [無EN/ENO] [帶EN/ENO] ST [無EN/ENO] [帶EN/ENO]

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行, FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	BOOL
ENO	輸出狀態 (TRUE: 正常, FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	WORD

■功能

■運算處理

- 將 (s) 中輸入的BOOL型的資料轉換為WORD型的資料後從 (d) 中輸出。
- 輸入值為FALSE的情況下，以WORD型的資料值輸出0H。
- 輸入值為TRUE的情況下，以WORD型的資料值輸出1H。

■運算結果

執行條件	運算結果
EN	ENO (d)
TRUE (運算執行)	TRUE 運算輸出值
FALSE (運算停止)	FALSE*1 不定值

*1 在ENO輸出了FALSE的情況下，從 (d) 中輸出的資料將變為不定值。在此情況下，應進行不使用從 (d) 中輸出的資料的程式處理。

■出錯

沒有運算出錯。

1 表示函數符號。

關於函數符號中附帶括弧的通用函數/FB，表示多個函數。例如，“BOOL_TO_DINT(_E)”的情況下，相當於“BOOL_TO_DINT”、“BOOL_TO_DINT_E”這2個。

函數符號	符號含義
函數符號中附帶“(E)”	為可使用帶EN/ENO的記錄形式的通用函數/FB。

2 表示各CPU模組的通用函數/FB的使用可否。(附帶×的CPU模組中不能使用通用函數/FB。)

3 表示梯形圖語言、ST語言、FBD/LD語言中的記述形式。

在方框圍住處放入以下符號。

- 通用函數：相應的函數符號
- 通用FB：例項名及相應的FB符號

在EN(執行條件)中輸入控制通用函數/FB的執行的條件。在ENO(執行結果)中輸出通用函數/FB的執行結果。

FBD/LD的情況下，不顯示函數的返回值的名稱。

關於例項，請參閱下述手冊。

📖 MELSEC iQ-R 程式手冊(程式設計篇)

4 表示各引數的內容、類型、資料類型、標籤中的資料類型。

- 關於資料類型，請參閱下述內容。

📖 32頁 資料的指定方法

5 表示通用函數/FB的功能有關內容。

6 表示有通用函數/FB中特有運算出錯的情況下，執行時發生的出錯代碼及出錯內容有關內容。

出錯代碼欄中，記載了儲存出錯代碼的元件。出錯代碼被儲存到SDO中的情況下，出錯標誌(SMO)將變為ON。(從CPU模組的模組標籤也可以確認。)

關於各通用函數/FB中未記載的出錯，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

第1部分 概要

本部分由下述章節構成。

1 指令概要

1 指令概要

1.1 指令的構成

可程式控制器中可使用的指令，大部分可分為指令部分及操作數部分。

以下介紹指令部分及操作數部分的用途。

- 指令部分：該指令的功能如下所示。
- 操作數部分：指令中使用的資料如下所示。

操作數部分分為源資料、目標資料、數值資料。

源(s)

源是運算中使用的資料。

根據各指令中指定的標籤及元件，其情況如下所示。

種類	內容
常數	指定運算中使用的數值。 由於是在建立程式時進行設定，因此在程式執行中不能更改。 當常數在可變資料中使用的情況下，應進行變址修飾。*1
元件 標籤	指定儲存了運算中使用的資料的元件或標籤。 在執行運算之前需要將資料預先儲存到指定的元件或標籤中。 程式執行中，透過更改指定的元件或標籤中儲存的資料，可以更改該指令中使用的資料。

*1 關於變址修飾，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

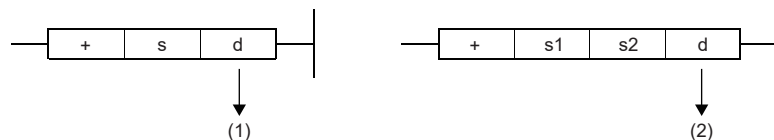
目標(d)

目標中儲存運算後的資料。

但是，根據指令，有時在運算前需要在目標中儲存在運算中會使用的資料。

例

在BIN16位元資料的加法運算指令的情況下



(1): 運算執行前，儲存使用的資料。

(2): 只儲存運算結果。

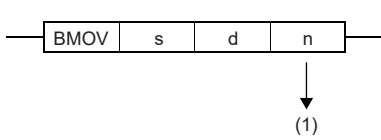
目標必須設定用於儲存資料的標籤及元件。

元件數/傳送數/資料數/字元串數等的數值(n)

對於元件數、傳送數、資料數、字元串數等的數值，指定使用多個元件的指令及指定重複次數、指定處理的資料數・字元串數等的指令中使用的元件數、傳送數、資料數、字元串數等。

例

在塊傳送指令的情況下



(1)：指定BMOV指令中傳送的傳送數。

元件數、傳送數、字元數等的容量指定允許範圍為0~65535、0~4294967295。^{*1}

但是，在元件數、傳送數、字元數等的容量被指定為0的情況下，該指令將變為無處理。

另外，根據元件記憶體的容量或檔案儲存的容量，可能會有小於上述指定上限值的情況。^{*2}

*1 設定範圍根據指令而有所不同。詳細內容，請參閱各指令。

*2 元件記憶體的容量或檔案儲存容量會依機種不同而有所差異。有關詳細內容，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(入門篇)

要點

在傳送數等的數值中使用較大值的情況下，掃描時間將延遲，應加以注意。

1.2 資料的指定方法

CPU模組的各指令中可使用的資料有下述幾種類型。

資料	分類
位元資料	位元資料
16位元資料(字元資料)	帶符號BIN16位元資料
	無符號BIN16位元資料
32位元資料(雙字元資料)	帶符號BIN32位元資料
	無符號BIN32位元資料
實數資料(浮點資料)	單精度實數資料
	雙精度實數資料
BCD資料	BCD4位資料
	BCD8位資料
	BCD16位資料
字元串資料	字元串
	Unicode字元串

作為元件處理的資料

在將設定資料透過元件進行指定的情況下，可指定的元件/常數的類型如下所示。

資料類型	內容	可指定的元件/常數*1
位元	可以處理位元資料。 ☞ 36頁 位元資料	<ul style="list-style-type: none"> 位元元件 字元元件的位元指定
字元	可以處理字元資料。 ☞ 38頁 16位元資料(字元資料)	<ul style="list-style-type: none"> 字元元件 位元元件的位指定(K1~K4)*2 10進制常數 16進制常數
帶符號BIN16位元	可以處理16位元資料。	<ul style="list-style-type: none"> 10進制常數 16進制常數
無符號BIN16位元	在帶符號及無符號中，值的範圍有所不同。 ☞ 38頁 16位元資料(字元資料)	
雙字元	可以處理雙字元資料。 ☞ 41頁 32位元資料(雙字元資料)	<ul style="list-style-type: none"> 字元元件 雙字元元件 位元元件的位指定(K1~K8)*2 10進制常數 16進制常數
帶符號BIN32位元	32位元資料或16位元資料可以處理2個連續的資料。	<ul style="list-style-type: none"> 10進制常數 16進制常數
無符號BIN32位元	在帶符號及無符號中，值的範圍有所不同。 ☞ 41頁 32位元資料(雙字元資料)	
BCD4位	可以處理BCD4位資料。 將16位元資料以4位進行分割，將各位以0~9進行指定。	<ul style="list-style-type: none"> 字元元件 位元元件的位指定(K1~K4)*2 10進制常數 16進制常數
BCD8位	可以處理BCD8位資料。 將32位元資料以8位進行分割，將各位以0~9進行指定。	<ul style="list-style-type: none"> 字元元件 雙字元元件 位元元件的位指定(K1~K8)*2 10進制常數 16進制常數
單精度實數	可以處理單精度實數資料(單精度浮點資料)。 ☞ 44頁 單精度實數資料的構成	<ul style="list-style-type: none"> 字元元件 雙字元元件 實數常數
雙精度實數	可以處理雙精度實數資料(雙精度浮點資料)。 ☞ 45頁 雙精度實數資料的構成	<ul style="list-style-type: none"> 字元元件 雙字元元件 實數常數
字元串	可以處理ASCII代碼、移位JIS代碼的字元串資料。 ☞ 47頁 字元串資料	<ul style="list-style-type: none"> 字元元件 字元串常數
Unicode字元串	可以處理Unicode的字元串資料。 ☞ 47頁 字元串資料	<ul style="list-style-type: none"> 字元元件 字元串常數
元件名	可以直接指定元件。	<ul style="list-style-type: none"> 可以使用的元件中對應的元件名

*1 透過指令進行源(s)及數值資料(n)指定的資料中，可以使用常數。

*2 關於指定方法，請參閱各資料類型的詳細頁面。

作為標籤處理的資料


在設定資料透過標籤進行指定的情況下，可指定的標籤類型如下所示。

■基本資料類型

資料類型(標籤)	可指定的標籤
位元 (BOOL)	<ul style="list-style-type: none">位元型標籤字元[無符號]/位元串[16位元]型標籤的位元指定字元[帶符號]型標籤的位元指定定時器/累計定時器/超長定時器/超長累計定時器型標籤的觸點・線圈計數器/超長計數器型標籤的觸點・線圈
字元[無符號]/位元串[16位元] (WORD)	<ul style="list-style-type: none">字元[無符號]/位元串[16位元]型標籤位元型標籤的位指定(K1~K4)定時器/累計定時器型標籤的當前值計數器型標籤的當前值
雙字元[無符號]/位元串[32位元] (DWORD)	<ul style="list-style-type: none">雙字元[無符號]/位元串[32位元]型標籤位元型標籤的位指定(K1~K8)超長定時器/超長累計定時器型標籤的當前值超長計數器型標籤的當前值
字元[帶符號] (INT)	<ul style="list-style-type: none">字元[帶符號]型標籤位元型標籤的位指定(K1~K4)定時器/累計定時器型標籤的當前值計數器型標籤的當前值
雙字元[帶符號] (DINT)	<ul style="list-style-type: none">雙字元[帶符號]型標籤位元型標籤的位指定(K1~K8)超長定時器/超長累計定時器型標籤的當前值超長計數器型標籤的當前值
單精度實數 (REAL)	<ul style="list-style-type: none">單精度實數型標籤
雙精度實數 (LREAL)	<ul style="list-style-type: none">雙精度實數型標籤
時間 (TIME)	<ul style="list-style-type: none">時間型標籤
字元串 (STRING)	<ul style="list-style-type: none">字元串型標籤
字元串[Unicode] (WSTRING)	<ul style="list-style-type: none">字元串[Unicode]型標籤
指標 (POINTER)	<ul style="list-style-type: none">指標型標籤

要點

關於各標籤的內容，請參閱下述手冊。

 MELSEC iQ-R CPU模組用戶手冊(應用篇)

■總稱資料類型

是彙總了若干個基本資料類型的標籤的資料類型。

在允許函數及FB的引數、恢復值等多個資料類型的情況下，使用總稱資料類型。

對於定義為總稱資料類型的標籤，在低位的資料類型的任意類型中均可使用。

資料類型 (標籤)				可指定的資料類型			
ANY*1	ANY_ELEMENTARY	ANY_BIT		ANY_BOOL	位元		
				ANY_BITADDR*1	位元		
				ANY16_U	字元[無符號]/位元串[16位元]		
				ANY32_U	雙字元[無符號]/位元串[32位元]		
		ANY_WORDADDR		ANY_NUM	ANY_INT	ANY16_S	字元[帶符號]
						ANY16_U	字元[無符號]/位元串[16位元]
					ANY32	ANY32_S	雙字元[帶符號]、時間
						ANY32_U	雙字元[無符號]/位元串[32位元]
				ANY_REAL		ANYREAL_32	單精度實數
						ANYREAL_64	雙精度實數
				ANY_STRING		ANYSTRING_SINGLE	字元串
						ANYSTRING_DOUBLE	字元串[Unicode]
		ANY16_OR_STRING_SINGLE		ANY16_S	字元[帶符號]		
				ANY16_U	字元[無符號]/位元串[16位元]		
				ANYSTRING_SINGLE	字元串		
		ANY_DT		字元[帶符號]、字元[無符號]/位元串[16位元]			
		ANY_TM		字元[帶符號]、字元[無符號]/位元串[16位元]			
ANY_STRUCT*1		結構體					
STRUCT		結構體					

*1 也可作為數組使用。

■總稱資料類型(數組)

關於下述總稱資料類型，應定義數組要素數。

資料類型 (標籤)			可指定的資料類型
ANYBIT_ARRAY			位元的數組
ANYWORD_ARRAY	ANY16_ARRAY	ANY16_S_ARRAY	字元[帶符號]的數組
		ANY16_U_ARRAY	字元[無符號]/位元串[16位]的數組
	ANY32_ARRAY	ANY32_S_ARRAY	雙字元[帶符號]的數組、時間的數組
		ANY32_U_ARRAY	雙字元[無符號]/位元串[32位元]的數組
	ANY_REAL_ARRAY	ANY_REAL_32_ARRAY	單精度實數的數組
		ANY_REAL_64_ARRAY	雙精度實數的數組
	ANY_STRING_ARRAY	ANY_STRING_SINGLE_ARRAY	字元串的數組
		ANY_STRING_DOUBLE_ARRAY	字元串[Unicode]的數組
STRUCT_ARRAY			結構體的數組

位元資料

資料容量及資料的範圍

位元資料是將觸點及線圈等以1位元單位處理的資料。

資料名稱	資料容量	值的範圍
位元資料	1位元	0、1

位元元件/標籤中位元資料的處理

1點位元元件/標籤可處理1位元資料。

字元元件中位元資料的處理

對字元元件進行位元No. 指定，可以處理指定位元No. 的位元資料。

位元指定的記載方法為“字元元件編號.位元No.”。

位元No. 的指定是以16進制數在0~F的範圍內進行指定。

例如，D0的位元5(b5)以“D0.5”進行指定，D0的位元10(b10)以“D0.A”進行指定。

可進行字元元件的位元指定的元件如下所示。

項目	元件
可進行字元元件的位元指定的元件	<ul style="list-style-type: none">資料寄存器 (D)連結寄存器 (W、J□\W)特殊連結寄存器 (SW、J□\SW)功能寄存器 (FD)特殊寄存器 (SD)模組存取元件 (U□\G)CPU緩衝記憶體存取元件 (U3E□\G、U3E□\HG)檔案寄存器 (R、ZR)模組更新用寄存器 (RD)

SIL2過程CPU及安全CPU的安全程式中所使用的安全元件的位元No. 的指定，也以16進制數在0~F的範圍內進行指定。

例如，SA\D0的位元5(b5)以“SA\D0.5”、SA\D0的位元10(b10)以“SA\D0.A”進行指定。

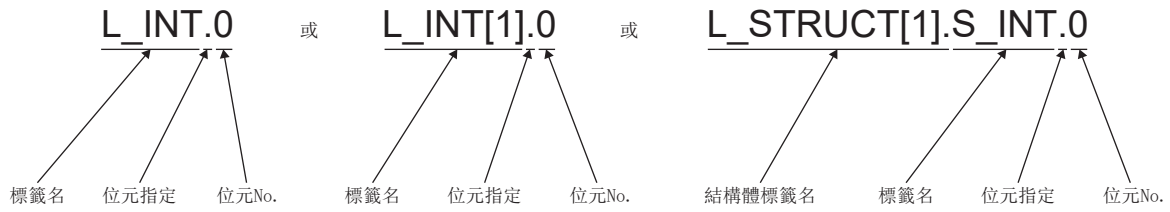
可進行字元元件的位元指定的安全元件如下所示。

項目	元件
可進行字元元件的位元指定的元件	<ul style="list-style-type: none">安全資料寄存器 (SA\D)安全連結寄存器 (SA\W)安全特殊寄存器 (SA\SD)

字型標籤中位元資料的處理

在字型標籤中，透過進行位元No. 指定，可以處理指定位元No. 的位元資料。
位元指定的記載方法為“標籤名.位元No.”。

例



可進行位元指定的標籤的資料類型如下所示。

項目	資料類型
可進行位元指定的標籤的資料類型	<ul style="list-style-type: none"> • 字元[帶符號](INT型) • 字元[無符號]/位元串[16位元](WORD型) • 定時器(TIMER型)的當前值(N)^{*1} • 累計定時器(RETENTIVETIMER型)的當前值(N)^{*1} • 計數器(COUNTER型)的當前值(N)^{*1}

*1 梯形圖的情況下不能指定。

16位元資料(字元資料)

資料容量及資料的範圍

16位元資料中，有帶符號16位元資料和無符號16位元資料。

對於帶符號16位元資料，負數以2的補數表示。

資料名稱	資料容量	值的範圍	
		10進制記載	16進制記載
帶符號16位元資料	16位元(1字元)	-32768~32767	0000H~FFFFH
無符號16位元資料		0~65535	

位元元件中16位元資料的處理

對位元元件進行位指定，可以作為16位元資料處理。

項目		記載	示例
位元元件	直接連結元件以外	K□位元元件起始編號 □：位數(在1~4的範圍內指定)	K4X10 K2M113
	直接連結元件	J□\K□位元元件起始編號 左側□：網路No. 右側□：位數(在1~4的範圍內指定)	J1\K3B10 J10\K2Y10

對於SIL2過程CPU及安全CPU的安全程式中使用的位元元件，透過位指定，也可作為16位元資料處理。

項目	記載	示例
位元元件	SA\K□位元元件起始編號 □：位數(在1~4的範圍內指定)	SA\K4X10 SA\K2M113

位元型數組的標籤中16位元資料的處理

對位元型數組的標籤進行位指定，可以作為16位元資料處理。

對位元型數組的標籤進行位指定，作為16位元資料處理時的記載方法如下所示。

項目	記載	示例
位元型數組的標籤	K□標籤名 □：位數(在1~4的範圍內指定) 標籤的位指定時，以無要素指定方式進行指定。	K1L_BOOL

位指定的範圍

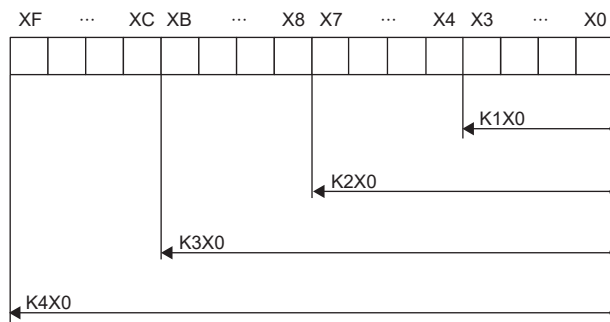
各位指定的16位元資料的範圍如下所示。

位指定	10進制記載	16進制記載
K1	0~15	0H~FH
K2	0~255	00H~FFH
K3	0~4095	000H~FFFH
K4	帶符號16位元資料：-32768~32767 無符號16位元資料：0~65535	0000H~FFFFH

例

對X0進行了位指定情況下的對象點數如下所示。

- K1X0→X0~X3的4點為對象
- K2X0→X0~X7的8點為對象
- K3X0→X0~XB的12點為對象
- K4X0→X0~XF的16點為對象



在源(s)側指定了位指定的位元元件的情況下

在指令的源側指定了位指定的位元元件的情況下，目標側的字元元件中，源側進行了位指定的位元以後的高位位元中將儲存0。

梯形圖示例	處理
<p>• 在16位元資料指令的情況下</p> <p>(S)</p>	

在目標(d)側指定指定了位的位元元件的情況下

在指令的目標側有位指定的情況下，位指定的點數將成為目標側的對象。位指定點數以後的高位位元元件不變化。

梯形圖示例	處理
<p>• 在源資料為字元元件的情況下</p> <p>(d)</p>	<p>(1): 沒有變化。</p>

字元元件/標籤中16位元資料的處理

■字元元件

1點字元件可處理16位元資料。

■字元型標籤

1點字元型標籤中可處理16位元資料。

32位元資料(雙字元資料)

資料容量及資料的範圍

32位元資料中，有帶符號32位元資料和無符號32位元資料。

對於帶符號32位元資料，負數以2的補數表示。

資料名稱	資料容量	值的範圍	
		10進制記載	16進制記載
帶符號32位元資料	32位元(2字元)	-2147483648~2147483647	00000000H~FFFFFFFH
無符號32位元資料		0~4294967295	

位元元件中32位元資料的處理

對位元元件進行位指定，可以作為32位元資料處理。

項目	記載	示例
位元元件 直接連結元件以外		K8X80 K6B018
直接連結元件		J1\K7B30 J10\K5X128

對於SIL2過程CPU及安全CPU的安全程式中使用的位元元件，透過位指定，也可作為32位元資料處理。

項目	記載	示例
位元元件		SA\K8X80 SA\K6B018

位元型數組的標籤中32位元資料的處理

對位元型數組的標籤進行位指定，可以作為32位元資料處理。

對位元型數組的標籤進行位指定作為32位元資料處理時的記載方法如下所示。

項目	記載	示例
位元型數組的標籤	<p>標籤的位指定時，以無要素指定方式進行指定。 梯形圖以外的語言，應在K5~K8的範圍進行指定。(梯形圖以外的語言的情況下，K1~K4的範圍的位指定將作為ANY16進行處理。)</p>	K8L_BOOL

位指定的範圍

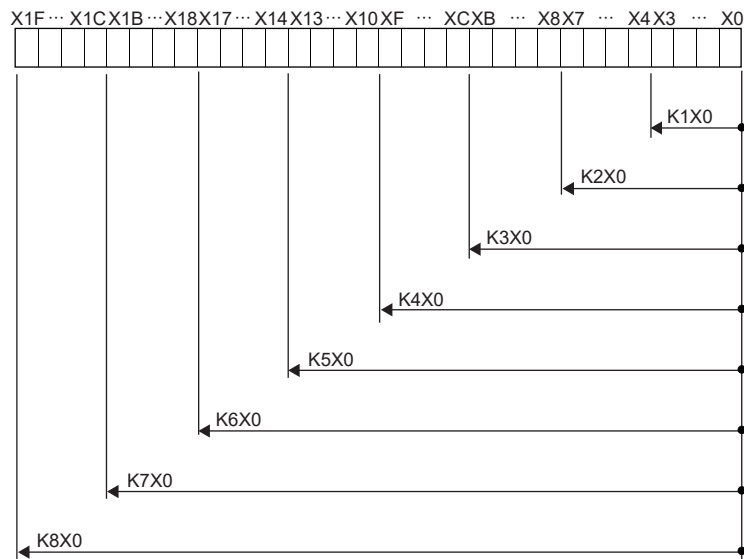
各位指定的32位元資料的範圍如下所示。

位指定	10進制記載	16進制記載
K1	0~15	0H~FH
K2	0~255	00H~FFH
K3	0~4095	000H~FFFH
K4	0~65535	0000H~FFFFH
K5	0~1048575	00000H~FFFFFH
K6	0~16777215	000000H~FFFFFFH
K7	0~268435455	0000000H~FFFFFFFH
K8	帶符號32位元資料：-2147483648~2147483647 無符號32位元資料：0~4294967295	00000000H~FFFFFFFH

例

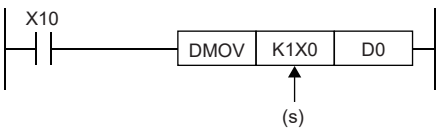
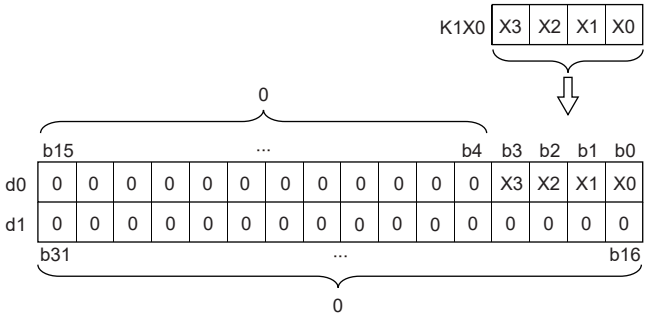
對X0進行了位指定情況下的對象點數如下所示。

- K1X0→X0~X3的4點為對象
- K2X0→X0~X7的8點為對象
- K3X0→X0~XB的12點為對象
- K4X0→X0~XF的16點為對象
- K5X0→X0~X13的20點為對象
- K6X0→X0~X17的24點為對象
- K7X0→X0~X1B的28點為對象
- K8X0→X0~X1F的32點為對象



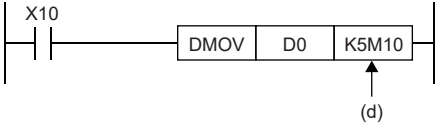
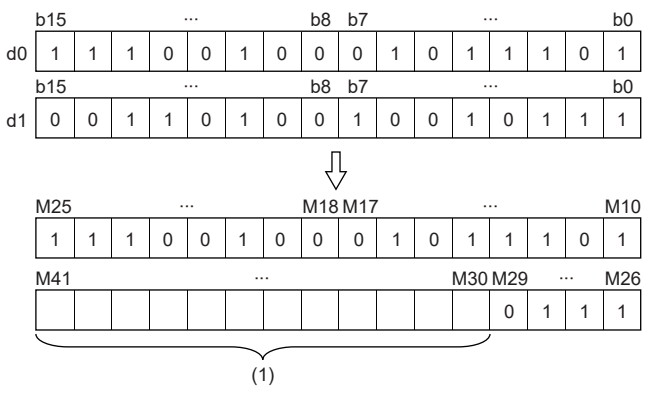
■在源(s)側指定了位指定的位元元件的情況下

在指令的源側指定了位指定的位元元件的情況下，目標側的字元元件中，源側進行了位指定的位元以後的高位位元中將儲存0。

梯形圖示例	處理
<p>• 在32位元資料指令的情況下</p> 	 <p>Diagram illustrating the bit shift operation. The source bits X3, X2, X1, and X0 are shifted into the destination bits b3, b2, b1, and b0. The bits b15 through b4 are set to 0.</p>

■在目標(d)側指定指定了位的位元元件的情況下

在指令的目標側有位指定的情況下，位指定的點數將成為目標側的對象。
位指定點數以後的高位位元元件不變化。

梯形圖示例	處理
<p>• 在源資料為字元元件的情況下</p> 	 <p>Diagram illustrating the bit shift operation. The source bits b15 through b0 are shifted into the destination bits M18 through M10. The bits M25 through M18 are set to 1, and the bits M17 through M10 are set to 0. The bits M41 through M26 are unchanged (1).</p> <p>(1): 沒有變化。</p>

字元元件/標籤中32位元資料的處理

■字元元件

2點的字元元件可作為32位元資料處理。

但是，對於下述元件，1點元件可處理32位元資料。

- 超長定時器(LT)
- 超長累計定時器(LST)
- 超長計數器(LC)
- 超長變址寄存器(LZ)

■雙字型標籤

雙字型標籤的1點可作為32位元資料處理。

實數資料(浮點資料)

資料容量及資料的範圍

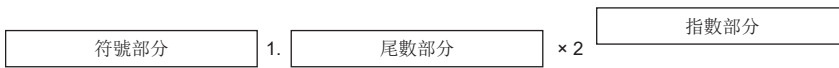
實數資料中，有以32位元表示的單精度實數資料及以64位元表示的雙精度實數資料。

實數資料只能儲存到位元元件以外的元件或單精度實數型標籤、雙精度實數型標籤中。

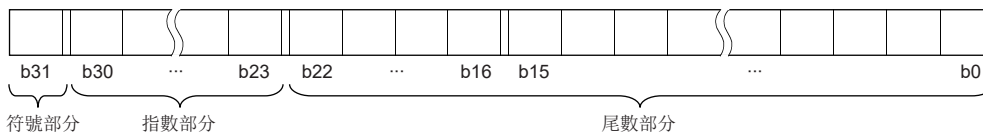
資料名稱	資料容量	值的範圍
單精度實數資料(單精度浮點資料)	正數	$2^{-126} \leq \text{實數} < 2^{128}$
	零	0
	負數	$-2^{128} < \text{實數} \leq -2^{-126}$
雙精度實數資料(雙精度浮點資料)	正數	$2^{-1022} \leq \text{實數} < 2^{1024}$
	零	0
	負數	$-2^{1024} < \text{實數} \leq -2^{-1022}$

單精度實數資料的構成

單精度實數資料由符號部、尾數部、指數部所構成，按以下方式表示。



單精度實數資料內部表示時的位構成及含義如下所示。



■符號部(1位元)

數值的正負以1位元表示。0表示正數或0。1表示負數。

■尾數部(23位元)

將單精度實數以2進制數 $1.XXXXX \dots \times 2^N$ 表示時的XXXXX...如下所示。

■指數部(8位元)

將單精度實數以2進制數 $1.XXXXX \dots \times 2^N$ 表示時的N如下所示。單精度實數中指數部的值與N的關係如下所示。

指數部(b24~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

在工程工具的監視功能中，可以監視CPU模組的實數資料。

實數資料中表示“0”時，將下述範圍全部設定為0。


- 在單精度實數資料的情況下：b0～b31
- 在雙精度實數資料的情況下：b0～b63

實數資料的設定範圍如下所示。^{*1}

- 在單精度實數資料的情況下： $-2^{128} < [\text{單精度實數資料}] \leq -2^{-126}$ ， 0 ， $2^{-126} \leq [\text{單精度實數資料}] < 2^{128}$
- 在雙精度實數資料的情況下： $-2^{1024} < [\text{雙精度實數資料}] \leq -2^{-1022}$ ， 0 ， $2^{-1022} \leq [\text{雙精度實數資料}] < 2^{1024}$

實數資料中請勿指定“-0”（僅最高位元1）。以-0進行實數運算時將變為運算出錯。

*1 關於上溢及下溢時的動作及輸入了特殊值時的動作，請參閱下述手冊。

 MELSEC iQ-R CPU模組用戶手冊(應用篇)

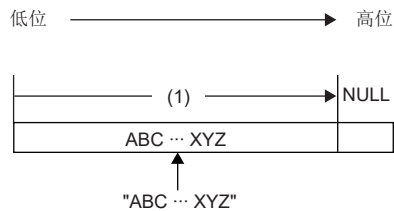
字元串資料

字元串資料的形式

字元串資料中有下述類型，可以將以NULL代碼結束的字元代碼串作為字元串處理。

種類	字元代碼	最終字元
字元串	ASCII代碼、移位JIS代碼	NULL (00H)
Unicode字元串	Unicode (UTF-16 (小端字節序))	NULL (0000H)

將字元串資料儲存到元件或數組中時，將以從元件編號或數組的要素編號的低位至高位的順序儲存字元串資料。



(1): 字元代碼串

字元串的記載方法

梯形圖程式中字元串的記載方法如下所示。

資料類型	記載方法	記載示例
字元串	STRING	將字元串 (ASCII代碼、移位JIS代碼)、Unicode字元串用雙引號(") 圍住。 "ABC"
字元串 [Unicode]	WSTRING	

ST程式中的字元串的記載方法如下所示。

資料類型	記載方法	記載示例
字元串	STRING	將字元串 (ASCII代碼、移位JIS代碼)用單引號(') 圍住。 'ABC'
字元串 [Unicode]	WSTRING	將Unicode字元串用雙引號(") 圍住。 "ABC"

FBD/LD程式中字元串的記載方法如下所示。

資料類型	記載方法	記載示例
字元串	STRING	將字元串 (ASCII代碼、移位JIS代碼)用單引號(') 圍住。
字元串 [Unicode]	WSTRING	將Unicode字元串用雙引號(") 圍住。

資料的範圍

字元串資料的範圍如下所示。

種類	標籤中可設置的最大字元數	字元串常數中可使用的最大字元數
字元串	半形255字元 (不包括終端的NULL。)	半形255字元 (不包括終端的NULL。)
Unicode字元串*1	255字元 (不包括終端的NULL。)	255字元 (不包括終端的NULL。)

*1 Unicode字元串可在基本多語言中使用。

資料儲存中必要的字數

字元串資料可儲存到字元件中。

字元串資料的儲存必要字數如下所示。

字元串的字節數	用於儲存字元串的必要字數	用於儲存Unicode字元串的必要字數
0字節	1[字]	1[字]
奇數字節	(字元串的字節數+1)÷2[字]	一(由於1字元為偶數字節。)
偶數字節	(字元串的字節數+2)÷1[字]	字元數+1[字]

字元串資料的儲存目標

字元串資料的儲存目標示意圖如下所示。

■字元串的情況下

在字元串的儲存示意圖中，NULL表示NULL代碼(00H)。

儲存的字元串	從D0開始儲存了字元串資料時的示意圖	從字型標籤的數組arrayA[0]開始儲存了字元串資料時的示意圖																
空字元串 (“ ”或’ ’)	D0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td><td>NULL</td></tr></table>	NULL	NULL	arrayA[0] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td><td>NULL</td></tr></table>	NULL	NULL												
NULL	NULL																	
NULL	NULL																	
ABC	D0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td><td>A</td></tr><tr><td>NULL</td><td>C</td></tr></table>	B	A	NULL	C	arrayA[0] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td><td>A</td></tr><tr><td>NULL</td><td>C</td></tr></table> arrayA[1] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td><td>NULL</td></tr></table>	B	A	NULL	C	NULL	NULL						
B	A																	
NULL	C																	
B	A																	
NULL	C																	
NULL	NULL																	
ABCD	D0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td><td>A</td></tr><tr><td>D</td><td>C</td></tr><tr><td>NULL</td><td>NULL</td></tr></table>	B	A	D	C	NULL	NULL	arrayA[0] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td><td>A</td></tr><tr><td>D</td><td>C</td></tr><tr><td>NULL</td><td>NULL</td></tr></table> arrayA[1] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td><td>NULL</td></tr></table> arrayA[2] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td><td>NULL</td></tr></table>	B	A	D	C	NULL	NULL	NULL	NULL	NULL	NULL
B	A																	
D	C																	
NULL	NULL																	
B	A																	
D	C																	
NULL	NULL																	
NULL	NULL																	
NULL	NULL																	

■Unicode字元串的情況下





在Unicode字元串的儲存示意圖中，NULL表示NULL代碼(0000H)。

儲存的字元串	從D0開始儲存了字元串資料時的示意圖	從字型標籤的數組arrayA[0]開始儲存了字元串資料時的示意圖														
空字元串(“ ”)	D0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td></tr></table>	NULL	arrayA[0] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td></tr></table>	NULL												
NULL																
NULL																
ABCD	D0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>A</td></tr><tr><td>B</td></tr><tr><td>C</td></tr><tr><td>D</td></tr><tr><td>NULL</td></tr></table>	A	B	C	D	NULL	arrayA[0] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>A</td></tr><tr><td>B</td></tr><tr><td>C</td></tr><tr><td>D</td></tr><tr><td>NULL</td></tr></table> arrayA[1] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td></tr></table> arrayA[2] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td></tr></table> arrayA[3] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td></tr></table> arrayA[4] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td></tr></table>	A	B	C	D	NULL	NULL	NULL	NULL	NULL
A																
B																
C																
D																
NULL																
A																
B																
C																
D																
NULL																
NULL																
NULL																
NULL																
NULL																

1.3 執行條件

執行條件的類型

指令的執行條件有以下類型。

執行條件		說明*1
ON時執行		在ON中執行型的指令中，僅在指令的前條件為ON期間時執行該指令。在前條件為OFF的情況下，不執行該指令，不進行處理。
上升沿執行		在ON時執行1次型的指令中，僅在指令的前條件的上升沿時 (OFF→ON) 執行指令，以後即使條件為ON也不執行該指令，不進行處理。
OFF時執行		在OFF中執行型的指令中，僅在指令的前條件為OFF期間時執行該指令。在前條件為ON的情況下，不執行該指令，不進行處理。
下降沿執行		在OFF時執行1次型的指令中，僅在指令的前條件的下降沿時 (ON→OFF) 執行指令，以後即使條件為OFF也不執行該指令，不進行處理。
常時執行	—	在常時執行的指令中，與指令的前條件的ON/OFF無關，始終執行。在前條件為OFF的情況下，該指令將進行OFF處理。

*1 在ST語言、FBD/LD語言中，指令中的EN將成為指令的前條件。

各指令的執行條件

各指令中執行條件有所不同。關於各指令的執行條件，請參閱各指令的頁面。

在ST語言、FBD/LD語言中，EN將成為執行條件，僅在EN為TRUE的情況下才執行。ENO變為與EN相同的狀態。

此外，對於通用函數/FB，根據EN的有無其執行條件有所不同。無EN的情況下將常時執行。關於帶EN的情況，請參閱通用函數/FB的頁面。

1.4 指令處理時間的高速化

子集處理

子集處理是指，在可進行子集處理的指令中，各操作數指定的元件及標籤在滿足指定條件的情況下，進行步數的減少及指令處理時間的高速化處理。

無論是否處於子集處理狀態，指令符號及操作數的數不變化。

可進行子集處理的指令

關於各指令的子集處理的可否，請參閱下述內容。

☞ 1525頁 基本步數及子集處理的可否

操作數的條件

滿足變為子集處理狀態的操作數的條件如下所示。

■指定了元件的情況下

操作數中指定了元件情況下的條件如下表所示。

操作數的資料類型	條件*1
位元資料	滿足下述之一。 <ul style="list-style-type: none">• 用戶元件• CPU緩衝記憶體存取元件的本機指定(至“U3En”的變址修飾時除外)*2• CPU緩衝記憶體存取元件的恆定週期通訊區域的其它機號指定*3• 檔案寄存器• 局部元件• 更新資料寄存器
帶符號16位元資料 無符號16位元資料 帶符號32位元資料 無符號32位元資料	滿足下述之一。 <ul style="list-style-type: none">• 用戶元件• CPU緩衝記憶體存取元件的本機指定(至“U3En”的變址修飾時除外)*2• CPU緩衝記憶體存取元件的恆定週期通訊區域的其它機號指定*3• 變址寄存器• 檔案寄存器• 局部元件• 更新資料寄存器• 常數(10進制數、16進制數)
單精度實數	滿足下述之一。 <ul style="list-style-type: none">• 用戶元件• CPU緩衝記憶體存取元件的本機指定(至“U3En”的變址修飾時除外)*2• CPU緩衝記憶體存取元件的恆定週期通訊區域的其它機號指定*3• 變址寄存器• 檔案寄存器• 局部元件• 更新資料寄存器• 常數(單精度實數)

*1 也包括指定了位元No. 指定、位指定、間接指定、變址修飾的元件的情況下。

*2 本機的CPU緩衝記憶體存取元件中，使用了U3En\G□、U3En\G□Zn、U3En\HG□、U3En\HG□Zn的情況下。

*3 其它機號的CPU緩衝記憶體存取元件中，使用了U3En\HG□、U3En\HG□Zn的情況下。

■指定了已分配元件的標籤的情況下

與指定了元件情況下的條件相同。

■指定了各標籤區域中分配的標籤的情況下

操作數中指定了標籤區域、鎖存標籤區域中分配的標籤的情況下，如果是可進行子集處理的指令，與操作數的資料類型無關，將進行子集處理。(也包括進行了位元No. 指定、位指定的情況下。)

1.5 程式時的注意事項

指令通用的出錯

執行指令時出錯的條件如下所示。

出錯內容*1	出錯代碼
指定了超出範圍的輸入輸出編號(000H~FFFH、3E0H~3E3H以外)。	2800H
指定了不存在對象模組的輸入輸出編號。	2801H
指定了指令中不能指定的模組的輸入輸出編號。	2803H
指定了超出範圍的網路編號(1~239以外)。	2804H
指定了不存在的網路編號。	2805H
<ul style="list-style-type: none"> 指令中指定的元件或標籤超出了允許使用範圍。 在CPU參數的檔案設定中未設定檔案寄存器，或在程式中未設定使用的檔案寄存器的狀態下，存取了檔案寄存器。 	2820H
<ul style="list-style-type: none"> 超出指令中指定的模組具有的緩衝記憶體的范围。 指令中指定的模組是不具有緩衝記憶體的模組。 	2823H

*1 在觸點指令中未檢測出出錯，運算結果變為非導通狀態。

執行指令時的元件、標籤的范围檢查

元件、標籤的范围檢查

透過指令指定了元件及標籤的情況下，因為不進行範圍的檢查，因此應建立其運算結果不會超出相應元件/標籤的范围的程式。

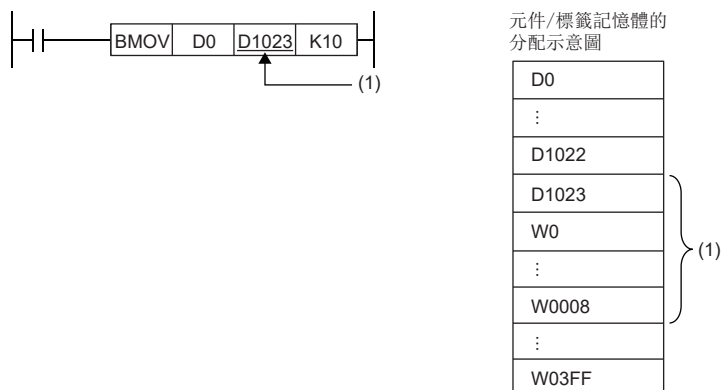
指定時超出相應元件/標籤范围的情況下，因為不發生出錯，資料將被寫入其它元件及標籤中。

但是，超出區域的情況下將發生出錯(出錯代碼：2820H)。

將元件中分配的標籤透過程式中的指令進行了指定的情況下也相同。

例

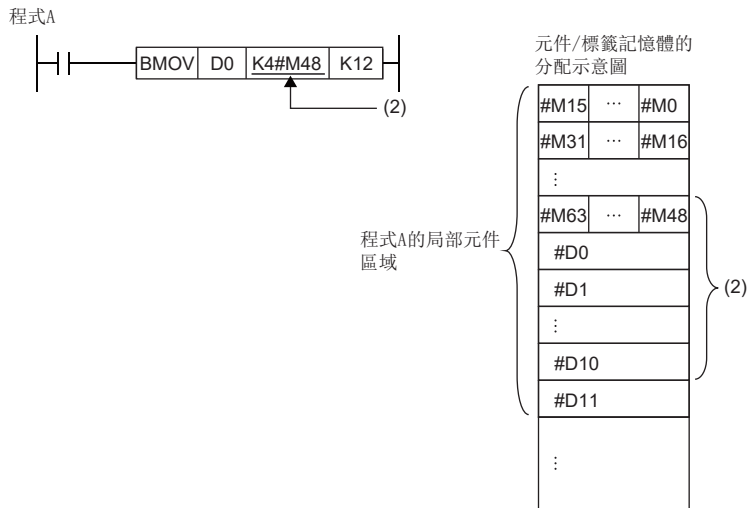
分配元件/標籤記憶體時，全局元件的D1023的下個已變為W0的情況下



(1) 傳送目標的相應範圍為D1023~D1032。即使D1024~D1032不存在，也將寫入資料，並覆蓋W0~W0008的資料。

例

已在M0~M63，D0~D11的範圍設定局部元件的情況下

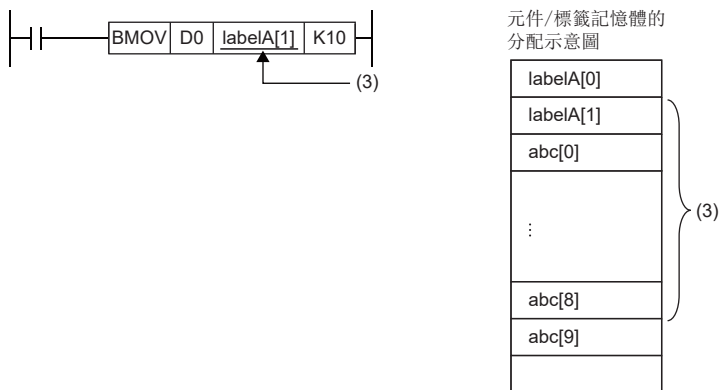


(2) 傳送目標的相應範圍為程式A的局部元件#M48~#M239。即使#M64~#M239不存在，也將寫入資料，並覆蓋#D0~#D10的資料。

例

標籤依下列數組進行設定的情況下

標籤名	資料類型	數組要素
labelA	位元串[16位元]	2
abc	位元串[16位元]	10



(3) 傳送目標的相應範圍為labelA[1]開始算起10個點。即使labelA[2]~labelA[9]不存在，也將寫入資料，並覆蓋abc[0]~abc[8]的資料。

檔案寄存器的範圍檢查

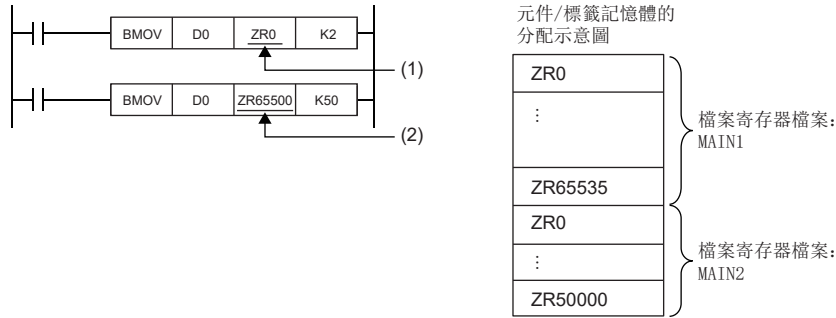
透過指令指定了檔案寄存器的情況下，因為不進行範圍的檢查，因此應建立其運算結果不會超出相應檔案寄存器的範圍的程式。

指定時超出檔案寄存器 (ZR) 範圍的情況下，將發生出錯 (出錯代碼：2820H)。

指定時超出檔案寄存器 (R) 中使用的塊No. 的檔案寄存器範圍的情況下，將發生出錯 (出錯代碼：2820H)。

例

在指定了檔案寄存器 (ZR) 的情況下

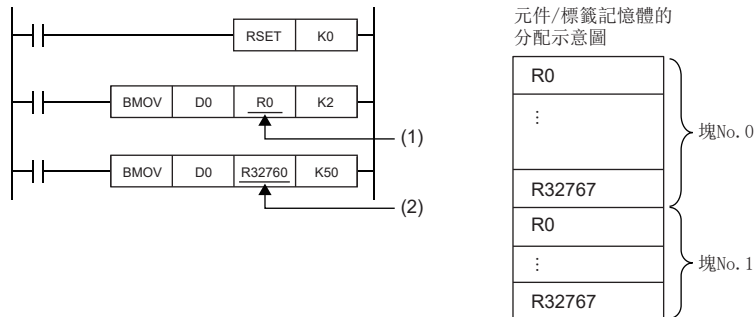


(1) 傳送目標在檔案寄存器MAIN1的範圍內。資料將被寫入ZR0~ZR1。

(2) 傳送目標超出檔案寄存器MAIN1的範圍。由於超出檔案寄存器MAIN1的區域範圍，因此發生出錯。

例

指定了檔案寄存器 (R) 的情況下



(1) 傳送目標在塊No. 0的R元件的範圍內。資料將被寫入R0~R1。

(2) 傳送目標超出了塊No. 0的R元件的範圍。由於超出塊No. 0的R元件的區域範圍，因此發生出錯。

使用超長定時器、超長累計定時器元件時的動作

對於超長定時器、超長累計定時器，處理資料超出當前值的寬度(32位元)的情況下，不僅是當前值的區域，還將使用上次值、觸點・線圈的區域執行動作。

元件	構成
定時器(T)	
累計定時器(ST)	
計數器(C)	
超長定時器(LT)	
超長累計定時器(LST)	
超長計數器(LC)	

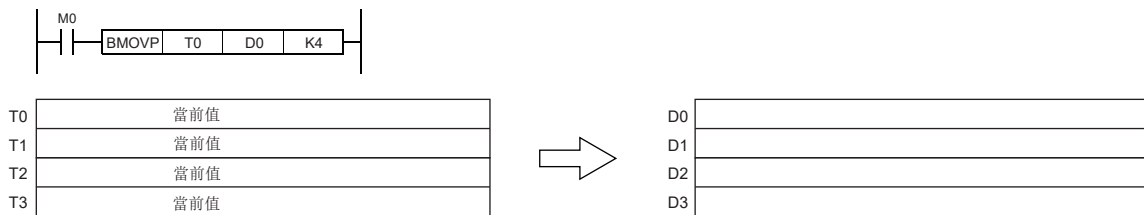
當前值的批量傳送中使用BMOV指令的情況下，無法進行僅當前值的批量傳送，因此應對當前值、觸點、線圈進行批量傳送後，傳送後的資料中只使用當前值。

當前值的批量傳送中使用DMOV指令的情況下，僅將當前值的傳送透過FOR~NEXT指令重複進行，應傳送當前值。

例

對定時器元件的當前值進行批量傳送的情況下

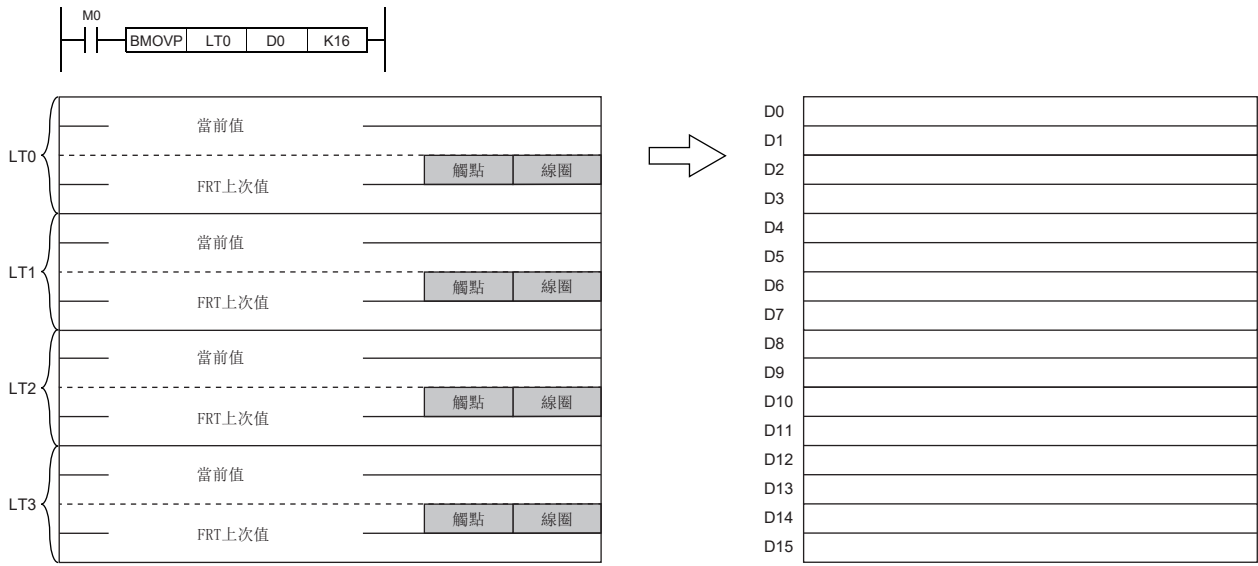
當前值的批量傳送中使用BMOV指令的情況下，只能批量傳送當前值。



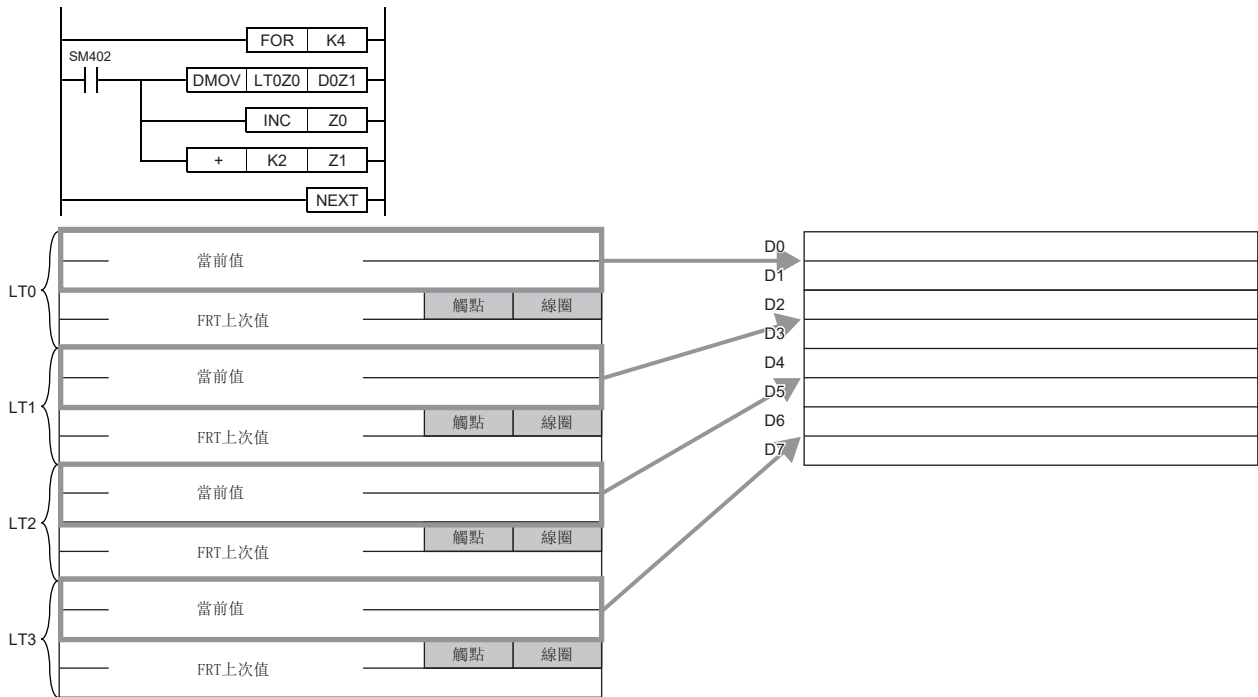
例

對超長定時器元件的當前值進行批量傳送的情況下

當前值的批量傳送中使用BMOV指令的情況下，只能批量傳送當前值、觸點、線圈。



當前值的批量傳送中使用DMOV指令的情況下，只能批量傳送當前值。



使用同一元件的OUT指令、SET/RST指令、PLS/PLF指令時的動作

以下介紹將使用了同一元件的OUT指令、SET/RST指令、PLS/PLF指令在1個掃描中執行多次情況下的動作有關內容。

要點

關於SIL2過程CPU及安全CPU的安全程式中1個安全週期處理中多次執行了使用同一元件的OUT指令、SET/RST指令、PLS/PLF指令情況下的動作，應按以下方式進行替換閱讀。

- “掃描” → “安全週期處理”
- “X0” → “SA\X0”、“X1” → “SA\X1”、“M0” → “SA\M0”
- 圖中X0的上升沿的“END” → “安全週期處理開始” *1
- 圖中X0的下降沿的“END” → “安全週期處理結束” *1

*1 在PLF指令中，無論是X0的上升沿還是下降沿，均應進行“END” → “安全週期處理開始”的替換閱讀。

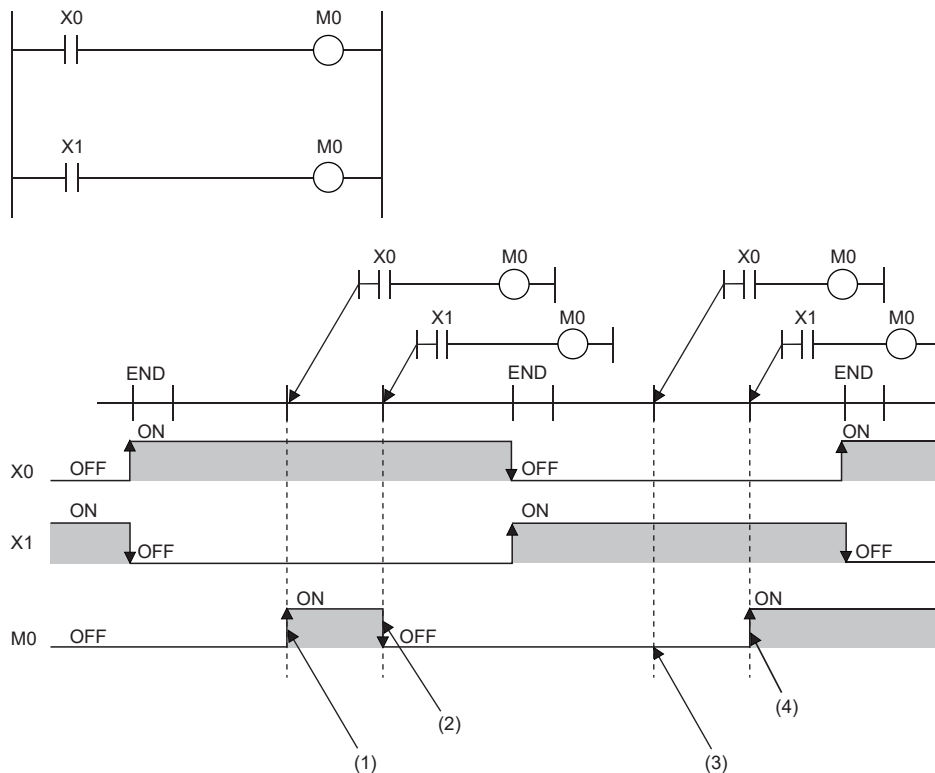
同一元件的OUT指令的情況下

1個掃描中應勿多次執行同一元件的OUT指令。

1個掃描中多次執行了同一元件的OUT指令的情況下，執行各個OUT指令時，根據OUT指令為止的運算結果，指定元件將變為ON/OFF。

執行各個OUT指令時，因為指定元件的ON/OFF是確定的，因此1個掃描中有可能重複進行ON/OFF。

輸入的X0及X1中，建立了將同一內部繼電器(M0)設為ON/OFF的梯形圖情況下的動作如下所示。



- (1) X0為ON，因此M0變為ON。
- (2) X1為OFF，因此M0變為OFF。
- (3) X0為OFF，因此M0保持為OFF不變。
- (4) X1為ON，因此M0變為ON。

透過OUT指令指定輸出(Y)時，1個掃描的最後執行的OUT指令的ON/OFF狀態將被輸出。

使用了同一元件的SET/RST指令的情況下

■SET指令的情況下

SET指令在執行指令為ON時將指定元件設為ON、在執行指令為OFF時無處理。

因此將同一元件的SET指令在1個掃描中執行了多次的情況下，如果1個執行指令為ON，指定元件將變為ON。

■RST指令的情況下

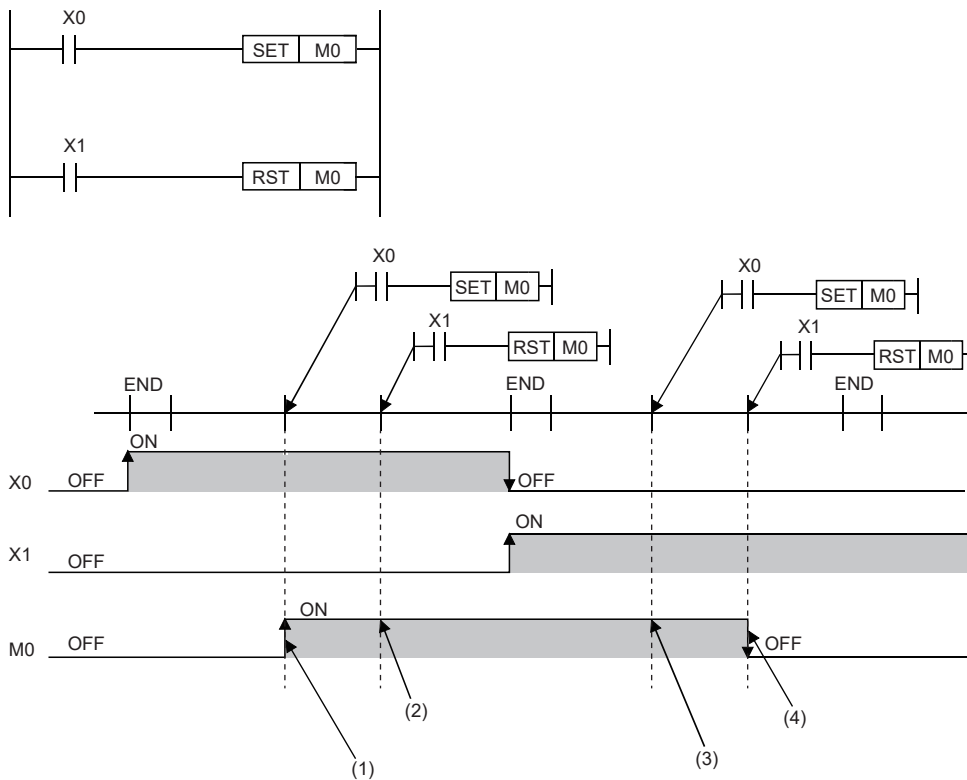
RST指令在執行指令為ON時將指定元件設為OFF、在執行指令為OFF時無處理。

因此將同一元件的RST指令在1個掃描中執行了多次的情況下，如果1個執行指令為ON，指定元件將變為OFF。

■1個掃描中存在有同一元件的SET指令及RST指令的情況下

1個掃描中存在有同一元件的SET指令及RST指令的情況下，SET指令在執行指令為ON時將指定元件設為ON、RST指令在執行指令為ON時將指定元件設為OFF。

SET指令及RST指令的執行指令為OFF的情況下，指定元件的ON/OFF狀態不變化。



(1) X0為ON，因此M0變為ON。

(2) X1為OFF，因此M0保持為ON不變。(RST指令將變為無處理。)

(3) X0為OFF，因此M0保持為ON不變。(SET指令將變為無處理。)

(4) X1為ON，因此M0變為OFF。

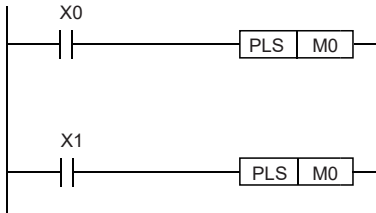
透過SET/RST指令指定輸出(Y)時，1個掃描的最後執行的SET/RST指令的ON/OFF狀態將被輸出。

使用了同一元件的PLS指令的情況下

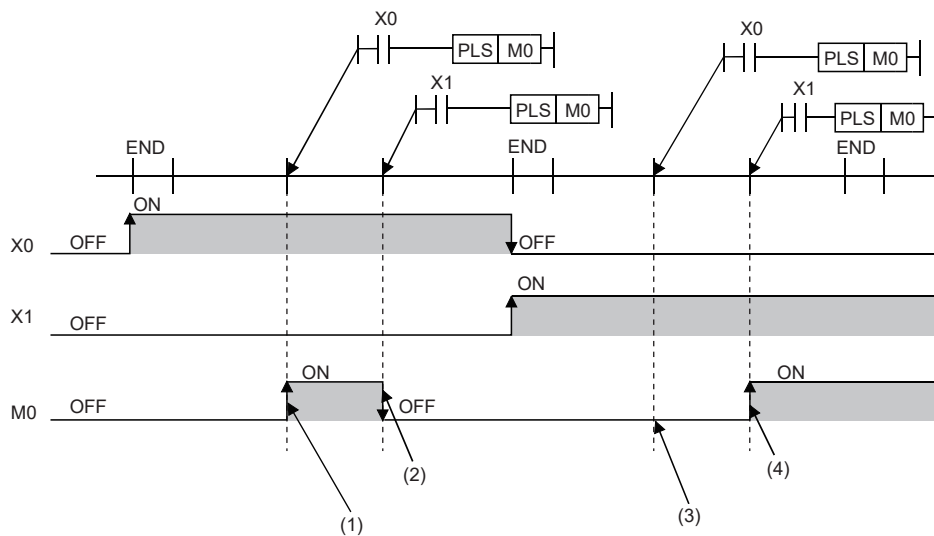
PLS指令在執行指令的OFF→ON時將指定元件設為ON。OFF→ON以外(OFF→OFF、ON→ON、ON→OFF)時，將指定元件設為OFF。

在1個掃描中多次執行了同一元件的PLS指令的情況下，在各PLS指令的執行指令的OFF→ON時，將指定元件設為ON。OFF→ON以外時將指定元件設為OFF。

因此，在1個掃描中多次執行了同一元件的PLS指令的情況下，透過PLS指令設為ON的元件有可能不執行1個掃描ON。

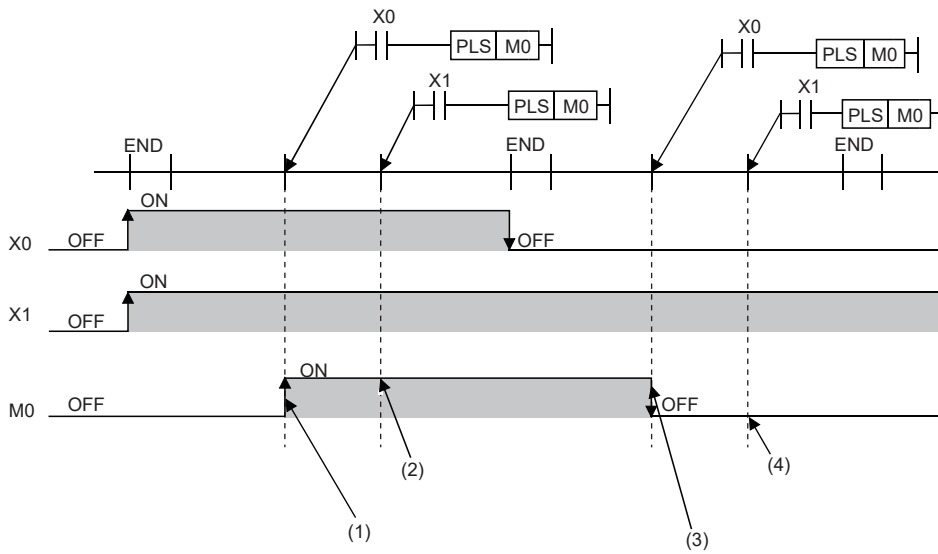


- X0與X1的ON/OFF時機不相同時(指定元件未1個掃描ON。)



- (1) X0為OFF→ON，因此M0變為ON。
- (2) X1為OFF→ON以外，因此M0變為OFF。
- (3) X0為OFF→ON以外，因此M0保持為OFF不變。
- (4) X1為OFF→ON，因此M0變為ON。

- X0及X1的OFF→ON為同一時機時



- (1) X0為OFF→ON，因此M0變為ON。
- (2) X1為OFF→ON，因此M0保持為OFF不變。
- (3) X0為OFF→ON以外，因此M0變為OFF。
- (4) X1為OFF→ON以外，因此M0保持為OFF不變。

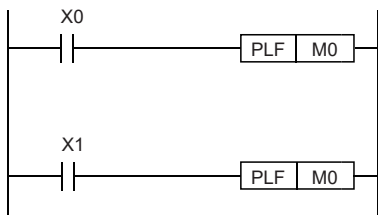
透過PLS指令指定輸出(Y)時，1個掃描的最後執行的PLS指令的ON/OFF狀態將被輸出。

使用了同一元件的PLF指令的情況下

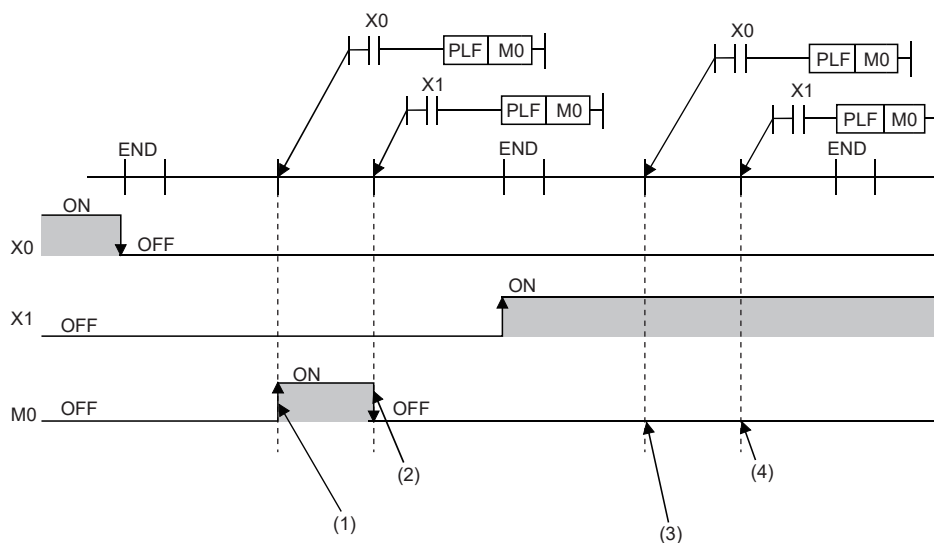
PLF指令在執行指令的ON→OFF時將指定元件設為ON。ON→OFF以外(OFF→OFF、OFF→ON、ON→ON)時，將指定元件設為OFF。

在1個掃描中多次執行了同一元件的PLF指令的情況下，各PLF指令的執行指令為ON→OFF時，將指定元件設為ON。ON→OFF以外時將指定元件設為OFF。

因此，在1個掃描中多次執行了同一元件的PLF指令的情況下，透過PLF指令設為ON的元件有可能不執行1個掃描ON。

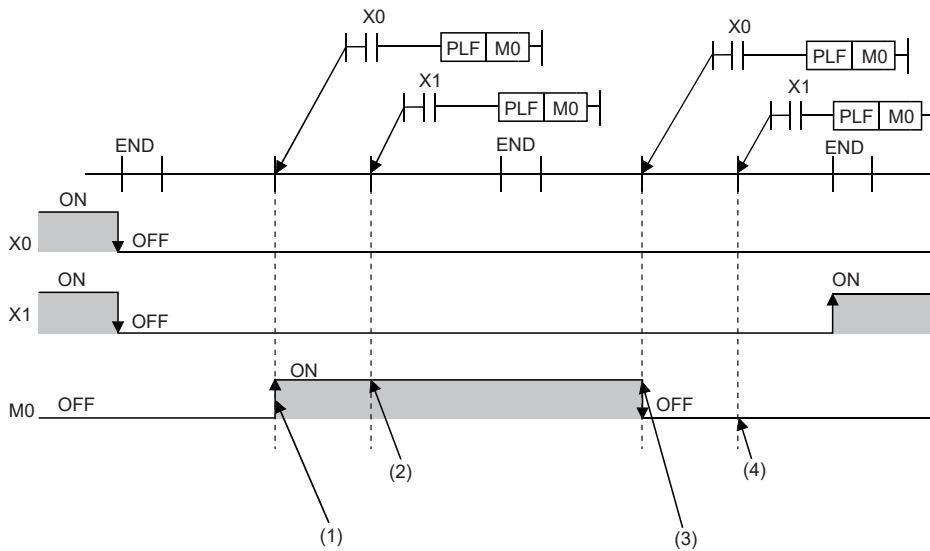


- X0與X1的ON/OFF時機不相同時(指定元件未1個掃描ON。)



- (1) X0為ON→OFF，因此M0變為ON。
- (2) X1為ON→OFF以外，因此M0變為OFF。
- (3) X0為ON→OFF以外，因此M0保持為OFF不變。
- (4) X1為ON→OFF以外，因此M0保持為OFF不變。

- X0及X1的ON→OFF為同一時機時



- (1) X0為ON→OFF，因此M0變為ON。
- (2) X1為ON→OFF，因此M0保持為ON不變。
- (3) X0為ON→OFF以外，因此M0變為OFF。
- (4) X1為ON→OFF以外，因此M0保持為OFF不變。

透過PLF指令指定輸出(Y)時，1個掃描最後執行的PLF指令的ON/OFF狀態將被輸出。

使用檔案寄存器時的限制事項

更新元件中指定了檔案寄存器的情況下，應注意下述限制事項。

指定了與程式同名的檔案寄存器的情況下

參數中指定為使用與程式同名的檔案寄存器的情況下，將無法正確更新。使用與程式同名的檔案寄存器的情況下，將被更新為與程式設定中設定為最後編號的程式同名的檔案寄存器。

希望讀寫更新資料的情況下，應使用QDRSET指令，切換為相應的檔案寄存器之後在進行指定。

透過QDRSET指令更改了檔案名及驅動器No. 的情況下

透過QDRSET指令更改了檔案寄存器的檔案名的情況下及更改了驅動器No. 的情況下，將被連結到更新之前的設定檔案。

希望讀寫更新資料的情況下，應指定為更新之前的設定檔案。

透過RSET指令切換了塊No. 的情況下

透過RSET指令切換了塊No. 的情況下，應注意以下事項。

- 將被更新為切換後的塊No. 的檔案寄存器(R)。
- 將被更新為更新之前的塊No. 的檔案寄存器(R)。

希望讀寫更新資料的情況下，應指定為更新之前的塊No.。

第2部分 指令/FUN/FB一覽

本部分由下述章節構成。

2 CPU模組用指令

3 模組專用指令

4 通用函數/通用FB

2 CPU模組用指令

一覽表的閱讀方法如下所示。

項目	內容
指令符號	表示指令名。
處理內容	表示指令的概要。
參閱目標	表示詳細說明的參閱目標。

2.1 順控程式指令

觸點指令

■運算開始、串聯連接、並聯連接

指令符號	處理內容	參閱目標
LD	將指定元件的ON/OFF資訊作為運算結果。(常開觸點運算開始指令)	148頁 LD、LDI、AND、ANI、OR、ORI
LDI	將指定元件的ON/OFF資訊作為運算結果。(常閉觸點運算開始指令)	
AND	將指定元件的ON/OFF資訊與至當時為止的運算結果的AND運算作為運算結果。(常開觸點串聯連接指令)	
ANI	將指定元件的ON/OFF資訊與至當時為止的運算結果的AND運算作為運算結果。(常閉觸點串聯連接指令)	
OR	將指定元件的ON/OFF資訊與至當時為止的運算結果的OR運算作為運算結果。(1個常開觸點的並聯連接指令)	
ORI	將指定元件的ON/OFF資訊與至當時為止的運算結果的OR運算作為運算結果。(1個常閉觸點的並聯連接指令)	

■脈衝運算開始、脈衝串聯連接、脈衝並聯連接

指令符號	處理內容	參閱目標
LDP	僅指定位元元件的上升沿時(OFF→ON)導通。(上升沿脈衝運算開始指令)	150頁 LDP、LDF、ANDP、ANDF、ORP、ORF
LDF	僅指定位元元件的下降沿時(ON→OFF)導通。(下降沿脈衝運算開始指令)	
ANDP	與至當時為止的運算結果進行AND運算。(上升沿脈衝串聯連接指令)	
ANDF	與至當時為止的運算結果進行AND運算。(下降沿脈衝串聯連接指令)	
ORP	與至當時為止的運算結果進行OR運算。(上升沿脈衝並聯連接指令)	
ORF	與至當時為止的運算結果進行OR運算。(下降沿脈衝並聯連接指令)	

■脈衝否定運算開始、脈衝否定串聯連接、脈衝否定並聯連接

指令符號	處理內容	參閱目標
LDPI	指定元件的OFF時、ON時、下降沿時(ON→OFF)的情況下導通。(上升沿脈衝否定運算開始指令)	153頁 LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI
LDFI	指定位元元件的上升沿時(OFF→ON)、OFF時、ON時導通。(下降沿脈衝否定運算開始指令)	
ANDPI	與至當時為止的運算結果進行AND運算。(上升沿脈衝否定串聯連接指令)	
ANDFI	與至當時為止的運算結果進行AND運算。(下降沿脈衝否定串聯連接指令)	
ORPI	與至當時為止的運算結果進行OR運算。(上升沿脈衝否定並聯連接指令)	
ORFI	與至當時為止的運算結果進行OR運算。(下降沿脈衝否定並聯連接指令)	

合併指令

■梯形圖塊串聯連接、並聯連接

指令符號	處理內容	參閱目標
ANB	進行邏輯塊之間的AND運算(邏輯塊之間的串聯連接)。	156頁 ANB、ORB
ORB	進行邏輯塊之間的OR運算(邏輯塊之間的串聯連接)。	

■運算結果推入、讀取、彈出

指令符號	處理內容	參閱目標
MPS	儲存MPS指令之前的運算結果(ON/OFF)。	157頁 MPS、MRD、MPP
MRD	讀取透過MPS指令儲存的運算結果。	
MPP	提取透過MPS指令儲存的運算結果。	

■運算結果反轉

指令符號	處理內容	參閱目標
INV	對INV指令之前的運算結果進行反轉。	159頁 INV

■運算結果脈衝化

指令符號	處理內容	參閱目標
MEP	MEP指令之前的運算結果的上升沿時(OFF→ON)導通。	160頁 MEP、MEF
MEF	MEF指令之前的運算結果的下降沿時(ON→OFF)導通。	

■邊緣繼電器運算結果脈衝化

指令符號	處理內容	參閱目標
EGP	將EGP指令之前的運算結果透過邊緣繼電器(V)進行儲存。在運算結果的上升沿時(OFF→ON)導通。	161頁 EGP、EGF
EGF	將EGF指令之前的運算結果透過邊緣繼電器(V)進行儲存。在運算結果的下降沿時(ON→OFF)導通。	

輸出指令

■OUT(定時器、計數器、報警器除外)

指令符號	處理內容	參閱目標
OUT	將運算結果輸出至指定的元件。	163頁 OUT

■定時器、超長定時器

指令符號	處理內容	參閱目標
OUT T	OUT指令之前的運算結果為ON時，線圈ON後進行定時器的計測。如果時限到，常開觸點將導通、常閉觸點將變為非導通。 <ul style="list-style-type: none"> • OUT T：低速定時器指令 • OUTH T：高速定時器指令 • OUT ST：低速累計定時器指令 • OUTH ST：高速累計定時器指令 • OUT LT：低速超長定時器指令 • OUT LST：低速超長累計定時器指令 	165頁 OUT T、OUTH T、OUT ST、OUTH ST
OUTH T		
OUT ST		
OUTH ST		
OUT LT		
OUT LST	168頁 OUT LT、OUT LST	

■計數器、超長計數器

指令符號	處理內容	參閱目標
OUT C	OUT指令之前的運算結果由OFF→ON變化時，計數器的當前值(計數值)將被+1。如果計數到，計數器的常開觸點將導通、常閉觸點將變為非導通。 <ul style="list-style-type: none"> • OUT C：計數器 • OUT LC：超長計數器 	171頁 OUT C
OUT LC		173頁 OUT LC

■報警器

指令符號	處理內容	參閱目標
OUT F	將OUT F指令之前的運算結果輸出至指定的報警器。	175頁 OUT F

■元件的設定(報警器除外)

指令符號	處理內容	參閱目標
SET	將指定位元設為ON。	176頁 SET

■元件的重設(報警器除外)

指令符號	處理內容	參閱目標
RST	將指定元件設為0(OFF)。對於定時器、計數器，將當前值設定為0、將觸點/線圈設為OFF。	178頁 RST

■報警器的設定、重設

指令符號	處理內容	參閱目標
SET F	將指定的報警器設為ON。	180頁 SET F
RST F	將指定的報警器設為OFF。	182頁 RST F

■上升沿、下降沿輸出

指令符號	處理內容	參閱目標
PLS	PLS指令的OFF→ON時使指定元件1個掃描ON。	184頁 PLS
PLF	PLF指令的ON→OFF時使指定元件1個掃描ON。	186頁 PLF

■位元元件輸出反轉

指令符號	處理內容	參閱目標
FF	將指定的元件狀態反轉。	188頁 FF

■直接輸出的脈衝化

指令符號	處理內容	參閱目標
DELTA	對指定的直接存取輸出(DY)進行脈衝輸出。	190頁 DELTA(P)
DELTAP		

移位指令

■位元元件移位

指令符號	處理內容	參閱目標
SFT	將1個小編號的元件的ON/OFF狀態移位至指定的元件後，將1個小編號的元件設為OFF。	192頁 SFT(P)
SFTP		

主控制指令

■主控制的設定、重設

指令符號	處理內容	參閱目標
MC	開始主控制。	194頁 MC、MCR
MCR	結束主控制。	

階段處理指令

■階段處理

指令符號	處理內容	參閱目標
PHASE	開始執行階段處理。	199頁 PHASE
PHASECHG	結束執行中的階段處理，移轉至指定的階段。	201頁 PHASECHG
PHASEEND	結束執行中的階段處理。	203頁 PHASEEND

結束指令

■主程式結束

指令符號	處理內容	參閱目標
FEND	將主程式與子程式、中斷程式分割時使用。	204頁 FEND

■順控程式結束

指令符號	處理內容	參閱目標
END	表示程式的最終。	205頁 END

停止指令

■順控程式停止

指令符號	處理內容	參閱目標
STOP	停止CPU模組的運算。(與將開關設為STOP側的情況相同。)	207頁 STOP

無處理指令

■無處理

指令符號	處理內容	參閱目標
NOP	在設定程式偵錯用空間等情況下使用。	208頁 NOP
NOPLF	在無處理的指令的情況下，對到當時的運算將無任何影響。	209頁 NOPLF

2.2 基本指令

比較運算指令

■BIN16位元資料比較

指令符號	處理內容	參閱目標
LD=、AND=、OR=	將對指定的2個BIN16位元資料透過常開觸點處理進行比較。	212頁 LD□(U)、AND□(U)、OR□(U)
LD=U、AND=U、OR=U		
LD<>、AND<>、OR<>		
LD<>U、AND<>U、OR<>U		
LD>、AND>、OR>		
LD>U、AND>U、OR>U		
LD<=、AND<=、OR<=		
LD<=U、AND<=U、OR<=U		
LD<、AND<、OR<		
LD<U、AND<U、OR<U		
LD>=、AND>=、OR>=		
LD>=U、AND>=U、OR>=U		

■BIN32位元資料比較

指令符號	處理內容	參閱目標
LDD=、ANDD=、ORD=	將對指定的2個BIN32位元資料透過常開觸點處理進行比較。	214頁 LDD□(U)、ANDD□(U)、ORD□(U)
LDD=U、ANDD=U、ORD=U		
LDD<>、ANDD<>、ORD<>		
LDD<>U、ANDD<>U、ORD<>U		
LDD>、ANDD>、ORD>		
LDD>U、ANDD>U、ORD>U		
LDD<=、ANDD<=、ORD<=		
LDD<=U、ANDD<=U、ORD<=U		
LDD<、ANDD<、ORD<		
LDD<U、ANDD<U、ORD<U		
LDD>=、ANDD>=、ORD>=		
LDD>=U、ANDD>=U、ORD>=U		

■BIN16位元資料比較輸出

指令符號	處理內容	參閱目標
CMP	比較(s1)中指定的BIN16位元資料與(s2)中指定的BIN16位元資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為0N。	216頁 CMP(P)(U)
CMPP		
CMP_U		
CMPP_U		

■BIN32位元資料比較輸出

指令符號	處理內容	參閱目標
DCMP	比較(s1)中指定的BIN32位元資料與(s2)中指定的BIN32位元資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為ON。	218頁 DCMP(P) (_U)
DCMPP		
DCMP_U		
DCMPP_U		

■BIN16位元資料區域比較輸出

指令符號	處理內容	參閱目標
ZCP	對下限值(s1)的BIN16位元資料與上限值(s2)的BIN16位元資料的區域，在比較資料(s3)的BIN16位元資料中進行比較，根據比較的結果(下、區域內、上)，(d)、(d)+1、(d)+2其中之一將變為ON。	220頁 ZCP(P) (_U)
ZCPP		
ZCP_U		
ZCPP_U		

■BIN32位元資料區域比較輸出

指令符號	處理內容	參閱目標
DZCP	對下限值(s1)的BIN32位元資料與上限值(s2)的BIN32位元資料的區域，在比較資料(s3)的BIN32位元資料中進行比較，根據比較的結果(下、區域內、上)，(d)、(d)+1、(d)+2其中之一將變為ON。	222頁 DZCP(P) (_U)
DZCPP		
DZCP_U		
DZCPP_U		

■BIN16位元塊資料比較

指令符號	處理內容	參閱目標
BKCMPE、BKCMPE<>、BKCMPE>、 BKCMPE<=、BKCMPE<、BKCMPE>=	對指定的2個BIN16位元塊資料進行比較。	224頁 BKCMPE(P) (_U)
BKCMPE=P、BKCMPE<>P、 BKCMPE>P、BKCMPE<=P、 BKCMPE<P、BKCMPE>=P		
BKCMPE=_U、BKCMPE<>_U、 BKCMPE>_U、BKCMPE<=_U、 BKCMPE<_U、BKCMPE>=_U		
BKCMPE=P_U、BKCMPE<>P_U、 BKCMPE>P_U、BKCMPE<=P_U、 BKCMPE<P_U、BKCMPE>=_U		

■BIN32位元塊資料比較

指令符號	處理內容	參閱目標
DBKCMPE、DBKCMPE<>、 DBKCMPE>、DBKCMPE<=、 DBKCMPE<、DBKCMPE>=	對指定的2個BIN32位元塊資料進行比較。	226頁 DBKCMPE(P) (_U)
DBKCMPE=P、DBKCMPE<>P、 DBKCMPE>P、DBKCMPE<=P、 DBKCMPE<P、DBKCMPE>=P		
DBKCMPE=_U、DBKCMPE<>_U、 DBKCMPE>_U、DBKCMPE<=_U、 DBKCMPE<_U、DBKCMPE>=_U		
DBKCMPE=P_U、DBKCMPE<>P_U、 DBKCMPE>P_U、DBKCMPE<=P_U、 DBKCMPE<P_U、DBKCMPE>=P_U		

算術運算指令

■BIN16位元加減法運算

指令符號	處理內容	參閱目標
+	對指定的2個BIN16位元資料進行加法運算。(操作數為2個的情況下)	229頁 +(P) (U) [操作數為2個的情況下]
+P		
+_U		
+P_U		
+	對指定的2個BIN16位元資料進行加法運算。(操作數為3個的情況下)	231頁 +(P) (U) [操作數為3個的情況下]
+P		
+_U		
+P_U		
-	對指定的2個BIN16位元資料進行減法運算。(操作數為2個的情況下)	233頁 -(P) (U) [操作數為2個的情況下]
-P		
-_U		
-P_U		
-	對指定的2個BIN16位元資料進行減法運算。(操作數為3個的情況下)	235頁 -(P) (U) [操作數為3個的情況下]
-P		
-_U		
-P_U		

■BIN32位元加減法運算

指令符號	處理內容	參閱目標
D+	對指定的2個BIN32位元資料進行加法運算。(操作數為2個的情況下)	237頁 D+(P) (U) [操作數為2個的情況下]
D+P		
D+_U		
D+P_U		
D+	對指定的2個BIN32位元資料進行加法運算。(操作數為3個的情況下)	239頁 D+(P) (U) [操作數為3個的情況下]
D+P		
D+_U		
D+P_U		
D-	對指定的2個BIN32位元資料進行減法運算。(操作數為2個的情況下)	241頁 D-(P) (U) [操作數為2個的情況下]
D-P		
D-_U		
D-P_U		
D-	對指定的2個BIN32位元資料進行減法運算。(操作數為3個的情況下)	243頁 D-(P) (U) [操作數為3個的情況下]
D-P		
D-_U		
D-P_U		

■BIN16位元乘除法運算

指令符號	處理內容	參閱目標
*	對指定的2個BIN16位元資料進行乘法運算。	245頁 *(P) (U)
*P		
*_U		
*P_U		
/	對指定的2個BIN16位元資料進行除法運算。	247頁 /(P) (U)
/P		
/_U		
/P_U		

■BIN32位元乘法運算

指令符號	處理內容	參閱目標
D*	對指定的2個BIN32位元資料進行乘法運算。	249頁 D*(P) (U)
D*P		
D*_U		
D*P_U		
D/	對指定的2個BIN32位元資料進行除法運算。	251頁 D/(P) (U)
D/P		
D/_U		
D/P_U		

■BCD4位加減法運算

指令符號	處理內容	參閱目標
B+	對指定的2個BCD4位資料進行加法運算。(操作數為2個的情況下)	253頁 B+(P) [操作數為2個的情況下]
B+P		
B+	對指定的2個BCD4位資料進行加法運算。(操作數為3個的情況下)	254頁 B+(P) [操作數為3個的情況下]
B+P		
B-	對指定的2個BCD4位資料進行減法運算。(操作數為2個的情況下)	256頁 B-(P) [操作數為2個的情況下]
B-P		
B-	對指定的2個BCD4位資料進行減法運算。(操作數為3個的情況下)	257頁 B-(P) [操作數為3個的情況下]
B-P		

■BCD8位加減法運算

指令符號	處理內容	參閱目標
DB+	對指定的2個BCD8位資料進行加法運算。(操作數為2個的情況下)	259頁 DB+(P) [操作數為2個的情況下]
DB+P		
DB+	對指定的2個BCD8位資料進行加法運算。(操作數為3個的情況下)	260頁 DB+(P) [操作數為3個的情況下]
DB+P		
DB-	對指定的2個BCD8位資料進行減法運算。(操作數為2個的情況下)	262頁 DB-(P) [操作數為2個的情況下]
DB-P		
DB-	對指定的2個BCD8位資料進行減法運算。(操作數為3個的情況下)	263頁 DB-(P) [操作數為3個的情況下]
DB-P		

■BCD4位乘法運算

指令符號	處理內容	參閱目標
B*	對指定的2個BCD4位資料進行乘法運算。	265頁 B*(P)
B*P		
B/	對指定的2個BCD4位資料進行除法運算。	267頁 B/(P)
B/P		

■BCD8位乘法運算

指令符號	處理內容	參閱目標
DB*	對指定的2個BCD8位資料進行乘法運算。	269頁 DB*(P)
DB*P		
DB/	對指定的2個BCD8位資料進行除法運算。	271頁 DB/(P)
DB/P		

■BIN16位元塊資料加減法運算

指令符號	處理內容	參閱目標
BK+	對指定的2個BIN16位元資料塊進行加法運算。	273頁 BK+(P) (_U)
BK+P		
BK+_U		
BK+P_U		
BK-	對指定的2個BIN16位元資料塊進行減法運算。	275頁 BK-(P) (_U)
BK-P		
BK-_U		
BK-P_U		

■BIN32位元塊資料加減法運算

指令符號	處理內容	參閱目標
DBK+	對指定的2個BIN32位元資料塊進行加法運算。	277頁 DBK+(P) (_U)
DBK+P		
DBK+_U		
DBK+P_U		
DBK-	對指定的2個BIN32位元資料塊進行減法運算。	280頁 DBK-(P) (_U)
DBK-P		
DBK-_U		
DBK-P_U		

■16位元BIN資料增量、減量

指令符號	處理內容	參閱目標
INC	對指定的BIN16位元資料進行+1。	283頁 INC(P) (_U)
INCP		
INC_U		
INCP_U		
DEC	對指定的BIN16位元資料進行-1。	285頁 DEC(P) (_U)
DECP		
DEC_U		
DECP_U		

■32位元BIN資料增量、減量

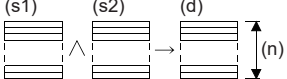
指令符號	處理內容	參閱目標
DINC	對指定的BIN32位元資料進行+1。	287頁 DINC(P) (_U)
DINCP		
DINC_U		
DINCP_U		
DDEC	對指定的BIN32位元資料進行-1。	289頁 DDEC(P) (_U)
DDECP		
DDEC_U		
DDECP_U		

邏輯運算指令

■16位元/32位元資料邏輯且

指令符號	處理內容	參閱目標
WAND	進行指定的2個BIN16位元資料的邏輯且。(操作數為2個的情況下)	291頁 WAND(P) [操作數為2個的情況下]
WANDP		
WAND	進行指定的2個BIN16位元資料的邏輯且。(操作數為3個的情況下)	293頁 WAND(P) [操作數為3個的情況下]
WANDP		
DAND	進行指定的2個BIN32位元資料的邏輯且。(操作數為2個的情況下)	295頁 DAND(P) [操作數為2個的情況下]
DANDP		
DAND	進行指定的2個BIN32位元資料的邏輯且。(操作數為3個的情況下)	297頁 DAND(P) [操作數為3個的情況下]
DANDP		

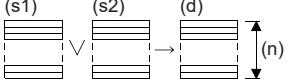
■16位元塊資料邏輯且

指令符號	處理內容	參閱目標
BKAND	對指定的2個BIN16位元資料塊的邏輯且。 	299頁 BKAND(P)
BKANDP		

■16位元/32位元資料邏輯或

指令符號	處理內容	參閱目標
WOR	進行指定的2個BIN16位元資料的邏輯或。(操作數為2個的情況下)	301頁 WOR(P) [操作數為2個的情況下]
WORP		
WOR	進行指定的2個BIN16位元資料的邏輯或。(操作數為3個的情況下)	303頁 WOR(P) [操作數為3個的情況下]
WORP		
DOR	進行指定的2個BIN32位元資料的邏輯或。(操作數為2個的情況下)	305頁 DOR(P) [操作數為2個的情況下]
DORP		
DOR	進行指定的2個BIN32位元資料的邏輯或。(操作數為3個的情況下)	307頁 DOR(P) [操作數為3個的情況下]
DORP		

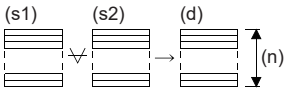
■16位元塊資料邏輯或

指令符號	處理內容	參閱目標
BKOR	進行指定的2個BIN16位元資料塊的邏輯或。 	309頁 BKOR(P)
BKORP		

■16位元/32位元資料排他邏輯或

指令符號	處理內容	參閱目標
WXOR	進行指定的2個BIN16位元資料的排他邏輯或。(操作數為2個的情況下)	311頁 WXOR(P) [操作數為2個的情況下]
WXORP		
WXOR	進行指定的2個BIN16位元資料的排他邏輯或。(操作數為3個的情況下)	313頁 WXOR(P) [操作數為3個的情況下]
WXORP		
DXOR	進行指定的2個BIN32位元資料的排他邏輯或。(操作數為2個的情況下)	315頁 DXOR(P) [操作數為2個的情況下]
DXORP		
DXOR	進行指定的2個BIN32位元資料的排他邏輯或。(操作數為3個的情況下)	317頁 DXOR(P) [操作數為3個的情況下]
DXORP		

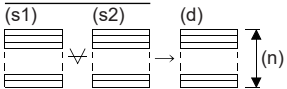
■16位元塊資料排他邏輯或

指令符號	處理內容	參閱目標
BKXOR	進行指定的2個BIN16位元資料塊的排他邏輯或。	319頁 BKXOR (P)
BKXORP		

■16位元/32位元資料否定排他邏輯或

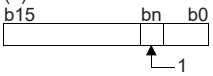
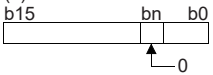
指令符號	處理內容	參閱目標
WXNR	進行指定的2個BIN16位元資料的否定排他邏輯或。(操作數為2個的情況下)	321頁 WXNR (P) [操作數為2個的情況下]
WXNRP		
WXNR	進行指定的2個BIN16位元資料的否定排他邏輯或。(操作數為3個的情況下)	323頁 WXNR (P) [操作數為3個的情況下]
WXNRP		
DXNR	進行指定的2個BIN32位元資料的否定排他邏輯或。(操作數為2個的情況下)	325頁 DXNR (P) [操作數為2個的情況下]
DXNRP		
DXNR	進行指定的2個BIN32位元資料的否定排他邏輯或。(操作數為3個的情況下)	327頁 DXNR (P) [操作數為3個的情況下]
DXNRP		

■16位元塊資料否定排他邏輯或

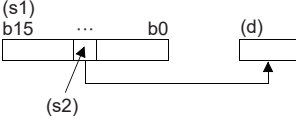
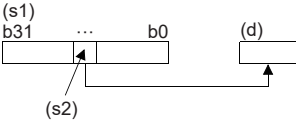
指令符號	處理內容	參閱目標
BKXNR	進行指定的2個BIN16位元資料塊的否定排他邏輯或。	329頁 BKXNR (P)
BKXNRP		

位元處理指令

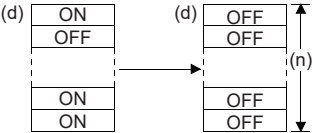
■字元元件的位元設定/重設

指令符號	處理內容	參閱目標
BSET	將指定的字元元件內的第n位元進行設定(1)。	331頁 BSET(P)
BSETP	(d) 	
BRST	將指定的字元元件內的第n位元進行重設(0)。	333頁 BRST(P)
BRSTP	(d) 	

■位元測試

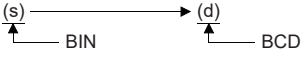

指令符號	處理內容	參閱目標
TEST	提取指定的字元元件內的第n位元。	335頁 TEST(P)
TESTP	(s1) 	
DTEST	提取指定的2字元元件內的第n位元。	337頁 DTEST(P)
DTESTP	(s1) 	

■位元元件的批量重設

指令符號	處理內容	參閱目標
BKRST	將指定的位元元件開始的n點的位元元件進行重設。	339頁 BKRST(P)
BKRSTP	(d) 	

資料轉換指令

■BIN資料→BCD4位/8位轉換

指令符號	處理內容	參閱目標
BCD	將指定的BIN16位元資料(0~9999)轉換為BCD4位資料。	405頁 BCD(P)
BCDP	(s) → (d) 	
DBCDC	將指定的BIN32位元資料(0~99999999)轉換為BCD8位資料。	407頁 DBCD(P)
DBCDCP	(s)+1, (s) → (d)+1, (d) 	

■BCD4位/8位→BIN16位元/32位元資料轉換

指令符號	處理內容	參閱目標
BIN BINP	將指定的BCD4位資料 (0~9999) 轉換為BIN16位元資料。 $\begin{array}{ccc} (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & \text{BCD} & \text{BIN} \end{array}$	409頁 BIN (P)
DBIN DBINP	將指定的BCD8位資料 (0~99999999) 轉換為BIN32位元資料。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & \text{BCD} & \text{BIN} \end{array}$	411頁 DBIN (P)

■單精度實數→帶符號BIN16位元/32位元資料

指令符號	處理內容	參閱目標
FLT2INT FLT2INTP	將指定的單精度實數 (-32768~32767) 轉換為帶符號BIN16位元資料。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 實數	413頁 FLT2INT (P)
FLT2DINT FLT2DINTP	將指定的單精度實數 (-2147483648~2147483647) 轉換為帶符號BIN32位元資料。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 實數	417頁 FLT2DINT (P)

■單精度實數→無符號BIN16位元/32位元資料

指令符號	處理內容	參閱目標
FLT2UINT FLT2UINTP	將指定的單精度實數 (0~65535) 轉換為無符號BIN16位元資料。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 實數	415頁 FLT2UINT (P)
FLT2UDINT FLT2UDINTP	將指定的單精度實數 (0~4294967295) 轉換為無符號BIN32位元資料。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 實數	419頁 FLT2UDINT (P)

■雙精度實數→帶符號BIN16位元/32位元資料

指令符號	處理內容	參閱目標
DBL2INT DBL2INTP	將指定的雙精度實數 (-32768~32767) 轉換為帶符號BIN16位元資料。 $\begin{array}{ccc} (s)+3, (s)+2, (s)+1, (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 實數	421頁 DBL2INT (P)
DBL2DINT DBL2DINTP	將指定的雙精度實數 (-2147483648~2147483647) 轉換為帶符號BIN32位元資料。 $\begin{array}{ccc} (s)+3, (s)+2, (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 實數	425頁 DBL2DINT (P)

■雙精度實數→無符號BIN16位元/32位元資料

指令符號	處理內容	參閱目標
DBL2UINT	將指定的雙精度實數 (0~65535) 轉換為無符號BIN16位元資料。	423頁 DBL2UINT (P)
DBL2UINTP	$\begin{array}{ccc} (s)+3, (s)+2, (s)+1, (s) & \xrightarrow{\hspace{1cm}} & (d) \\ \uparrow & & \uparrow \\ & & \text{BIN} \\ (1) & & \end{array}$ <p>(1): 實數</p>	
DBL2UDINT	將指定的雙精度實數 (0~4294967295) 轉換為無符號BIN32位元資料。	427頁 DBL2UDINT (P)
DBL2UDINTP	$\begin{array}{ccc} (s)+3, (s)+2, (s)+1, (s) & \xrightarrow{\hspace{1cm}} & (d)+1, (d) \\ \uparrow & & \uparrow \\ & & \text{BIN} \\ (1) & & \end{array}$ <p>(1): 實數</p>	

■帶符號BIN16位元資料→無符號BIN16位元/32位元資料轉換

指令符號	處理內容	參閱目標
INT2UINT	將 (s) 中指定的帶符號BIN16位元資料轉換為無符號BIN16位元資料後，儲存到 (d) 中。	429頁 INT2UINT (P)
INT2UINTP		
INT2UDINT	將 (s) 中指定的帶符號BIN16位元資料轉換為無符號BIN32位元資料後，儲存到 (d) 中。	433頁 INT2UDINT (P)
INT2UDINTP		

■帶符號BIN16位元資料→帶符號BIN32位元資料轉換

指令符號	處理內容	參閱目標
INT2DINT	將 (s) 中指定的帶符號BIN16位元資料轉換為帶符號BIN32位元資料後，儲存到 (d) 中。	431頁 INT2DINT (P)
INT2DINTP		

■無符號BIN16位元資料→帶符號BIN16位元/32位元資料轉換

指令符號	處理內容	參閱目標
UINT2INT	將 (s) 中指定的無符號BIN16位元資料轉換為帶符號BIN16位元資料後，儲存到 (d) 中。	435頁 UINT2INT (P)
UINT2INTP		
UINT2DINT	將 (s) 中指定的無符號BIN16位元資料轉換為帶符號BIN32位元資料後，儲存到 (d) 中。	437頁 UINT2DINT (P)
UINT2DINTP		

■無符號BIN16位元資料→無符號BIN32位元資料轉換

指令符號	處理內容	參閱目標
UINT2UDINT	將 (s) 中指定的無符號BIN16位元資料轉換為無符號BIN32位元資料後，儲存到 (d) 中。	439頁 UINT2UDINT (P)
UINT2UDINTP		

■帶符號BIN32位元資料→帶符號BIN16位元資料轉換

指令符號	處理內容	參閱目標
DINT2INT	將 (s) 中指定的帶符號BIN32位元資料轉換為帶符號BIN16位元資料後，儲存到 (d) 中。	441頁 DINT2INT (P)
DINT2INTP		

■帶符號BIN32位元資料→無符號BIN16位元/32位元資料轉換

指令符號	處理內容	參閱目標
DINT2UINT	將 (s) 中指定的帶符號BIN32位元資料轉換為無符號BIN16位元資料後，儲存到 (d) 中。	443頁 DINT2UINT (P)
DINT2UINTP		
DINT2UDINT	將 (s) 中指定的帶符號BIN32位元資料轉換為無符號BIN32位元資料後，儲存到 (d) 中。	445頁 DINT2UDINT (P)
DINT2UDINTP		

■無符號BIN32位元資料→帶符號BIN16位元/32位元資料轉換

指令符號	處理內容	參閱目標
UDINT2INT	將 (s) 中指定的無符號BIN32位元資料轉換為帶符號BIN16位元資料後，儲存到 (d) 中。	447頁 UDINT2INT (P)
UDINT2INTP		
UDINT2DINT	將 (s) 中指定的無符號BIN32位元資料轉換為帶符號BIN32位元資料後，儲存到 (d) 中。	451頁 UDINT2DINT (P)
UDINT2DINTP		

■無符號BIN32位元資料→無符號BIN16位元資料轉換

指令符號	處理內容	參閱目標
UDINT2UINT	將 (s) 中指定的無符號BIN32位元資料轉換為無符號BIN16位元資料後，儲存到 (d) 中。	449頁 UDINT2UINT (P)
UDINT2UINTP		

■BIN16位元/32位元資料→BIN16位元/32位元格雷碼資料轉換

指令符號	處理內容	參閱目標
GRY	將指定的BIN16位元資料 (-32768~32767) 轉換為BIN16位元格雷碼資料。 $\begin{array}{ccc} (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & \text{BIN} & \text{Gc} \end{array}$ Gc: 格雷碼	453頁 GRY (P) (_U)
GRYP		
GRY_U	將指定的BIN16位元資料 (0~65535) 轉換為BIN16位元格雷碼資料。 $\begin{array}{ccc} (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & \text{BIN} & \text{Gc} \end{array}$ Gc: 格雷碼	
GRYP_U		
DGRY	將指定的BIN32位元資料 (-2147483648~2147483647) 轉換為BIN32位元格雷碼資料。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & \text{BIN} & \text{Gc} \end{array}$ Gc: 格雷碼	455頁 DGRY (P) (_U)
DGRYP		
DGRY_U	將指定的BIN32位元資料 (0~4294967295) 轉換為BIN32位元格雷碼資料。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & \text{BIN} & \text{Gc} \end{array}$ Gc: 格雷碼	
DGRYP_U		

■BIN16位元/32位元格雷碼資料→BIN16位元/32位元資料轉換

指令符號	處理內容	參閱目標
GBIN	將指定的BIN16位元格雷碼資料 (-32768~32767) 轉換為BIN16位元資料。 $\begin{array}{ccc} (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & \text{Gc} & \text{BIN} \end{array}$ Gc: 格雷碼	457頁 GBIN (P) (_U)
GBINP		
GBIN_U	將指定的BIN16位元格雷碼資料 (0~65535) 轉換為BIN16位元資料。 $\begin{array}{ccc} (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & \text{Gc} & \text{BIN} \end{array}$ Gc: 格雷碼	
GBINP_U		
DGBIN	將指定的BIN32位元格雷碼資料 (-2147483648~2147483647) 轉換為BIN32位元資料。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & \text{Gc} & \text{BIN} \end{array}$ Gc: 格雷碼	459頁 DGBIN (P) (_U)
DGBINP		
DGBIN_U	將指定的BIN32位元格雷碼資料 (0~4294967295) 轉換為BIN32位元資料。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & \text{Gc} & \text{BIN} \end{array}$ Gc: 格雷碼	
DGBINP_U		

■BIN16位元資料塊→BCD4位資料轉換塊

指令符號	處理內容	參閱目標
BKBCD	將從 (s) 開始的 (n) 點的BIN資料批量轉換為BCD資料後，儲存到 (d) 及其以後。	461頁 BKBCD (P)
BKBCDP		

■塊BCD4位資料→塊BIN16位元轉換資料

指令符號	處理內容	參閱目標
BKBIN	將從 (s) 開始的 (n) 點的BCD資料批量轉換為BIN資料後，儲存到 (d) 及其以後。	463頁 BKBIN (P)
BKBINP		

■10進制ASCII碼資料→BIN16位元/32位元資料轉換

指令符號	處理內容	參閱目標
DABIN	將 (s) 中指定的10進制ASCII值轉換為1字元BIN值後，儲存到 (d) 中指定的字元元件編號中。	465頁 DABIN (P) (_U)
DABINP		
DABIN_U		
DABINP_U		
DDABIN	將 (s) 中指定的10進制ASCII值轉換為2字元BIN值後，儲存到 (d) 中指定的字元元件編號中。	468頁 DDABIN (P) (_U)
DDABINP		
DDABIN_U		
DDABINP_U		

■16進制ASCII碼資料→BIN16位元/32位元資料轉換

指令符號	處理內容	參閱目標
HABIN	將 (s) 中指定的16進制ASCII值轉換為1字元BIN值後，儲存到 (d) 中指定的字元元件編號中。	471頁 HABIN (P)
HABINP		
DHABIN	將 (s) 中指定的16進制ASCII值轉換為2字元BIN值後，儲存到 (d) 中指定的字元元件編號中。	473頁 DHABIN (P)
DHABINP		

■10進制ASCII碼資料→BCD4位/8位轉換

指令符號	處理內容	參閱目標
DABCD	將 (s) 中指定的10進制ASCII值轉換為1字元BCD值後，儲存到 (d) 中指定的字元元件編號中。	476頁 DABCD (P)
DABCDP		
DDABCD	將 (s) 中指定的10進制ASCII值轉換為2字元BCD值後，儲存到 (d) 中指定的字元元件編號中。	479頁 DDABCD (P)
DDABCDP		

■10進制字元串→BIN16位元/32位元資料轉換

指令符號	處理內容	參閱目標
VAL	將 (s) 中指定包含小數點的字元串轉換為1字元BIN值及小數部位數後，儲存到 (d1)、(d2) 中指定的元件中。	482頁 VAL (P) (_U)
VALP		
VAL_U		
VALP_U		
DVAL	將 (s) 中指定包含小數點的字元串轉換為2字元BIN值及小數部位數後，儲存到 (d1)、(d2) 中指定的元件中。	485頁 DVAL (P) (_U)
DVALP		
DVAL_U		
DVALP_U		



■16進制ASCII碼→16進制資料BIN轉換

指令符號	處理內容	參閱目標
ASC2INT	將 (s) 中指定的字元元件以後的16進制ASCII碼資料以 (n) 中指定的字元數轉換為BIN值後，儲存到 (d) 中指定的元件編號及其以後。	488頁 ASC2INT (P)
ASC2INTP		

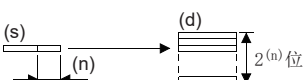
■單精度實數→BCD分解

指令符號	處理內容	參閱目標
EMOD	將 (s1) 的32位元浮點資料轉換為 (s2) 中指定的小數部位數的BCD後，儲存到 (d) 中指定的元件中。	490頁 EMOD (P)
EMODP		

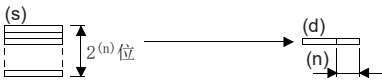
■BIN16位元/32位元資料2的補數(符號反轉)

指令符號	處理內容	參閱目標
NEG	對BIN16位元元件的符號進行反轉。	492頁 NEG (P)
NEGP		
DNEG	對BIN32位元元件的符號進行反轉。	493頁 DNEG (P)
DNEGP		


■8→256位元解碼

指令符號	處理內容	參閱目標
DECO	對指定的元件的低位(n)位元進行解碼。	494頁 DECO (P)
DECOP		

■256→8位元編碼

指令符號	處理內容	參閱目標
ENCO	對2的(n)次方位元的資料進行編碼。	496頁 ENCO (P)
ENCOP		

■7段解碼

指令符號	處理內容	參閱目標
SEG	將元件的低4位元中指定的0~F的資料解碼為7段顯示資料。	498頁 SEG (P)
SEGP		

■BIN16位元資料的4位元分離

指令符號	處理內容	參閱目標
DIS	將(s)中指定的16位元資料以4位元單位分離，儲存到(d)開始的(n)點的低位4位元中。(n<4)	500頁 DIS (P)
DISP		

■BIN16位元資料的4位元合併

指令符號	處理內容	參閱目標
UNI	將(s)中指定的元件開始的(n)點的低位4位元資料合併後，儲存到(d)中指定的元件中。(n<4)	502頁 UNI (P)
UNIP		

■任意資料的位元分離、合併

指令符號	處理內容	參閱目標
NDIS	將(s1)中指定的元件以後的資料分離為(s2)及其以後中指定的位元後，從(d)中指定的元件開始依次儲存。	504頁 NDIS (P)
NDISP		
NUNI	將(s1)中指定的元件以後的資料合併到(s2)及其以後中指定的各個位元中後，從(d)中指定的元件開始依次儲存。	506頁 NUNI (P)
NUNIP		

■字節單位資料分離、合併

指令符號	處理內容	參閱目標
WTOB	從(s)中指定的元件開始，將(n)點的16位元資料分解為8位元單位後，從(d)中指定的元件開始依次儲存。	508頁 WTOB (P)
WTOBP		
BTOW	從(s)中指定的元件開始，將(n)點的16位元資料的低8位元合併為16位元後，從(d)中指定的元件開始依次儲存。	510頁 BTOW (P)
BTOWP		

資料移位指令

■BIN16位元資料的n位元右移、左移

指令符號	處理內容	參閱目標
SFR	將指定元件的BIN16位元資料右移。移位後的空餘資料將儲存0。	341頁 SFR (P)
SFRP		
SFL	將指定元件的BIN16位元資料左移。移位後的空餘資料將儲存0。	343頁 SFL (P)
SFLP		

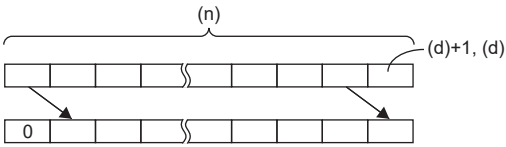
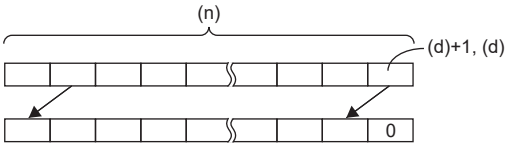
■n位元資料的1位元右移、左移

指令符號	處理內容	參閱目標
BSFR	從指定的元件開始將n點向右移1位元。移位後的空餘資料將儲存0。	345頁 BSFR (P)
BSFRP		
BSFL	從指定的元件開始將n點向左移1位元。移位後的空餘資料將儲存0。	347頁 BSFL (P)
BSFLP		

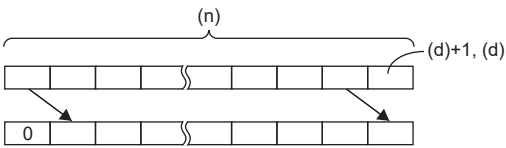
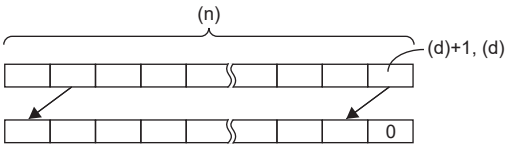
■n字元資料的1字元右移、左移

指令符號	處理內容	參閱目標
DSFR	從指定的元件開始將n點右移1字元。移位後的空餘資料將儲存0。	349頁 DSFR (P)
DSFRP		
DSFL	從指定的元件開始將n點左移1字元。移位後的空餘資料將儲存0。	351頁 DSFL (P)
DSFLP		

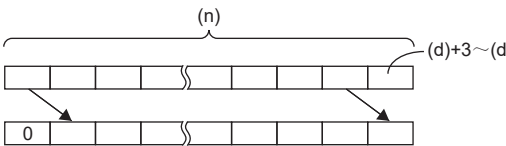
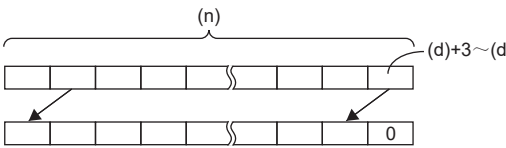
■n雙字元資料的1雙字元右移、左移

指令符號	處理內容	參閱目標
DDSFR DDSFRP	從指定的元件開始將n雙字元的資料向右移1雙字元。移位後的空餘資料將儲存0。 	353頁 DDSFR (P)
DDSFL DDSFLP	從指定的元件開始將n雙字元的資料向左移1雙字元。移位後的空餘資料將儲存0。 	355頁 DDSFL (P)

■單精度實數n點的1點右移、左移

指令符號	處理內容	參閱目標
ESFR ESFRP	從指定的元件開始將單精度實數n點的資料向右移1點。移位後的空餘資料將儲存0。 	357頁 ESFR (P)
ESFL ESFLP	從指定的元件開始將單精度實數n點的資料向左移1點。移位後的空餘資料將儲存0。 	359頁 ESFL (P)

■雙精度實數n點的1點右移、左移

指令符號	處理內容	參閱目標
EDSFR EDSFRP	從指定的元件開始將雙精度實數n點的資料向右移1點。移位後的空餘資料將儲存0。 	361頁 EDSFR (P)
EDSFL EDSFLP	從指定的元件開始將雙精度實數n點的資料向左移1點。移位後的空餘資料將儲存0。 	363頁 EDSFL (P)

■n位元資料的n位元右移、左移

指令符號	處理內容	參閱目標
SFTBR SFTBRP	<p>從指定的元件開始將n位元的資料右移n位元。移位後的空餘資料將儲存0。</p>	365頁 SFTBR (P)
SFTR SFTRP	<p>從指定的元件開始在(n1)位元的資料範圍內向右移(n2)位元。移位後的空餘資料將儲存指定資料。</p>	367頁 SFTR (P)
SFTBL SFTBLP	<p>從指定的元件開始將n位元的資料左移n位元。移位後的空餘資料將儲存0。</p>	369頁 SFTBL (P)
SFTL SFTLP	<p>從指定的元件開始在(n1)位元的資料範圍內向左移(n2)位元。移位後的空餘資料將儲存指定資料。</p>	371頁 SFTL (P)

■n字元資料的n字元右移、左移

指令符號	處理內容	參閱目標
SFTWR SFTWRP	從指定的元件開始將n字元的資料右移n字元。移位後的空餘資料將儲存0。 	373頁 SFTWR (P)
WSFR WSFRP	從指定的元件開始在(n1)字元的資料範圍內向右移(n2)字元。移位後的空餘資料將儲存指定資料。 	375頁 WSFR (P)
SFTWL SFTWLP	從指定的元件開始將n字元的資料左移n字元。移位後的空餘資料將儲存0。 	377頁 SFTWL (P)
WSFL WSFLP	從指定的元件開始在(n1)字元的資料範圍內向左移(n2)字元。移位後的空餘資料將儲存指定資料。 	379頁 WSFL (P)

■n雙字元資料的n雙字元右移、左移

指令符號	處理內容	參閱目標
SFTDWR SFTDWRP	從指定的元件開始在(n1)雙字元的資料範圍內向右移(n2)雙字元。移位後的空餘資料將儲存0。 	381頁 SFTDWR (P)
DWSFTR DWSFTRP	從指定的元件開始在(n1)雙字元的資料範圍內向右移(n2)雙字元。移位後的空餘資料將儲存指定資料。 	383頁 DWSFTR (P)
SFTDWL SFTDWLP	從指定的元件開始在(n1)雙字元的資料範圍內向左移(n2)雙字元。移位後的空餘資料將儲存0。 	385頁 SFTDWL (P)

指令符號	處理內容	參閱目標
DWSFTL DWSFTLP	<p>從指定的元件開始在(n1)雙字元的資料範圍內向左移(n2)雙字元。移位後的空餘資料將儲存指定資料。</p>	387頁 DWSFTL(P)

■單精度實數n點的n點右移、左移

指令符號	處理內容	參閱目標
SFTR SFTRP	<p>從指定的元件開始在單精度實數(n1)點的資料範圍內向右移(n2)點。移位後的空餘資料將儲存0。</p>	389頁 SFTR(P)
ESFTR ESFTRP	<p>從指定的元件開始在單精度實數(n1)點的資料範圍內向右移(n2)點。移位後的空餘資料將儲存指定資料。</p>	391頁 ESFTR(P)
SFTL SFTLP	<p>從指定的元件開始在單精度實數(n1)點的資料範圍內向左移(n2)點。移位後的空餘資料將儲存0。</p>	393頁 SFTL(P)
ESFTL ESFTLP	<p>從指定的元件開始在單精度實數(n1)點的資料範圍內向左移(n2)點。移位後的空餘資料將儲存指定資料。</p>	395頁 ESFTL(P)

■雙精度實數n點的n點右移、左移

指令符號	處理內容	參閱目標
SFTEDR SFTEDRP	<p>從指定的元件開始在雙精度實數(n1)點的資料範圍內向右移(n2)點。移位後的空餘資料將儲存0。</p>	397頁 SFTEDR(P)
EDSFTR EDSFTRP	<p>從指定的元件開始在雙精度實數(n1)點的資料範圍內向右移(n2)點。移位後的空餘資料將儲存指定資料。</p>	399頁 EDSFTR(P)
SFTEDL SFTEDLP	<p>從指定的元件開始在雙精度實數(n1)點的資料範圍內向左移(n2)點。移位後的空餘資料將儲存0。</p>	401頁 SFTEDL(P)
EDSFTL EDSFTLP	<p>從指定的元件開始在雙精度實數(n1)點的資料範圍內向左移(n2)點。移位後的空餘資料將儲存指定資料。</p>	403頁 EDSFTL(P)

資料傳送指令

■BIN16位元/32位元資料傳送

指令符號	處理內容	參閱目標
MOV	對指定的元件的BIN16位元資料進行傳送。	512頁 MOV (P)
MOVP	(s) \longrightarrow (d)	
DMOV	對指定的元件的BIN32位元資料進行傳送。	514頁 DMOV (P)
DMOVP	(s)+1, (s) \longrightarrow (d)+1, (d)	

■BIN16位元/32位元資料否定傳送

指令符號	處理內容	參閱目標
CML	將指定的BIN16位元資料逐位元反轉後進行傳送。	516頁 CML (P)
CMLP	(s) \longrightarrow (d)	
DCML	將指定的BIN32位元資料逐位元反轉後進行傳送。	518頁 DCML (P)
DCMLP	(s)+1, (s) \longrightarrow (d)+1, (d)	

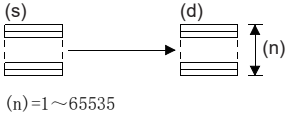
■位移動

指令符號	處理內容	參閱目標
SMOV	在位單位(4位元)中進行資料的分配/合成。	520頁 SMOV (P)
SMOVP		

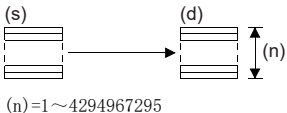
■1位元資料反轉傳送

指令符號	處理內容	參閱目標
CMLB	將(s)中指定的位元資料反轉後，儲存到(d)中。	523頁 CMLB (P)
CMLBP		

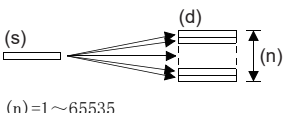
■BIN16位元資料塊傳送(16位元)

指令符號	處理內容	參閱目標
BMOV	對指定的元件開始的(n)點的BIN16位元資料進行批量傳送。	525頁 BMOV (P)
BMOVP	 (n)=1~65535	

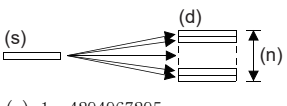
■BIN16位元資料塊傳送(32位元)

指令符號	處理內容	參閱目標
BMOVL	對指定的元件開始的(n)點的BIN16位元資料進行批量傳送。	527頁 BMOVL (P)
BMOVLP	 (n)=1~4294967295	

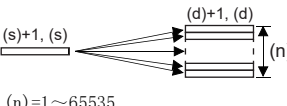
■同一BIN16位元資料塊傳送(16位元)

指令符號	處理內容	參閱目標
FMOV	將BIN16位元資料傳送到指定的元件開始的(n)點處。	529頁 FMOV (P)
FMOVP	 (n)=1~65535	

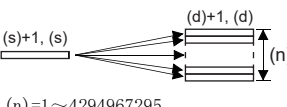
■同一BIN16位元資料塊傳送(32位元)

指令符號	處理內容	參閱目標
FMOVL	將BIN16位元資料傳送到指定的元件開始的(n)點處。	531頁 FMOVL(P)
FMOVLP	 <p>(n)=1~4294967295</p>	

■同一BIN32位元資料塊傳送(16位元)

指令符號	處理內容	參閱目標
DFMOV	將BIN32位元資料傳送到指定的元件開始的(n)點處。	533頁 DFMOV(P)
DFMOVLP	 <p>(n)=1~65535</p>	

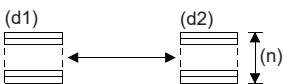
■同一BIN32位元資料塊傳送(32位元)

指令符號	處理內容	參閱目標
DFMOVL	將BIN32位元資料傳送到指定的元件開始的(n)點處。	535頁 DFMOVL(P)
DFMOVLP	 <p>(n)=1~4294967295</p>	

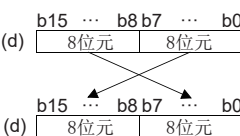
■BIN16位元/32位元資料交換

指令符號	處理內容	參閱目標
XCH	對指定的BIN16位元資料進行交換。	537頁 XCH(P)
XCHP	(d1) ←→ (d2)	
DXCH	對指定的BIN32位元資料進行交換。	539頁 DXCH(P)
DXCHP	(d1)+1, (d1) ←→ (d2)+1, (d2)	

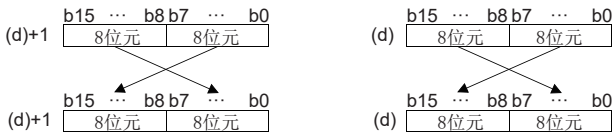
■BIN16位元資料塊交換

指令符號	處理內容	參閱目標
BXCH	對指定的元件開始的(n)點的BIN16位元資料進行交換。	541頁 BXCH(P)
BXCHP		

■BIN16位元資料上下字節交換

指令符號	處理內容	參閱目標
SWAP	對指定的元件的上下各8位元的值進行交換。	543頁 SWAP(P)
SWAPP		

■BIN32位元資料上下字節交換

指令符號	處理內容	參閱目標
DSWAP	對指定的元件的上下各8位元的值進行交換。	544頁 DSWAP (P)
DSWAPP		

■1位元資料傳送

指令符號	處理內容	參閱目標
MOVB	將 (s) 中指定的位元資料儲存到 (d) 中。	546頁 MOVB (P)
MOVBP		

■n位元資料傳送

指令符號	處理內容	參閱目標
BLKMOVB	將從 (s) 開始的 (n) 點的位元資料批量傳送到 (d) 開始的 (n) 點的位元資料中。	548頁 BLKMOVB (P)
BLKMOVBP		

2.3 應用指令

程式控制

程式分支指令

■指針分支

指令符號	處理內容	參閱目標
CJ	執行同一程式檔案內指定指針編號的程式。	553頁 CJ、SCJ、JMP
SCJ	從下一個掃描開始執行同一程式檔案內指定指針編號的程式。	
JMP	無條件執行同一程式檔案內指定指針編號的程式。	

■跳轉至END

指令符號	處理內容	參閱目標
GOEND	跳轉至同一程式檔案內的FEND或END指令處。	556頁 GOEND

程式執行控制指令

■中斷禁止、中斷允許

指令符號	處理內容	參閱目標
DI	禁止中斷程式的執行。	557頁 DI、EI
EI	解除中斷禁止狀態。	

■指定優先度及其以下優先度的中斷禁止

指令符號	處理內容	參閱目標
DI	即使發生了(s)中指定的優先度以下的中斷程式的啟動原因，在執行EI指令之前也將禁止中斷程式的執行。	559頁 DI

■中斷程式屏蔽

指令符號	處理內容	參閱目標
IMASK	將指定的中斷指針編號的中斷程式設為允許執行狀態/禁止執行狀態。	564頁 IMASK

■指定中斷指針的禁止/允許

指令符號	處理內容	參閱目標
SIMASK	將指定的中斷指針編號的中斷程式設為允許執行狀態/禁止執行狀態。	566頁 SIMASK

■從中斷程式的恢復

指令符號	處理內容	參閱目標
IRET	表示中斷程式處理的結束。	568頁 IRET

■WDT重設

指令符號	處理內容	參閱目標
WDT	對看門狗定時器進行重設。	569頁 WDT(P)
WDTP		

結構化指令

■FOR～NEXT

指令符號	處理內容	參閱目標
FOR	將FOR～NEXT指令之間的處理執行(n)次。	570頁 FOR、NEXT
NEXT		

■FOR～NEXT強制結束

指令符號	處理內容	參閱目標
BREAK	強制結束透過FOR～NEXT指令進行的重複處理後，將執行移轉至指定的指針處。	572頁 BREAK (P)
BREAKP		

■子程式調用

指令符號	處理內容	參閱目標
CALL	透過輸入條件成立執行(P)的子程式。 (s1)～(s5)為至子程式的引數。)	574頁 CALL (P)
CALLP		

■從子程式返回

指令符號	處理內容	參閱目標
RET	表示子程式的結束。	578頁 RET

■子程式的輸出OFF調用

指令符號	處理內容	參閱目標
FCALL	輸入條件不成立時進行不執行(P)的子程式的處理。 (s1)～(s5)為至子程式的引數。)	579頁 FCALL (P)
FCALLP		

■程式檔案之間子程式調用

指令符號	處理內容	參閱目標
ECALL	透過輸入條件成立執行指定程式的(P)的子程式。 (s1)～(s5)為至子程式的引數。)	583頁 ECALL (P)
ECALLP		

■程式檔案之間子程式輸出OFF調用

指令符號	處理內容	參閱目標
EFCALL	輸入條件不成立時進行不執行(P)的子程式的處理。(s1)～(s5)為至子程式的引數。)	588頁 EFCALL (P)
EFCALLP		

■子程式調用

指令符號	處理內容	參閱目標
XCALL	透過輸入條件成立執行(P)的子程式。 輸入條件不成立時進行不執行(P)的子程式的處理。 (s1)～(s5)為至子程式的引數。)	592頁 XCALL

程式控制用指令

■程式待機

指令符號	處理內容	參閱目標
PSTOP	將指定的程式設為待機類型。	597頁 PSTOP (P)
PSTOPP		

■程式輸出OFF待機

指令符號	處理內容	參閱目標
POFF	將指定程式的OUT指令的線圈設為OFF，設為待機類型。	599頁 POFF (P)
POFFP		

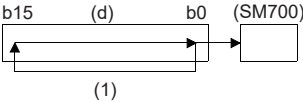
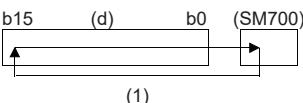
■程式掃描執行登錄

指令符號	處理內容	參閱目標
PSCAN	將指定的程式登錄為掃描執行類型。	603頁 PSCAN(P)
PSCANP		

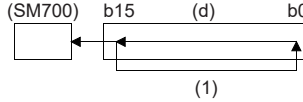
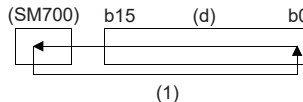
資料處理

旋轉指令

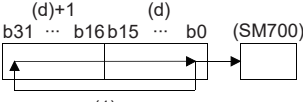
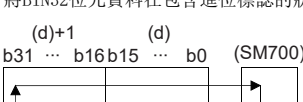
■BIN16位元資料的右旋轉

指令符號	處理內容	參閱目標
ROR	將BIN16位元資料在不包含進位標誌的狀況下右旋轉(n)位元。	603頁 ROR(P)、RCR(P)
RORP	 <p>(1)</p> <p>(1): (n)位元右旋轉</p>	
RCR	將BIN16位元資料在包含進位標誌的狀況下右旋轉(n)位元。	
RCRP	 <p>(1)</p> <p>(1): (n)位元右旋轉</p>	

■BIN16位元資料的左旋轉

指令符號	處理內容	參閱目標
ROL	將BIN16位元資料在不包含進位標誌的狀況下左旋轉(n)位元。	606頁 ROL(P)、RCL(P)
ROLP	 <p>(1)</p> <p>(1): (n)位元左旋轉</p>	
RCL	將BIN16位元資料在包含進位標誌的狀況下左旋轉(n)位元。	
RCLP	 <p>(1)</p> <p>(1): (n)位元左旋轉</p>	

■BIN32位元資料的右旋轉

指令符號	處理內容	參閱目標
DROR	將BIN32位元資料在不包含進位標誌的狀況下右旋轉(n)位元。	609頁 DROR(P)、DRCR(P)
DRORP	 <p>(1)</p> <p>(1): (n)位元右旋轉</p>	
DRCR	將BIN32位元資料在包含進位標誌的狀況下右旋轉(n)位元。	
DRCRP	 <p>(1)</p> <p>(1): (n)位元右旋轉</p>	

■BIN32位元資料的左旋轉

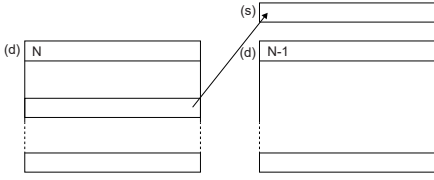
指令符號	處理內容	參閱目標
DROL DROLP	<p>將BIN32位元資料在不包含進位標誌的狀況下左旋轉(n)位元。</p> <p>(1): (n)位元左旋轉</p>	611頁 DROL (P)、 DRCL (P)
DRCL DRCLP	<p>將BIN32位元資料在包含進位標誌的狀況下左旋轉(n)位元。</p> <p>(1): (n)位元左旋轉</p>	

資料表操作指令

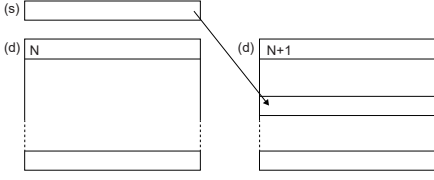
■從資料表的先入資料讀取

指令符號	處理內容	參閱目標
FIFR FIFRP	<p>將最先儲存到表中的資料儲存到指定的元件中。</p> <p>N: 資料數</p>	613頁 FIFR (P)

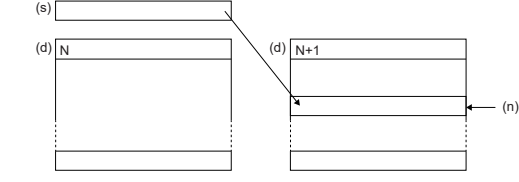
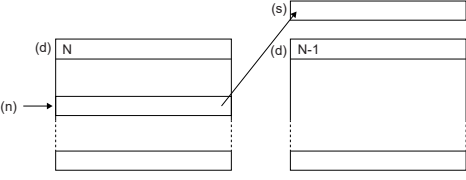
■從資料表的後入資料讀取

指令符號	處理內容	參閱目標
FPOP FPOPP	將最後儲存到表中的資料儲存到指定的元件中。  N: 資料數	615頁 FPOP (P)

■至資料表的資料寫入

指令符號	處理內容	參閱目標
FIFW FIFWP	將BIN16位元資料儲存到指定的資料表中。  N: 資料數	617頁 FIFW (P)

■資料表的資料插入、刪除

指令符號	處理內容	參閱目標
FINS FINSP	將BIN16位元資料插入到指定資料表的第(n)個中。  N: 資料數	619頁 FINS (P)
FDEL FDELP	將資料表的第(n)個資料刪除。  N: 資料數	621頁 FDEL (P)

資料讀取/寫入指令

■從資料記憶體的資料讀取

指令符號	處理內容	參閱目標
S.DEVLD SP.DEVLD	從資料記憶體上的元件資料儲存用檔案中讀取資料。	624頁 S(P).DEVLD

■至資料記憶體的資料寫入

指令符號	處理內容	參閱目標
SP.DEVST	將資料寫入到資料記憶體上的元件資料儲存用檔案中。	626頁 SP.DEVST

檔案操作指令

■從指定檔案的資料讀取

指令符號	處理內容	參閱目標
SP. FREAD	從指定的檔案中讀取資料。	628頁 SP. FREAD

■至指定檔案的資料寫入

指令符號	處理內容	參閱目標
SP. FWRITE	將資料寫入到指定的檔案中。	645頁 SP. FWRITE

■刪除指定檔案

指令符號	處理內容	參閱目標
SP. FDELETE	刪除指定檔案或資料夾。	661頁 SP. FDELETE

■複製指定檔案

指令符號	處理內容	參閱目標
SP. FCOPY	複製指定檔案或資料夾。	664頁 SP. FCOPY

■移動指定檔案

指令符號	處理內容	參閱目標
SP. FMOVE	移動指定檔案或資料夾。	669頁 SP. FMOVE

■更改指定檔案名

指令符號	處理內容	參閱目標
SP. FRENAME	更改指定檔案名稱或資料夾名稱。	674頁 SP. FRENAME

■取得指定檔案的狀態

指令符號	處理內容	參閱目標
SP. FSTATUS	取得指定檔案或資料夾的狀態。	677頁 SP. FSTATUS

資料控制指令

■BIN16位元/32位元資料上下限限位控制

指令符號	處理內容	參閱目標
LIMIT	根據指定的輸入值 (BIN16位元值) 是否在上下限限位值的範圍內，對輸出值進行控制。	681頁 LIMIT (P) (_U)
LIMITP		
LIMIT_U		
LIMITP_U		
DLIMIT	根據指定的輸入值 (BIN32位元值) 是否在上下限限位值的範圍內，對輸出值進行控制。	683頁 DLIMIT (P) (_U)
DLIMITP		
DLIMIT_U		
DLIMITP_U		

■BIN16位元/32位元資料死區控制

指令符號	處理內容	參閱目標
BAND	根據指定的輸入值 (BIN16位元值) 是否在死區的上下限範圍內，對輸出值進行控制。	685頁 BAND (P) (_U)
BANDP		
BAND_U		
BANDP_U		
DBAND	根據指定的輸入值 (BIN32位元值) 是否在死區的上下限範圍內，對輸出值進行控制。	687頁 DBAND (P) (_U)
DBANDP		
DBAND_U		
DBANDP_U		

■BIN16位元/32位元資料區域控制

指令符號	處理內容	參閱目標
ZONE	對指定的輸入值 (BIN16位元值) 附加偏置值。	689頁 ZONE (P) (_U)
ZONEP		
ZONE_U		
ZONEP_U		
DZONE	對指定的輸入值 (BIN32位元值) 附加偏置值。	691頁 DZONE (P) (_U)
DZONEP		
DZONE_U		
DZONEP_U		

■BIN16位元/32位元單位定標 (各點座標資料)

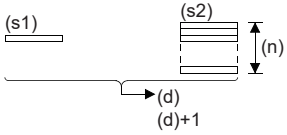
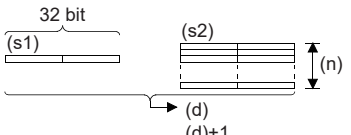
指令符號	處理內容	參閱目標
SCL	對 (s2) 中指定的定標用轉換資料 (16位元資料單位) 以 (s1) 中指定的輸入值進行定標, 將運算結果儲存到 (d) 中指定的元件中。 定標轉換是基於 (s2) 中指定的元件及其以後儲存的定標用轉換資料進行。	693頁 SCL (P) (_U)
SCLP		
SCL_U		
SCLP_U		
DSCL	對 (s2) 中指定的定標用轉換資料 (32位元資料單位) 以 (s1) 中指定的輸入值進行定標, 將運算結果儲存到 (d) 中指定的元件中。 定標轉換是基於 (s2) 中指定的元件及其以後儲存的定標用轉換資料進行。	696頁 DSCL (P) (_U)
DSCLP		
DSCL_U		
DSCLP_U		

■BIN16位元/32位元單位定標 (各X/Y座標資料)

指令符號	處理內容	參閱目標
SCL2	對 (s2) 中指定的定標用轉換資料 (16位元資料單位) 以 (s1) 中指定的輸入值進行定標, 將運算結果儲存到 (d) 中指定的元件中。 定標轉換是基於 (s2) 中指定的元件及其以後儲存的定標用轉換資料進行。	699頁 SCL2 (P) (_U)
SCL2P		
SCL2_U		
SCL2P_U		
DSCL2	對 (s2) 中指定的定標用轉換資料 (32位元資料單位) 以 (s1) 中指定的輸入值進行定標, 將運算結果儲存到 (d) 中指定的元件中。 定標轉換是基於 (s2) 中指定的元件及其以後儲存的定標用轉換資料進行。	701頁 DSCL2 (P) (_U)
DSCL2P		
DSCL2_U		
DSCL2P_U		

資料處理指令

■BIN16位元/32位元資料搜尋

指令符號	處理內容	參閱目標
SERDATA	<p>搜尋 (s1) 中指定的BIN16位元資料, 搜尋 (s2) 中指定的BIN16位元資料開始的 (n) 點分。</p>  <p>(d): 一致的位置 (d)+1: 一致個數</p>	703頁 SERDATA (P)
SERDATAP		
DSERDATA	<p>搜尋 (s1) 中指定的BIN32位元資料, 搜尋 (s2) 中指定的BIN32位元資料開始的 (n) 點分。</p>  <p>(d): 一致的位置 (d)+1: 一致個數</p>	705頁 DSERDATA (P)
DSERDATAP		

■BIN16位元/32位元資料搜尋(最小・相同・最大)

指令符號	處理內容	參閱目標
SERMM SERMMP	<p>搜尋從(s1)中指定的BIN16位元資料開始的(n)點分、(s2)中指定的BIN16位元資料與相同資料及最大值、最小值。</p>	707頁 SERMM(P)
DSERMM DSERMMP	<p>搜尋從(s1)中指定的BIN32位元資料開始的(n)點分、(s2)中指定的BIN32位元資料與相同資料及最大值、最小值。</p>	709頁 DSERMM(P)

■BIN16位元/32位元資料位元檢查

指令符號	處理內容	參閱目標
SUM SUMP	<p>將儲存指定的元件的BIN16位元資料中，變為1的位元的總數。</p> <p>(d): 1的總數</p>	711頁 SUM(P)
DSUM DSUMP	<p>將儲存指定的元件的BIN32位元資料中，變為1的位元的總數。</p> <p>(d): 1的總數</p>	712頁 DSUM(P)

■BIN16位元/32位元資料的位元判定

指令符號	處理內容	參閱目標
BON BONP	<p>檢查指定的元件的(n)位元的狀態是ON還是OFF後，將結果儲存至(d)中指定的元件中。</p>	713頁 BON(P)
DBON DBONP	<p>檢查指定的元件的(n)位元的狀態是ON還是OFF後，將結果儲存至(d)中指定的元件中。</p>	715頁 DBON(P)

■BIN16位元/32位元資料最大值搜尋

指令符號	處理內容	參閱目標
MAX MAXP MAX_U MAXP_U	<p>從(s)中指定的元件開始的(n)點的資料以16位元單位進行搜尋，將最大值儲存到(d)中指定的軟件中。</p>	717頁 MAX(P) (U)
DMAX DMAXP DMAX_U DMAXP_U	<p>從(s)中指定的元件開始的(n)點的資料以32位元單位進行搜尋，將最大值儲存到(d)中指定的軟件中。</p>	719頁 DMAX(P) (U)

■BIN16位元/32位元資料最小值搜尋

指令符號	處理內容	參閱目標
MIN MINP MIN_U MINP_U	<p>從(s)中指定的元件開始的(n)點的資料以16位元單位進行搜尋，將最小值儲存到(d)中指定的元件中。</p>	721頁 MIN(P) (U)
DMIN DMINP DMIN_U DMINP_U	<p>從(s)中指定的元件開始的(n)點的資料以32位元單位進行搜尋，將最小值儲存到(d)中指定的元件中。</p>	723頁 DMIN(P) (U)

■BIN16位元/32位元資料排序

指令符號	處理內容	參閱目標
SORTD	將從(s1)中指定的元件開始的(n)點的資料以16位元單位進行排序。 (需要 $(n) \times ((n)-1) \div 2$ 個掃描)	725頁 SORTD(_U)
SORTD_U		
DSORTD	將從(s1)中指定的元件開始的(n)點的資料以32位元單位進行排序。 (需要 $(n) \times ((n)-1) \div 2$ 個掃描)	727頁 DSORTD(_U)
DSORTD_U		

■BIN16位元資料表排序

指令符號	處理內容	參閱目標
SORTTBL	將(s)指定的縱向方向連續的((n1)×(n2))點的BIN16位元資料表(排序源)的資料列,以(n3)欄的資料為基準按遞增或遞減進行排序,並儲存至(d)指定的((n1)×(n2))點的BIN16位元資料表(排序後)。	729頁 SORTTBL(_U)
SORTTBL_U		

■BIN16位元/32位元資料表排序2

指令符號	處理內容	參閱目標
SORTTBL2	將(s)指定的橫向方向連續的((n1)×(n2))點的BIN16位元資料表(排序源)的資料列,以(n3)欄的資料為基準按遞增或遞減進行排序,並儲存至(d)指定的((n1)×(n2))點的BIN16位元資料表(排序後)。	733頁 SORTTBL2(_U)
SORTTBL2_U		
DSORTTBL2	將(s)指定的橫向方向連續的((n1)×(n2))點的BIN32位元資料表(排序源)的資料列,以(n3)欄的資料為基準按遞增或遞減進行排序,並儲存至(d)指定的((n1)×(n2))點的BIN32位元資料表(排序後)。	737頁 DSORTTBL2(_U)
DSORTTBL2_U		

■BIN16位元資料合計值計算

指令符號	處理內容	參閱目標
WSUM	將從(s)中指定的元件開始的(n)點的16位元BIN資料全部進行加法運算後,儲存到(d)中指定的元件中。	741頁 WSUM(P)(_U)
WSUM_U		
WSUMP		
WSUMP_U		

■BIN32位元資料合計值計算

指令符號	處理內容	參閱目標
DWSUM	將從(s)中指定的元件開始的(n)點的32位元BIN資料全部進行加法運算後,儲存到(d)中指定的元件中。	743頁 DWSUM(P)(_U)
DWSUM_U		
DWSUMP		
DWSUMP_U		

■BIN16位元/32位元資料平均值計算

指令符號	處理內容	參閱目標
MEAN	對從(s)中指定的元件開始的(n)點的(16位元單位)的平均值進行計算後,儲存到(d)中指定的元件中。	745頁 MEAN(P)(_U)
MEANP		
MEAN_U		
MEANP_U		
DMEAN	對從(s)中指定的元件開始的(n)點的(32位元單位)的平均值進行計算後,儲存到(d)中指定的元件中。	747頁 DMEAN(P)(_U)
DMEANP		
DMEAN_U		
DMEANP_U		

■BIN16位元/32位元平方根計算

指令符號	處理內容	參閱目標
SQRT	對指定的BIN16位元資料的平方根進行運算。 $\sqrt{(s)} \rightarrow (d)$	749頁 SQRT(P)
SQRTP		
DSQRT	對指定的BIN32位元資料的平方根進行運算。 $\sqrt{(s)+1}, (s) \rightarrow (d)$	750頁 DSQRT(P)
DSQRTP		

■CRC運算

指令符號	處理內容	參閱目標
CRC	關於以(s)中指定的元件起始的(n)點的8位元資料(字節單位),生成CRC值並儲存到(d)。	751頁 CRC(P)
CRCP		

檢查碼指令

■檢查碼

指令符號	處理內容	參閱目標
CCD	計算 (s) ~ (s)+(n)-1 中儲存的資料的加法運算資料與水平奇偶校驗，加法運算資料儲存在 (d) 中，水平奇偶校驗儲存至 (d)+1 中。	753 頁 CCD (P)
CCDP		

偵錯、故障診斷

偵錯、故障診斷指令

■出錯顯示或報警器重設

指令符號	處理內容	參閱目標
LEDR	對CPU模組的報警器顯示及可繼續運轉的自我診斷出錯的顯示進行重設。	756頁 LEDR

■運轉繼續異常指令

指令符號	處理內容	參閱目標
PALERT	CPU模組為繼續運行型出錯。	758頁 PALERT (P)
PALERTP		

■運轉停止異常指令

指令符號	處理內容	參閱目標
PABORT	停止程式的執行、將CPU模組設為停止型出錯。	760頁 PABORT

字元串處理

字元串處理指令

■字元串比較

指令符號	處理內容	參閱目標
LD\$=、AND\$=、OR\$=	將 (s1) 的字元串與 (s2) 的字元串進行逐個字元的比較。	762頁 LD\$□、AND\$□、OR\$□
LD\$<>、AND\$<>、OR\$<>		
LD\$>、AND\$>、OR\$>		
LD\$<=、AND\$<=、OR\$<=		
LD\$<、AND\$<、OR\$<		
LD\$>=、AND\$>=、OR\$>=		

■字元串的合併

指令符號	處理內容	參閱目標
\$+	將 (d) 中指定的字元串與 (s) 中指定的字元串合併後，儲存到 (d) 及其以後。	766頁 \$(P) [操作數為2個的情況下]
\$(P)		
\$+	將 (s1) 中指定的字元串與 (s2) 中指定的字元串合併後，儲存到 (d) 及其以後。	768頁 \$(P) [操作數為3個的情況下]
\$(P)		

■字元串傳送

指令符號	處理內容	參閱目標
\$MOV	將 (s) 中指定的字元串傳送至 (d) 中指定的元件及其以後。	770頁 \$MOV (P)
\$MOV P		
\$MOV_WS	將 (s) 中指定的Unicode字元串傳送至 (d) 中指定的元件及其以後。	772頁 \$MOV (P)_WS
\$MOV P_WS		

■BIN16位元/32位元資料→10進制ASCII碼轉換

指令符號	處理內容	參閱目標
BINDA	將 (s) 中指定的1字元BIN轉換為10進制ASCII值後，儲存到 (d) 中指定的字元元件編號及其以後。	774頁 BINDA (P) (_U)
BINDAP		
BINDA_U		
BINDAP_U		
DBINDA	將 (s) 中指定的2字元BIN轉換為10進制ASCII值後，儲存到 (d) 中指定的字元元件編號及其以後。	778頁 DBINDA (P) (_U)
DBINDAP		
DBINDA_U		
DBINDAP_U		

■BIN16位元/32位元資料→16進制ASCII碼轉換

指令符號	處理內容	參閱目標
BINHA	將 (s) 中指定的1字元BIN轉換為16進制ASCII值後，儲存到 (d) 中指定的字元元件編號及其以後。	783頁 BINHA (P)
BINHAP		
DBINHA	將 (s) 中指定的2字元BIN轉換為16進制ASCII值後，儲存到 (d) 中指定的字元元件編號及其以後。	787頁 DBINHA (P)
DBINHAP		

■BIN16位元/32位元資料→字元串轉換

指令符號	處理內容	參閱目標
STR	將 (s2) 中指定的1字元BIN值轉換為 (s1) 中指定的全部位數及小數部位數的10進制字元串後，儲存到 (d) 中指定的元件中。	791頁 STR (P) (_U)
STRP		
STR_U		
STRP_U		
DSTR	將 (s2) 中指定的2字元BIN值轉換為 (s1) 中指定的全部位數及小數部位數的10進制字元串後，儲存到 (d) 中指定的元件中。	794頁 DSTR (P) (_U)
DSTRP		
DSTR_U		
DSTRP_U		

■BCD4位/8位資料→10進制ASCII碼轉換

指令符號	處理內容	參閱目標
BCDDA	將 (s) 中指定的1字元BCD值轉換為10進制ASCII值後，儲存到 (d) 中指定的字元元件編號及其以後。	797頁 BCDDA (P)
BCDDAP		
DBCDDA	將 (s) 中指定的2字元BCD值轉換為10進制ASCII值後，儲存到 (d) 中指定的字元元件編號及其以後。	801頁 DBCDDA (P)
DBCDDAP		

■單精度實數→字元串轉換

指令符號	處理內容	參閱目標
ESTR	將 (s1) 中指定的單精度實數資料轉換為字元串後，儲存到 (d) 中指定的元件中。	806頁 ESTR (P)
ESTRP		

■16進制BIN資料→16進制ASCII碼轉換

指令符號	處理內容	參閱目標
INT2ASC	將 (s) 中指定的元件編號以後的1字元BIN值轉換為16進制ASCII碼後，以 (n) 中指定的字元數儲存到 (d) 中指定的字元元件編號及其以後。	810頁 INT2ASC (P)
INT2ASCP		

■Unicode字元串→移位JIS字元串轉換

指令符號	處理內容	參閱目標
WS2SJIS	將 (s) 中指定的Unicode字元串轉換為移位JIS字元串後，儲存到 (d) 中。	812頁 WS2SJIS (P)
WS2SJISP		

■移位JIS字元串→Unicode字元串轉換(無字節順序標記)

指令符號	處理內容	參閱目標
SJIS2WS	將 (s) 中指定的移位JIS字元串轉換為Unicode字元串後，儲存到 (d) 中。	814頁 SJIS2WS (P)
SJIS2WSP		

■移位JIS→Unicode轉換(有字節順序標記)

指令符號	處理內容	參閱目標
SJIS2WSB	將 (s) 中指定的移位JIS字元串轉換為Unicode字元串後，在起始處附加字節順序標記後儲存到 (d) 中。	816頁 SJIS2WSB (P)
SJIS2WSBP		

■字元串的長度檢測

指令符號	處理內容	參閱目標
LEN	將 (s) 中指定的元件中儲存的字元串資料的長度(字元數) 儲存到 (d) 中指定的元件中。	818頁 LEN (P)
LENP		

■從字元串的右側/左側提取

指令符號	處理內容	參閱目標
RIGHT	將 (s) 中指定的字元串的最終字元開始的 (n) 字元儲存到 (d) 的指定元件中。	820頁 RIGHT (P)
RIGHTP		
LEFT	將 (s) 中指定的字元串的起始字元開始的 (n) 字元儲存到 (d) 的指定元件中。	822頁 LEFT (P)
LEFTP		

■字元串中的任意提取、替換

指令符號	處理內容	參閱目標
MIDR	從 (s1) 中指定的字元串的 (s2) 中指定的位置開始，將指定的字元數儲存到 (d) 中指定的元件中。	824頁 MIDR (P)
MIDRP		
MIDW	將從 (s1) 的字元串開始的指定的字元數儲存到 (d) 的字元串的 (s2) 中指定的位置處。	826頁 MIDW (P)
MIDWP		

■字元串搜尋

指令符號	處理內容	參閱目標
INSTR	從 (s2) 的字元串的第 (s3) 字元開始搜尋 (s1) 的字元串，將一致的位置儲存到 (d) 中。	829頁 INSTR (P)
INSTRP		

■字元串插入

指令符號	處理內容	參閱目標
STRINS	將 (s1) 中指定的字元串資料，插入到從 (d) 中指定的字元串資料的起始開始的第 (s2) 字元 (插入位置) 處。	831頁 STRINS (P)
STRINSP		

■字元串刪除

指令符號	處理內容	參閱目標
STRDEL	從 (d) 中指定的字元串資料的起始開始，根據第 (s) 字元中指定的位置 (刪除開始位置)，刪除 (n) 字元的資料。	833頁 STRDEL (P)
STRDELP		

實數處理

浮點指令

■單精度實數比較

指令符號	處理內容	參閱目標
LDE=、ANDE=、ORE=	將單精度實數透過常開觸點處理進行比較運算。	835頁 LDE□、ANDE□、ORE□
LDE<>、ANDE<>、ORE<>		
LDE>、ANDE>、ORE>		
LDE<=、ANDE<=、ORE<=		
LDE<、ANDE<、ORE<		
LDE>=、ANDE>=、ORE>=		

■雙精度實數比較

指令符號	處理內容	參閱目標
LDED=、ANDED=、ORED=	將雙精度實數透過常開觸點處理進行比較運算。	838頁 LDED□、ANDED□、ORED□
LDED<>、ANDED<>、ORED<>		
LDED>、ANDED>、ORED>		
LDED<=、ANDED<=、ORED<=		
LDED<、ANDED<、ORED<		
LDED>=、ANDED>=、ORED>=		

■單精度實數比較輸出

指令符號	處理內容	參閱目標
ECMP	比較 (s1) 中指定的單精度實數資料與 (s2) 中指定的單精度實數資料，根據比較的結果 (小、一致、大)，(d)、(d)+1、(d)+2 其中之一將變為 ON。	841頁 ECMP (P)
ECMPP		

■雙精度實數比較輸出

指令符號	處理內容	參閱目標
EDCMP	比較(s1)中指定的雙精度實數資料與(s2)中指定的雙精度實數資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為0N。	843頁 EDCMP (P)
EDCMP		

■單精度實數區域比較輸出

指令符號	處理內容	參閱目標
EZCP	對於下限值(s1)的單精度實數與上限值(s2)的單精度實數的頻率，比較比較資料(s3)的單精度實數，根據比較的結果(下、區域內、上)，(d)、(d)+1、(d)+2其中之一將變為0N。	845頁 EZCP (P)
EZCPP		

■雙精度實數區域比較輸出

指令符號	處理內容	參閱目標
EDZCP	對於下限值(s1)的雙精度實數與上限值(s2)的雙精度實數的頻率，比較比較資料(s3)的雙精度實數，根據比較的結果(下、區域內、上)，(d)、(d)+1、(d)+2其中之一將變為0N。	847頁 EDZCP (P)
EDZCPP		

■單精度實數加減法運算

指令符號	處理內容	參閱目標
E+	對單精度實數進行加法運算。(操作數為2個的情況下)	849頁 E+(P) [操作數為2個的情況下]
E+P		
E+	對單精度實數進行加法運算。(操作數為3個的情況下)	851頁 E+(P) [操作數為3個的情況下]
E+P		
E-	對單精度實數進行減法運算。(操作數為2個的情況下)	853頁 E-(P) [操作數為2個的情況下]
E-P		
E-	對單精度實數進行減法運算。(操作數為3個的情況下)	855頁 E-(P) [操作數為3個的情況下]
E-P		

■雙精度實數加減法運算

指令符號	處理內容	參閱目標
ED+	對雙精度實數進行加法運算。(操作數為2個的情況下)	857頁 ED+(P) [操作數為2個的情況下]
ED+P		
ED+	對雙精度實數進行加法運算。(操作數為3個的情況下)	859頁 ED+(P) [操作數為3個的情況下]
ED+P		
ED-	對雙精度實數進行減法運算。(操作數為2個的情況下)	861頁 ED-(P) [操作數為2個的情況下]
ED-P		
ED-	對雙精度實數進行減法運算。(操作數為3個的情況下)	863頁 ED-(P) [操作數為3個的情況下]
ED-P		

■單精度實數乘除法運算

指令符號	處理內容	參閱目標
E*	對單精度實數進行乘法運算。	865頁 E*(P)
E*P		
E/	對單精度實數進行除法運算。	867頁 E/(P)
E/P		

■雙精度實數乘除法運算

指令符號	處理內容	參閱目標
ED*	對雙精度實數進行乘法運算。	869頁 ED*(P)
ED*P		
ED/	對雙精度實數進行除法運算。	871頁 ED/(P)
ED/P		

■帶符號BIN16位元/32位元資料→單精度實數轉換

指令符號	處理內容	參閱目標
INT2FLT	將 (s) 中指定的帶符號BIN16位元資料轉換為單精度實數後，儲存到 (d) 中。	873頁 INT2FLT (P)
INT2FLTP		
DINT2FLT	將 (s) 中指定的帶符號BIN32位元資料轉換為單精度實數後，儲存到 (d) 中。	875頁 DINT2FLT (P)
DINT2FLTP		

■無符號BIN16位元/32位元資料→單精度實數轉換

指令符號	處理內容	參閱目標
UINT2FLT	將 (s) 中指定的無符號BIN16位元資料轉換為單精度實數後，儲存到 (d) 中。	874頁 UINT2FLT (P)
UINT2FLTP		
UDINT2FLT	將 (s) 中指定的無符號BIN32位元資料轉換為單精度實數後，儲存到 (d) 中。	877頁 UDINT2FLT (P)
UDINT2FLTP		

■雙精度→單精度轉換

指令符號	處理內容	參閱目標
DBL2FLT	將 (s) 中指定的雙精度實數轉換為單精度實數後，儲存到 (d) 中。	879頁 DBL2FLT (P)
DBL2FLTP		

■帶符號BIN16位元/32位元資料→雙精度實數轉換

指令符號	處理內容	參閱目標
INT2DBL	將 (s) 中指定的帶符號BIN16位元資料轉換為雙精度實數後，儲存到 (d) 中。	881頁 INT2DBL (P)
INT2DBLP		
DINT2DBL	將 (s) 中指定的帶符號BIN32位元資料轉換為雙精度實數後，儲存到 (d) 中。	883頁 DINT2DBL (P)
DINT2DBLP		

■無符號BIN16位元/32位元資料→雙精度實數轉換

指令符號	處理內容	參閱目標
UINT2DBL	將 (s) 中指定的無符號BIN16位元資料轉換為雙精度實數後，儲存到 (d) 中。	882頁 UINT2DBL (P)
UINT2DBLP		
UDINT2DBL	將 (s) 中指定的無符號BIN32位元資料轉換為雙精度實數後，儲存到 (d) 中。	884頁 UDINT2DBL (P)
UDINT2DBLP		

■單精度→雙精度轉換

指令符號	處理內容	參閱目標
FLT2DBL	將 (s) 中指定的單精度實數轉換為雙精度實數後，儲存到 (d) 中。	885頁 FLT2DBL (P)
FLT2DBLP		

■字元串→單精度實數轉換

指令符號	處理內容	參閱目標
EVAL	將 (s) 中指定的字元串轉換為單精度實數後，儲存到 (d) 中指定的元件中。	887頁 EVAL (P)
EVALP		

■BCD格式資料→單精度實數資料轉換

指令符號	處理內容	參閱目標
EREXP	將 (s1) 的BCD資料以 (s2) 中指定的小數部位數轉換為單精度實數後，儲存到 (d) 中指定的元件中。	891頁 EREXP (P)
EREXPP		

■單精度實數符號反轉

指令符號	處理內容	參閱目標
ENEG	對單精度實數資料的符號進行反轉。 $\overline{(d)+1, (d)} \longrightarrow (d)+1, (d)$ <p>(1): 實數</p>	893頁 ENEG (P)
ENEGP		

■雙精度實數符號反轉

指令符號	處理內容	參閱目標
EDNEG	對雙精度實數資料的符號進行反轉。	894頁 EDNEG (P)
EDNEGP	$\overbrace{(d)+3, (d)+2, (d)+1, (d)}^{(1)} \longrightarrow (d)+3, (d)+2, (d)+1, (d)$ <p>(1): 實數</p>	

■單精度實數資料傳送

指令符號	處理內容	參閱目標
EMOV	將單精度實數資料傳送至指定的元件中。	895頁 EMOV (P)
EMOVP	$\overbrace{(s)+1, (s)}^{(1)} \longrightarrow (d)+1, (d)$ <p>(1): 實數</p>	

■雙精度實數資料傳送

指令符號	處理內容	參閱目標
EDMOV	將雙精度實數資料傳送至指定的元件中。	896頁 EDMOV (P)
EDMOVP	$\overbrace{(s)+3, (s)+2, (s)+1, (s)}^{(1)} \longrightarrow (d)+3, (d)+2, (d)+1, (d)$ <p>(1): 實數</p>	

■單精度實數SIN運算

指令符號	處理內容	參閱目標
SIN	對單精度實數中指定角度的SIN(正弦)值進行運算。	897頁 SIN (P)
SINP		

■單精度實數COS運算

指令符號	處理內容	參閱目標
COS	對單精度實數中指定角度的COS(餘弦)值進行運算。	899頁 COS (P)
COSP		

■單精度實數TAN運算

指令符號	處理內容	參閱目標
TAN	對單精度實數中指定角度的TAN(正切)值進行運算。	901頁 TAN (P)
TANP		

■單精度實數 SIN^{-1} 運算

指令符號	處理內容	參閱目標
ASIN	透過單精度實數中指定的SIN值進行角度運算。	903頁 ASIN (P)
ASINP		

■單精度實數 COS^{-1} 運算

指令符號	處理內容	參閱目標
ACOS	透過單精度實數中指定的COS值進行角度運算。	905頁 ACOS (P)
ACOSP		

■單精度實數 TAN^{-1} 運算

指令符號	處理內容	參閱目標
ATAN	透過單精度實數中指定的TAN值進行角度運算。	907頁 ATAN (P)
ATANP		

■雙精度實數SIN運算

指令符號	處理內容	參閱目標
SIND	對雙精度實數中指定的角度的SIN(正弦)值進行運算。	909頁 SIND (P)
SINDP		

■雙精度實數COS運算

指令符號	處理內容	參閱目標
COSD	對雙精度實數中指定的角度的COS(餘弦)值進行運算。	911頁 COSD(P)
COSDP		

■雙精度實數TAN運算

指令符號	處理內容	參閱目標
TAND	對雙精度實數中指定的角度的TAN(正切)值進行運算。	913頁 TAND(P)
TANDP		

■雙精度實數 SIN^{-1} 運算

指令符號	處理內容	參閱目標
ASIND	透過雙精度實數中指定的SIN值進行角度運算。	915頁 ASIND(P)
ASINDP		

■雙精度實數 COS^{-1} 運算

指令符號	處理內容	參閱目標
ACOSD	透過雙精度實數中指定的COS值進行角度運算。	917頁 ACOSD(P)
ACOSDP		

■雙精度實數 TAN^{-1} 運算

指令符號	處理內容	參閱目標
ATAND	透過雙精度實數中指定的TAN值進行角度運算。	919頁 ATAND(P)
ATANDP		

■BCD型SIN運算

指令符號	處理內容	參閱目標
BSIN	對以BCD值指定的角度的SIN(正弦)值進行運算。	921頁 BSIN(P)
BSINP		

$\text{SIN}(s) \longrightarrow \begin{array}{|c|} \hline (d) \\ \hline (d)+1 \\ \hline (d)+2 \\ \hline \end{array}$

(d): 符號
 (d)+1: 整數部分
 (d)+2: 小數部分

■BCD型COS運算

指令符號	處理內容	參閱目標
BCOS	對以BCD值指定的角度的COS(餘弦)值進行運算。	923頁 BCOS(P)
BCOSP		

$\text{COS}(s) \longrightarrow \begin{array}{|c|} \hline (d) \\ \hline (d)+1 \\ \hline (d)+2 \\ \hline \end{array}$

(d): 符號
 (d)+1: 整數部分
 (d)+2: 小數部分

■BCD型TAN運算

指令符號	處理內容	參閱目標
BTAN	對以BCD值指定的角度的TAN(正切)值進行運算。	925頁 BTAN(P)
BTANP		

$\text{TAN}(s) \longrightarrow \begin{array}{|c|} \hline (d) \\ \hline (d)+1 \\ \hline (d)+2 \\ \hline \end{array}$

(d): 符號
 (d)+1: 整數部分
 (d)+2: 小數部分

■BCD型 SIN^{-1} 運算

指令符號	處理內容	參閱目標		
BASIN	對以BCD值指定的值的 SIN^{-1} (反正弦)值進行運算。	927頁 BASIN (P)		
BASINP	$\text{SIN}^{-1}(s) \longrightarrow$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>(d)</td></tr> <tr><td>(d)+1</td></tr> <tr><td>(d)+2</td></tr> </table> (d): 符號 (d)+1: 整數部分 (d)+2: 小數部分		(d)	(d)+1
(d)				
(d)+1				
(d)+2				

■BCD型 COS^{-1} 運算

指令符號	處理內容	參閱目標		
BACOS	對以BCD值指定的值的 COS^{-1} (反餘弦)值進行運算。	929頁 BACOS (P)		
BACOSP	$\text{COS}^{-1}(s) \longrightarrow$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>(d)</td></tr> <tr><td>(d)+1</td></tr> <tr><td>(d)+2</td></tr> </table> (d): 符號 (d)+1: 整數部分 (d)+2: 小數部分		(d)	(d)+1
(d)				
(d)+1				
(d)+2				

■BCD型 TAN^{-1} 運算

指令符號	處理內容	參閱目標		
BATAN	對以BCD值指定的值的 TAN^{-1} (反正切)值進行運算。	931頁 BATAN (P)		
BATAMP	$\text{TAN}^{-1}(s) \longrightarrow$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>(d)</td></tr> <tr><td>(d)+1</td></tr> <tr><td>(d)+2</td></tr> </table> (d): 符號 (d)+1: 整數部分 (d)+2: 小數部分		(d)	(d)+1
(d)				
(d)+1				
(d)+2				

■單精度實數角度→弧度轉換

指令符號	處理內容	參閱目標
RAD	將角度的大小單位從單精度實數中指定的度單位轉換為弧度單位。	933頁 RAD (P)
RADP	度→弧度轉換 $(s)+1, (s) \longrightarrow (d)+1, (d)$	

■單精度實數弧度→角度轉換

指令符號	處理內容	參閱目標
DEG	將角度的大小單位從單精度實數中指定的弧度單位轉換為度單位。	935頁 DEG (P)
DEGP	弧度→度轉換 $(s)+1, (s) \longrightarrow (d)+1, (d)$	

■雙精度實數角度→弧度轉換

指令符號	處理內容	參閱目標
RADD	將角度的大小單位從雙精度實數中指定的度單位轉換為弧度單位。	937頁 RADD (P)
RADDP	度→弧度轉換 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)+3, (d)+2, (d)+1, (d)$	

■雙精度實數弧度→角度轉換

指令符號	處理內容	參閱目標
DEGD	將角度的大小單位從雙精度實數中指定的弧度單位轉換為度單位。	939頁 DEGD (P)
DEGDP	弧度→度轉換 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)+3, (d)+2, (d)+1, (d)$	

■單精度實數平方根

指令符號	處理內容	參閱目標
ESQRT	對單精度實數中指定的值的平方根進行運算。	941頁 ESQRT (P)
ESQRTP	$\sqrt{(s)+1, (s)}$ \longrightarrow (d)+1, (d)	

■雙精度實數平方根

指令符號	處理內容	參閱目標
EDSQRT	對雙精度實數中指定的值的平方根進行運算。	943頁 EDSQRT (P)
EDSQRTP	$\sqrt{(s)+3, (s)+2, (s)+1, (s)}$ \longrightarrow (d)+3, (d)+2, (d)+1, (d)	

■單精度實數指數運算

指令符號	處理內容	參閱目標
EXP	對單精度實數中指定的值的指數進行運算。	945頁 EXP (P)
EXPP		

■雙精度實數指數運算

指令符號	處理內容	參閱目標
EXPD	對雙精度實數中指定的值的指數進行運算。	947頁 EXPD (P)
EXPDP		

■單精度實數自然對數運算

指令符號	處理內容	參閱目標
LOG	對以單精度實數中指定的值的自然對數(e)為底時的對數進行運算。	949頁 LOG (P)
LOGP		

■雙精度實數自然對數運算

指令符號	處理內容	參閱目標
LOGD	對以雙精度實數中指定的值的自然對數(e)為底時的對數進行運算。	951頁 LOGD (P)
LOGDP		

■BCD4位/8位平方根

指令符號	處理內容	參閱目標	
BSQRT	對以BCD值(4位)指定的值的平方根進行運算。	953頁 BSQRT (P)	
BSQRTP	$\sqrt{(s)}$ \longrightarrow <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>(d)</td></tr><tr><td>(d)+1</td></tr></table> (d): 整數部分 (d)+1: 小數部分		(d)
(d)			
(d)+1			
BDSQRT	對以BCD值(8位)指定的值的平方根進行運算。	955頁 BDSQRT (P)	
BDSQRTP	$\sqrt{(s)+1, (s)}$ \longrightarrow <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>(d)</td></tr><tr><td>(d)+1</td></tr></table> (d): 整數部分 (d)+1: 小數部分		(d)
(d)			
(d)+1			

■單精度實數冪運算

指令符號	處理內容	參閱目標
POW	對單精度實數的冪進行運算。	957頁 POW (P)
POWP		

■雙精度實數冪運算

指令符號	處理內容	參閱目標
POWD	對雙精度實數的冪進行運算。	959頁 POWD (P)
POWDP		

■單精度實數常用對數運算

指令符號	處理內容	參閱目標
LOG10	對單精度實數中指定的值的常用對數(以10為底的對數)進行運算。	961頁 LOG10(P)
LOG10P		

■雙精度實數常用對數運算

指令符號	處理內容	參閱目標
LOG10D	對雙精度實數中指定的值的常用對數(以10為底的對數)進行運算。	963頁 LOG10D(P)
LOG10DP		

■單精度實數最大值搜尋

指令符號	處理內容	參閱目標
EMAX	將從(s)開始的(n)點中指定的單精度實數的塊資料的最大值儲存到(d)中指定的搜尋結果(最大值)中。	965頁 EMAX(P)
EMAXP		

■雙精度實數最大值搜尋

指令符號	處理內容	參閱目標
EDMAX	將從(s)開始的(n)點中指定的雙精度實數的塊資料的最大值儲存到(d)中指定的搜尋結果(最大值)中。	967頁 EDMAX(P)
EDMAXP		

■單精度實數最小值搜尋

指令符號	處理內容	參閱目標
EMIN	將從(s)開始的(n)點中指定的單精度實數的塊資料的最小值儲存到(d)中指定的搜尋結果(最小值)中。	969頁 EMIN(P)
EMINP		

■雙精度實數最小值搜尋

指令符號	處理內容	參閱目標
EDMIN	將從(s)開始的(n)點中指定的雙精度實數的塊資料的最小值儲存到(d)中指定的搜尋結果(最小值)中。	971頁 EDMIN(P)
EDMINP		

隨機數

隨機數指令

■隨機數發生、系列更改

指令符號	處理內容	參閱目標
RND	發生0~32767(不含32767)的隨機數, 儲存到(d)中指定的元件中。	973頁 RND(P)
RNDP		
SRND	根據(s)中指定的元件中儲存的16位元BIN資料的內容, 更改隨機數系列。	974頁 SRND(P)
SRNDP		

元件操作

變址寄存器指令

■變址寄存器的批量儲存、恢復

指令符號	處理內容	參閱目標
ZPUSH	將變址寄存器的內容儲存到 (d) 中指定的區域中。	975頁 ZPUSH (P)
ZPUSHP		
ZPOP	將 (d) 中指定的區域中儲存的資料讀取到變址寄存器中。	977頁 ZPOP (P)
ZPOPP		

■變址寄存器/超長變址寄存器的選擇儲存、恢復

指令符號	處理內容	參閱目標
ZPUSH	將 (s) 中指定的變址寄存器、超長變址寄存器的內容儲存到 (d) 中指定的區域中。	978頁 ZPUSH (P)
ZPUSHP		
ZPOP	將 (d) 中指定的區域中儲存的資料讀取到變址寄存器、超長變址寄存器中。	981頁 ZPOP (P)
ZPOPP		

檔案寄存器操作指令

■檔案寄存器的塊No. 切換

指令符號	處理內容	參閱目標
RSET	將程式中使用的檔案寄存器的塊No. 更改為 (s) 中指定的元件中儲存的塊No.。	983頁 RSET (P)
RSETP		

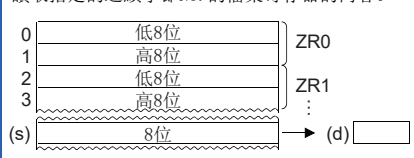
■檔案寄存器用檔案設定

指令符號	處理內容	參閱目標
QDRSET	更改程式中使用的檔案寄存器的檔案名。	985頁 QDRSET (P)
QDRSETP		

檔案寄存器的1字節單位的讀取/寫入指令

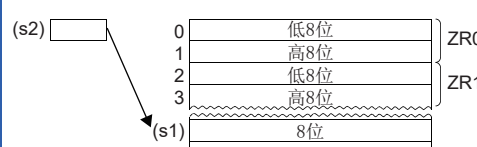
■檔案寄存器的1字節資料讀取

指令符號	處理內容	參閱目標
ZRRDB	讀取指定的連續字節No. 的檔案寄存器的內容。	987頁 ZRRDB (P)
ZRRDBP		



■檔案寄存器的1字節資料寫入

指令符號	處理內容	參閱目標
ZRWRB	將指定的元件的低位的位元的內容寫入到連續字節No. 的檔案寄存器中。	989頁 ZRWRB (P)
ZRWRBP		



間接地址讀取指令

■間接地址讀取

指令符號	處理內容	參閱目標
ADRSET	讀取指定的元件的間接地址。	991頁 ADRSET (P)
ADRSETP	<p>(1): 指定元件的間接地址 (2): 元件名</p>	

定時器、計數器

特殊計數器指令

■單相輸入升值/降值計數器

指令符號	處理內容	參閱目標
UDCNT1	更新指定計數器的當前值。 <p>(1): Cn當前值 (2): Cn觸點</p>	993頁 UDCNT1

■2相輸入升值/降值計數器

指令符號	處理內容	參閱目標
UDCNT2	根據指定A相脈衝及B相脈衝的狀態，更新計數器的當前值。 <p>(1): Cn當前值 (2): Cn觸點</p>	996頁 UDCNT2

特殊定時器指令

■示教定時器

指令符號	處理內容	參閱目標
TTMR	將計測指令ON中的時間以秒為單位進行測定後，儲存與乘數相乘後的值。 <p>(s)=0:1, (s)=1:10, (s)=2:100 T_{ON}: TTMR的ON時間</p>	998頁 TTMR

■特殊功能定時器

指令符號	處理內容	參閱目標
STMR	根據STMR指令的輸入條件的ON/OFF，(d)中指定的位元元件開始的4點將進行以下動作。 <ul style="list-style-type: none"> • (d)+0: OFF延遲定時器輸出 • (d)+1: OFF後單次觸發定時器輸出 • (d)+2: ON後單次觸發定時器輸出 • (d)+3: ON延遲+OFF延遲定時器 	1000頁 STMR

脈衝系統指令

■脈衝密度的測定

指令符號	處理內容	參閱目標
SPD	將 (s1) 中指定的元件的脈衝輸入進行 (s2) 中指定的時間計數後，儲存到 (d) 中指定的元件中。	1002頁 SPD

■恆定週期脈衝輸出

指令符號	處理內容	參閱目標
PLSY	將 (s) 中指定的頻率的脈衝以 (n) 中指定的次數輸出到 (d) 中指定的輸出編號 (Y) 中。	1004頁 PLSY

■脈衝寬度調制

指令符號	處理內容	參閱目標
PWM	將 (s1) 中指定的ON時間及 (s2) 中指定的週期的脈衝輸出到 (d) 中指定的輸出編號 (Y) 中。	1006頁 PWM

就近控制

就近控制指令

■旋轉檯的就近控制

指令符號	處理內容	參閱目標
ROTC	在以 (n1) 分割的旋轉檯中，從停止位置開始向 (s)+1 中指定的位置進行就近旋轉。	1008頁 ROTC

斜坡信號

斜坡信號指令

■斜坡信號

指令符號	處理內容	參閱目標
RAMPQ	從 (s1) 中指定的值開始向 (s2) 中指定的值以 (n) 中指定的次數進行移行。 當前值將被儲存到 (d1)+0 中指定的元件中。	1011頁 RAMPQ

矩陣輸入

矩陣輸入指令

■矩陣輸入

指令符號	處理內容	參閱目標
MTR	將從 (s) 中指定的元件開始依次獲取16點×(n) 列的資料，儲存到 (d2) 中指定的元件及其以後。	1014頁 MTR

CPU模組內建資料庫存取功能

資料庫存取指令

■資料庫連接

指令符號	處理內容	參閱目標
DBOPEN	連接 (s) 中指定的資料夾中儲存的資料庫，將資料庫設為允許使用狀態。	1023頁 DBOPEN (P)
DBOPENP		

■資料庫斷開

指令符號	處理內容	參閱目標
DBCLOSE	解除 (s) 中指定的識別編號及資料庫的分配。	1025頁 DBCLOSE (P)
DBCLOSEP		

■資料庫記錄添加

指令符號	處理內容	參閱目標
DBINSERT	在與 (s1) 中設定的識別編號對應的資料庫的 (s2) 中設定的表中添加記錄。	1027頁 DBINSERT (P)
DBINSERTP		

■資料庫記錄更新

指令符號	處理內容	參閱目標
DBUPDATE	在以 (s1) 中指定的識別編號指定的資料庫內的 (s2) 中指定的表中，對符合 (s5) 中指定的條件的所有記錄進行更新。	1034頁 DBUPDATE (P)
DBUPDATEP		

■資料庫記錄搜尋

指令符號	處理內容	參閱目標
DBSELECT	對 (s1) 中與指定的識別編號對應的資料庫的 (s2) 中指定的表的記錄進行搜尋。	1040頁 DBSELECT (P)
DBSELECTP		

■資料庫記錄刪除

指令符號	處理內容	參閱目標
DBDELETE	在與 (s1) 中指定的識別編號對應的資料庫的 (s2) 中指定的表中，將符合 (s3) 中設定的條件的記錄刪除。	1048頁 DBDELETE (P)
DBDELETEP		

■資料庫導入

指令符號	處理內容	參閱目標
DBIMPORT	對 (s) 中指定的路徑中儲存的Unicode文字檔案中設定的資料進行提取、構築資料庫。	1018頁 DBIMPORT (P)
DBIMPORTP		

■資料庫導出

指令符號	處理內容	參閱目標
DBEXPORT	將資料庫中設定的資料寫入 (s) 中指定的路徑中儲存的Unicode文字檔案中。	1021頁 DBEXPORT (P)
DBEXPORTP		

■資料庫事務處理開始

指令符號	處理內容	參閱目標
DBTRANS	對 (s) 中指定的識別編號對應的資料庫中的事務開始進行聲明。	1052頁 DBTRANS (P)
DBTRANSP		

■資料庫提交

指令符號	處理內容	參閱目標
DBCOMMIT	對與 (s) 中指定的識別編號對應的資料庫的相關事務進行確定。	1054頁 DBCOMMIT (P)
DBCOMMITP		

■資料庫回滾

指令符號	處理內容	參閱目標
DBROLBAK	執行 (s) 中指定的識別編號對應的資料庫中的回滾。	1056頁 DBROLBAK (P)
DBROLBAKP		

時鐘

時鐘用指令

■時鐘資料的讀取

指令符號	處理內容	參閱目標
DATERD	從CPU模組的時鐘單元中讀取“年、月、日、時、分、秒、星期”。	1062頁 DATERD (P)
DATERDP		

(d)
(d)+1
(d)+2
(d)+3
(d)+4
(d)+5
(d)+6

(d): 年
(d)+1: 月
(d)+2: 日
(d)+3: 時
(d)+4: 分
(d)+5: 秒
(d)+6: 星期

■時鐘資料的寫入

指令符號	處理內容	參閱目標
DATEWR	將指定的元件及其以後儲存的時鐘資料寫入到CPU模組的時鐘單元中。	1064頁 DATEWR (P)
DATEWRP		

(s)
(s)+1
(s)+2
(s)+3
(s)+4
(s)+5
(s)+6

(s): 年
(s)+1: 月
(s)+2: 日
(s)+3: 時
(s)+4: 分
(s)+5: 秒
(s)+6: 星期

■時鐘資料的加法運算

指令符號	處理內容	參閱目標
DATE+	對時間資料進行加法運算。	1066頁 DATE+ (P)
DATE+P		

(s1)		(s2)	+		(d)
hour		hour		→	hour
minute		minute			minute
second		second			second

■時鐘資料的減法運算

指令符號	處理內容	參閱目標
DATE-	對時間資料進行減法運算。	1068頁 DATE-(P)
DATE-P	$\begin{array}{ c } \hline (s1) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array} - \begin{array}{ c } \hline (s2) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array} \rightarrow \begin{array}{ c } \hline (d) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array}$	

■時間資料的轉換(時分秒→秒)

指令符號	處理內容	參閱目標
TIME2SEC	將時間資料(時、分、秒)轉換為秒資料。	1070頁 TIME2SEC(P)
TIME2SECP	$\begin{array}{ c } \hline (s) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array} \rightarrow \overbrace{\text{second}}^{(d)+1} \quad (d)$	

■時間資料的轉換(秒→時分秒)

指令符號	處理內容	參閱目標
SEC2TIME	將秒資料轉換為時間資料(時、分、秒)。	1072頁 SEC2TIME(P)
SEC2TIMEP	$\overbrace{\text{second}}^{(s)+1} \quad (s) \rightarrow \begin{array}{ c } \hline (d) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array}$	

■日期時間資料的轉換

指令符號	處理內容	參閱目標
DATE2SEC	將日期時間資料(年、月、日、時、分、秒)轉換為秒資料。	1074頁 DATE2SEC(P) (_U)
DATE2SECP	$\begin{array}{ c } \hline (s) \\ \hline \text{year} \\ \hline \text{month} \\ \hline \text{day} \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array} \rightarrow \overbrace{\text{second}}^{(d)+1} \quad (d)$	
DATE2SEC_U		
DATE2SECP_U		
DATE2SECP_U		
SEC2DATE	將秒資料轉換為日期時間資料(年、月、日、時、分、秒、星期)。	1076頁 SEC2DATE(P) (_U)
SEC2DATEP	$\overbrace{\text{second}}^{(s)+1} \quad (s) \rightarrow \begin{array}{ c } \hline (d) \\ \hline \text{year} \\ \hline \text{month} \\ \hline \text{day} \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \text{day of week} \\ \hline \end{array}$	
SEC2DATE_U		
SEC2DATEP_U		
SEC2DATEP_U		

■日期比較

指令符號	處理內容	參閱目標
LDDT=、ANDDT=、ORDT=	進行指定日期資料的比較，或對日期資料與當前日期進行比較。	1078頁 LDDT□、ANDDT□、ORDT□
LDDT<>、ANDDT<>、ORDT<>		
LDDT>、ANDDT>、ORDT>		
LDDT<=、ANDDT<=、ORDT<=		
LDDT<、ANDDT<、ORDT<		
LDDT>=、ANDDT>=、ORDT>=		

■時間比較

指令符號	處理內容	參閱目標
LDTM=、ANDTM=、ORTM=	進行指定時間資料的比較，或對指定的時間資料與當前時間進行比較。	1082頁 LDTM□、ANDTM□、ORTM□
LDTM<>、ANDTM<>、ORTM<>		
LDTM>、ANDTM>、ORTM>		
LDTM<=、ANDTM<=、ORTM<=		
LDTM<、ANDTM<、ORTM<		
LDTM>=、ANDTM>=、ORTM>=		

■時間比較輸出

指令符號	處理內容	參閱目標
TCMP	比較(s1)、(s2)、(s3)中指定的比較時間資料與(s4)中指定的時間資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為ON。	1085頁 TCMP(P)
TCMPP		

■時間區域比較輸出

指令符號	處理內容	參閱目標
TZCP	對於下限值(s1)的時間資料與上限值(s2)的時間資料的區域，以比較時間資料(s3)進行比較，根據比較的結果(下、區域內、上)，(d)、(d)+1、(d)+2其中之一將變為ON。	1087頁 TZCP(P)
TZCPP		

■擴展時鐘資料的讀取

指令符號	處理內容	參閱目標
S.DATERD	從CPU內部的時鐘單元中讀取包含1/1000秒的時鐘資料。	1089頁 S(P).DATERD
SP.DATERD		

(d)
(d)+1
(d)+2
(d)+3
(d)+4
(d)+5
(d)+6
(d)+7

(d): 年
(d)+1: 月
(d)+2: 日
(d)+3: 時
(d)+4: 分
(d)+5: 秒
(d)+6: 星期
(d)+7: 1/1000秒

■擴展時鐘資料的加法運算

指令符號	處理內容	參閱目標
S.DATE+	對時間資料進行加法運算。	1091頁 S(P).DATE+
SP.DATE+		

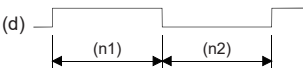
(s1)	+	(s2)	→	(d)
hour		hour		hour
minute		minute		minute
second		second		second
—		—		—
1/1000 second		1/1000 second		1/1000 second

■擴展時鐘資料的減法運算

指令符號	處理內容	參閱目標															
S. DATE-	對時間資料進行減法運算。 (s1) (s2) (d) <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>hour</td></tr> <tr><td>minute</td></tr> <tr><td>second</td></tr> <tr><td>—</td></tr> <tr><td>1/1000 second</td></tr> </table> - <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>hour</td></tr> <tr><td>minute</td></tr> <tr><td>second</td></tr> <tr><td>—</td></tr> <tr><td>1/1000 second</td></tr> </table> → <table border="1" style="display: inline-table;"> <tr><td>hour</td></tr> <tr><td>minute</td></tr> <tr><td>second</td></tr> <tr><td>—</td></tr> <tr><td>1/1000 second</td></tr> </table>	hour	minute	second	—	1/1000 second	hour	minute	second	—	1/1000 second	hour	minute	second	—	1/1000 second	1093頁 S(P).DATE-
hour																	
minute																	
second																	
—																	
1/1000 second																	
hour																	
minute																	
second																	
—																	
1/1000 second																	
hour																	
minute																	
second																	
—																	
1/1000 second																	
SP. DATE-																	

時機計測指令

■發生時機脈衝

指令符號	處理內容	參閱目標
DUTY	將用戶用時機時鐘進行指定掃描的ON、指定掃描的OFF。 (d)  (n1): (n1) 掃描 (n2): (n2) 掃描 (d): SM420~SM424	1095頁 DUTY

■指定資料的時間計測

指令符號	處理內容	參閱目標
TIMCHK	對輸入條件的ON時間進行計測，如果連續ON的時間超過設定的時間，則將(d)中指定的元件設為ON。	1097頁 TIMCHK

■小時針

指令符號	處理內容	參閱目標
HOURM	以1小時為單位計測啟動觸點為ON的時間，在ON時間的累計達到(s)中指定的時間(BIN16位元資料)的時點下，(d2)中指定的元件為ON。	1099頁 HOURM
DHOURM	以1小時為單位計測啟動觸點為ON的時間，在ON時間的累計達到(s)中指定的時間(BIN32位元資料)的時點下，(d2)中指定的元件為ON。	1101頁 DHOURM

模組存取

模組存取指令

■I/O更新

指令符號	處理內容	參閱目標
RFS	在1個掃描的途中進行相應的輸入輸出的部分更新。	1103頁 RFS(P)
RFSP		

■選擇更新

指令符號	處理內容	參閱目標
COM	輸入條件成立時進行各種模組的更新處理及服務處理等。	1105頁 COM(P)
COMP		

■模組更新

指令符號	處理內容	參閱目標
S. ZCOM	進行指定的各種模組的更新處理。	1107頁 S(P).ZCOM
SP. ZCOM		

■從模組中進行1字元/2字元資料讀取(16位元指定)

指令符號	處理內容	參閱目標
FROM	從智能功能模組及其它機號的CPU模組內的緩衝記憶體中，以16位元單位讀取(n)字元的資料。	1109頁 FROM(P)、DFROM(P)
FROMP		
DFROM	從智能功能模組及其它機號的CPU模組內的緩衝記憶體中，以16位元單位讀取(n)×2字元的資料。	
DFROMP		

■至模組的1字元/2字元資料寫入(16位元指定)

指令符號	處理內容	參閱目標
TO	在智能功能模組及本機的CPU模組內的緩衝記憶體中，以16位元單位寫入(n)字元的資料。	1113頁 TO(P)、DT0(P)
TOP		
DT0	在智能功能模組及本機的CPU模組內的緩衝記憶體中，以16位元單位寫入(n)×2字元的資料。	
DTOP		

■從模組中進行1字元/2字元資料讀取(32位元指定)

指令符號	處理內容	參閱目標
FROMD	從智能功能模組及其它機號的CPU模組內的緩衝記憶體中，以32位元單位讀取(n)字元的資料。	1118頁 FROMD(P)、DFROMD(P)
FROMDP		
DFROMD	從智能功能模組及其它機號的CPU模組內的緩衝記憶體中，以32位元單位讀取(n)×2字元的資料。	
DFROMDP		

■至模組的1字元/2字元資料寫入(32位元指定)

指令符號	處理內容	參閱目標
TOD	在智能功能模組及本機的CPU模組內的緩衝記憶體中，以32位元單位寫入(n)字元的資料。	1122頁 TOD(P)、DTOD(P)
TODP		
DTOD	在智能功能模組及本機的CPU模組內的緩衝記憶體中，以32位元單位寫入(n)×2字元的資料。	
DTODP		

■模組型號讀取

指令符號	處理內容	參閱目標
TYPERD	讀取(H)中指定的起始輸入輸出編號的模組型號，儲存到(d)中指定的元件及其以後。	1127頁 TYPERD(P)
TYPERDP		

■模組特有資訊讀取

指令符號	處理內容	參閱目標
UNIINFRD	從(H)中指定的模組中，將(n)中指定的點數的模組資訊儲存到(d)中指定的元件及其以後。	1131頁 UNIINFRD(P)
UNIINFRDP		

參數設定操作

路由資訊指令

■路由資訊的讀取

指令符號	處理內容	參閱目標
S. RTREAD	對路由參數中設定的資料進行讀取。	1136頁 S(P).RTREAD
SP. RTREAD		

■路由資訊的登錄

指令符號	處理內容	參閱目標
S. RTWRITE	將路由資訊登錄到路由參數的指定區域中。	1138頁 S(P).RTWRITE
SP. RTWRITE		

CPU模組資料記錄功能

記錄用指令

■觸發記錄設定/重設

指令符號	處理內容	參閱目標
LOGTRG	發生觸發記錄中觸發條件。將工程工具的觸發記錄設定中設定的記錄數的資料採集結果儲存到資料記錄檔案中。	1140頁 LOGTRG
LOGTRGR	對觸發條件進行重設。	1142頁 LOGTRGR

記錄功能

資料採集指令

■資料採集觸發設定

指令符號	處理內容	參閱目標
DATATRG	在記錄功能中對於指定的設定No. 進行資料採集。	1143頁 DATATRG

內建乙太網路功能用指令

打開/關閉處理指令

■連接的建立

指令符號	處理內容	參閱目標
SP. SOCOOPEN	進行 (s1) 中指定的連接的打開處理。	1145頁 SP. SOCOOPEN

■連接的斷開

指令符號	處理內容	參閱目標
SP. SOCCLOSE	對 (s1) 中指定的連接進行關閉處理。(連接的斷開)	1148頁 SP. SOCCLOSE

套接字通訊用指令

■接收資料的END處理時讀取

指令符號	處理內容	參閱目標
SP. SOCRCV	END處理時將 (s1) 中指定的連接的接收資料從套接字通訊接收資料區域中讀取。	1150頁 SP. SOCRCV

■執行接收資料的指令時讀取

指令符號	處理內容	參閱目標
S. SOCRCVS	執行指令時將 (s) 中指定的連接的接收資料從套接字通訊接收資料區域中讀取。	1153頁 S. SOCRCVS

■資料發送

指令符號	處理內容	參閱目標
SP. SOCSND	將 (s3) 中設定的資料發送至 (s1) 中指定的連接的對象設備中。	1155頁 SP. SOCSND

■連接資訊的讀取

指令符號	處理內容	參閱目標
SP. SOCCINF	讀取 (s1) 中指定的連接的連接資訊。	1158頁 SP. SOCCINF

■連接的通訊目標更改 (UDP/IP)

指令符號	處理內容	參閱目標
SP. SOCCSET	更改 (s1) 中指定的連接的通訊對象IP地址、通訊對象埠編號。 (僅UDP/IP通訊時)	1160頁 SP. SOCCSET

■連接的接收模式更改

指令符號	處理內容	參閱目標
SP. SOCRMODE	針對 (s1) 中指定的連接，更改TCP接收模式及接收資料容量。	1162頁 SP. SOCRMODE

■套接字通訊接收資料讀取

指令符號	處理內容	參閱目標
S. SOCRDATA	從 (s1) 中指定的連接的套接字通訊接收資料區域中，讀取 (n) 中指定的字的資料後，儲存在 (d) 中指定的元件及其以後。	1165頁 S(P). SOCRDATA
SP. SOCRDATA		

通訊協定支援功能指令

■登錄協定執行

指令符號	處理內容	參閱目標
SP. ECPRTCL	執行工程工具的通訊協定支援功能中設定的協定。	1167頁 SP. ECPRTCL

SLMP幀發送指令

■SLMP幀發送

指令符號	處理內容	參閱目標
SP. SLMPSEND	對支援SLMP的設備發送SLMP的報文。	1174頁 SP. SLMPSEND

檔案傳送功能用指令

■發送FTP客戶端檔案

指令符號	處理內容	參閱目標
SP. FTPPUT	發送 (s2) 中指定的CPU模組的檔案至 (s3) 中指定的FTP伺服器的資料夾路徑。	1181頁 SP. FTPPUT

■獲取FTP客戶端檔案

指令符號	處理內容	參閱目標
SP. FTPGET	在 (s3) 中指定的CPU模組的資料夾路徑中獲取 (s2) 中指定的FTP伺服器的檔案。	1186頁 SP. FTPGET

PID運算指令

PID運算

指令符號	處理內容	參閱目標
PID	從(s1)、(s2)、(s3)開始設定的值進行PID運算，所有的採樣時間的運算結果都將儲存至(d)。	1203頁 PID

PID控制指令

PID控制指令(不完全微分)

■PID控制用資料的設定

指令符號	處理內容	參閱目標
S.PIDINIT	將(s)中指定的元件編號及其以後設定的使用環路數的PID控制用資料批量登錄到CPU模組內部，設為允許PID控制狀態。	1216頁 S(P).PIDINIT
SP.PIDINIT		

■PID運算

指令符號	處理內容	參閱目標
S.PIDCONT	進行採樣週期的計測及PID運算。	1219頁 S(P).PIDCONT
SP.PIDCONT		

■指定環路No.的運算停止/開始

指令符號	處理內容	參閱目標
S.PIDSTOP	停止(s)中指定的環路No.的PID運算。	1222頁 S(P).PIDSTOP
SP.PIDSTOP		
S.PIDRUN	開始(s)中指定的環路No.的PID運算。	1223頁 S(P).PIDRUN
SP.PIDRUN		

■指定環路No.的參數更改

指令符號	處理內容	參閱目標
S.PIDPRMW	將(s1)中指定的環路No.的運算參數更改為(s2)中指定的元件編號及其以後儲存的PID控制用資料。	1224頁 S(P).PIDPRMW
SP.PIDPRMW		

PID控制指令(完全微分)

■PID控制用資料的設定

指令符號	處理內容	參閱目標
PIDINIT	將(s)中指定的元件編號及其以後設定的使用環路數的PID控制用資料批量登錄到CPU模組內部，設為允許PID控制狀態。	1228頁 PIDINIT(P)
PIDINITP		

■PID運算

指令符號	處理內容	參閱目標
PIDCONT	進行採樣週期的計測及PID運算。	1230頁 PIDCONT(P)
PIDCONTP		

■指定環路No.的運算停止/開始


指令符號	處理內容	參閱目標
PIDSTOP	停止(s)中指定的環路No.的PID運算。	1233頁 PIDSTOP(P)
PIDSTOPP		
PIDRUN	開始(s)中指定的環路No.的PID運算。	1234頁 PIDRUN(P)
PIDRUNP		

■指定環路No. 的參數更改

指令符號	處理內容	參閱目標
PIDPRMW	將 (s1) 中指定的環路No. 的運算參數更改為 (s2) 中指定的元件編號及其以後儲存的PID控制用資料。	1235頁 PIDPRMW (P)
PIDPRMWP		

過程控制指令

關於過程控制指令的詳細內容，請參閱下述手冊。

 MELSEC iQ-R 程式手冊 (過程控制FB/指令篇)


要點

建立過程控制用程式時，推薦使用過程控制FB。

過程控制FB具有如下所示的特點。

- 透過FB部件的配置和接線可以簡單地建立程式，由於準備了多種進行過程控制所需的FB類型，使得程式編輯變得簡單。
- 因為FB的初始值可以在工程工具的“FB內容”的畫面中設定，因此不需要初始值設定用的程式。
- 無需理會元件的地址，可以透過標籤指定FB的引數。
- 可以透過工程工具的面板等存取標籤資料，確認和調整標籤FB的執行狀態。

有關過程控制FB的詳細內容，請參閱下述手冊。

 MELSEC iQ-R 程式手冊 (過程控制FB/指令篇)

多CPU之間專用指令

從其它機號CPU模組中的元件讀取


指令符號	處理內容	參閱目標
D. DDRD	配置多CPU系時，將(n)中指定的其它機號CPU模組中指定的資料讀取到本機的CPU模組中。	1241頁 D(P). DDRD、 M(P). DDRD
DP. DDRD		
M. DDRD		
MP. DDRD		

至其它機號CPU模組的元件寫入

指令符號	處理內容	參閱目標
D. DDWR	配置多CPU系統時，將本機的CPU模組中指定的資料寫入到(n)中指定的其它機號CPU模組中。	1244頁 D(P). DDWR、 M(P). DDWR
DP. DDWR		
M. DDWR		
MP. DDWR		

運動CPU專用指令

關於可使用的運動CPU專用指令，請參閱下述手冊。

 MELSEC iQ-R Motion Controller Programming Manual (Program Design)

SFC程式用指令

SFC控制指令

■步活性檢查

指令符號	處理內容	參閱目標
LD [S□/BL□\S□]	將指定步的激活/非激活當作運算結果。(常閉觸點指令)	1247頁 LD、LDI、AND、ANI、OR、ORI [S□/BL□\S□]
LDI [S□/BL□\S□]	將指定步的激活/非激活當作運算結果。(常閉觸點指令)	
AND [S□/BL□\S□]	將指定步的激活/非激活、與至當時為止的運算結果之AND運算當作運算結果。(常閉觸點串聯連接指令)	
ANI [S□/BL□\S□]	將指定步的激活/非激活、與至當時為止的運算結果之AND運算當作運算結果。(常閉觸點串聯連接指令)	
OR [S□/BL□\S□]	將指定步的激活/非激活、與至當時為止的運算結果之OR運算當作運算結果。(1個常閉觸點的並聯連接指令)	
ORI [S□/BL□\S□]	將指定步的激活/非激活、與至當時為止的運算結果之OR運算當作運算結果。(1個常閉觸點的並聯連接指令)	

■檢查塊激活

指令符號	處理內容	參閱目標
LD [BL□]	將指定塊的激活/非激活作為運算結果。(常閉觸點指令)	1249頁 LD、LDI、AND、ANI、OR、ORI [BL□]
LDI [BL□]	將指定塊的激活/非激活作為運算結果。(常閉觸點指令)	
AND [BL□]	將指定塊的激活/非激活、與至當時為止的運算結果之AND運算當作運算結果。(常閉觸點串聯連接指令)	
ANI [BL□]	將指定塊的激活/非激活、與至當時為止的運算結果之AND運算當作運算結果。(常閉觸點串聯連接指令)	
OR [BL□]	將指定塊的激活/非激活、與至當時為止的運算結果之OR運算當作運算結果。(1個常閉觸點的並聯連接指令)	
ORI [BL□]	將指定塊的激活/非激活、與至當時為止的運算結果之OR運算當作運算結果。(1個常閉觸點的並聯連接指令)	

■激活步批量讀取

指令符號	處理內容	參閱目標
MOV [K4S□/BL□\K4S□]	指定塊的步激活狀態，作為位元資訊在BIN16位元資料單位中讀取至指定元件。	1251頁 MOV (P) [K4S□/BL□\K4S□]
MOVP [K4S□/BL□\K4S□]		
DMOV [K8S□/BL□\K8S□]	指定塊的步激活狀態，作為位元資訊在BIN32位元資料單位中讀取至指定元件。	1253頁 DMOV (P) [K8S□/BL□\K8S□]
DMOVP [K8S□/BL□\K8S□]		
BMOV [K4S□/BL□\K4S□]	指定塊的步激活狀態，從指定步中透過指定字部分批量讀取。	1256頁 BMOV (P) [K4S□/BL□\K4S□]
BMOVP [K4S□/BL□\K4S□]		

■塊啟動

指令符號	處理內容	參閱目標
SET [BL□]	單獨激活指定塊，從初始步開始執行。	1259頁 SET [BL□]

■結束塊

指令符號	處理內容	參閱目標
RST [BL□]	單獨非激活指定塊。	1261頁 RST [BL□]

■塊停止

指令符號	處理內容	參閱目標
PAUSE [BL□]	指定塊為暫時停止狀態。	1263頁 PAUSE [BL□]

■再次開始塊

指令符號	處理內容	參閱目標
RSTART [BL□]	解除指定塊的暫時停止，再次開始從停止步執行。	1265頁 RSTART [BL□]

■步啟動

指令符號	處理內容	參閱目標
SET [S□/BL□\S□]	激活指定步。	1267頁 SET [S□/BL□\S□]

■步結束

指令符號	處理內容	參閱目標
RST [S□/BL□\S□]	非激活指定步。	1269頁 RST [S□/BL□\S□]

■對象塊切換

指令符號	處理內容	參閱目標
BRSET	指定SFC控制指令的對象塊。	1271頁 BRSET

SFC專用指令

■移轉條件虛擬輸出

指令符號	處理內容	參閱目標
TRAN	使移轉條件成立的虛擬輸出。	1273頁 TRAN

二重化系統用指令

系統切換

指令符號	處理內容	參閱目標
SP.CONTSW	執行了SP.CONTSW指令的掃描的END處理時，切換控制系統和待機系統。	1274頁 SP.CONTSW

系統切換禁止/允許

指令符號	處理內容	參閱目標
DCONTSW	禁止系統切換(用戶切換)。	1278頁 DCONTSW、
ECONTSW	允許系統切換(用戶切換)。	ECONTSW

由待機系統向控制系統寫入資料

指令符號	處理內容	參閱目標
CONTWR	透過雙系統執行程式，由待機系統向控制系統寫入資料。	1280頁 CONTWR(P)
CONTWRP		

安全系統用指令

安全資料一致性檢查資訊讀取

指令符號	處理內容	參閱目標
SP.SIDRD	從指定檔案讀取安全資料一致性檢查用識別元。	1284頁 SP.SIDRD

3 模組專用指令

關於模組專用指令的詳細內容，請參閱下述手冊。

📖 MELSEC iQ-R 程式手冊(模組專用指令篇)

4 通用函數/通用FB

一覽表的閱讀方法如下所示。

項目	內容
函數符號、FB符號	表示函數、FB名。
處理內容	表示函數、FB的概要。
參閱目標	表示詳細說明的參閱目標。

4.1 通用函數

類型轉換函數

■BOOL型→WORD型/DWORD型轉換

函數符號	處理內容	參閱目標
BOOL_TO_WORD	將BOOL型資料轉換為WORD型資料。	1292頁 BOOL_TO_WORD(_E)
BOOL_TO_WORD_E		
BOOL_TO_DWORD	將BOOL型資料轉換為DWORD型資料。	1293頁 BOOL_TO_DWORD(_E)
BOOL_TO_DWORD_E		

■BOOL型→INT型/DINT型轉換

函數符號	處理內容	參閱目標
BOOL_TO_INT	將BOOL型資料轉換為INT型資料。	1294頁 BOOL_TO_INT(_E)
BOOL_TO_INT_E		
BOOL_TO_DINT	將BOOL型資料轉換為DINT型資料。	1295頁 BOOL_TO_DINT(_E)
BOOL_TO_DINT_E		

■BOOL型→TIME型轉換

函數符號	處理內容	參閱目標
BOOL_TO_TIME	將BOOL型資料轉換為TIME型資料。	1296頁 BOOL_TO_TIME(_E)
BOOL_TO_TIME_E		

■BOOL型→STRING型轉換

函數符號	處理內容	參閱目標
BOOL_TO_STRING	將BOOL型資料轉換為STRING型資料。	1297頁 BOOL_TO_STRING(_E)
BOOL_TO_STRING_E		

■WORD型→BOOL型轉換

函數符號	處理內容	參閱目標
WORD_TO_BOOL	將WORD型資料轉換為BOOL型資料。	1298頁 WORD_TO_BOOL(_E)
WORD_TO_BOOL_E		

■WORD型→DWORD型轉換

函數符號	處理內容	參閱目標
WORD_TO_DWORD	將WORD型資料轉換為DWORD型資料。	1299頁 WORD_TO_DWORD(_E)
WORD_TO_DWORD_E		

■WORD型→INT型/DINT型轉換

函數符號	處理內容	參閱目標
WORD_TO_INT	將WORD型資料轉換為INT型資料。	1300頁
WORD_TO_INT_E		WORD_TO_INT(_E)
WORD_TO_DINT	將WORD型資料轉換為DINT型資料。	1301頁
WORD_TO_DINT_E		WORD_TO_DINT(_E)

■WORD型→TIME型轉換

函數符號	處理內容	參閱目標
WORD_TO_TIME	將WORD型資料轉換為TIME型資料。	1303頁
WORD_TO_TIME_E		WORD_TO_TIME(_E)

■WORD型→STRING型轉換

函數符號	處理內容	參閱目標
WORD_TO_STRING	將WORD型資料轉換為STRING型資料。	1304頁
WORD_TO_STRING_E		WORD_TO_STRING(_E)

■DWORD型→BOOL型轉換

函數符號	處理內容	參閱目標
DWORD_TO_BOOL	將DWORD型資料轉換為BOOL型資料。	1305頁
DWORD_TO_BOOL_E		DWORD_TO_BOOL(_E)

■DWORD型→WORD型轉換

函數符號	處理內容	參閱目標
DWORD_TO_WORD	將DWORD型資料轉換為WORD型資料。	1306頁
DWORD_TO_WORD_E		DWORD_TO_WORD(_E)

■DWORD型→INT型/DINT型轉換

函數符號	處理內容	參閱目標
DWORD_TO_INT	將DWORD型資料轉換為INT型資料。	1308頁
DWORD_TO_INT_E		DWORD_TO_INT(_E)
DWORD_TO_DINT	將DWORD型資料轉換為DINT型資料。	1310頁
DWORD_TO_DINT_E		DWORD_TO_DINT(_E)

■DWORD型→TIME型轉換

函數符號	處理內容	參閱目標
DWORD_TO_TIME	將DWORD型資料轉換為TIME型資料。	1311頁
DWORD_TO_TIME_E		DWORD_TO_TIME(_E)

■DWORD型→STRING型轉換

函數符號	處理內容	參閱目標
DWORD_TO_STRING	將DWORD型資料轉換為STRING型資料。	1312頁
DWORD_TO_STRING_E		DWORD_TO_STRING(_E)

■INT型→BOOL型轉換

函數符號	處理內容	參閱目標
INT_TO_BOOL	將INT型資料轉換為BOOL型資料。	1313頁
INT_TO_BOOL_E		INT_TO_BOOL(_E)

■INT型→WORD型/DWORD型轉換

函數符號	處理內容	參閱目標
INT_TO_WORD	將INT型資料轉換為WORD型資料。	1314頁
INT_TO_WORD_E		INT_TO_WORD(_E)
INT_TO_DWORD	將INT型資料轉換為DWORD型資料。	1315頁
INT_TO_DWORD_E		INT_TO_DWORD(_E)

■INT型→DINT型轉換

函數符號	處理內容	參閱目標
INT_TO_DINT	將INT型資料轉換為DINT型資料。	1317頁
INT_TO_DINT_E		INT_TO_DINT(_E)

■INT型→BCD型轉換

函數符號	處理內容	參閱目標
INT_TO_BCD	將INT型資料轉換為BCD型資料。	1318頁
INT_TO_BCD_E		INT_TO_BCD(_E)

■INT型→REAL型/LREAL型轉換

函數符號	處理內容	參閱目標
INT_TO_REAL	將INT型資料轉換為REAL型資料。	1320頁
INT_TO_REAL_E		INT_TO_REAL(_E)
INT_TO_LREAL	將INT型資料轉換為LREAL型資料。	1321頁
INT_TO_LREAL_E		INT_TO_LREAL(_E)

■INT型→TIME型轉換

函數符號	處理內容	參閱目標
INT_TO_TIME	將INT型資料轉換為TIME型資料。	1322頁
INT_TO_TIME_E		INT_TO_TIME(_E)

■INT型→STRING型轉換

函數符號	處理內容	參閱目標
INT_TO_STRING	將INT型資料轉換為STRING型資料。	1323頁
INT_TO_STRING_E		INT_TO_STRING(_E)

■DINT型→BOOL型轉換

函數符號	處理內容	參閱目標
DINT_TO_BOOL	將DINT型資料轉換為BOOL型資料。	1325頁
DINT_TO_BOOL_E		DINT_TO_BOOL(_E)

■DINT型→WORD型/DWORD型轉換

函數符號	處理內容	參閱目標
DINT_TO_WORD	將DINT型資料轉換為WORD型資料。	1326頁
DINT_TO_WORD_E		DINT_TO_WORD(_E)
DINT_TO_DWORD	將DINT型資料轉換為DWORD型資料。	1328頁
DINT_TO_DWORD_E		DINT_TO_DWORD(_E)

■DINT型→INT型轉換

函數符號	處理內容	參閱目標
DINT_TO_INT	將DINT型資料轉換為INT型資料。	1329頁
DINT_TO_INT_E		DINT_TO_INT(_E)

■DINT型→BCD型轉換

函數符號	處理內容	參閱目標
DINT_TO_BCD	將DINT型資料轉換為BCD型資料。	1330頁 DINT_TO_BCD(_E)
DINT_TO_BCD_E		

■DINT型→REAL型/LREAL型轉換

函數符號	處理內容	參閱目標
DINT_TO_REAL	將DINT型資料轉換為REAL型資料。	1332頁 DINT_TO_REAL(_E)
DINT_TO_REAL_E		
DINT_TO_LREAL	將DINT型資料轉換為LREAL型資料。	1333頁 DINT_TO_LREAL(_E)
DINT_TO_LREAL_E		

■DINT型→TIME型轉換

函數符號	處理內容	參閱目標
DINT_TO_TIME	將DINT型資料轉換為TIME型資料。	1334頁 DINT_TO_TIME(_E)
DINT_TO_TIME_E		

■DINT型→STRING型轉換

函數符號	處理內容	參閱目標
DINT_TO_STRING	將DINT型資料轉換為STRING型資料。	1335頁 DINT_TO_STRING(_E)
DINT_TO_STRING_E		

■BCD型→INT型/DINT型轉換

函數符號	處理內容	參閱目標
BCD_TO_INT	將BCD型資料轉換為INT型資料。	1337頁 BCD_TO_INT(_E)
BCD_TO_INT_E		
BCD_TO_DINT	將BCD型資料轉換為DINT型資料。	1339頁 BCD_TO_DINT(_E)
BCD_TO_DINT_E		

■BCD型→STRING型轉換

函數符號	處理內容	參閱目標
BCD_TO_STRING	將BCD型資料轉換為STRING型資料。	1342頁 BCD_TO_STRING(_E)
BCD_TO_STRING_E		

■REAL型→INT型/DINT型轉換

函數符號	處理內容	參閱目標
REAL_TO_INT	將REAL型資料轉換為INT型資料。	1344頁 REAL_TO_INT(_E)
REAL_TO_INT_E		
REAL_TO_DINT	將REAL型資料轉換為DINT型資料。	1346頁 REAL_TO_DINT(_E)
REAL_TO_DINT_E		

■REAL型→LREAL型轉換

函數符號	處理內容	參閱目標
REAL_TO_LREAL	將REAL型資料轉換為LREAL型資料。	1348頁 REAL_TO_LREAL(_E)
REAL_TO_LREAL_E		

■REAL型→STRING型轉換

函數符號	處理內容	參閱目標
REAL_TO_STRING	將REAL型資料轉換為STRING型(指數形式)資料。	1350頁 REAL_TO_STRING(_E)
REAL_TO_STRING_E		

■LREAL型→INT型/DINT型轉換

函數符號	處理內容	參閱目標
LREAL_TO_INT	將LREAL型資料轉換為INT型資料。	1353頁
LREAL_TO_INT_E		LREAL_TO_INT(_E)
LREAL_TO_DINT	將LREAL型資料轉換為DINT型資料。	1354頁
LREAL_TO_DINT_E		LREAL_TO_DINT(_E)

■LREAL型→REAL型轉換

函數符號	處理內容	參閱目標
LREAL_TO_REAL	將LREAL型資料轉換為REAL型資料。	1355頁
LREAL_TO_REAL_E		LREAL_TO_REAL(_E)

■TIME型→BOOL型轉換

函數符號	處理內容	參閱目標
TIME_TO_BOOL	將TIME型資料轉換為BOOL型資料。	1356頁
TIME_TO_BOOL_E		TIME_TO_BOOL(_E)

■TIME型→WORD型/DWORD型轉換

函數符號	處理內容	參閱目標
TIME_TO_WORD	將TIME型資料轉換為WORD型資料。	1357頁
TIME_TO_WORD_E		TIME_TO_WORD(_E)
TIME_TO_DWORD	將TIME型資料轉換為DWORD型資料。	1358頁
TIME_TO_DWORD_E		TIME_TO_DWORD(_E)

■TIME型→INT型/DINT型轉換

函數符號	處理內容	參閱目標
TIME_TO_INT	將TIME型資料轉換為INT型資料。	1359頁
TIME_TO_INT_E		TIME_TO_INT(_E)
TIME_TO_DINT	將TIME型資料轉換為DINT型資料。	1360頁
TIME_TO_DINT_E		TIME_TO_DINT(_E)

■TIME型→STRING型轉換

函數符號	處理內容	參閱目標
TIME_TO_STRING	將TIME型資料轉換為STRING型資料。	1361頁
TIME_TO_STRING_E		TIME_TO_STRING(_E)

■STRING型→BOOL型轉換

函數符號	處理內容	參閱目標
STRING_TO_BOOL	將STRING型資料轉換為BOOL型資料。	1363頁
STRING_TO_BOOL_E		STRING_TO_BOOL(_E)

■STRING型→WORD型/DWORD型轉換

函數符號	處理內容	參閱目標
STRING_TO_WORD	將STRING型資料轉換為WORD型資料。	1364頁
STRING_TO_WORD_E		STRING_TO_WORD(_E)
STRING_TO_DWORD	將STRING型資料轉換為DWORD型資料。	1365頁
STRING_TO_DWORD_E		STRING_TO_DWORD(_E)

■STRING型→INT型/DINT型轉換

函數符號	處理內容	參閱目標
STRING_TO_INT	將STRING型資料轉換為INT型資料。	1366頁
STRING_TO_INT_E		STRING_TO_INT(_E)
STRING_TO_DINT	將STRING型資料轉換為DINT型資料。	1368頁
STRING_TO_DINT_E		STRING_TO_DINT(_E)

■STRING型→BCD型轉換

函數符號	處理內容	參閱目標
STRING_TO_BCD	將STRING型資料轉換為BCD型資料。	1370頁
STRING_TO_BCD_E		STRING_TO_BCD(_E)

■STRING型→REAL型轉換

函數符號	處理內容	參閱目標
STRING_TO_REAL	將STRING型資料轉換為REAL型資料。	1372頁
STRING_TO_REAL_E		STRING_TO_REAL(_E)

■STRING型→TIME型轉換

函數符號	處理內容	參閱目標
STRING_TO_TIME	將STRING型資料轉換為TIME型資料。	1375頁
STRING_TO_TIME_E		STRING_TO_TIME(_E)

■位元數組→INT型/DINT型轉換

函數符號	處理內容	參閱目標
BITARR_TO_INT	將從位元數組指定的位元數轉換為INT型資料。	1376頁
BITARR_TO_INT_E		BITARR_TO_INT(_E)
BITARR_TO_DINT	將從位元數組指定的位元數轉換為DINT型資料。	1377頁
BITARR_TO_DINT_E		BITARR_TO_DINT(_E)

■INT型/DINT型→位元數組轉換

函數符號	處理內容	參閱目標
INT_TO_BITARR	將INT型資料的低n位元輸出到位元數組中。	1378頁
INT_TO_BITARR_E		INT_TO_BITARR(_E)
DINT_TO_BITARR	將DINT型資料的低n位元輸出到位元數組中。	1379頁
DINT_TO_BITARR_E		DINT_TO_BITARR(_E)

■位元數組的複製

函數符號	處理內容	參閱目標
CPY_BITARR	將位元數組進行指定位元容量的複製。	1380頁
CPY_BITARR_E		CPY_BITARR(_E)

■字標籤的指定位元讀取/寫入/複製

函數符號	處理內容	參閱目標
GET_BIT_OF_INT	讀取字標籤的指定位元。	1381頁
GET_BIT_OF_INT_E		GET_BIT_OF_INT(_E)
SET_BIT_OF_INT	對字標籤的指定位元進行寫入。	1383頁
SET_BIT_OF_INT_E		SET_BIT_OF_INT(_E)
CPY_BIT_OF_INT	將字標籤的指定位元複製到其它字標籤的指定位元。	1385頁
CPY_BIT_OF_INT_E		CPY_BIT_OF_INT(_E)

■起始資料的獲取

函數符號	處理內容	參閱目標
GET_BOOL_ADDR	將指定資料的起始資料作為BOOL型、INT型或WORD型資料進行輸出。	1387頁 GET_BOOL_ADDR、 GET_INT_ADDR、 GET_WORD_ADDR
GET_INT_ADDR		
GET_WORD_ADDR		

單數值變數函數

■絕對值

函數符號	處理內容	參閱目標
ABS	輸出輸入值的絕對值。	1388頁 ABS(_E)
ABS_E		

■平方根

函數符號	處理內容	參閱目標
SQRT	運算輸入值的平方根。	1390頁 SQRT(_E)
SQRT_E		

■自然對數運算

函數符號	處理內容	參閱目標
LN	輸出輸入值的自然對數運算結果。	1391頁 LN(_E)
LN_E		

■常用對數運算

函數符號	處理內容	參閱目標
LOG	輸出輸入值的常用對數(以10為底的對數)的運算結果。	1392頁 LOG(_E)
LOG_E		

■指數運算

函數符號	處理內容	參閱目標
EXP	輸出輸入值的指數運算結果。	1394頁 EXP(_E)
EXP_E		

■SIN運算/COS運算/TAN運算

函數符號	處理內容	參閱目標
SIN	輸出輸入值的SIN(正弦)值。	1395頁 SIN(_E)
SIN_E		
COS	輸出輸入值的COS(餘弦)值。	1396頁 COS(_E)
COS_E		
TAN	輸出輸入值的TAN(正切)值。	1397頁 TAN(_E)
TAN_E		

■SIN⁻¹運算/COS⁻¹運算/TAN⁻¹運算

函數符號	處理內容	參閱目標
ASIN	輸出輸入值的SIN ⁻¹ (反正弦)值。	1398頁 ASIN(_E)
ASIN_E		
ACOS	輸出輸入值的COS ⁻¹ (反餘弦)值。	1399頁 ACOS(_E)
ACOS_E		
ATAN	輸出輸入值的TAN ⁻¹ (反正切)值。	1400頁 ATAN(_E)
ATAN_E		

算術運算函數

■加法運算

函數符號	處理內容	參閱目標
ADD	輸出輸入值的和((s1)+(s2)+...+(s28))。	1401頁 ADD(_E)
ADD_E		

■乘法運算

函數符號	處理內容	參閱目標
MUL	輸出輸入值的積((s1)×(s2)×...×(s28))。	1404頁 MUL(_E)
MUL_E		

■減法運算

函數符號	處理內容	參閱目標
SUB	輸出輸入值的差((s1)-(s2))。	1406頁 SUB(_E)
SUB_E		

■除法運算

函數符號	處理內容	參閱目標
DIV	輸出輸入值的商((s1)÷(s2))。	1409頁 DIV(_E)
DIV_E		

■餘數

函數符號	處理內容	參閱目標
MOD	輸出輸入值的餘數((s1)÷(s2))。	1411頁 MOD(_E)
MOD_E		

■冪

函數符號	處理內容	參閱目標
EXPT	輸出輸入值的冪。	1413頁 EXPT(_E)
EXPT_E		

■代入

函數符號	處理內容	參閱目標
MOVE	輸出輸入值的代入。	1414頁 MOVE(_E)
MOVE_E		

位元位移函數

■n位元左移、右移

函數符號	處理內容	參閱目標
SHL	將輸入值左移(n)位元數後輸出。	1416頁 SHL(_E)
SHL_E		
SHR	將輸入值右移(n)位元數後輸出。	1418頁 SHR(_E)
SHR_E		

■n位元左旋轉、右旋轉

函數符號	處理內容	參閱目標
ROL	將輸入值左旋轉(n)位元數後輸出。	1420頁 ROL(_E)
ROL_E		
ROR	將輸入值右旋轉(n)位元數後輸出。	1422頁 ROR(_E)
ROR_E		

位元型布爾函數

■邏輯且、邏輯或、排他邏輯或

函數符號	處理內容	參閱目標
AND	輸出輸入值的邏輯且。	1424頁 AND(_E)、 OR(_E)、XOR(_E)
AND_E		
OR		
OR_E	輸出輸入值的邏輯或。	
XOR	輸出輸入值的排他邏輯或。	
XOR_E		

■邏輯否定

函數符號	處理內容	參閱目標
NOT	輸出輸入值的邏輯否定。	1427頁 NOT(_E)
NOT_E		

選擇函數

■選擇值

函數符號	處理內容	參閱目標
SEL	輸出選擇的輸入值。	1428頁 SEL(_E)
SEL_E		

■最大值、最小值選擇

函數符號	處理內容	參閱目標
MAX	輸出輸入值的最大值。	1430頁 MAX(_E)、 MIN(_E)
MAX_E		
MIN	輸出輸入值的最小值。	
MIN_E		

■上下限限位控制

函數符號	處理內容	參閱目標
LIMIT	輸出上下限限位控制的輸入值。	1432頁 LIMIT(_E)
LIMIT_E		

■多路復用器

函數符號	處理內容	參閱目標
MUX	輸出多個輸入值的一個。	1435頁 MUX(_E)
MUX_E		

比較函數

■比較

函數符號	處理內容	參閱目標
GT	輸出輸入值的資料比較結果。	1437頁 GT(_E)、 GE(_E)、EQ(_E)、 LE(_E)、LT(_E)
GT_E		
GE		
GE_E		
EQ		
EQ_E		
LE		
LE_E		
LT		
LT_E		
NE		
NE_E		

字元串函數

■字元串的長度檢測

函數符號	處理內容	參閱目標
LEN	檢測並輸出輸入的字元串的長度。	1441頁 LEN(_E)
LEN_E		

■從字元串的左側、右側提取

函數符號	處理內容	參閱目標
LEFT	從輸入的字元串資料的左側開始輸出指定字元。	1443頁 LEFT(_E)、 RIGHT(_E)
LEFT_E		
RIGHT	從輸入的字元串資料的右側開始輸出指定字元。	
RIGHT_E		

■字元串的提取

函數符號	處理內容	參閱目標
MID	從輸入的字元串的任意位置開始輸出指定字元。	1445頁 MID(_E)
MID_E		

■字元串的合併

函數符號	處理內容	參閱目標
CONCAT	合併字元串後輸出。	1447頁 CONCAT(_E)
CONCAT_E		

■字元串的插入

函數符號	處理內容	參閱目標
INSERT	在字元串之間插入字元串後輸出。	1449頁 INSERT(_E)
INSERT_E		

■字元串的刪除

函數符號	處理內容	參閱目標
DELETE	刪除字元串的任意範圍後輸出。	1451頁 DELETE(_E)
DELETE_E		

■字元串的替換

函數符號	處理內容	參閱目標
REPLACE	替換字元串的任意範圍後輸出。	1453頁 REPLACE(_E)
REPLACE_E		

■字元串的搜尋

函數符號	處理內容	參閱目標
FIND	搜尋字元串並輸出搜尋結果。	1456頁 FIND(_E)
FIND_E		

時間資料類型函數

■加法運算

函數符號	處理內容	參閱目標
ADD_TIME	輸出輸入值 (TIME型) 的和 ((s1)+(s2))。	1458頁 ADD_TIME(_E)
ADD_TIME_E		

■減法運算

函數符號	處理內容	參閱目標
SUB_TIME	輸出輸入值 (TIME型) 的差 ((s1)-(s2))。	1460頁 SUB_TIME(_E)
SUB_TIME_E		

■乘法運算

函數符號	處理內容	參閱目標
MUL_TIME	輸出輸入值 (TIME型) 的積 ((s1)×(s2))。	1462頁 MUL_TIME(_E)
MUL_TIME_E		

■除法運算

函數符號	處理內容	參閱目標
DIV_TIME	輸出輸入值 (TIME型) 的商 ((s1)÷(s2))。	1464頁 DIV_TIME(_E)
DIV_TIME_E		

4.2 通用FB

雙穩態FB

■雙穩態FB(設定優先)

FB符號	處理內容	參閱目標
SR	判斷2個輸入值，輸出1(TRUE)或者0(FALSE)。	1468頁 SR(_E)
SR_E		

■雙穩態FB(重設優先)

FB符號	處理內容	參閱目標
RS	判斷2個輸入值，輸出1(TRUE)或者0(FALSE)。	1470頁 RS(_E)
RS_E		

邊緣檢測FB

■上升沿邊緣檢測

FB符號	處理內容	參閱目標
R_TRIG	檢測信號的上升沿並輸出脈衝信號。	1472頁 R_TRIG(_E)
R_TRIG_E		

■下降沿檢測

FB符號	處理內容	參閱目標
F_TRIG	檢測信號的下降沿並輸出脈衝信號。	1474頁 F_TRIG(_E)
F_TRIG_E		

計數器FB

■升值計數器

FB符號	處理內容	參閱目標
CTU	對信號的上升沿次數進行遞增計數。	1476頁 CTU(_E)
CTU_E		

■降值計數器

FB符號	處理內容	參閱目標
CTD	對信號的上升沿次數進行遞減計數。	1478頁 CTD(_E)
CTD_E		

■升值降值計數器

FB符號	處理內容	參閱目標
CTUD	對信號的上升沿次數進行遞增/遞減計數。	1480頁 CTUD(_E)
CTUD_E		

■計數器FB

FB符號	處理內容	參閱目標
COUNTER_FB_M	執行條件成立時，執行遞增計數。	1483頁 COUNTER_FB_M

定時器FB

■脈衝定時器

FB符號	處理內容	參閱目標
TP	在指定時間期間將信號設為ON。	1485頁 TP(_E)
TP_E		

■ON延遲定時器

FB符號	處理內容	參閱目標
TON	在指定的時間後將信號設為ON。	1487頁 TON(_E)
TON_E		

■OFF延遲定時器

FB符號	處理內容	參閱目標
TOF	在指定的時間後將信號設為OFF。	1489頁 TOF(_E)
TOF_E		

■定時器FB

FB符號	處理內容	參閱目標
TIMER_10_FB_M	執行條件成立後，至設定的時間為止執行定時器計數。	1491頁 TIMER_□_M
TIMER_100_FB_M		
TIMER_HIGH_FB_M		
TIMER_LOW_FB_M		
TIMER_CONT_FB_M		
TIMER_CONTHFB_M		

第3部分 順控程式指令

本部分由下述章節構成。

5 順控程式指令

5 順控程式指令

5.1 觸點指令

運算開始、串聯連接、並聯連接

LD、LDI、AND、ANI、OR、ORI

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

- LD：常開觸點運算開始指令/LDI：常閉觸點運算開始指令
將指定元件的ON/OFF資訊作為運算結果。
- AND：常開觸點串聯連接指令/ANI：常閉觸點串聯連接指令
將指定元件的ON/OFF資訊與至當時為止的運算結果的AND運算作為運算結果。
- OR：1個常開觸點的並聯連接指令/ORI：1個常閉觸點的並聯連接指令
將指定元件的ON/OFF資訊與至當時為止的運算結果的OR運算作為運算結果。

梯形圖	ST
	不對應。

FBD/LD
不對應。

■執行條件

指令	執行條件
LD	常時執行
LDI	
AND	
ANI	
OR	
ORI	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	作為觸點使用的元件	—	位元	ANY_BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它(DX)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	○	○	○	—	○	—	○	—	—	○

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	—

5

功能

■LD、LDI

- LD指令透過常開觸點運算開始指令、LDI指令透過常閉觸點運算開始指令來獲取指定元件的ON/OFF資訊*1，將該值作為運算結果。

*1 字元元件的位元指定時，根據指定位元的1/0而ON/OFF。

■AND、ANI

- AND指令透過常開觸點串聯連接指令、ANI指令透過常閉觸點串聯連接指令來獲取指定位元元件的ON/OFF資訊*1，與至當時為止的運算結果進行AND運算，將該值作為運算結果。

*1 字元元件的位元指定時，根據指定位元的1/0而ON/OFF。

- 工程工具的梯形圖編輯模式內容如下所示。

- 寫入：AND指令、ANI指令在串聯連接的情況下，最多可以建立24級的梯形圖。
- 讀取：AND指令、ANI指令在串聯連接的情況下，最多可以顯示24級的梯形圖。超過了24級的情況下最多顯示24級。

■OR、ORI

- OR指令透過1個常開觸點的並聯連接指令、ORI指令透過1個常閉觸點的並聯連接指令來獲取指定元件的ON/OFF資訊*1，與至當時為止的運算結果進行OR運算，將該值作為運算結果。

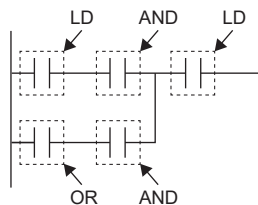
*1 字元元件的位元指定時，根據指定位元的1/0而ON/OFF。

- 工程工具的梯形圖編輯模式內容如下所示。

- 寫入：OR指令、ORI指令最多可以建立23個連續連接的梯形圖。
- 讀取：OR指令、ORI指令最多可以顯示23個連續連接的梯形圖。超過了23個的梯形圖將無法正常顯示。

■透過組合LD、LDI、AND、ANI、OR、ORI的動作

透過組合LD、AND、OR的動作示例如下所示。LDI、ANI、ORI也相同。



要點

- 字元元件的位元指定時，位元的指定是以16進制數進行。(例如，D0的b11將變為“D0.0B”。)

出錯

沒有運算出錯。

脈衝運算開始、脈衝串聯連接、脈衝並聯連接

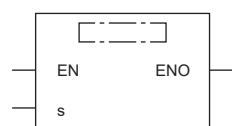
LDP、LDF、ANDP、ANDF、ORP、ORF

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

- LDP: 上升沿脈衝運算開始指令
僅在指定位元元件的上升沿時 (OFF→ON) 導通。
- LDF: 下降沿脈衝運算開始指令
僅在指定位元元件的下降沿時 (ON→OFF) 導通。
- ANDP: 上升沿脈衝串聯連接指令/ANDF: 下降沿脈衝串聯連接指令
與至當時為止的運算結果進行AND運算。
- ORP: 上升沿脈衝並聯連接指令/ORF: 下降沿脈衝並聯連接指令
與至當時為止的運算結果進行OR運算。

梯形圖	ST
	<pre> ENO:=LDP (EN, s); ENO:=LDF (EN, s); ENO:=ANDP (EN, s); ENO:=ANDF (EN, s); ENO:=ORP (EN, s); ENO:=ORF (EN, s); </pre>

FBD/LD



■執行條件

指令	執行條件
LDP LDF ANDP ANDF ORP ORF	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	作為觸點使用的元件	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其它(DX)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	○	○	○	—	○	—	○	—	—	—	○	

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

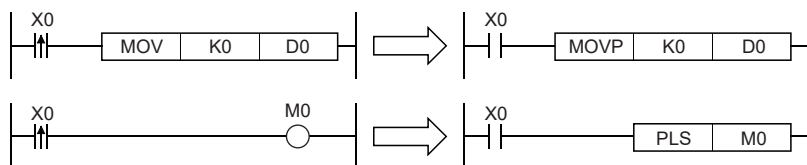
操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	—

功能

■LDP、LDF

- LDP指令透過上升沿脈衝運算開始指令，僅在指定位元元件的上升沿時(OFF→ON)導通。字元元件的位元指定時，僅在指定位元由0→1變化時導通。僅LDP指令的情況下，與ON中執行指令的脈衝化指令(□P)相同。

將已使用LDP指令的梯形圖替換為未使用LDP指令的梯形圖時，其情況如下所示。



- LDF指令透過下降沿脈衝運算開始指令，在指定位元元件的下降沿時(ON→OFF)導通。字元元件的位元指定時，指定位元由1→0變化時導通。
- 透過ST及FBD/LD使用LDP指令的情況下，指定位元元件(s)的上升沿時(OFF→ON)，ENO將ON。
- 透過ST及FBD/LD使用LDF指令的情況下，指定位元元件(s)的下降沿時(ON→OFF)，ENO將ON。
- 透過ST使用LDP、LDF指令時，應常時在EN中指定ON。
- 透過FBD/LD使用LDP、LDF指令時，應在EN中指定左母線或常時ON的變數部件/常數部件。

■ANDP、ANDF

- ANDP指令是上升沿脈衝串聯連接指令，ANDF指令是下降沿脈衝串聯連接指令，與至當時為止的運算結果進行AND運算並作為運算結果。ANDP指令、ANDF指令中使用的ON/OFF資訊如下所示。

ANDP、ANDF中指定的元件		ANDP的狀態	ANDF的狀態
位元元件	字元元件的位元指定		
OFF→ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON→OFF	1→0	OFF	ON

- 將ANDP指令透過ST及FBD/LD使用的情況下，EN與指定位元元件(s)上升沿的AND運算結果為ON時，ENO將ON。EN不成為執行條件。
- 將ANDF指令透過ST及FBD/LD使用的情況下，EN與指定位元元件(s)下降沿的AND運算結果為ON時，ENO將ON。EN不成為執行條件。

■ORP、ORF

- ORP指令是上升沿脈衝並聯連接指令、ORF指令是下降沿脈衝並聯連接指令，與至當時為止的運算結果進行OR運算並作為運算結果。ORP指令、ORF指令中使用的ON/OFF資訊如下所示。

ORP、ORF中指定的元件		ORP的狀態	ORF的狀態
位元元件	字元元件的位元指定		
OFF→ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON→OFF	1→0	OFF	ON

- 將ORP指令透過ST及FBD/LD使用的情況下，EN與指定位元元件(s)上升沿的OR運算結果為ON時，ENO將ON。EN不成為執行條件。
- 將ORF指令透過ST及FBD/LD使用的情況下，EN與指定位元元件(s)下降沿的OR運算結果為ON時，ENO將ON。EN不成為執行條件。

■透過組合LDP、LDF、ANDP、ANDF、ORP、ORF的動作

透過組合LDP、LDF、ANDP、ANDF、ORP、ORF的動作的示例與透過組合LD、AND、OR的動作相同。(☞ 148頁 LD、LDI、AND、ANI、OR、ORI)

出錯

沒有運算出錯。

脈衝否定運算開始、脈衝否定串聯連接、脈衝否定並聯連接

LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI

RnCPU RnENCPU RnPCPU (通用) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

- LDPI: 上升沿脈衝否定運算開始指令
在指定元件的OFF時、ON時、下降沿時 (ON→OFF) 的情況下導通。
- LDFI: 下降沿脈衝否定運算開始指令
在指定元件的上升沿時 (OFF→ON)、OFF時、ON時導通。
- ANDPI: 上升沿脈衝否定串聯連接指令/ANDFI: 下降沿脈衝否定串聯連接指令
與至當時為止的運算結果進行AND運算。
- ORPI: 上升沿脈衝否定並聯連接指令/ORFI: 下降沿脈衝否定並聯連接指令
與至當時為止的運算結果進行OR運算。

梯形圖	ST
<p>The diagram shows six instruction types in a ladder logic format:</p> <ul style="list-style-type: none"> LDPI: A normally open contact with a pulse symbol (upward arrow) and a dashed box labeled (s). LDFI: A normally closed contact with a pulse symbol (downward arrow) and a dashed box labeled (s). ANDPI: A normally open contact followed by a normally open contact with a pulse symbol (upward arrow) and a dashed box labeled (s). ANDFI: A normally open contact followed by a normally closed contact with a pulse symbol (downward arrow) and a dashed box labeled (s). ORPI: A normally open contact with a pulse symbol (upward arrow) and a dashed box labeled (s) in parallel with another normally open contact. ORFI: A normally open contact with a pulse symbol (downward arrow) and a dashed box labeled (s) in parallel with another normally open contact. 	<pre> ENO:=LDPI (EN, s); ENO:=LDFI (EN, s); ENO:=ANDPI (EN, s); ENO:=ANDFI (EN, s); ENO:=ORPI (EN, s); ENO:=ORFI (EN, s); </pre>

FBD/LD
<p>The FBD shows a rectangular block with an input terminal 'EN' on the left and an output terminal 'ENO' on the right. A parameter 's' is located below the 'EN' terminal.</p>

■執行條件

指令	執行條件
LDPI LDFI ANDPI ANDFI ORPI ORFI	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	作為觸點使用的元件	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其它(DX)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	\$	
(s)	○	○	○	○	—	○	—	○	—	—	—	○

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	—

功能

■LDPI、LDFI

- LDPI指令是上升沿脈衝否定運算開始指令，在指定位元元件的OFF時、ON時、下降沿時(ON→OFF)的情況下導通。字元元件的位元指定時，在指定位元為0的情況下、1的情況下、1→0的變化的情況下導通。
- LDFI指令是下降沿脈衝否定運算開始指令，在指定位元元件的上升沿時(OFF→ON)、OFF時、ON時的情況下導通。字元元件的位元指定時，在指定位元為0的情況下、1的情況下、0→1的變化的情況下導通。LDPI指令、LDFI指令中使用的ON/OFF資訊如下所示。

LDPI、LDFI中指定的元件		LDPI的狀態	LDFI的狀態
位元元件	字元元件的位元指定		
OFF→ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON→OFF	1→0	ON	OFF

- 透過ST及FBD/LD使用LDPI指令的情況下，指定位元元件(s)的上升沿以外時，ENO將ON。
- 透過ST及FBD/LD使用LDFI指令的情況下，指定位元元件(s)的下降沿以外時，ENO將ON。
- 透過ST使用LDPI、LDFI指令時，應當時在EN中指定ON。
- 透過FBD/LD使用LDPI、LDFI指令時，應在EN中指定左母線或常時ON的變數部件/常數部件。

■ANDPI、ANDFI

- ANDPI指令透過上升沿脈衝否定串聯連接指令、ANDFI指令透過下降沿脈衝否定串聯連接指令，與至當時為止的運算結果進行AND運算並將該值作為運算結果。ANDPI指令、ANDFI指令中使用的ON/OFF資訊如下所示。

ANDPI、ANDFI中指定的元件		ANDPI的狀態	ANDFI的狀態
位元元件	字元元件的位元指定		
OFF→ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON→OFF	1→0	ON	OFF

- 透過ST及FBD/LD使用ANDPI指令的情況下，EN與指定位元元件(s)上升沿的AND運算結果為ON以外時，ENO將ON。EN不成為執行條件。
- 透過ST及FBD/LD使用ANDFI指令的情況下，EN與指定位元元件(s)下降沿的AND運算結果為ON以外時，ENO將ON。EN不成為執行條件。

■ORPI、ORFI

- ORPI指令是上升沿脈衝否定並聯連接指令、ORFI指令是下降沿脈衝否定並聯連接指令，與至當時為止的運算結果進行OR運算並作為運算結果。ORPI指令、ORFI指令中使用的ON/OFF資訊如下所示。

ORPI、ORFI中指定的元件		ORPI的狀態	ORFI的狀態
位元元件	字元元件的位元指定		
OFF→ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON→OFF	1→0	ON	OFF

- 透過ST及FBD/LD使用ORPI指令的情況下，EN與指定位元元件(s)上升沿的OR運算結果為ON以外時，ENO將ON。EN不成為執行條件。
- 透過ST及FBD/LD使用ORFI指令的情況下，EN與指定位元元件(s)下降沿的OR運算結果為ON以外時，ENO將ON。EN不成為執行條件。

■透過組合LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI的動作

透過組合LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI的動作的示例與透過組合LD、AND、OR的動作相同。(☞ 148頁 LD、LDI、AND、ANI、OR、ORI)

出錯

沒有運算出錯。

5.2 合併指令

梯形圖塊串聯連接、並聯連接

ANB、ORB

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

- ANB：梯形圖塊串聯連接指令
進行A塊與B塊的AND運算。
- ORB：梯形圖塊並聯連接指令
進行A塊與B塊的OR運算。

梯形圖	ST
<p>A: A塊 B: B塊</p>	不對應。

FBD/LD

不對應。

■執行條件

指令	執行條件
ANB ORB	常時執行

功能

■ANB

- 進行A塊與B塊的AND運算，並作為運算結果。
- ANB指令的符號不是觸點符號，而是連接符號。

■ORB

- 進行A塊與B塊的OR運算，並作為運算結果。
- ORB指令對2觸點及其以上的梯形圖塊進行並聯連接。僅1個觸點的並聯連接使用OR指令、ORI指令，無需ORB指令。
- ORB指令的符號不是觸點符號，而是連接符號。

出錯

沒有運算出錯。

運算結果推入、讀取、彈出

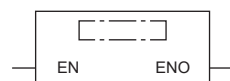
MPS、MRD、MPP

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

- MPS: 運算結果推入指令
儲存MPS指令之前的運算結果 (ON/OFF)。
- MRD: 運算結果讀取指令
讀取透過MPS指令儲存的運算結果。
- MPP: 運算結果彈出指令
獲取透過MPS指令儲存的運算結果。

梯形圖	ST
<p>(梯形圖顯示的情況下，不能顯示MPS、MRD、MPP。)</p>	ENO:=MPS (EN) ; ENO:=MRD (EN) ; ENO:=MPP (EN) ;

FBD/LD



■執行條件

指令	執行條件
MPS	常時執行
MRD	
MPP	

功能

■MPS

- 儲存MPS指令之前的運算結果 (ON/OFF)。
- MPS指令最多可以連續使用16次。中途使用了MPP指令的情況下，MPS指令的使用數將被-1。

■MRD

- 讀取透過MPS指令儲存的運算結果，以該運算結果從下一個步開始進行運算。

■MPP

- 讀取透過MPS指令儲存的運算結果，以該運算結果從下一個步開始進行運算。
- 清除透過MPS指令儲存的運算結果。
- MPS指令的使用數將被-1。

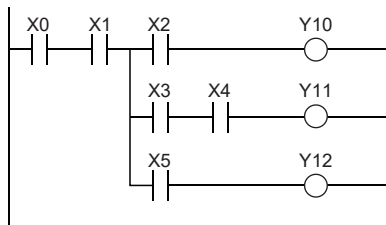
出錯

沒有運算出錯。

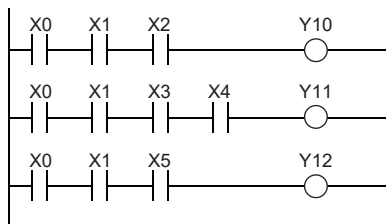
要點

- 使用及未使用MPS指令、MRD指令、MPP指令的情況的梯形圖如下所示。

[使用了MPS指令、MRD指令、MPP指令的梯形圖]



[未使用MPS指令、MRD指令、MPP指令的梯形圖]



- MPS指令、MPP指令的使用數應設定為相同。MPS指令、MPP指令的使用數不相同的情況下，工程工具的梯形圖模式中可能無法正常顯示梯形圖。

運算結果反轉

INV

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對INV指令之前的運算結果進行反轉。

梯形圖	ST
	ENO:=INV(EN);

FBD/LD

執行條件

指令	執行條件
INV	常時執行

功能

- 對INV指令之前的運算結果進行反轉。

INV指令之前為止的運算結果	執行INV指令後的運算結果
OFF	ON
ON	OFF

出錯

沒有運算出錯。

要點

- INV指令是以INV指令之前為止的運算結果執行動作，因此應在與AND(148頁 運算開始、串聯連接、並聯連接)指令同一位置使用。INV指令無法在LD、OR(148頁 運算開始、串聯連接、並聯連接)的位置使用。
- 在使用了梯形圖塊的情況下，對梯形圖塊的範圍的運算結果進行反轉。在並用INV指令及ANB指令使得梯形圖動作的情況下，應注意反轉的範圍。



虛線部：反轉範圍

關於ANB指令的詳細內容，請參閱下述內容。

156頁 ANB、ORB

運算結果脈衝化

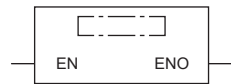
MEP、MEF

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

- MEP：運算結果脈衝化指令(上升沿)
在MEP指令之前的運算結果的上升沿時(OFF→ON)導通。
- MEF：運算結果脈衝化指令(下降沿)
在MEF指令之前的運算結果的下降沿時(ON→OFF)導通。

梯形圖	ST
	<pre>ENO:=MEP(EN); ENO:=MEF(EN);</pre>

FBD/LD



執行條件

指令	執行條件
MEP MEF	常時執行

功能

MEP

- 在MEP指令之前的運算結果為上升沿時(OFF→ON)，變為ON(導通狀態)。MEP指令之前的運算結果為上升沿以外的情況下，變為OFF(非導通狀態)。
- 使用MEP指令時，對多個觸點進行了串聯連接的情況下，脈衝化處理將易於進行。

MEF

- 在MEF指令之前的運算結果為下降沿時(ON→OFF)，變為ON(導通狀態)。MEF指令之前的運算結果為下降沿以外的情況下，變為OFF(非導通狀態)。
- 使用MEF指令時，對多個觸點進行了串聯連接的情況下，脈衝化處理將易於進行。

出錯

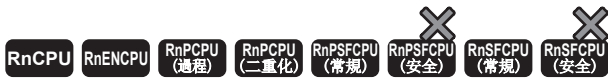
沒有運算出錯。

要點

- 對於MEP指令、MEF指令，如果透過子程式及FOR~NEXT指令等進行變址修飾後的觸點的脈衝化，將可能無法正常動作。若是透過子程式、FOR~NEXT指令進行變址修飾後的觸點的脈衝化的情況下，請參閱下述內容。
☞ 161頁 EGP、EGF
- 對於MEP指令、MEF指令，由於是以從MEP指令、MEF指令之前的LD指令開始至MEP指令、MEF指令之前為止的運算結果來執行動作，因此應在與AND(☞ 148頁 運算開始、串聯連接、並聯連接)指令相同的位置使用。MEP指令、MEF指令無法在LD、OR(☞ 148頁 運算開始、串聯連接、並聯連接)的位置使用。

變址繼電器運算結果脈衝化

EGP、EGF



• EGP：變址繼電器運算結果脈衝化指令(上升沿)

將EGP指令之前的運算結果透過變址繼電器(V)儲存。在運算結果的上升沿時(OFF→ON)導通。

• EGF：變址繼電器運算結果脈衝化指令(下降沿)

將EGF指令之前的運算結果透過變址繼電器(V)儲存。在運算結果的下降沿時(ON→OFF)導通。

梯形圖	ST
	<pre>ENO:=EGP(EN, d); ENO:=EGF(EN, d);</pre>

FBD/LD

■執行條件

指令	執行條件
EGP	
EGF	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存運算結果的變址繼電器編號	—	位元	ANY_BOOL*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用元件(V)中分配的位元型標籤。

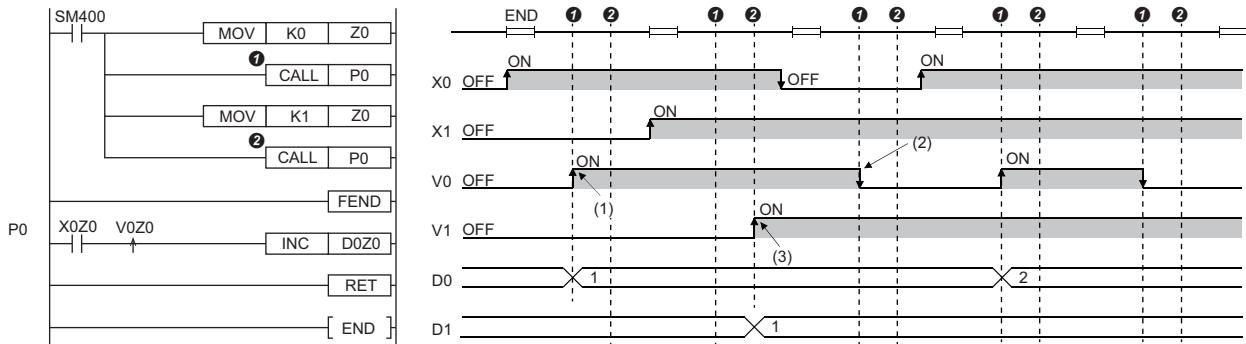
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它(V)	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(d)	—	—	—	—	—	—	—	—	—	—	—	○

功能

■EGP

- 將EGP指令之前的運算結果透過變址繼電器(V)儲存。
- 在EGP指令之前的運算結果為上升沿時(OFF→ON)，變為ON(導通狀態)。EGP指令之前的運算結果為上升沿以外(ON→ON、ON→OFF、OFF→OFF)的情況下，變為OFF(非導通狀態)。
- 對於EGP指令，透過子程式及FOR~NEXT指令，在進行變址修飾後的程式的脈衝運算的情況下使用。
- EGP指令不能在AND指令處理中使用。
- 在子程式中使用了EGP指令的情況下，其動作如下所示。



- (1): 在X0的上升沿為ON。
(2): 在X0的下降沿為OFF。
(3): 在X1的上升沿為ON。

■EGF

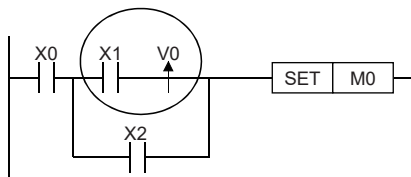
- 將EGF指令之前的運算結果透過變址繼電器(V)儲存。
- 在EGF指令之前的運算結果為下降沿時(ON→OFF)，變為ON(導通狀態)。在EGF指令之前的運算結果為下降沿以外(OFF→ON、ON→ON、OFF→OFF)的情況下，變為OFF(非導通狀態)。
- 對於EGF指令，透過子程式及FOR~NEXT指令進行變址修飾後的程式的脈衝運算的情況下使用。
- 對於EGF指令，不能在AND指令處理中使用。

出錯

沒有運算出錯。

要點

- 對於EGP及EGF指令，由於是以從EGP、EGF指令之前的LD指令開始至EGP指令、EGF指令為止的運算結果來執行動作，因此應在與AND指令(148頁 運算開始、串聯連接、並聯連接)相同的位置使用。EGP指令、EGF指令無法在LD、OR(148頁 運算開始、串聯連接、並聯連接)的位置使用。
- 對於EGP指令、EGF指令，在下述梯形圖塊的位置中不能使用。



5.3 輸出指令

OUT(定時器、計數器、報警器除外)

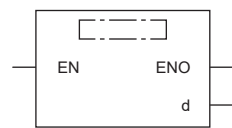
OUT

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將運算結果輸出到指定的元件中。

梯形圖	ST
	ENO:=OUT(EN, d);

FBD/LD



■執行條件

指令	執行條件
OUT	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	ON/OFF的元件編號	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元		Z	雙字元		間接指定	常數				其它 (DY)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□		LT、LST、LC	LZ		K、H	E	\$		
(d)	○*1	○	○*2	○	—	—	—	○	—	—	—	○	

- *1 在使用F的情況下，請參閱下述指令。
☞ 175頁 OUT F
- *2 使用T、ST的情況下，請參閱下述指令。
☞ 165頁 OUT T、OUTH T、OUT ST、OUTH ST
使用C的情況下，請參閱下述指令。
☞ 171頁 OUT C

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	○	○*3	—

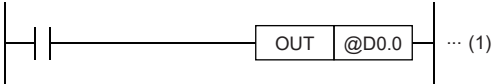
- *3 使用SA\T、SA\ST的情況下，請參閱下述指令。
☞ 165頁 OUT T、OUTH T、OUT ST、OUTH ST
使用SA\C的情況下，請參閱下述指令。
☞ 171頁 OUT C

功能

- 將OUT指令之前的運算結果輸出到指定的元件中。

條件	運算結果	線圈/指定位元
使用位元件時	OFF	OFF
	ON	ON
字元件的位元指定時	OFF	0
	ON	1

- 使用間接指定的情況下，按以下方式進行位元指定。



(1) 輸出到D0中儲存的間接地址的第0位元。

出錯

沒有運算出錯。

定時器

OUT T、OUTH T、OUT ST、OUTH ST

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用時，在系統切換的動作中有限制。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

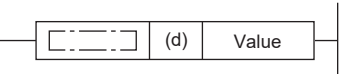
要點

在SIL2過程CPU及安全CPU的安全程式中使用安全定時器的情況下，除特別標明的情況以外，將按下述方式替換閱讀。

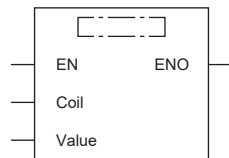
- “定時器” → “安全定時器”
- “累計定時器” → “安全累計定時器”
- “T” → “SA\T”、“ST” → “SA\ST”、“MO” → “SA\MO”
- “掃描” → “安全週期處理”
- “掃描時間” → “安全週期時間”

- OUT T: 低速定時器指令
- OUTH T: 高速定時器指令
- OUT ST: 低速累計定時器指令
- OUTH ST: 高速累計定時器指令

當OUT指令之前的運算結果為ON時，線圈將ON並進行定時器的計測。如果時限到，常開觸點將導通，常閉觸點將變為非導通。

梯形圖	ST
 <p>Value: 設定值</p>	<pre>ENO:=OUT_T(EN,Coil,Value); ENO:=OUTH(EN,Coil,Value);</pre>

FBD/LD



Value: 設定值
(□中放入OUT_T、OUTH。)

執行條件

指令	執行條件
OUT T OUTH T OUT ST OUTH ST	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d) Coil	定時器元件或定時器型標籤	—	位元	ANY_BOOL
Value	定時器的設定值	0~32767	無符號BIN16位元*1	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在ST、FBD/LD的情況下，資料類型變為ANY_INT。

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其它(DY)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(d) Coil	—	—	○*1	—	—	—	—	—	—	—	—	—	—
Value	—	—	○*2	○	—	—	—	—	○*3	—	—	—	—

*1 只能使用T、ST。

*2 不能使用T、ST、C。

*3 只能使用10進制常數(K)。

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元		字元		常數	
	SA\X、SA\Y、SA\M、SA\SM、SA\B		SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD		K、H	
(d) Coil	—		○*4		—	
Value	—		○*5		○*6	

*4 只能使用SA\T、SA\ST。

*5 不能使用SA\T、SA\ST、SA\C。

*6 只能使用10進制常數(K)。

功能

- 當OUT指令之前的運算結果為ON時，(d)(ST、FBD/LD的情況下為Coil)中指定的定時器的線圈將變為ON並進行計測直至設定值為止。如果時限到(當前值≥設定值)，常閉觸點將導通，常開觸點將變為非導通。
- 當OUT指令之前的運算結果由ON→OFF變化時，其情況如下所示。

定時器的類型	定時器線圈	定時器的當前值	時限到前		時限到後	
			常開觸點	常閉觸點	常開觸點	常閉觸點
低速定時器	OFF	0	非導通	導通	非導通	導通
高速定時器						
低速累計定時器	OFF	保持當前值	非導通	導通	導通	非導通
高速累計定時器						

- 時限到後，透過RST指令進行累計定時器的當前值的清除及觸點的OFF。
- 當設定值為0時，執行OUT指令時將時限到。
- 當執行OUT指令時，進行下述處理。
 - OUT T指令、OUTH T指令、OUT ST指令、OUTH ST指令的線圈的ON/OFF
 - OUT T指令、OUTH T指令、OUT ST指令、OUTH ST指令的觸點的ON/OFF
 - OUT T指令、OUTH T指令、OUT ST指令、OUTH ST指令的當前值的更改
- OUT T指令、OUTH T指令、OUT ST指令、OUTH ST指令為ON中，透過JMP指令等跳過了OUT T指令的情況下，將不進行當前值的更新及觸點的ON/OFF。
- 將同一OUT T指令、OUTH T指令、OUT ST指令、OUTH ST指令在同一掃描內執行了2次及其以上的情況下，將按照執行的次數進行當前值更新。

要點

- 定時器的時限設定是在工程工具的參數設定中進行。
低速定時器/低速累計定時器：1~1000ms(預設：100ms、設定單位：1ms)
高速定時器/高速累計定時器：0.01~100.0ms(預設：10.0ms、設定單位：0.01ms)
- 關於定時器的計數方法，請參閱下述手冊。
MELSEC iQ-R CPU模組用戶手冊(應用篇)

注意事項

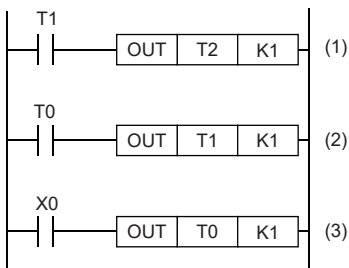
在建立透過定時器的觸點進行其它定時器的計測的程式的情況下，應從之後計測的定時器開始依次程式。

在下述情況下，如果按計測順序進行程式，所有的定時器將在同一掃描中變為ON。

- 當設定值短於掃描時間時
- 在設定值為1的情況下

例

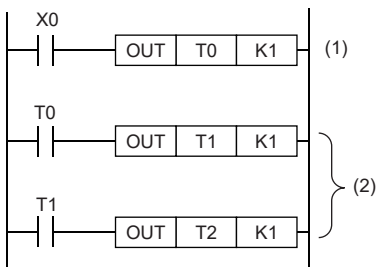
在T0~T2的定時器按照依後計測的定時器開始的順序進行了程式的情況下



- (1) 自T1的觸點變為ON的下一個掃描起，T2的定時器將開始計測。
- (2) 自T0的觸點變為ON的下一個掃描起，T1的定時器將開始計測。
- (3) 當X0為ON時，T0的定時器將開始計測。

例

在T0~T2的定時器按照計測順序進行了程式的情況下



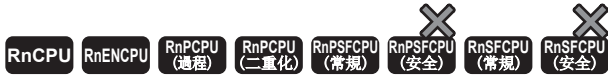
- (1) X0為ON時，T0的定時器將開始計測。
- (2) T0的觸點變為ON時，T1、T2的定時器的觸點也將變為ON。

出錯

沒有運算出錯。

超長定時器

OUT LT、OUT LST



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統將不進行動作。(□MELSEC iQ-R CPU模組用戶手冊(應用篇))

- OUT LT: 低速超長定時器指令
- OUT LST: 低速超長累計定時器指令

當OUT指令之前的運算結果為ON時，線圈將ON並進行定時器的計測。如果時限到，常開觸點將導通，常閉觸點將變為非導通。

梯形圖	ST
<p>Value: 設定值</p>	ENO:=OUT_T(EN,Coil,Value);

FBD/LD
<p>Value: 設定值 (□中放入OUT_T。)</p>

■執行條件

指令	執行條件
OUT LT	常時執行
OUT LST	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	超長定時器元件或超長定時器型標籤	—	位元	ANY_BOOL
Coil				
Value	超長定時器的設定值	0~4294967295	無符號BIN32位元*1	ANY32*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在ST、FBD/LD的情況下，資料類型變為ANY_INT。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(d)	—	—	—	—	—	○*1	—	—	—	—	—
Coil											
Value	—	—	○*2	○	—	—	—	—	—	○*3	—

*1 只能使用LT、LST。

*2 不能使用T、ST、C。

*3 只能使用10進制常數(K)。

功能

- 當OUT指令之前的運算結果為ON時，(d) (ST、FBD/LD的情況下為Coil)中指定的超長定時器的線圈將變為ON並進行計測直至設定值為止。如果時限到(當前值 \geq 設定值)，常開觸點將導通，常閉觸點將變為非導通。
- 當OUT指令之前的運算結果由ON \rightarrow OFF變化時，其情況如下所示。

定時器的類型	定時器線圈	定時器的當前值	時限到前		時限到後	
			常開觸點	常閉觸點	常開觸點	常閉觸點
超長定時器	OFF	0	非導通	導通	非導通	導通
超長累計定時器	OFF	保持當前值	非導通	導通	導通	非導通

- 時限到後，透過RST指令進行超長累計定時器的當前值清除及觸點的OFF。
- 當設定值為0時，執行OUT指令時將時限到。
- 當執行OUT指令時，進行下述處理。
 - OUT LT指令、OUT LST指令的線圈的ON/OFF
 - OUT LT指令、OUT LST指令的觸點的ON/OFF
 - OUT LT指令、OUT LST指令的當前值的更改
- 在OUT LT指令、OUT LST指令為ON中，透過JMP指令等跳過了OUT LT指令的情況下，將不進行當前值的更新及觸點的ON/OFF。
- 當同一OUT LT指令、OUT LST指令在同一掃描內執行了2次及其以上的情況下，將按照執行的次數進行當前值的更新。

要點

- 定時器的時限設定是在工程工具的參數設定中進行。
超長定時器/超長累計定時器：0.001~1000ms(預設：0.001ms、設定單位：0.001ms)
- 關於定時器的計數方法，請參閱下述手冊。
MELSEC iQ-R CPU模組用戶手冊(應用篇)

注意事項

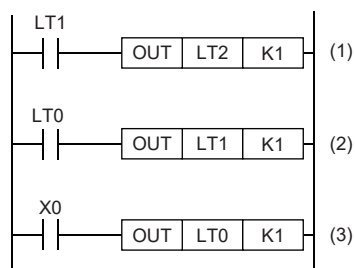
在建立透過超長定時器的觸點進行其它超長定時器的計測的程式的情況下，應按照從之後計測的超長定時器開始的順序進行程式。

在下述情況下，如果按計測順序進行程式，所有的定時器將在同一掃描中變為ON。

- 當設定值短於掃描時間時
- 在設定值為1的情況下

例

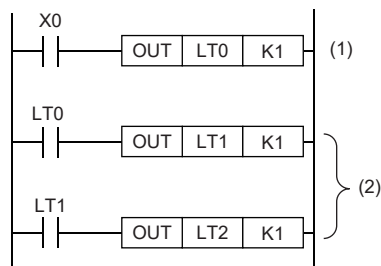
在將LT0~LT2的定時器按照從後計測的超長定時器開始的順序進行程式的情況下



- 自LT1的觸點變為ON的下一個掃描起，LT2的定時器將開始計測。
- 自LT0的觸點變為ON的下一個掃描起，LT1的定時器將開始計測。
- 當X0變為ON時，LT0的定時器將開始計測。

例

在將LT0~LT2的定時器按照計測順序進行程式的情況下



- 當X0變為ON時，LT0的定時器將開始計測。
- 當LT0的觸點變為ON時，LT1、LT2的定時器的觸點也將變為ON。

出錯

沒有運算出錯。

計數器

OUT C

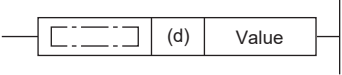
RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

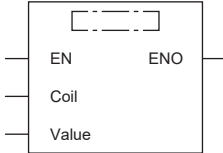
要點

在SIL2過程CPU及安全CPU的安全程式中使用安全計數器的情況下，除特別標明的情況以外，將按下述方式替換閱讀。

- “OUT C” → “OUT SA\C”

OUT指令之前的運算結果由OFF→ON變化時，計數器的當前值(計數值)將被+1。如果計數到，計數器的常開觸點將導通，常閉觸點將變為非導通。

梯形圖	ST
 <p>Value: 設定值</p>	ENO:=OUT_C(EN,Coil,Value);

FBD/LD
 <p>Value: 設定值</p>

執行條件

指令	執行條件
OUT C	常時執行

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	計數器編號	—	位元	ANY_BOOL*1
Coil				
Value	計數器的設定值	0~65535	無符號BIN16位元*2	ANY16*2
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用計數器型標籤。

*2 在ST、FBD/LD的情況下，資料類型變為ANY_INT。

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	—	—	○*1	—	—	—	—	—	—	—	—	—
Coil												
Value	—	—	○*2	○	—	—	—	—	○*3	—	—	—

*1 只能使用C。

*2 不能使用T、ST、C。

*3 只能使用10進制常數(K)。

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	—	○*4	—
Coil			
Value	—	○*5	○*6

*4 只能使用SA\C。

*5 不能使用SA\T、SA\ST、SA\C。

*6 只能使用10進制常數(K)。

功能

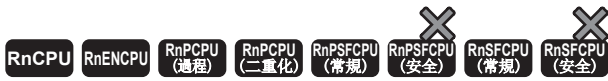
- 當OUT指令之前的運算結果由OFF→ON變化時，(d) (ST、FBD/LD的情況下為Coil)中指定的計數器的當前值(計數值)將被+1。如果計數到(當前值≥設定值)，常開觸點將導通，常閉觸點將變為非導通。
- 在運算結果維持ON不變的狀況下不進行計數。(計數輸入無需脈衝化。)
- 計數到後，在執行RST指令之前計數值及觸點的狀態不變化。
- 在設定值為0的情況下，其處理與1時相同。

出錯

沒有運算出錯。

超長計數器

OUT LC



當OUT指令之前的運算結果由OFF→ON變化時，超長計數器的當前值(計數值)將被+1。如果計數到，超長計數器的常開觸點將導通，常閉觸點將變為非導通。

梯形圖	ST
<p>Value: 設定值</p>	<pre>ENO:=OUT_C(EN,Coil,Value);</pre>

FBD/LD
<p>Value: 設定值 (□中放入OUT_C。)</p>

■執行條件

指令	執行條件
OUT LC	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	超長計數器編號	—	位元	ANY_BOOL*1
Coil				
Value	超長計數器的設定值	0~4294967295	無符號BIN32位元*2	ANY32*2
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用超長計數器型標籤。

*2 在ST、FBD/LD的情況下，資料類型變為ANY_INT。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(d)	—	—	—	—	—	○*1	—	—	—	—	—
Coil											
Value	—	—	○*2	○	—	—	—	—	—	○*3	—

*1 只能使用LC。

*2 不能使用T、ST、C。

*3 只能使用10進制常數(K)。

功能

- 當OUT指令之前的運算結果由OFF→ON變化時，(d) (ST、FBD/LD的情況下為Coil) 中指定的超長計數器的當前值(計數值)將被+1。如果計數到(當前值≥設定值)，常開觸點將導通，常閉觸點將變為非導通。
- 在運算結果維持ON不變的狀況下不進行計數。(計數輸入無需脈衝化。)
- 計數到後，在執行RST指令之前計數值及觸點的狀態不變化。
- 在設定值為0的情況下，其處理與1時相同。

出錯

沒有運算出錯。

報警器

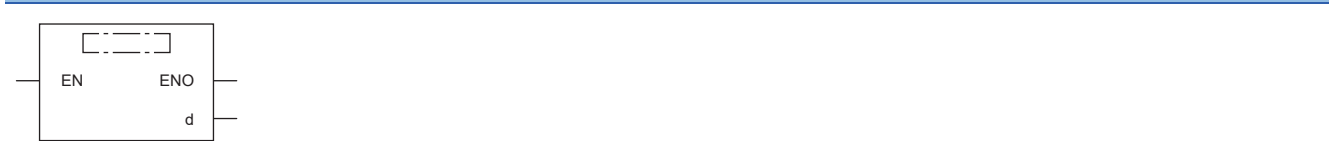
OUT F



將OUT F指令之前的運算結果輸出到指定的報警器。

梯形圖	ST
	ENO:=OUT (EN, d);

FBD/LD



(□中放入OUT F。)

■執行條件

指令	執行條件
OUT F	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	變為ON的報警器編號	—	位元	ANY_BOOL*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用報警器中分配的標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(d)	○*1	—	—	—	—	—	—	—	—	—	—	—

*1 只能使用F。

功能

- 將OUT F指令之前的運算結果輸出到指定的報警器。
- 透過OUT F指令將報警器(F)設為ON的情況如下所示。
 - CPU模組前面的USER LED將亮燈。
 - 變為ON的報警器編號(F編號)將被儲存到特殊寄存器(SD64~SD79)中。
 - SD63的內容將被+1。
- 在SD63的內容為16的情況下(報警器已有16個ON的情況時)，即使新的報警器ON，變為ON的報警器編號也不被儲存到SD64~SD79中。
- 透過OUT F指令將報警器設為OFF的情況如下所示。
 - 線圈將變為OFF，但USER LED的狀態、SD63~SD79的內容不變化。
 - USER LED的熄燈且從SD63 ~SD79中刪除透過OUT F指令而OFF的報警器的情況下，透過RST F指令進行。

出錯

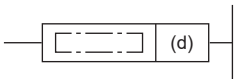
沒有運算出錯。

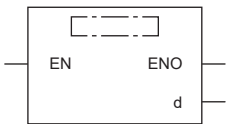
元件的設定(報警器除外)

SET


RnCPU RnENCPU RnPCPU (通規) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將指定位設為ON。

梯形圖	ST
	ENO:=SET(EN, d);

FBD/LD


■執行條件

指令	執行條件
SET	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	設定(ON)的位元元件編號/字元元件的位元指定	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它(DY)	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
(d)	○*1	○	—	—	—	—	—	○	—	—	—	○

*1 在使用F的情況下，請參閱下述指令。

☞ 180頁 SET F

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

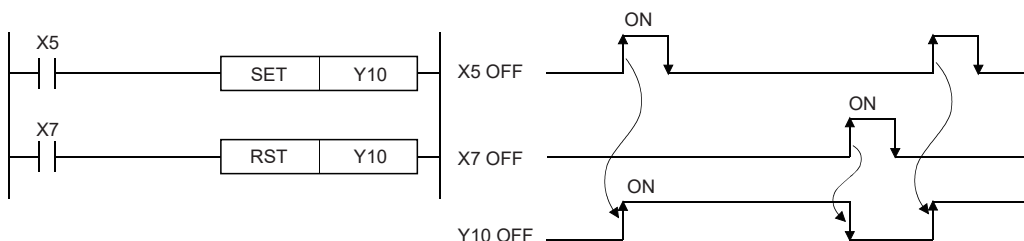
操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	○	—	—

功能

- 當執行指令為ON時，指定元件將變為下述狀態。

元件	元件的狀態
位元元件	將線圈、觸點設為ON。
字元元件的位元指定	將指定位設為1。

- 對於設為ON的元件，即使執行指令變為OFF也將保持為ON不變。對於透過SET指令設為ON的元件，可以透過RST指令設為OFF。



- 在執行指令為OFF的情況下，元件的狀態不變化。

出錯

沒有運算出錯。

要點

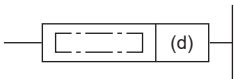
在元件中使用X的情況下，應使用實際輸入中未使用的元件編號。透過使用與實際輸入元件相同的編號，實際輸入的資料將被由SET指令指定的輸入X所覆蓋。

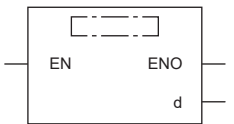
元件的重設(報警器除外)

RST

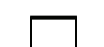
RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將指定元件設為0(OFF)。關於定時器、計數器，將當前值設為0，將觸點/線圈設為OFF。

梯形圖	ST
	ENO:=RST(EN, d);

FBD/LD


■執行條件

指令	執行條件
RST	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	重設的位元元件編號/字元元件的位元指定或重設的字元元件編號	—	位元/字元/雙字元	ANY_ELEMENTARY
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它(DY)	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
(d)	○*1	○	○	○	○	○	○	○	—	—	—	○

*1 在使用P的情況下，請參閱下述指令。

☞ 182頁 RST F

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

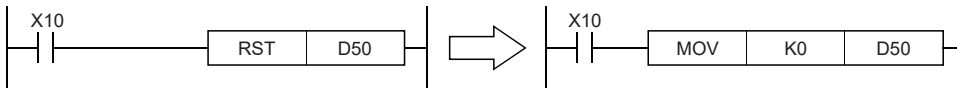
操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	○	○	—

功能

- 當執行指令為ON時，指定元件將變為下述狀態。

元件	元件的狀態
位元元件	將線圈、觸點設為OFF。
定時器、計數器	將當前值設為0，將線圈、觸點設為OFF。
字元元件的位元指定	將指定位元設為0。
定時器、計數器以外的字元元件	將內容設為0。

- 在執行指令為OFF的情況下，元件的狀態不變化。
- 對於透過RST指令指定字元元件時的指令，與下述梯形圖相同。

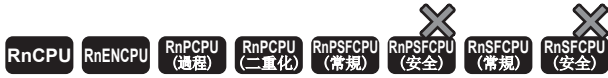


出錯

沒有運算出錯。

報警器的設定

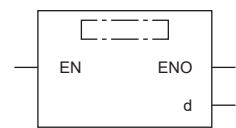
SET F



將指定的報警器設為ON。

梯形圖	ST
	ENO:=SET (EN, d) ;

FBD/LD



(□中放入SET。)

■執行條件

指令	執行條件
SET F	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	設定的報警器編號(F編號)	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

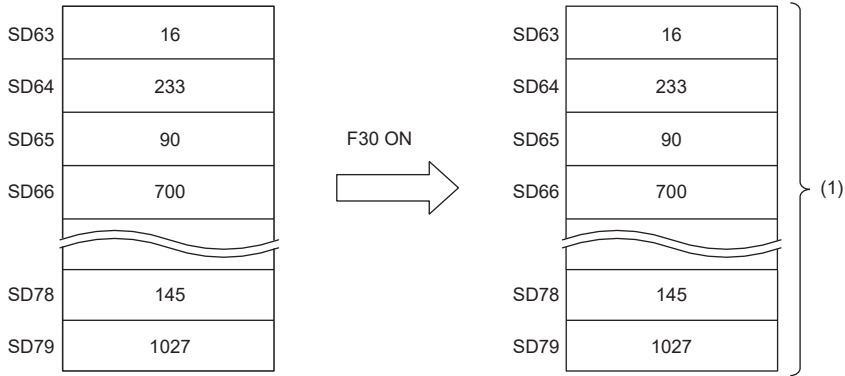
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(d)	○*1	—	—	—	—	—	—	—	—	—	—	—

*1 只能使用F。

功能

- 當執行指令為ON時，將(d)中指定的報警器設為ON。
- 將報警器(F)設為ON的情況如下所示。
 - USER LED將亮燈。
 - 變為ON的報警器編號(F編號)將被儲存到特殊寄存器(SD64~SD79)中。
 - SD63的內容將被+1。
- 在SD63的內容為16的情況下(報警器已有16個ON的情況時)，即使新的報警器ON，變為ON的報警器編號也不被儲存到SD64~SD79中。



(1): 不變化。

出錯

沒有運算出錯。

報警器的重設

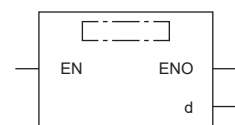
RST F



將指定的報警器設為OFF。

梯形圖	ST
	ENO:=RST (EN, d) ;

FBD/LD



(□中放入RST。)

■執行條件

指令	執行條件
RST F	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	重設的報警器編號(F編號)	—	位元 ^{*1}	ANY_BOOL ^{*1}
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在ST、FBD/LD的情況下，資料類型將變為ANY_ELEMENTARY。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(d)	○ ^{*1}	—	—	—	—	—	—	—	—	—	—

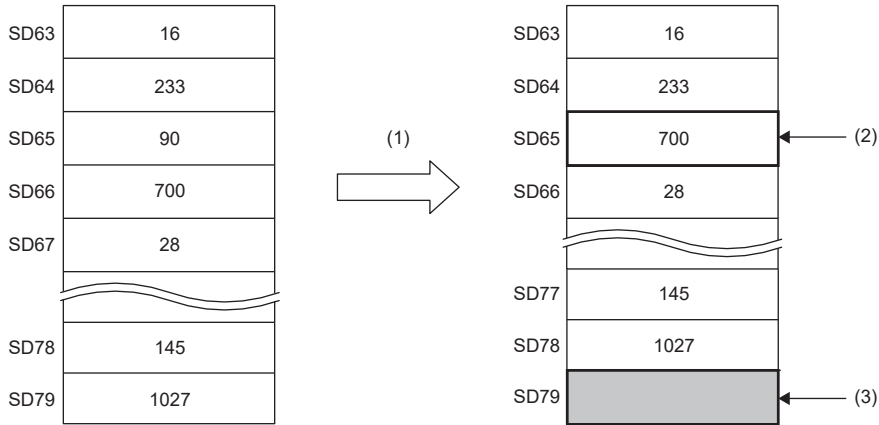
*1 只能使用F。

功能

- 當執行指令變為ON時，將(d)中指定的報警器設為OFF。
- 變為OFF的報警器編號(F編號)將從特殊寄存器(SD64~SD79)中被刪除，SD63的內容將被-1。
- 當SD63的內容為16時，透過RST指令，SD64~SD79報警器編號將被刪除。此外，當SD64~SD79中未登錄的編號的報警器變為ON時，該編號將被新建登錄。如果將SD64~SD79的報警器編號全部設為OFF，CPU模組前面的USER LED將熄燈。

例

在SD63為16、有未登錄的報警器編號的情況下



- (1): 重設F90
(2): 儲存已儲存在SD66的F編號。
(3): 儲存未登錄的F編號。

出錯

沒有運算出錯。

上升沿輸出

PLS

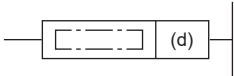
RnCPU RnENCPU RnPCPU (通規) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

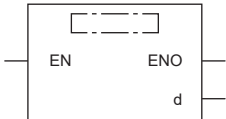
要點

在SIL2過程CPU及安全CPU的安全程式中使用本指令的情況下，除特別標明的情況以外，應按下述方式替換閱讀。


- “X0” → “SA\X0”、“X5” → “SA\X5”、“MO” → “SA\MO”
- “掃描” → “安全週期處理”

在PLS指令的OFF→ON時，將指定元件設為1個掃描ON。

梯形圖	ST
	ENO:=PLS (EN, d);

FBD/LD


執行條件

指令	執行條件
PLS	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	脈衝化的元件編號	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其它 (DY)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	○	○	○	○	—	—	—	○	—	—	—	○

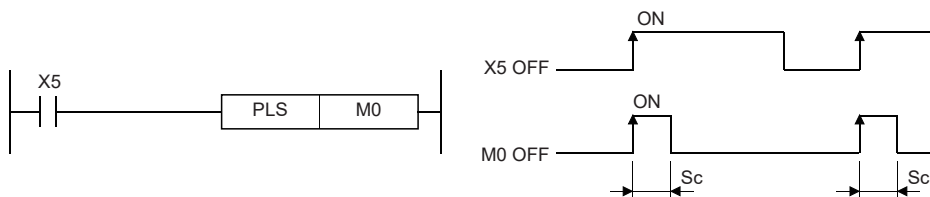
- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	○	○	—

功能

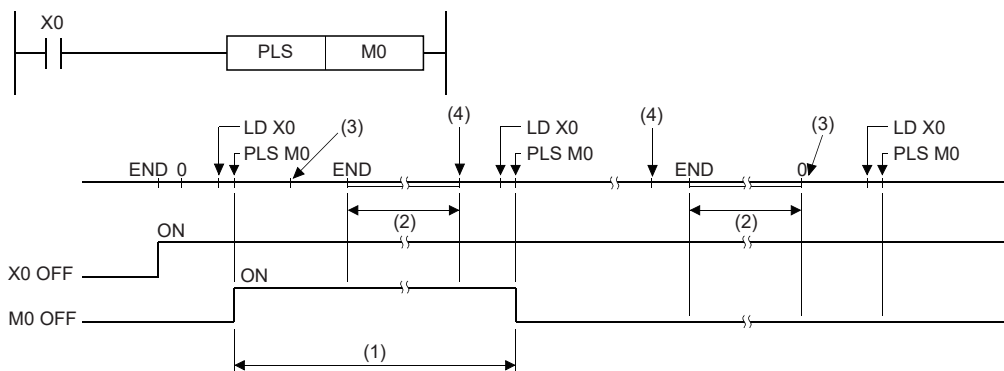
- 執行指令OFF→ON時將指定元件設為ON，執行指令OFF→ON以外(ON→ON、ON→OFF、OFF→OFF)時不設為OFF。在1個掃描中(d)中指定的元件的PLS指令為1個的情況下，指定元件將1個掃描ON。關於對同一元件在1個掃描中多次執行PLS指令情況下的動作，請參閱下述內容。

☞ 56頁 使用同一元件的OUT指令、SET/RST指令、PLS/PLF指令時的動作



Sc: 1個掃描

- 執行PLS指令後設為RUN→STOP，即使再次設為RUN也不執行PLS指令。



- (1) PLS M0的1個掃描
- (2) CPU模組的運算停止時間
- (3) 將CPU模組的RUN/STOP/RESET開關設為RUN→STOP。
- (4) 將CPU模組的RUN/STOP/RESET開關設為STOP→RUN。

- 在執行指令中指定鎖存繼電器(L)，鎖存繼電器為ON的狀態下將電源設為OFF→ON時，執行指令將在第1個掃描變為OFF→ON，執行PLS指令且指定元件將變為ON。對於電源ON後第1個掃描中變為ON的元件，透過下一個PLS指令(上升沿輸出)將變為OFF。
- 對於PLS指令，在執行指令後的下一個指令時進行OFF處理。但是，在SIL2過程CPU及安全CPU的安全程式中，PLS指令在下一個安全週期處理的安全程式動作，且在指令被執行前將指定的元件/標籤在1安全週期設為ON。對此，在PLS指令的引數中使用常規/安全共有標籤的情況下，根據安全週期中斷的時機，在對應的常規/安全共有標籤使用的程式或常規的程式中，可能會產生檢測不出常規/安全共有標籤為ON、以及檢測出複數週期為ON的狀況發生。

要點

- 將PLS指令透過CJ指令進行跳轉，執行的子程式未透過CALL(P)指令調用的情況下，(d)中指定的元件可能至少有1個掃描為ON，應加以注意。

出錯

沒有運算出錯。

下降沿輸出

PLF

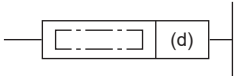
RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

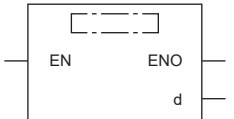
要點

在SIL2過程CPU及安全CPU的安全程式中使用本指令的情況下，除特別標明的情況以外，應按下述方式替換閱讀。


- “X5” → “SA\X5”
- “掃描” → “安全週期處理”

PLF指令ON→OFF時將指定元件設為1個掃描ON。

梯形圖	ST
	ENO:=PLF (EN, d);

FBD/LD


執行條件

指令	執行條件
PLF	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	脈衝化的元件編號	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其它 (DY)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	○	○	○	○	—	—	—	○	—	—	—	○

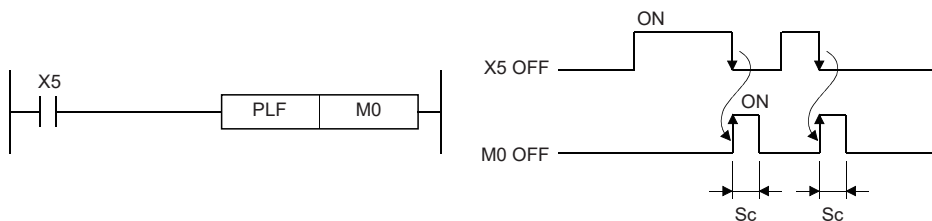
- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	○	○	—

功能

- 執行指令ON→OFF時將指定元件設為ON，執行指令為ON→OFF以外 (OFF→OFF、OFF→ON、ON→ON)時將變為OFF。在1個掃描中 (d)中指定的元件的PLF指令為1個的情況下，指定元件將1個掃描ON。關於在1個掃描中對同一元件執行了多次PLF指令時的動作，請參閱下述內容。

☞ 56頁 使用同一元件的OUT指令、SET/RST指令、PLS/PLF指令時的動作



Sc: 1個掃描

- 執行PLF指令後設為RUN→STOP，即使再次設為RUN也不執行PLF指令。
- 對於PLF指令，在指令執行後的下一個指令執行時進行OFF處理。但是，在SIL2過程CPU及安全CPU的安全程式中，PLF指令在下一個安全週期處理的安全程式動作，且在指令被執行前將指定的元件/標籤在1安全週期設為ON。對此，在PLF指令的引數中使用常規/安全共有標籤的情況下，根據安全週期中斷的時機，在對應的常規/安全共有標籤使用的程式或常規的程式中，可能會產生檢測不出常規/安全共有標籤為ON、以及檢測出複數週期為ON的狀況發生。

要點

- 將PLF指令透過CJ指令跳轉，在執行的子程式未進行CALL(P)指令調用的情況下，(d)中指定的元件可能至少有1個掃描為ON，因此應加以注意。

出錯

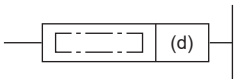
沒有運算出錯。

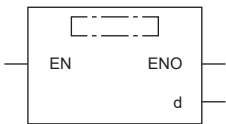
位元件輸出反轉

FF


RnCPU RnENCPU RnPCPU (通規) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的元件狀態進行反轉。

梯形圖	ST
	ENO:=FF(EN, d);

FBD/LD


■執行條件

指令	執行條件
FF	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	反轉的元件編號	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它 (DY)	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
(d)	○	○	○	○	—	—	—	○	—	—	—	○

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	○	○	—

功能

- 在執行指令OFF→ON時，對(d)中指定的元件狀態進行反轉。

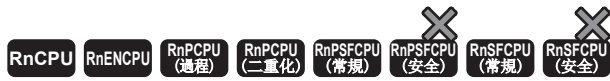
元件	元件的狀態	
	FF指令執行前	FF指令執行後
位元元件	OFF	ON
	ON	OFF
字元元件的位元指定	0	1
	1	0

出錯

沒有運算出錯。

直接輸出的脈衝化

DELTA (P)



對指定的直接存取輸出 (DY) 進行脈衝輸出。

梯形圖	ST
	ENO:=DELTA (EN, d) ; ENO:=DELTAP (EN, d) ;

FBD/LD

■執行條件

指令	執行條件
DELTA	
DELTAP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	脈衝化的元件編號	—	位元	ANY_BOOL* ¹
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用在元件 (DY) 中分配的標籤。

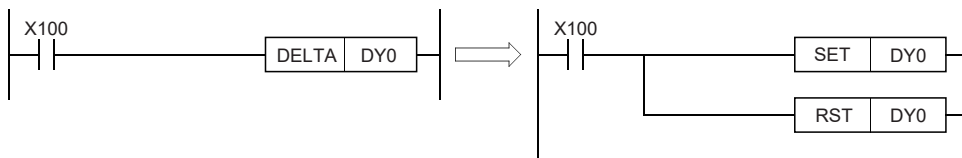
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它 (DY)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(d)	—	—	—	—	—	—	—	—	—	—	○

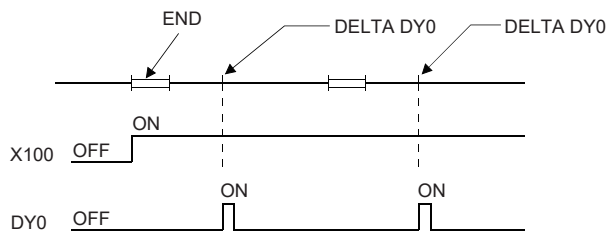
功能

- 對(d)中指定的直接存取輸出(DY)進行脈衝輸出。在(d)中指定DY0的情況下，其動作將與使用了SET指令、RST指令的梯形圖相同。

將已使用DELTA指令的梯形圖替換已使用SET/RST指令的梯形圖時，其情況如下所示。



動作如下所示。



注意事項

對於DELTA(P)指令，用於至智能功能模組的上升沿執行指令。
 在使用了輸出模組的實際輸出指令中不能使用。

出錯

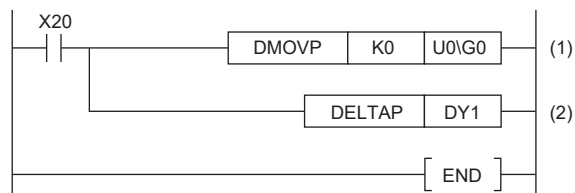
沒有運算出錯。

要點

對於DELTA(P)指令，用於高速計數器模組的預設值的設定等。

(例)

為X20變為ON時，進行基板的插槽0中安裝了高速計數器模組(RD62P2)的CH1的預設之程式。



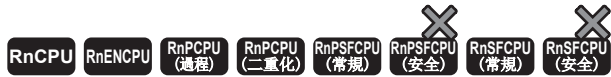
(1) RD62P2的緩衝記憶體0、1的地址中將儲存預置值(0)

(2) 預設指令的輸出

5.4 移位指令

位元元件移位

SFT(P)



將小1號的元件的ON/OFF狀態移位到指定的元件中，將小1號的元件設為OFF。

梯形圖	ST
	ENO:=SFT(EN, d); ENO:=SFTP(EN, d);

FBD/LD

■執行條件

指令	執行條件
SFT	
SFTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的元件	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其它 (DY)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(d)	○	○	○	○	—	—	—	○	—	—	—	○	

功能

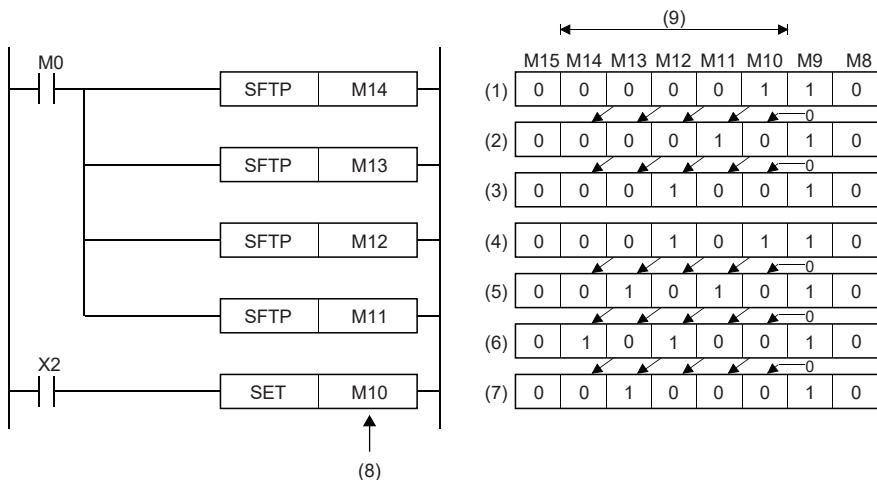
■在位元元件的情況下

- 將 (d) 中指定的元件的前一個元件的ON/OFF狀態移位到 (d) 中指定的元件中。(d) 中指定的元件的前一個元件將變為OFF。

例

透過SFTP指令指定了M11的情況下，執行SFTP指令時將M10的ON/OFF移位到M11中，將M10設為OFF。

- 對於移位的起始的元件應透過SET指令設為ON。
- 在連續使用SFT(P)指令的情況下，應從元件編號大的開始建立程式。



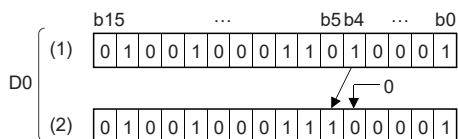
- (1): X02 ON
- (2): 第1次的移位輸入後
- (3): 第2次的移位輸入後
- (4): X02 ON
- (5): 第3次的移位輸入後
- (6): 第4次的移位輸入後
- (7): 第5次的移位輸入後
- (8): 移位的起始元件
- (9): 移位範圍

■在字元元件的位元指定的情況下

- 將 (d) 中指定位元的前一個位元的1/0狀態移位至 (d) 中指定的位元中。(d) 中指定的位元的前一個位元將變為0。

例

在透過SFT(P)指令指定了D0.5(D0的位元5(b5))的情況下，執行SFT(P)指令時將D0的b4的1/0移位到b5中，將b4設為0。



- (1): 移位執行前
- (2): 移位執行後

出錯

沒有運算出錯。

5.5 主控制指令

主控制的設定、重設

MC、MCR

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

要點

在SIL2過程CPU及安全CPU的安全程式中使用本指令的情況下，除特別標明的情況以外，應按下述方式替換閱讀。

- “定時器” → “安全定時器”
- “累計定時器” → “安全累計定時器”
- “計數器” → “安全計數器”
- 圖中的元件“X”、“M”裡應附加“SA\”。(例：“X0” → “SA\X0”，“M0” → “SA\M0”)
- “掃描時間” → “安全週期時間”

- MC：開始主控制。
- MCR：結束主控制。

梯形圖	ST
<p>(1)：主控制梯形圖</p>	<pre>ENO:=MC(EN, N, d); ENO:=MCR(EN, N);</pre>

FBD/LD	FBD/LD

■執行條件

指令	執行條件
MC	常時執行
MCR	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(N)	嵌套	N0~N14	元件名	ANY16_S*1
(d)	設為ON的元件編號	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用元件(N)中分配的標籤、或已分配常數的標籤。

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	N	DY
(N)	—	—	—	—	—	—	—	—	—	—	—	—	—
(d)	○	○	○	—	—	—	—	○	—	—	—	—	○

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

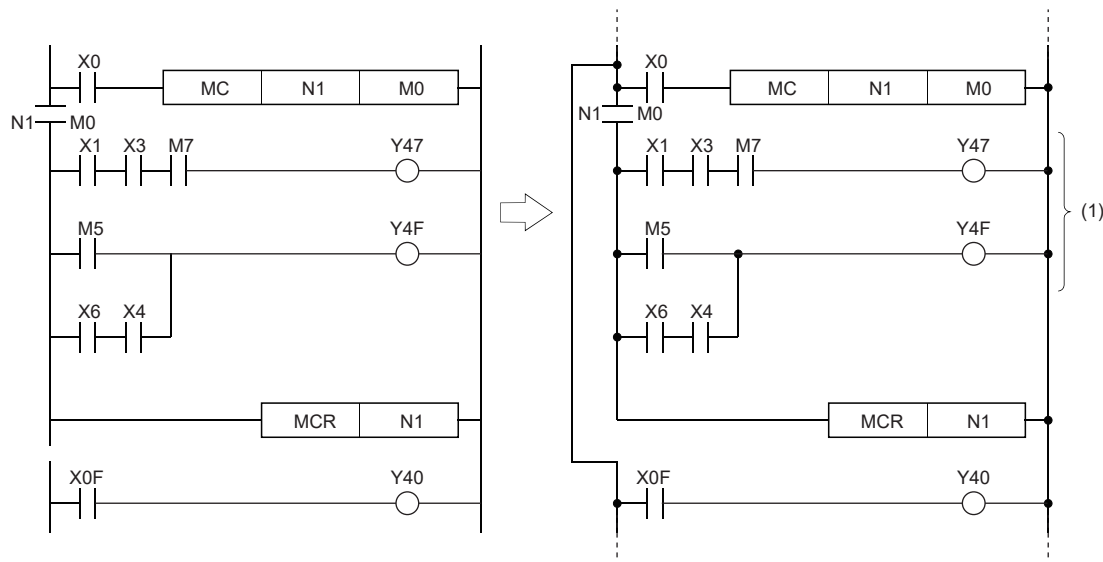
操作數	位元		字元		間接指定	常數		其他	
	SA\X、SA\Y、SA\M、SA\SM、SA\B		SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD			K、H	N		
(N)	—		—		—	—		—	○
(d)	○		○		○	—		—	—

功能

對於主控制指令，根據梯形圖的公共母線的開閉狀況，建立高效的梯形圖切換程式的指令。

使用了主控制的梯形圖如下所示。

(左：工程工具的顯示，右：實際的動作梯形圖)



(1) 僅在X0為ON時執行

■MC

• 在主控制的開始、MC指令的執行指令為ON的情況下，從MC指令開始到MCR指令之間的運算結果即為指令(梯形圖)的執行結果。在MC的執行指令為OFF的情況下，從MC指令開始到MCR指令之間的運算結果如下所示。

元件	元件的狀態
高速定時器 低速定時器	計數值變為0，線圈、觸點均變為OFF。
高速累計定時器 低速累計定時器 計數器	線圈變為OFF，但計數值、觸點均保持當前的狀態。
OUT指令中的元件	強制設為OFF。
SET指令、RST指令中的元件 SFT(P)指令中的元件 基本指令、應用指令中的元件	保持當前的狀態。

• 在MC指令為OFF的情況下，因為從MC指令開始到MCR指令之間的指令也將執行，因此掃描時間不會變短。

要點

- 在使用了主控制的梯形圖中，在存有不需要觸點指令的指令 (FOR~NEXT指令等) 的情況下，CPU 模組將與 MC 指令的執行指令無關下執行該指令。
- 為了建立易於理解的程式，成對的 MC 指令及 MCR 指令應在一個程式塊內使用。

- 對於 MC 指令，透過改變 (d) 的元件，可以多次使用同一嵌套 (N) 編號。
- 當 MC 指令為 ON 時，(d) 中指定的元件的線圈將變為 ON。此外，OUT 指令等中使用同一元件時將變為雙線圈，因此 (d) 中指定的元件請勿在其它指令中使用。

■MCR

- 透過執行主控制的解除指令，表示主控制範圍的結束。
- MCR 指令的前面請勿附加觸點指令。
- 使用時應設定同一嵌套編號的 MC 指令及 MCR 指令。但是，當 MCR 指令集中於 1 個位置的嵌套結構時，利用 1 個最小號的嵌套 (N) 編號，可以結束所有的主控制。(參閱注意事項)

出錯

沒有運算出錯。

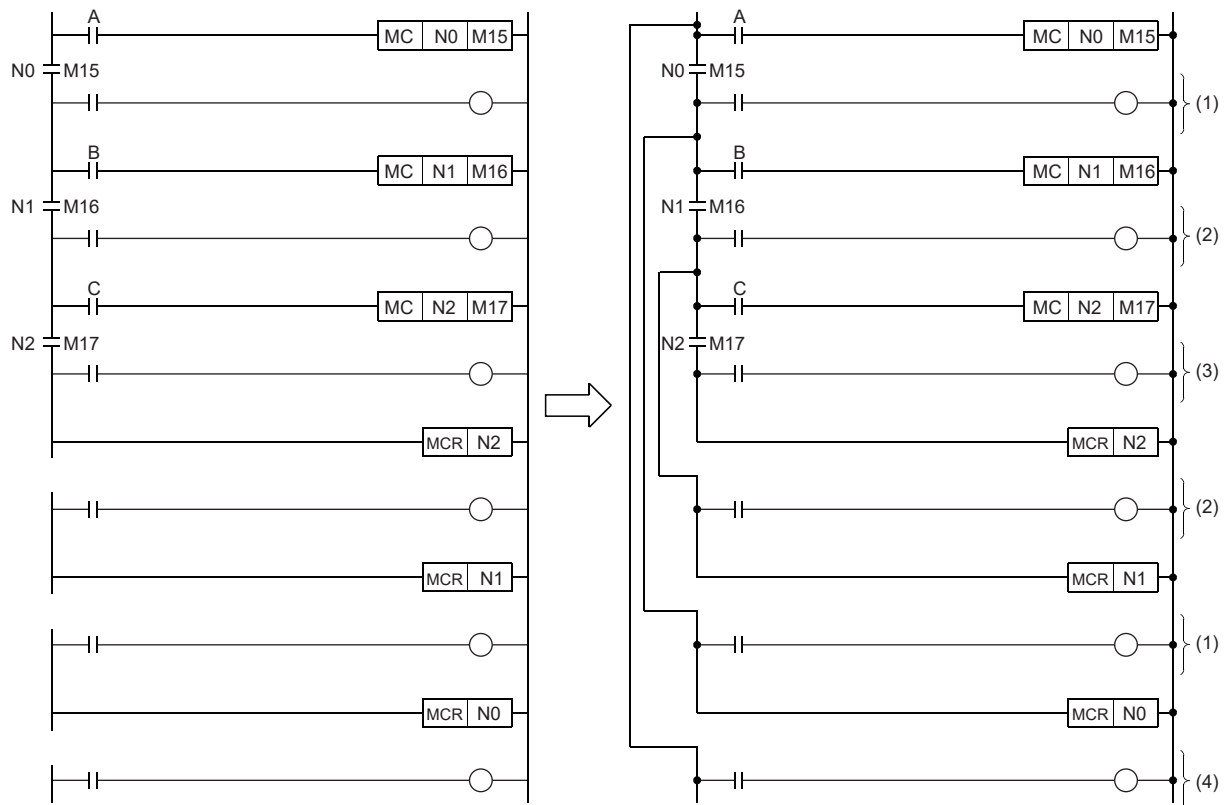
要點

主控制指令可透過嵌套結構使用。各個主控制區間透過嵌套 (N) 進行區分。嵌套可使用 N0~N14。

透過使用嵌套結構，可以建立對程式的執行條件依次進行限制的梯形圖。

使用了嵌套結構的梯形圖如下所示。

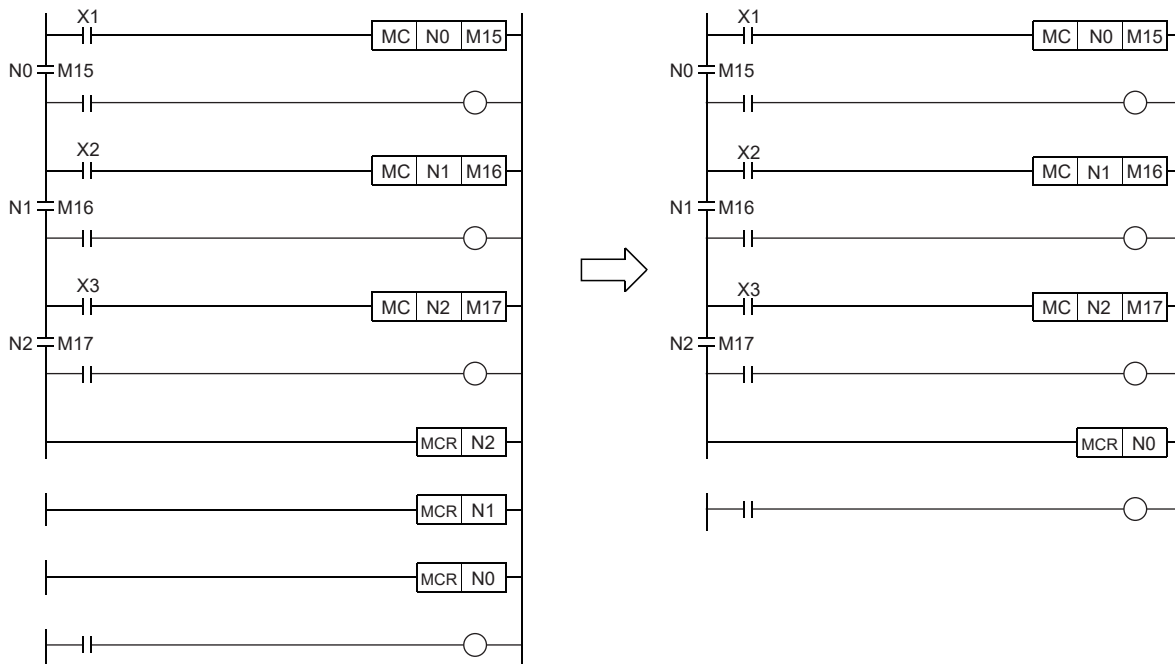
(左：工程工具的顯示，右：實際的動作梯形圖)



- (1) A 為 ON 時執行
- (2) A、B 為 ON 時執行
- (3) A、B、C 為 ON 時執行
- (4) 與 A、B、C 無關

注意事項

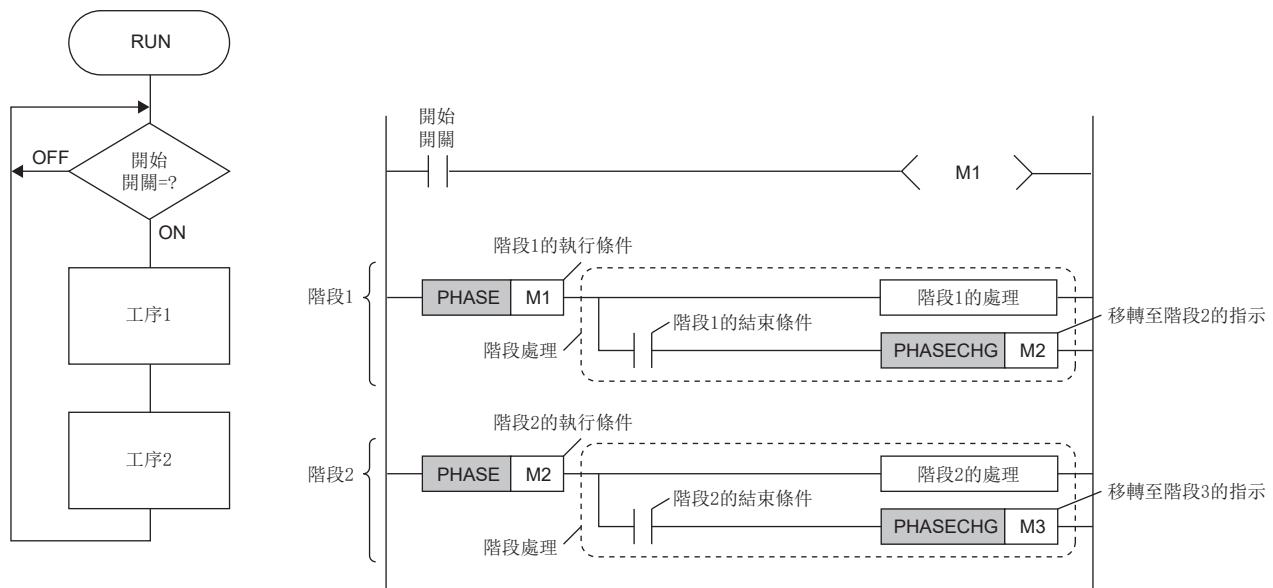
- 嵌套最多可以有15個(N0~N14)。在進行嵌套的情況下，MC指令中從嵌套(N)編號小的開始使用，而MCR指令是從編號大的開始使用。
- 當MCR指令集中於1個位置的嵌套結構時，利用1個最小號的嵌套(N)編號，可以結束所有的主控制。



5.6 階段處理指令

概要

階段處理指令是在梯形圖程式上根據工序進行依次處理時使用的指令。對每一個工序執行程式。



要點

- 根據工序進行依次處理的程式顯示有SFC程式。
透過步切換將一系列控制動作分割為多個步，可以明確地顯示程式的執行順序和執行條件，因此在控制複雜的工序時推薦使用SFC程式。
- PHASE指令將加法運算2步，因此在PHASE指令後的梯形圖程式上使用指定功能*1時，需要指定考慮了加法運算的2步(合計15步)的步No.。

*1 指定步No. 功能中有以下功能。

- 實時監視功能
- 執行條件的元件測試功能
- 資料記錄功能

工程工具的設定

使用階段處理指令需要在工程工具的選項中將“使用階段處理指令”設定為“是”。如果沒有進行本設定即轉換了輸入了階段處理指令的程式時，將變為出錯狀態。

[工具]⇒[選項]⇒[轉換]⇒[基本設定]⇒[動作設定]

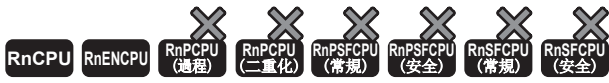
注意事項

下述為階段處理指令的通用的注意事項。

- 只能在梯形圖程式中使用。無法在SFC程式，ST程式，FBD/LD中使用。
- 只能在掃描執行類型程式中使用。初始執行類型程式、恆定週期執行類型程式、事件執行類型程式以及待機類型中無法使用。另外，輸入了階段處理指令的程式在工程工具上進行轉換後，請勿將其更改為掃描執行類型程式之外的程式執行類型。
- 只能在主程式中使用。子程式、中斷程式中無法使用。
- 只能在程式塊內使用。函數(FUN)、功能塊(FB)內無法使用。

階段處理開始

PHASE



• R00CPU、R01CPU、R02CPU可以在韌體版本為“24”及其以後的版本使用，RnCPU (R00CPU, R01CPU, R02CPU除外)、RnENCPU可以在韌體版本為“57”及其以後的版本使用。工程工具可以在版本為“1.075D”及其以後的版本使用。

開始執行階段處理。

梯形圖	ST
	不對應。

FBD/LD
不對應。

■執行條件

指令	執行條件
PHASE	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	階段處理開始的元件	—	位元	ANY_BOOL*1

*1 數組型標籤時，數組的添加文字中不能使用標籤或元件。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(s)	○*1*3	—	○*2*3	—	—	—	—	—	—	—	—	—

*1 不能使用F。

*2 不能使用T、ST、C、R、ZR。

*3 不能使用變址修飾。

功能

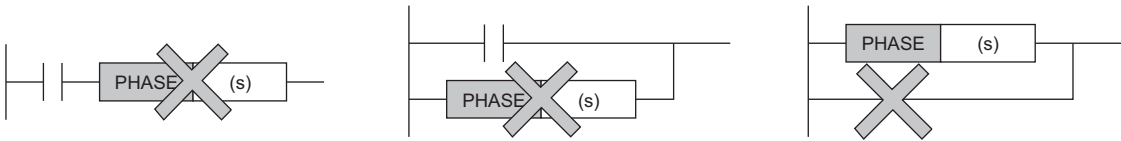
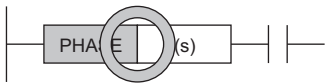
- (s)中指定的元件OFF→ON，ON→ON時，連接的階段處理的運算結果與指令(梯形圖)一致。
- (s)中指定的元件ON→OFF時，連接的階段處理的運算結果如下。

運算中使用的元件	運算結果(元件的狀態)
高速定時器 低速定時器	計數值變為0，線圈、觸點均變為OFF。
高速累計定時器 低速累計定時器 計數器	線圈變為OFF，但計數值、觸點均保持當前的狀態。
OUT指令中的元件	全部OFF。
SET/RST指令中的元件 SFT指令中的元件 基本/應用指令中的元件	保持當前的狀態。

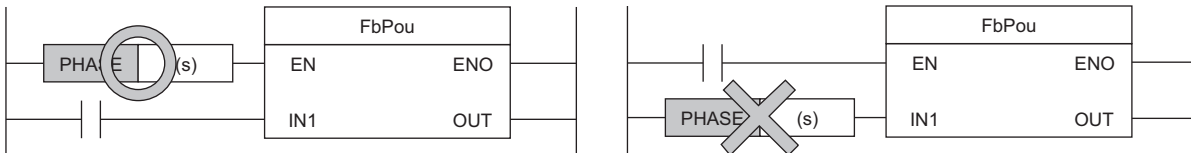
- (s)中指定的元件OFF→OFF時，連接的階段處理不進行處理。

注意事項

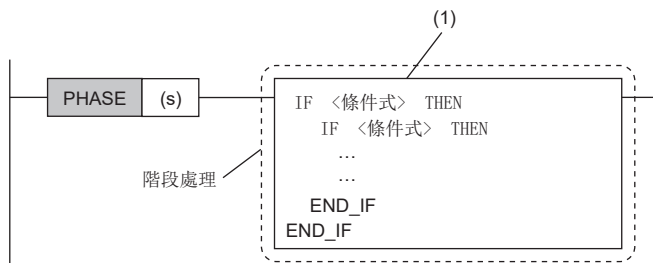
- PHASE指令僅可與母線直接連接。另外，不能與其他指令並聯連接。



- PHASE指令中連接FB/FUN時，僅可與FB/FUN的EN連接。



- 在PHASE指令連接的內嵌ST內使用控制語法(IF語句、CASE語句、FOR語句、WHILE語句、REPEAT語句)時，最大階層數不應超過127層。



(1) 階段處理中，內嵌ST的控制語法不應超過127層。

出錯

沒有運算出錯。

執行階段的移轉

PHASECHG



• R00CPU、R01CPU、R02CPU可以在韌體版本為“24”及其以後的版本使用，RnCPU (R00CPU, R01CPU, R02CPU除外)、RnENCPU可以在韌體版本為“57”及其以後的版本使用。工程工具可以在版本為“1.075D”及其以後的版本使用。

結束執行中的階段處理，移轉至指定的階段。

梯形圖	ST
	不對應。

FBD/LD
不對應。

執行條件

指令	執行條件
PHASECHG	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移轉目標階段的元件	—	位元	ANY_BOOL*1

*1 數組型標籤時，數組的添加文字中不能使用標籤或元件。

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(d)	○*1*3	—	○*2*3	—	—	—	—	—	—	—	—	—

*1 不能使用F。

*2 不能使用T、ST、C、R、ZR。

*3 不能使用變址修飾。

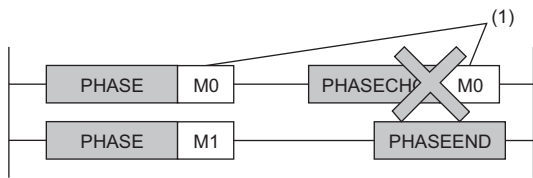
功能

• 結束執行中的階段(將之前的PHASE指令中指定的元件置為OFF*1)，移轉至(d)中指定的階段(將指定的元件置為ON)。

*1 PHASE指令連接的階段處理的非執行處理在本指令執行後的下一個掃描時執行。

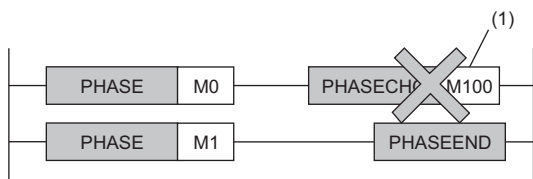
注意事項

- 應在階段處理內執行PHASECHG指令。在階段處理外執行時，將最後執行ON的PHASECHG指令的制定元件置為OFF*¹。(將執行階段的移轉(指定的元件的ON)。)
- *1 不包括PHASE指令執行ON之後到PHASECHG指令執行為止，進行了RUN中的梯形圖塊更改/RUN中的可程式控制器的寫入(順控程式、FB檔案、安全全局標籤設定檔案中任意一項的寫入)的情況。
- 請勿在(d)中指定與之前PHASE指令中指定的元件相同的元件。如果指定了，執行中的階段將不會結束，也不會進行階段移轉。



(1) PHASE指令與PHASECHG指令中指定相同的元件

- 請勿在(d)中指定PHASE指令中未使用的元件。如果指定了，雖然執行中的階段會結束，但不會進行階段移轉。(指定的元件ON。)



(1) 指定PHASE指令中未使用的元件

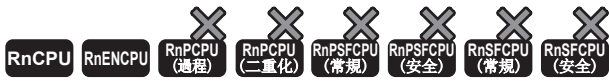
- 在元件中使用X的情況下，應使用實際輸入中未使用的元件編號。透過使用與實際輸入元件相同的編號，實際輸入的資料將被由PHASECHG指令指定的輸入X所覆蓋。

出錯

沒有運算出錯。

執行階段的結束

PHASEEND



• R00CPU、R01CPU、R02CPU可以在韌體版本為“24”及其以後的版本使用，RnCPU (R00CPU, R01CPU, R02CPU除外)、RnENCPU可以在韌體版本為“57”及其以後的版本使用。工程工具可以在版本為“1.075D”及其以後的版本使用。

結束執行中的階段。

梯形圖	ST
	不對應。

FBD/LD
不對應。

■執行條件

指令	執行條件
PHASEEND	

功能

- 結束執行中的階段 (將之前的PHASE指令中指定的元件置為OFF*1)。
- *1 PHASE指令連接的階段處理的非執行處理在本指令執行後的下一個掃描時執行。

注意事項

- 應在階段處理內執行PHASEEND指令。在階段處理外執行時，將最後執行ON的PHASECHG指令的制定元件置為OFF*1。
- *1 不包括PHASE指令執行ON之後到PHASEEND指令執行為止，進行了RUN中的梯形圖塊更改/RUN中的可程式控制器的寫入 (順控程式、FB檔案、安全全局標籤設定檔案中任意一項的寫入) 的情況。

出錯

沒有運算出錯。

5.7 結束指令

主程式結束

FEND

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

用於在程式檔案內將主程式與子程式、中斷程式分開。

梯形圖	ST
	不對應。

FBD/LD

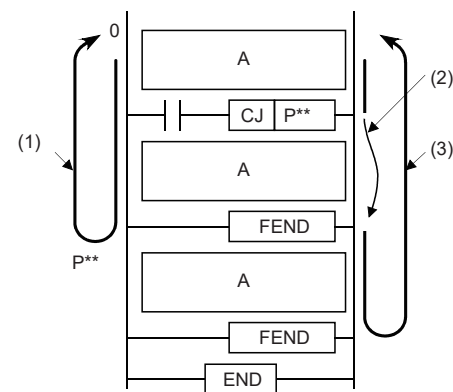
不對應。

■執行條件

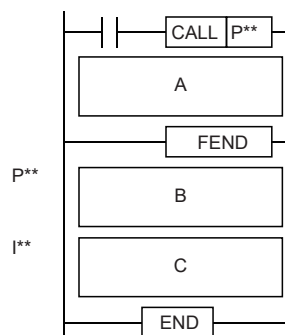
指令	執行條件
FEND	常時執行

功能

- 對於FEND指令，用於透過CJ指令等將順控程式的運算分支，以及將主程式子程式化的情況下，透過中斷指針(I)的中斷程式分開時使用。
- 執行FEND指令時，CPU模組將結束執行的程式。
- FEND指令以後的順控程式也可透過工程工具梯形圖顯示。使用CJ指令時



有子程式、中斷程式時



A: 主程式
B: 子程式
C: 中斷程式

- (1) 不執行CJ指令時的運算
- (2) 透過CJ指令進行跳轉
- (3) 執行了CJ指令時的運算

出錯

出錯代碼(SD0)	內容
3340H	在執行FOR指令後、執行NEXT指令之前，執行FEND指令時。
3381H	在執行CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令後、執行RET指令之前，執行FEND指令時。
33A1H	在中斷指針(I)的中斷程式中、執行IRET指令之前，執行了FEND指令時。

順控程式結束

END



表示程式的最終。

梯形圖	ST
	不對應。

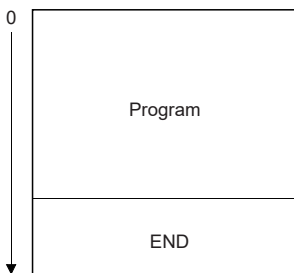
FBD/LD
不對應。

執行條件

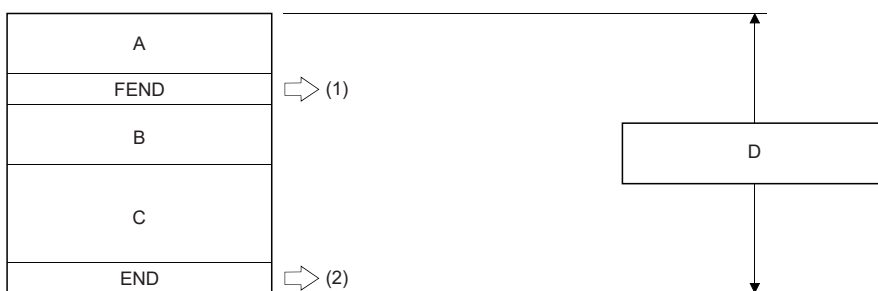
指令	執行條件
END	常時執行

功能

- 表示包含了主程式、子程式、中斷程式之程式的最終。
- 執行END指令時，CPU模組將結束正在執行的程式。



- 在程式的過程中需要END處理的情況下，應使用FEND指令。
- 在透過工程工具的梯形圖編輯模式進行程式的情況下，END指令將被自動輸入，不能進行編輯。
- 在存有主程式、子程式、中斷程式情況的結束指令的使用方法如下所示。



- A: 主程式
- B: 子程式
- C: 中斷程式
- D: 主順控程式區域
- (1): FEND指令是必要的。
- (2): END指令是必要的。

要點

對於將程式分開為多個程式塊情況下的END指令，表示程式塊的結束。

對於作為END處理執行的END指令，將變為程式設定中登錄的程式的最後執行的程式。

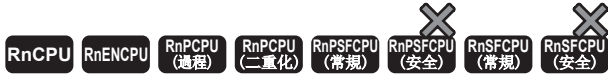
出錯

出錯代碼 (SD0)	內容
3340H	在執行FOR指令後、執行NEXT指令之前，執行END指令時。
3381H	在執行CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令後、執行RET指令之前，執行END指令時。
33A1H	在中斷指針(I)的中斷程式中、執行IRET指令之前，執行END指令時。

5.8 停止指令

順控程式停止

STOP



停止CPU模組的運算。(與將開關設為STOP側的情況相同。)

梯形圖	ST
	ENO:=STOP (EN);

FBD/LD

■執行條件

指令	執行條件
STOP	

功能

- 執行指令為ON時，重設輸出(Y)後，停止CPU模組的運算。(與將開關設為STOP側的情況相同。)
- 執行STOP指令後，重啟CPU模組運算時，將開關恢復為RUN→STOP後，再度置於RUN的位置。

出錯

出錯代碼(SD0)	內容
3340H	在執行FOR指令後、執行NEXT指令前，執行STOP指令時。
3381H	在執行CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令後、執行RET指令前，執行STOP指令時。
33A1H	在中斷指針(I)的中斷程式中、執行IRET指令前，執行STOP指令時。
33A3H	恆定週期執行類型程式中執行STOP指令時。
33A4H	事件執行類型程式中執行了STOP指令時。

5.9 無處理指令

無處理(NOP)

NOP

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

在設定程式的偵錯用空間等情況下使用。

梯形圖	ST
—	不對應。

FBD/LD

不對應。

■執行條件


指令	執行條件
NOP	常時執行

功能

- 透過無處理指令，對至當時為止的運算不會產生任何影響。
- NOP指令用於下述情況。
 - 設定程式的偵錯用空間。
 - 在不改變步數的狀況下刪除指令。(改寫為NOP指令。)
 - 暫時刪除指令。

要點

關於NOP指令的插入/刪除，請參閱下述手冊。

 GX Works3 操作手冊

出錯

沒有運算出錯。

無處理 (NOPLF)

NOPLF

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

透過無處理指令，對至當時為止的運算不會產生任何影響。

梯形圖	ST
	不對應。

FBD/LD

不對應。

■執行條件

指令	執行條件
NOPLF	常時執行

5

功能

- 透過無處理指令，對至當時為止的運算不會產生任何影響。

出錯

沒有運算出錯。

第4部分 基本指令

本部分由下述章節構成。

6 基本指令

6 基本指令

6.1 比較運算指令

BIN16位元資料比較

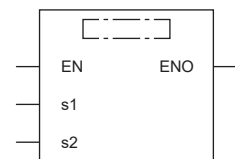
LD□(_U)、AND□(_U)、OR□(_U)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將指定的2個BIN16位元資料透過常開觸點處理進行比較。

梯形圖	ST*1
<p>(□中放入=(_U)、<(_U)、>(_U)、<=(_U)、<(_U)、>=(_U)。)</p>	<pre> ENO:=LD_□(EN, s1, s2); ENO:=AND_□(EN, s1, s2); ENO:=OR_□(EN, s1, s2); </pre> <pre> ENO:=LD_□_U(EN, s1, s2); ENO:=AND_□_U(EN, s1, s2); ENO:=OR_□_U(EN, s1, s2); </pre> <p>(□中放入EQ、NE、GT、LE、LT、GE。)*2</p>

FBD/LD



(□中放入LD_、AND_、OR_與EQ(_U)、NE(_U)、GT(_U)、LE(_U)、LT(_U)、GE(_U)的組合。)*2

*1 在版本為“1.035M”及其以後的工程工具中對應。
*2 EQ為=、NE為<>、GT為>、LE為<=、LT為<、GE為>=。

■執行條件

指令	執行條件
LD□(_U)、AND□(_U)、OR□(_U)	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	LD□、AND□、OR□	比較資料或儲存了比較資料的元件	-32768~32767	帶符號BIN16位元	ANY16_S
	LD□_U、AND□_U、OR□_U		0~65535	無符號BIN16位元	ANY16_U
(s2)	LD□、AND□、OR□	比較資料或儲存了比較資料的元件	-32768~32767	帶符號BIN16位元	ANY16_S
	LD□_U、AND□_U、OR□_U		0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○

功能

- 將(s1)中指定的元件的BIN16位元資料與(s2)中指定的元件的BIN16位元，透過常開觸點處理進行比較運算。
- 各指令的比較運算結果如下所示。

指令符號(梯形圖、FBD/LD)	條件	比較運算結果
=(_U)、EQ(_U)	(s1)=(s2)	導通狀態(ENO為ON)
<>(_U)、NE(_U)	(s1)≠(s2)	
>(_U)、GT(_U)	(s1)>(s2)	
<=(_U)、LE(_U)	(s1)≤(s2)	
<(_U)、LT(_U)	(s1)<(s2)	
>=(_U)、GE(_U)	(s1)≥(s2)	
=(_U)、EQ(_U)	(s1)≠(s2)	非導通狀態(ENO為OFF)
<>(_U)、NE(_U)	(s1)=(s2)	
>(_U)、GT(_U)	(s1)≤(s2)	
<=(_U)、LE(_U)	(s1)>(s2)	
<(_U)、LT(_U)	(s1)≥(s2)	
>=(_U)、GE(_U)	(s1)<(s2)	

- 透過LD□、AND□、OR□指令，在(s1)、(s2)中指定16進制數常數的情況下，如果指定最高位(b15)為1的數值(8~F)，將被視為BIN值的負數，進行比較運算。
- 在FBD/LD中使用了LD□指令時，應指定EN中的左母線或常時ON的變數組件/常數組件。
- 將OR□指令透過FBD/LD使用的情況下，將EN指定為TRUE時ENO將ON。EN不成為執行條件。

出錯

沒有運算出錯。

BIN32位元資料比較

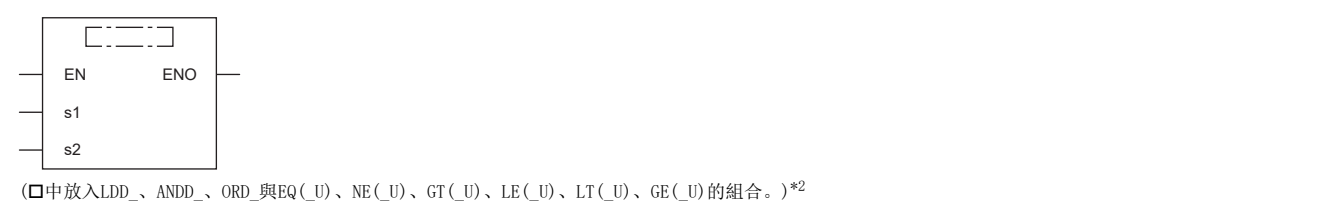
LDD□(_U)、ANDD□(_U)、ORD□(_U)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將指定的2個BIN32位元資料透過常開觸點處理進行比較。

梯形圖	ST*1
<p>(□中放入D=(_U)、D>(_U)、D>(_U)、D<(_U)、D<(_U)、D=(_U)。)</p>	<p>ENO:=LDD_□(EN, s1, s2); ENO:=ANDD_□(EN, s1, s2); ENO:=ORD_□(EN, s1, s2);</p> <p>ENO:=LDD_□_U(EN, s1, s2); ENO:=ANDD_□_U(EN, s1, s2); ENO:=ORD_□_U(EN, s1, s2);</p> <p>(□中放入EQ、NE、GT、LE、LT、GE。)*2</p>

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。
*2 EQ為=、NE為<>、GT為>、LE為<、LT為<、GE為>=。

執行條件

指令	執行條件
LDD□(_U)、ANDD□(_U)、ORD□(_U)	常時執行

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	LDD□、ANDD□、ORD□	比較資料或儲存了比較資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	LDD□_U、ANDD□_U、ORD□_U		0~4294967295	無符號BIN32位元	ANY32_U
(s2)	LDD□、ANDD□、ORD□	比較資料或儲存了比較資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	LDD□_U、ANDD□_U、ORD□_U		0~4294967295	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○

功能

- 將(s1)中指定的元件的BIN32位元資料與(s2)中指定的元件的BIN32位元資料透過常開觸點處理進行比較運算。
- 各指令的比較運算結果如下所示。

指令符號(梯形圖、FBD/LD)	條件	比較運算結果
D=(<u> </u>)、EQ(<u> </u>)	(s1)=(s2)	導通狀態(ENO為ON)
D<>(<u> </u>)、NE(<u> </u>)	(s1)≠(s2)	
D>(<u> </u>)、GT(<u> </u>)	(s1)>(s2)	
D<=(<u> </u>)、LE(<u> </u>)	(s1)≤(s2)	
D<(<u> </u>)、LT(<u> </u>)	(s1)<(s2)	
D>=(<u> </u>)、GE(<u> </u>)	(s1)≥(s2)	
D=(<u> </u>)、EQ(<u> </u>)	(s1)≠(s2)	非導通狀態(ENO為OFF)
D<>(<u> </u>)、NE(<u> </u>)	(s1)=(s2)	
D>(<u> </u>)、GT(<u> </u>)	(s1)≤(s2)	
D<=(<u> </u>)、LE(<u> </u>)	(s1)>(s2)	
D<(<u> </u>)、LT(<u> </u>)	(s1)≥(s2)	
D>=(<u> </u>)、GE(<u> </u>)	(s1)<(s2)	

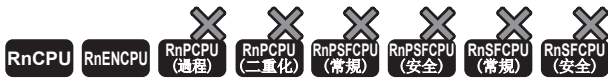
- 經由LDD□、ANDD□、ORD□指令，在(s1)、(s2)中指定了16進制數的常數的情況下，如果指定最高位位元(b31)為1的數值(8～F)，將被視為BIN值的負數，進行比較運算。
- 比較中使用的資料應透過處理32位元資料的指令(DMOV(P)指令等)指定。透過處理16位元資料的指令(MOV(P)指令等)指定的情況下，將無法正常進行大小比較。
- 透過FBD/LD使用LDD□指令的情況下，應指定EN中的左母線或常時ON的變數組件/常數組件。
- 透過FBD/LD使用ORD□指令的情況下，將EN指定為TRUE時ENO將ON。EN不成為執行條件。

出錯

沒有運算出錯。

BIN16位元資料比較輸出

CMP (P) (_U)

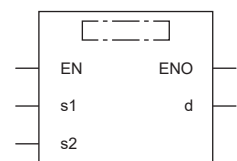


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

比較 (s1) 中指定的BIN16位元資料與 (s2) 中指定的BIN16位元資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為ON。

梯形圖	ST	
	ENO:=CMP (EN, s1, s2, d) ; ENO:=CMPP (EN, s1, s2, d) ;	ENO:=CMP_U (EN, s1, s2, d) ; ENO:=CMPP_U (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
CMP CMP_U	
CMPP CMPP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	CMP (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	CMP (P)_U	0~65535	無符號BIN16位元	ANY16_U
(s2)	CMP (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	CMP (P)_U	0~65535	無符號BIN16位元	ANY16_U
(d)	儲存了比較結果的起始元件	—	位元	ANYBIT_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

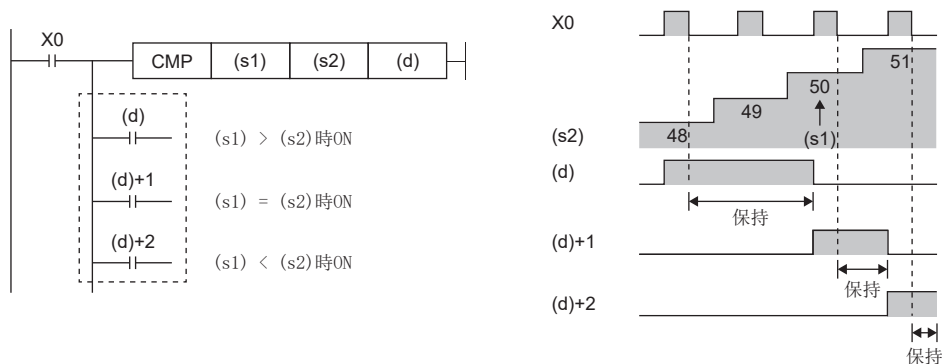
■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○*1	○	○	○	○	—	—	○	○	—	—	—
(s2)	○*1	○	○	○	○	—	—	○	○	—	—	—
(d)	○	—	○*2	—	—	—	—	○	—	—	—	—

*1 不能使用FX、FY。
 *2 不能使用T、ST、C。

功能

- 比較(s1)中指定的BIN16位元資料與(s2)中指定的BIN16位元資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為ON。

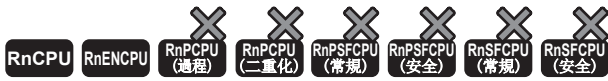


出錯

沒有運算出錯。

BIN32位元資料比較輸出

DCMP (P) (_U)

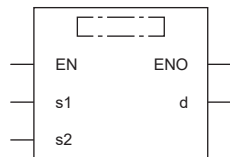


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

比較(s1)中指定的BIN32位元資料與(s2)中指定的BIN32位元資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為ON。

梯形圖	ST	
	ENO:=DCMP (EN, s1, s2, d) ; ENO:=DCMPP (EN, s1, s2, d) ;	ENO:=DCMP_U (EN, s1, s2, d) ; ENO:=DCMPP_U (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
DCMP DCMP_U	
DCMPP DCMPP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	DCMP (P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DCMP (P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(s2)	DCMP (P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DCMP (P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(d)	儲存了比較結果的起始元件	—	位元	ANYBIT_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

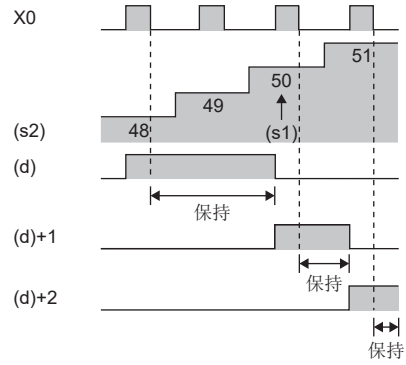
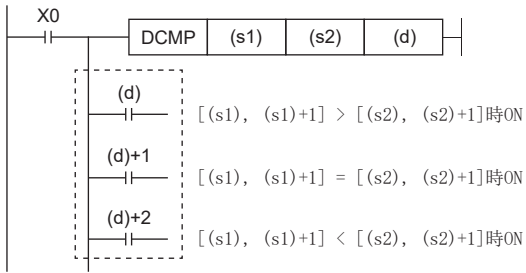
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○*1	○	○	○	○	○	○	○	○	—	—	—	
(s2)	○*1	○	○	○	○	○	○	○	○	—	—	—	
(d)	○	—	○*2	—	—	—	—	○	—	—	—	—	

*1 不能使用FX、FY。

*2 不能使用T、ST、C。

功能

- 比較(s1)中指定的BIN32位元資料與(s2)中指定的BIN32位元資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為0N。

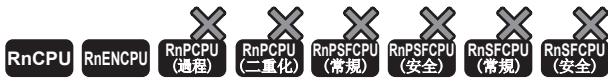


出錯

沒有運算出錯。

BIN16位元資料區域比較輸出

ZCP(P) (_U)



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

對於下限值(s1)的BIN16位元資料與上限值(s2)的BIN16位元資料的區域，透過比較資料(s3)的BIN16位元資料進行比較，根據結果(下、區域內、上)將(d)、(d)+1、(d)+2中的任意一個設定為ON狀態。

梯形圖	ST	
	ENO:=ZCP(EN, s1, s2, s3, d); ENO:=ZCPP(EN, s1, s2, s3, d);	ENO:=ZCP_U(EN, s1, s2, s3, d); ENO:=ZCPP_U(EN, s1, s2, s3, d);

FBD/LD

■執行條件

指令	執行條件
ZCP ZCP_U	
ZCPP ZCPP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	ZCP(P)	-32768~32767	帶符號BIN16位元	ANY16_S
	ZCP(P)_U	0~65535	無符號BIN16位元	ANY16_U
(s2)	ZCP(P)	-32768~32767	帶符號BIN16位元	ANY16_S
	ZCP(P)_U	0~65535	無符號BIN16位元	ANY16_U
(s3)	ZCP(P)	-32768~32767	帶符號BIN16位元	ANY16_S
	ZCP(P)_U	0~65535	無符號BIN16位元	ANY16_U
(d)	儲存了比較結果的起始元件	—	位元	ANYBIT_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

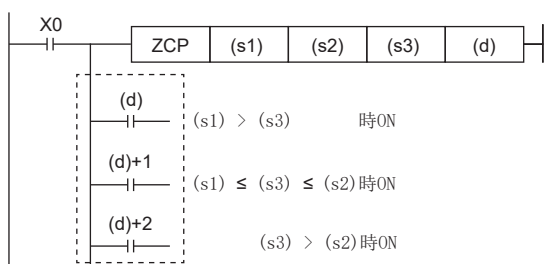
操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s1)	○*1	○	○	○	○	—	—	○	○	—	—	—
(s2)	○*1	○	○	○	○	—	—	○	○	—	—	—
(s3)	○*1	○	○	○	○	—	—	○	○	—	—	—
(d)	○	—	○*2	—	—	—	—	○	—	—	—	—

*1 不能使用FX、FY。

*2 不能使用T、ST、C。

功能

- 對於下限值(s1)的BIN16位元資料與上限值(s2)的BIN16位元資料的區域，透過比較資料(s3)的BIN16位元資料進行比較，根據結果(下、區域內、上)將(d)、(d)+1、(d)+2中的任意一個設定為ON狀態。



注意事項

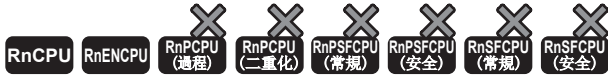
- 應將下限值(s1)設定成小於上限值(s2)的值。下限值(s1)大於上限值(s2)的值時，將上限值(s2)認作與下限值(s1)相同的值進行處理。

出錯

沒有運算出錯。

BIN32位元資料區域比較輸出

DZCP(P) (_U)



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

對於下限值(s1)的BIN32位元資料與上限值(s2)的BIN32位元資料的區域，透過比較資料(s3)的BIN32位元資料進行比較，根據結果(下、區域內、上)，將(d)、(d)+1、(d)+2中的任意一個設定為ON狀態。

梯形圖	ST	
	ENO:=DZCP(EN, s1, s2, s3, d); ENO:=DZCPP(EN, s1, s2, s3, d);	ENO:=DZCP_U(EN, s1, s2, s3, d); ENO:=DZCPP_U(EN, s1, s2, s3, d);

FBD/LD

■執行條件

指令	執行條件
DZCP DZCP_U	
DZCPP DZCPP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	DZCP(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DZCP(P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(s2)	DZCP(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DZCP(P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(s3)	DZCP(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DZCP(P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(d)	儲存了比較結果的起始元件	—	位元	ANYBIT_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

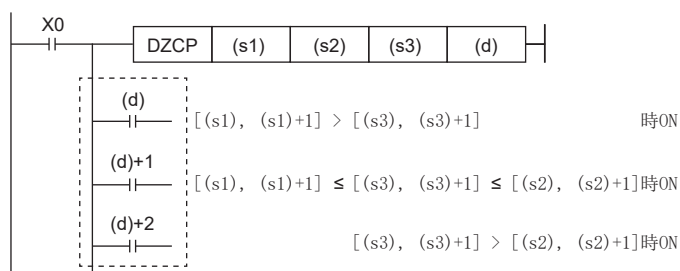
操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○*1	○	○	○	○	○	○	○	○	—	—	—
(s2)	○*1	○	○	○	○	○	○	○	○	—	—	—
(s3)	○*1	○	○	○	○	○	○	○	○	—	—	—
(d)	○	—	○*2	—	—	—	—	○	—	—	—	—

*1 不能使用FX、FY。

*2 不能使用T、ST、C。

功能

- 對於下限值(s1)的BIN32位元資料與上限值(s2)的BIN32位元資料的區域，透過比較資料(s3)的BIN32位元資料進行比較，根據結果(下、區域內、上)，將(d)、(d)+1、(d)+2中的任意一個設定為ON狀態。



注意事項

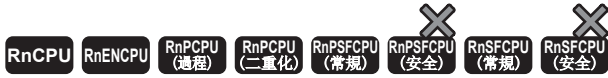
- 應將下限值(s1)設定成小於上限值(s2)的值。下限值(s1)大於上限值(s2)的值時，將上限值(s2)認作與下限值(s1)相同的值進行處理。

出錯

沒有運算出錯。

BIN16位元塊資料比較

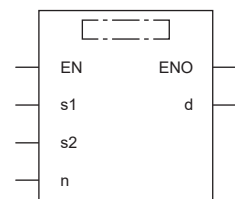
BKCOMP□(P) (_U)



對指定的2個BIN16位元塊資料進行比較。

梯形圖	ST*1
<p>(□中放入BKCOMP=(P) (_U)、BKCOMP<>(P) (_U)、BKCOMP>(P) (_U)、BKCOMP<=(P) (_U)、BKCOMP<(P) (_U)、BKCOMP>=(P) (_U)。))</p>	<p>ENO:=BKCOMP_□(EN, s1, s2, n, d); ENO:=BKCOMP_□P(EN, s1, s2, n, d);</p> <p>ENO:=BKCOMP_□_U(EN, s1, s2, n, d); ENO:=BKCOMP_□P_U(EN, s1, s2, n, d);</p> <p>(□中放入EQ、NE、GT、LE、LT、GE。)*2</p>

FBD/LD



(□中放入BKCOMP_與EQ(P) (_U)、NE(P) (_U)、GT(P) (_U)、LE(P) (_U)、LT(P) (_U)、GE(P) (_U)的組合。)*2

*1 在版本為“1.035M”及其以後的工程工具中對應。
*2 EQ為=、NE為<>、GT為>、LE為<=、LT為<、GE為>=。

■執行條件

指令	執行條件
BKCOMP□(_U)	
BKCOMP□P(_U)	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	BKCOMP□(P)	比較資料或儲存了比較資料的起始元件	-32768~32767	帶符號BIN16位元	ANY16_S*1
	BKCOMP□(P)_U		0~65535	無符號BIN16位元	ANY16_U*1
(s2)	BKCOMP□(P)	儲存了比較資料的起始元件	—	帶符號BIN16位元	ANY16_S*1
	BKCOMP□(P)_U			無符號BIN16位元	ANY16_U*1
(d)	儲存比較運算結果的起始元件	—	位元	ANY_BOOL*1	
(n)	比較的資料數	0~65535	無符號BIN16位元	ANY16	
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

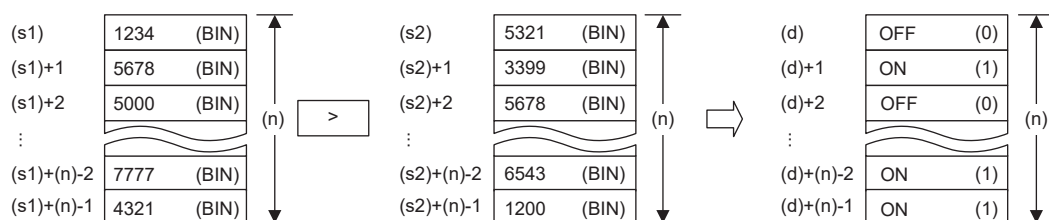
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

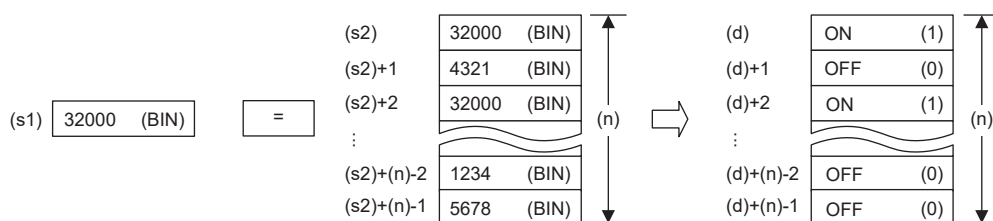
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	—	—	○	—	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—
(d)	○	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

功能

- 將(s1)中指定的元件開始的(n)點的BIN16位元資料與(s2)中指定的元件開始的(n)點的BIN16位元資料進行比較，將運算結果儲存到(d)中指定的元件及其以後。
- 在比較條件成立的情況下，(d)的相應元件將變為ON；比較條件不成立的情況下，(d)的相應元件將變為OFF。



- 比較運算是以16位元單位進行。
- (s1)可以指定常數。



- 各指令的比較運算結果如下所示。

指令符號(梯形圖、FBD/LD)	條件	比較運算結果
BKCMPE(P) (U)、BKCMPEQ(P) (U)	(s1) = (s2)	ON (1)
BKCMPC(P) (U)、BKCMNE(P) (U)	(s1) ≠ (s2)	
BKCMGP(P) (U)、BKCMGT(P) (U)	(s1) > (s2)	
BKCMLE(P) (U)、BKCMLE(P) (U)	(s1) ≤ (s2)	
BKCMCP(P) (U)、BKCMLT(P) (U)	(s1) < (s2)	
BKCMGE(P) (U)、BKCMGE(P) (U)	(s1) ≥ (s2)	
BKCMPE(P) (U)、BKCMPEQ(P) (U)	(s1) ≠ (s2)	OFF (0)
BKCMPC(P) (U)、BKCMNE(P) (U)	(s1) = (s2)	
BKCMGP(P) (U)、BKCMGT(P) (U)	(s1) ≤ (s2)	
BKCMLE(P) (U)、BKCMLE(P) (U)	(s1) > (s2)	
BKCMCP(P) (U)、BKCMLT(P) (U)	(s1) ≥ (s2)	
BKCMGE(P) (U)、BKCMGE(P) (U)	(s1) < (s2)	

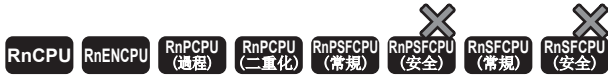
- 當(d)開始的(n)點中儲存的比較運算結果全部為ON(1)時，SM704(塊比較信號)將變為ON。

出錯

出錯代碼(SD0)	內容
2821H	當(s1)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍重複時。
	當(s2)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍重複時。

BIN32位元塊資料比較

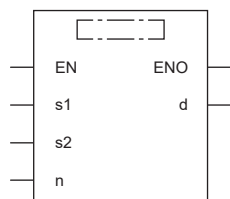
DBKCMP□(P) (_U)



對指定的2個BIN32位元塊資料進行比較。

梯形圖	ST*1				
<p>(□中放入DBKCMP=(P) (_U)、DBKCMP<>(P) (_U)、DBKCMP>(P) (_U)、DBKCMP<=(P) (_U)、DBKCMP<(P) (_U)、DBKCMP>=(P) (_U)。) </p>	<table border="1"> <tr> <td>ENO:=DBKCMP_□(EN, s1, s2, n, d) ;</td> <td>ENO:=DBKCMP_□_U(EN, s1, s2, n, d) ;</td> </tr> <tr> <td>ENO:=DBKCMP_□P(EN, s1, s2, n, d) ;</td> <td>ENO:=DBKCMP_□P_U(EN, s1, s2, n, d) ;</td> </tr> </table> <p>(□中放入EQ、NE、GT、LE、LT、GE。)*2</p>	ENO:=DBKCMP_□(EN, s1, s2, n, d) ;	ENO:=DBKCMP_□_U(EN, s1, s2, n, d) ;	ENO:=DBKCMP_□P(EN, s1, s2, n, d) ;	ENO:=DBKCMP_□P_U(EN, s1, s2, n, d) ;
ENO:=DBKCMP_□(EN, s1, s2, n, d) ;	ENO:=DBKCMP_□_U(EN, s1, s2, n, d) ;				
ENO:=DBKCMP_□P(EN, s1, s2, n, d) ;	ENO:=DBKCMP_□P_U(EN, s1, s2, n, d) ;				

FBD/LD



(□中放入DBKCMP_與EQ(P) (_U)、NE(P) (_U)、GT(P) (_U)、LE(P) (_U)、LT(P) (_U)、GE(P) (_U)的組合。)*2

*1 在版本為“1.035M”及其以後的工程工具中對應。
 *2 EQ為=、NE為<>、GT為>、LE為<=、LT為<、GE為>=。

■執行條件

指令	執行條件
DBKCMP□(_U)	
DBKCMP□P(_U)	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	DBKCMP□(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S*1
	DBKCMP□(P)_U	0~4294967295	無符號BIN32位元	ANY32_U*1
(s2)	DBKCMP□(P)	—	帶符號BIN32位元	ANY32_S*1
	DBKCMP□(P)_U		無符號BIN32位元	ANY32_U*1
(d)	儲存比較運算結果的起始元件	—	位元	ANY_BOOL*1
(n)	比較的資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

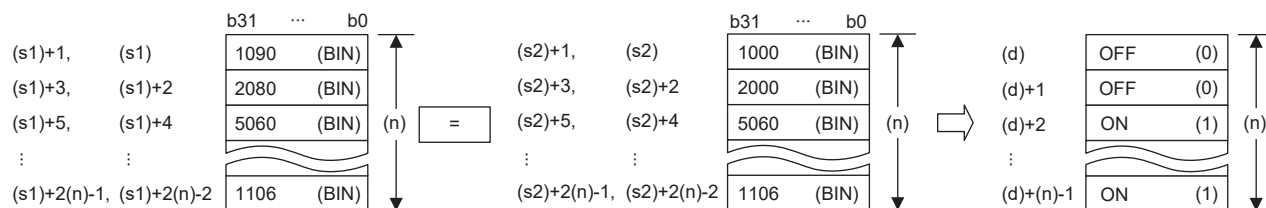
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

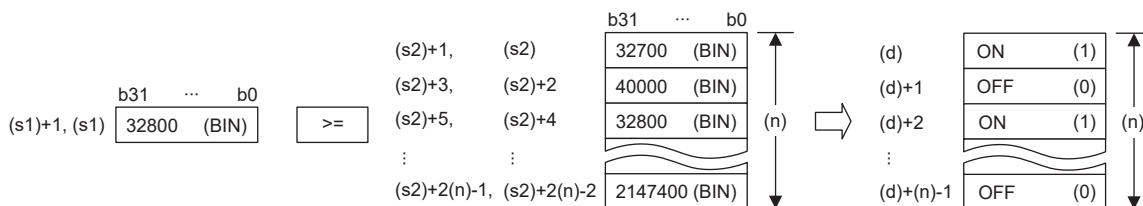
操作數	位元		字元				雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\(\H)G□	Z	LT、LST、LC	LZ	K、H		E	\$		
(s1)	—	—	○	—	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	○	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 將(s1)中指定的元件開始的(n)點的BIN32位元資料與(s2)中指定的元件開始的(n)點的BIN32位元資料進行比較，將運算結果儲存到(d)中指定的元件及其以後。
- 在比較條件成立的情況下，(d)的相應元件將變為ON；比較條件不成立的情況下，(d)的相應元件將變為OFF。



- 比較運算是以32位元單位進行。
- (s1)可以指定常數。



- (d)是在(s1)開始的(n)點的元件範圍及(s2)開始的(n)點的元件範圍以外進行指定。
- 各指令的比較運算結果如下所示。

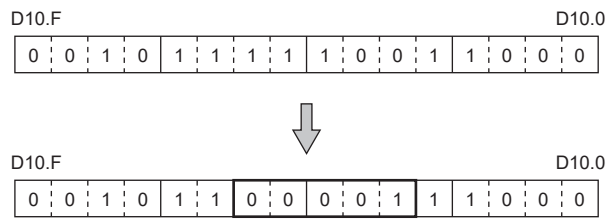
指令符號(梯形圖、FBD/LD)	條件	比較運算結果
DBKCOMP=(P) (U)、DBKCOMP_EQ(P) (U)	(s1)=(s2)	ON(1)
DBKCOMP<>(P) (U)、DBKCOMP_NE(P) (U)	(s1)≠(s2)	
DBKCOMP>(P) (U)、DBKCOMP_GT(P) (U)	(s1)>(s2)	
DBKCOMP<=(P) (U)、DBKCOMP_LE(P) (U)	(s1)≤(s2)	
DBKCOMP<(P) (U)、DBKCOMP_LT(P) (U)	(s1)<(s2)	
DBKCOMP>=(P) (U)、DBKCOMP_GE(P) (U)	(s1)≥(s2)	OFF(0)
DBKCOMP=(P) (U)、DBKCOMP_EQ(P) (U)	(s1)≠(s2)	
DBKCOMP<>(P) (U)、DBKCOMP_NE(P) (U)	(s1)=(s2)	
DBKCOMP>(P) (U)、DBKCOMP_GT(P) (U)	(s1)≤(s2)	
DBKCOMP<=(P) (U)、DBKCOMP_LE(P) (U)	(s1)>(s2)	
DBKCOMP<(P) (U)、DBKCOMP_LT(P) (U)	(s1)≥(s2)	
DBKCOMP>=(P) (U)、DBKCOMP_GE(P) (U)	(s1)<(s2)	

- 當(d)開始的(n)點中儲存的比較運算結果全部為ON(1)時，SM704(塊比較信號)將變為ON。
- 在(n)中指定的值為0的情況下，將變為無處理。

出錯

出錯代碼(SD0)	內容
2821H	當(s1)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍重複時。
	當(s2)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍重複時。

在進行字元件的位元指定的情況下，儲存運算結果的位元指定元件以外不變化。



6.2 算術運算指令

BIN16位元加法運算

+ (P) (_U) [操作數為2個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN16位元資料進行加法運算。

梯形圖	ST
	不對應。 (☞ 231頁 + (P) (_U) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 231頁 + (P) (_U) [操作數為3個的情況下])

■執行條件

指令	執行條件
+ +_U	
+P +P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	+ (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	+ (P)_U	0~65535	無符號BIN16位元	ANY16_U
(d)	+ (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	+ (P)_U	0~65535	無符號BIN16位元	ANY16_U

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

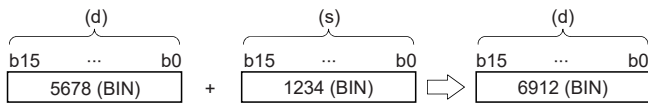
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

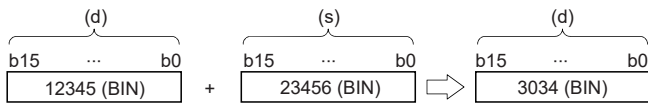
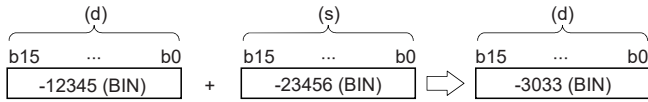
操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

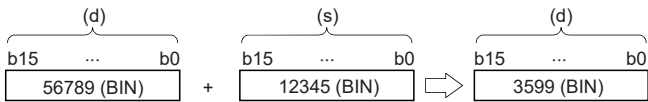
- 將(d)中指定的BIN16位元資料與(s)中指定的BIN16位元資料進行加法運算，將加法運算結果儲存到(d)中指定的元件中。



- 在加法運算結果中發生了上溢的情況下，位數上升將被忽略。在此情況下，SM700(進位標誌)不變為ON。
[+(P)的情況下]



[+(P)_U的情況下]



出錯

沒有運算出錯。

+ (P) (_U) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN16位元資料進行加法運算。

梯形圖	ST
	ENO:=PLUS (EN, s1, s2, d) ; ENO:=PLUSP (EN, s1, s2, d) ; ENO:=PLUS_U (EN, s1, s2, d) ; ENO:=PLUSP_U (EN, s1, s2, d) ;

FBD/LD
(□中放入PLUS、PLUSP、PLUS_U、PLUSP_U。)

■執行條件

指令	執行條件
+ +_U	
+P +P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)	
(s1)	+ (P)	被加法運算資料或儲存了被加法運算資料的元件	-32768~32767	帶符號BIN16位元	ANY16_S
	+ (P)_U		0~65535	無符號BIN16位元	ANY16_U
(s2)	+ (P)	加法運算資料或儲存了加法運算資料的元件	-32768~32767	帶符號BIN16位元	ANY16_S
	+ (P)_U		0~65535	無符號BIN16位元	ANY16_U
(d)	+ (P)	儲存運算結果的元件	—	帶符號BIN16位元	ANY16_S
	+ (P)_U			無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

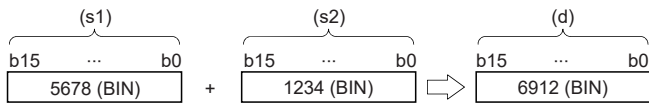
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

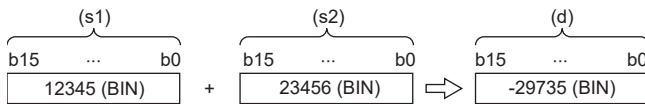
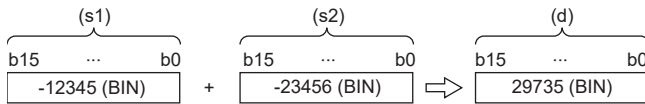
操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

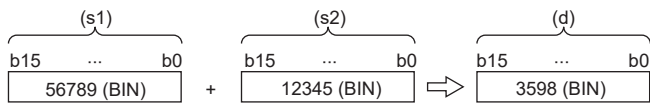
- 將 (s1) 中指定的BIN16位元資料與 (s2) 中指定的BIN16位元資料進行加法運算，將加法運算結果儲存到 (d) 中指定的元件中。



- 在加法運算結果中發生了上溢的情況下，位數上升將被忽略。在此情況下，SM700(進位標誌)不變為ON。
[+(P)的情況下]



[+(P)_U的情況下]



出錯

沒有運算出錯。

BIN16位元減法運算

-(P) (_U) [操作數為2個的情況下]

RnCPU RnENCPU RnPCPU (通稱) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN16位元資料進行減法運算。

梯形圖	ST
	不對應。 (☞ 235頁 -(P) (_U) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 235頁 -(P) (_U) [操作數為3個的情況下])

■執行條件

指令	執行條件
- -_U	
-P -P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	-(P)	-32768~32767	帶符號BIN16位元	ANY16_S
	-(P)_U	0~65535	無符號BIN16位元	ANY16_U
(d)	-(P)	-32768~32767	帶符號BIN16位元	ANY16_S
	-(P)_U	0~65535	無符號BIN16位元	ANY16_U

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

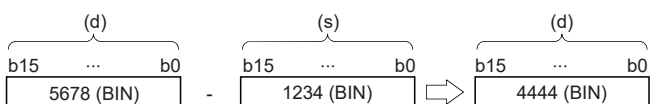
操作數	位元		字元				雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ	K、H		E	\$			
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—	

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

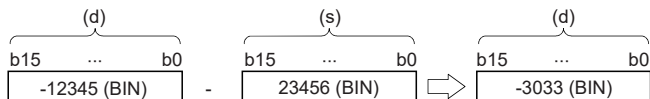
操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

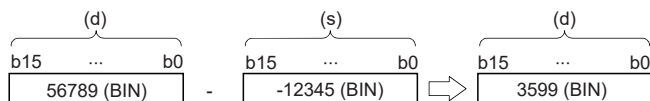
- 將 (d) 中指定的 BIN16 位元資料與 (s) 中指定的 BIN16 位元資料進行減法運算，將減法運算結果儲存到 (d) 中指定的元件中。



- 在減法運算結果中發生了下溢的情況下，位數下降將被忽略。在此情況下，SM700 (進位標誌) 不變為 0N。
[-(P) 的情況下]



[-(P) _U 的情況下]



出錯

沒有運算出錯。

-(P) (_U) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN16位元資料進行減法運算。

梯形圖	ST
	ENO:=MINUS (EN, s1, s2, d) ; ENO:=MINUSP (EN, s1, s2, d) ; ENO:=MINUS_U (EN, s1, s2, d) ; ENO:=MINUSP_U (EN, s1, s2, d) ;

FBD/LD
(□中放入MINUS、MINUSP、MINUS_U、MINUSP_U。)

■執行條件

指令	執行條件
- -_U	
-P -P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	-(P)	被減法運算資料或儲存了被減法運算資料的元件	-32768~32767	帶符號BIN16位元	ANY16_S
	-(P)_U		0~65535	無符號BIN16位元	ANY16_U
(s2)	-(P)	減法運算資料或儲存了減法運算資料的元件	-32768~32767	帶符號BIN16位元	ANY16_S
	-(P)_U		0~65535	無符號BIN16位元	ANY16_U
(d)	-(P)	儲存運算結果的元件	—	帶符號BIN16位元	ANY16_S
	-(P)_U			無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元		字元		常數	
	SA\X、SA\Y、SA\M、SA\SM、SA\B		SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD		K、H	
(s1)	○		○		○	
(s2)	○		○		○	
(d)	○		○		—	

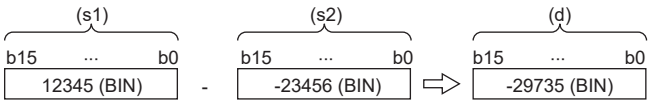
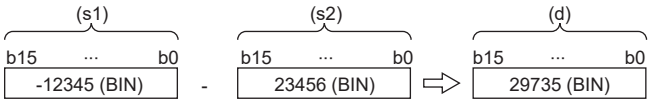
功能

- 將(s1)中指定的BIN16位元資料與(s2)中指定的BIN16位元資料進行減法運算，將減法運算結果儲存到(d)中指定的元件中。



- 在減法運算結果中發生了下溢的情況下，位數下降將被忽略。在此情況下，SM700(進位標誌)不變為ON。

[-(P)的情況下]



[-(P)_U的情況下]



出錯

沒有運算出錯。

BIN32位元加法運算

D+(P) (_U) [操作數為2個的情況下]

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN32位元資料進行加法運算。

梯形圖	ST
	不對應。 (☞ 239頁 D+(P) (_U) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 239頁 D+(P) (_U) [操作數為3個的情況下])

■執行條件

指令	執行條件
D+ D+_U	
D+P D+P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	D+(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	D+(P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(d)	D+(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	D+(P)_U	0~4294967295	無符號BIN32位元	ANY32_U

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

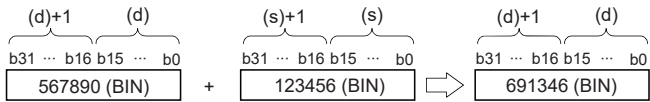
操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

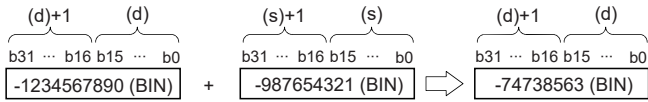
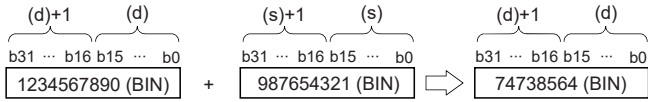
功能

- 將 (d) 中指定的 BIN32 位元資料與 (s) 中指定的 BIN32 位元資料進行加法運算，將加法運算結果儲存到 (d) 中指定的元件中。

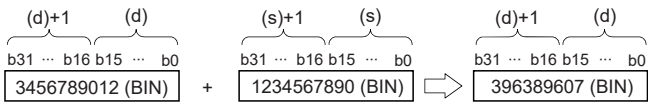


- 在加法運算結果中發生了上溢的情況下，位數上升將被忽略。在此情況下，SM700 (進位標誌) 不變為 ON。

[在 D+(P) 的情況下]



[在 D+(P)_U 的情況下]

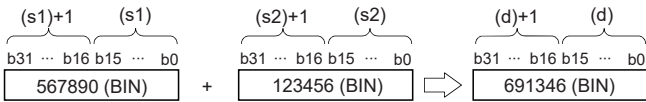


出錯

沒有運算出錯。

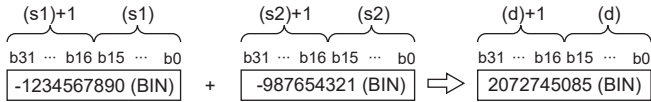
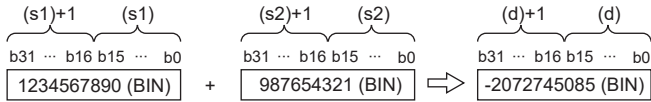
功能

- 將 (s1) 中指定的BIN32位元資料與 (s2) 中指定的BIN32位元資料進行加法運算，將加法運算結果儲存到 (d) 中指定的元件中。

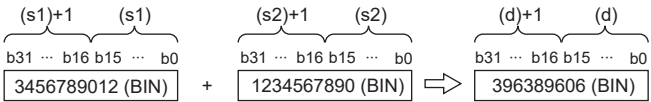


- 在加法運算結果中發生了上溢的情況下，位數上升將被忽略。在此情況下，SM700(進位標誌)不變為ON。

[在D+(P)的情況下]



[在D+(P)_U的情況下]



出錯

沒有運算出錯。

BIN32位元減法運算

D-(P) (_U) [操作數為2個的情況下]



對指定的2個BIN32位元資料進行減法運算。

梯形圖	ST
	不對應。 (☞ 243頁 D-(P) (_U) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 243頁 D-(P) (_U) [操作數為3個的情況下])

■執行條件

指令	執行條件
D- D-_U	
D-P D-P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	D-(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	D-(P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(d)	D-(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	D-(P)_U	0~4294967295	無符號BIN32位元	ANY32_U

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

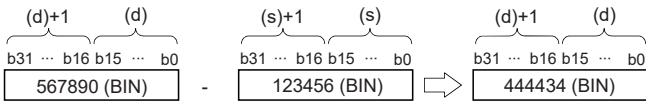
操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

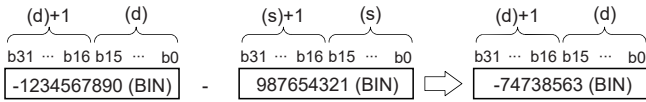
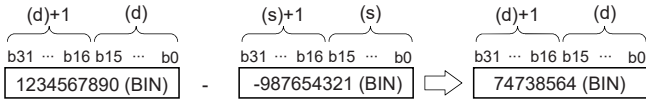
功能

- 將 (d) 中指定的 BIN32 位元資料與 (s) 中指定的 BIN32 位元資料進行減法運算，將減法運算結果儲存到 (d) 中指定的元件中。

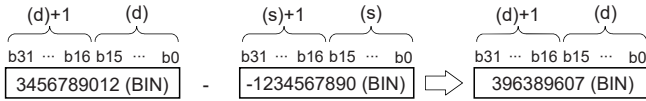


- 在減法運算結果中發生了下溢的情況下，位數上升將被忽略。在此情況下，SM700 (進位標誌) 不變為 0N。

[在 D-(P) 的情況下]



[在 D-(P)_U 的情況下]



出錯

沒有運算出錯。

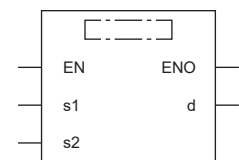
D-(P) (_U) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN32位元資料進行減法運算。

梯形圖	ST
	ENO:=DMINUS (EN, s1, s2, d) ; ENO:=DMINUSP (EN, s1, s2, d) ; ENO:=DMINUS_U (EN, s1, s2, d) ; ENO:=DMINUSP_U (EN, s1, s2, d) ;

FBD/LD



(□中放入DMINUS、DMINUSP、DMINUS_U、DMINUSP_U。)

■執行條件

指令	執行條件
D- D-_U	
D-P D-P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	D-(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	D-(P)_U		無符號BIN32位元	ANY32_U
(s2)	D-(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	D-(P)_U		無符號BIN32位元	ANY32_U
(d)	D-(P)	—	帶符號BIN32位元	ANY32_S
	D-(P)_U		無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

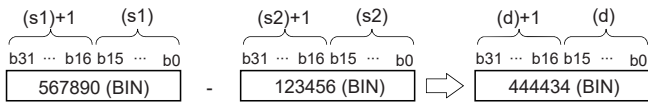
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

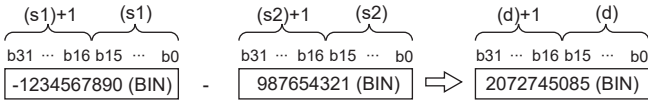
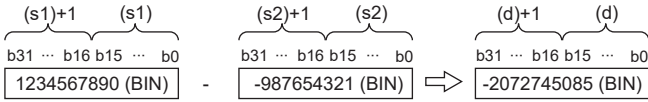
操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

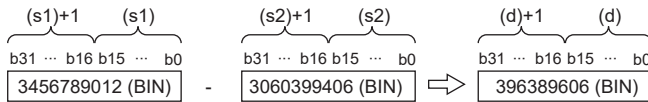
- 將(s1)中指定的BIN32位元資料與(s2)中指定的BIN32位元資料進行減法運算，將減法運算結果儲存到(d)中指定的元件中。



- 在減法運算結果中發生了下溢的情況下，位數上升將被忽略。在此情況下，SM700(進位標誌)不變為0N。
[在D-(P)的情況下]



[在D-(P)_U的情況下]



出錯

沒有運算出錯。

BIN16位元乘法運算

* (P) (_U)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN16位元資料進行乘法運算。

梯形圖	ST*1	
	ENO:=MULTI (EN, s1, s2, d) ; ENO:=MULTIP (EN, s1, s2, d) ;	ENO:=MULTI_U (EN, s1, s2, d) ; ENO:=MULTIP_U (EN, s1, s2, d) ;

FBD/LD



(□中放入MULTI、MULTIP、MULTI_U、MULTIP_U。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
* *_U	
*P *P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	* (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	* (P)_U	0~65535	無符號BIN16位元	ANY16_U
(s2)	* (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	* (P)_U	0~65535	無符號BIN16位元	ANY16_U
(d)	* (P)	—	帶符號BIN32位元	ANY32_S
	* (P)_U	—	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 將(s1)中指定的BIN16位元資料與(s2)中指定的BIN16位元資料進行乘法運算，將乘法運算結果儲存到(d)中指定的元件中。



- 在(d)為位元元件的情況下，指定將從低位的位元開始。

例

在(d)為位元元件情況下的乘法運算結果

- K1...低位的4位元 (b0~b3)
- K4...低位的16位元 (b0~b15)
- K8...低位的32位元 (b0~b31)

出錯

沒有運算出錯。

BIN16位元除法運算

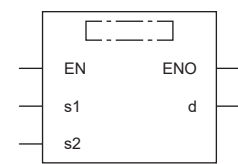
/ (P) (_U)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN16位元資料進行除法運算。

梯形圖	ST*1
	ENO:=DIVISION(EN, s1, s2, d); ENO:=DIVISIONP(EN, s1, s2, d); ENO:=DIVISION_U(EN, s1, s2, d); ENO:=DIVISIONP_U(EN, s1, s2, d);

FBD/LD



(□中放入DIVISION、DIVISIONP、DIVISION_U、DIVISIONP_U。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
/ /_U	
/P /P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	/ (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	/ (P)_U	0~65535	無符號BIN16位元	ANY16_U
(s2)	/ (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	/ (P)_U	0~65535	無符號BIN16位元	ANY16_U
(d)	/ (P)	—	帶符號BIN32位元	ANY16_S_ARRAY (要素數: 2)
	/ (P)_U	—	無符號BIN32位元	ANY16_U_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

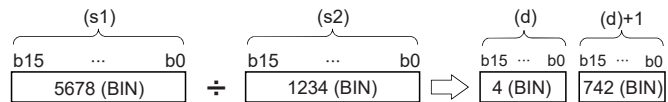
操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 將(s1)中指定的BIN16位元資料與(s2)中指定的BIN16位元資料進行除法運算，將除法運算結果儲存到(d)中指定的元件中。



(d): 商

(d)+1: 餘數

- 除法運算結果使用32位元儲存商及餘數。在位元元件的情況下，使用位指定中指定的位數，儲存商及餘數。
 - 商...被儲存到低位16位元中。
 - 餘數...被儲存到高位16位元中。

出錯

出錯代碼(SD0)	內容
3400H	當(s2)中指定的值(除數)為0時。

BIN32位元乘法運算

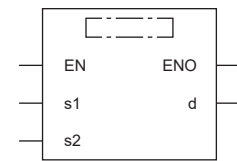
D*(P) (_U)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN32位元資料進行乘法運算。

梯形圖	ST*1	
	ENO:=DMULTI (EN, s1, s2, d); ENO:=DMULTIP (EN, s1, s2, d);	ENO:=DMULTI_U (EN, s1, s2, d); ENO:=DMULTIP_U (EN, s1, s2, d);

FBD/LD



(□中放入DMULTI、DMULTIP、DMULTI_U、DMULTIP_U。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
D* D*_U	
D*P D*P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	D*(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	D*(P)_U		無符號BIN32位元	ANY32_U
(s2)	D*(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	D*(P)_U		無符號BIN32位元	ANY32_U
(d)	D*(P)	—	帶符號BIN64位元	ANY32_S_ARRAY (要素數: 2)
	D*(P)_U		無符號BIN64位元	ANY32_U_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	—	○	—	—	○	—	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 將(s1)中指定的BIN32位元資料與(s2)中指定的BIN32位元資料進行乘法運算，將乘法運算結果儲存到(d)中指定的元件中。



- 在(d)為位元元件的情況下，乘法運算結果的低位32位元為止將成為對象，不能指定高位32位元。在位元元件中需要乘法運算結果的高位32位元資料的情況下，應預先將資料儲存到字元件中一次，將字元件的(d)+2、(d)+3的資料傳送到指定位元元件中。

例

在(d)為位元元件情況下的乘法運算結果

- K1---低位的4位元 (b0~b3)
- K4---低位的16位元 (b0~b15)
- K8---低位的32位元 (b0~b31)

出錯

沒有運算出錯。

BIN32位元除法運算

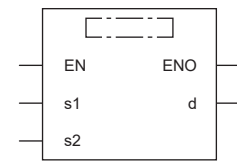
D/(P) (_U)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN32位元資料進行除法運算。

梯形圖	ST*1
	ENO:=DDIVISION(EN, s1, s2, d); ENO:=DDIVISIONP(EN, s1, s2, d); ENO:=DDIVISION_U(EN, s1, s2, d); ENO:=DDIVISIONP_U(EN, s1, s2, d);

FBD/LD



(□中放入DDIVISION、DDIVISIONP、DDIVISION_U、DDIVISIONP_U。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
D/ D/_U	
D/P D/P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	D/(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	D/(P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(s2)	D/(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	D/(P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(d)	D/(P)	—	帶符號BIN64位元	ANY32_S_ARRAY (要素數: 2)
	D/(P)_U	—	無符號BIN64位元	ANY32_U_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

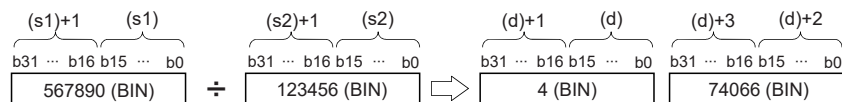
操作數	位元		字元				雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ	K、H		E	\$		
(s1)	○	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	—	○	—	—	○	—	○	○	○	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 將(s1)中指定的BIN32位元資料與(s2)中指定的BIN32位元資料進行除法運算，將除法運算結果儲存到(d)中指定的元件中。



- 在字元件的情況下，除法運算結果使用64位元儲存商及餘數。商被儲存到低位32位元中、餘數被儲存到高位32位元中。在位元元件的情況下，使用32位元並僅儲存商。

出錯

出錯代碼(SD0)	內容
3400H	當(s2)中指定的值(除數)為0時。

BCD4位加法運算

B+(P) [操作數為2個的情況下]



對指定的2個BCD4位資料進行加法運算。

梯形圖	ST
	不對應。 (☞ 254頁 B+(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 254頁 B+(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
B+	
B+P	

設置數據

■內容、範圍、資料類型

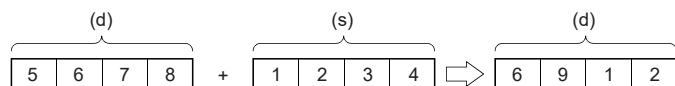
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	加法運算資料或儲存了加法運算資料的元件	0~9999	BCD4位	ANY16
(d)	儲存了被加法運算資料的元件	0~9999	BCD4位	ANY16

■可使用的元件

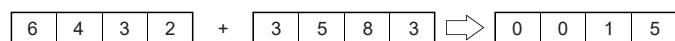
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- 將(d)中指定的BCD4位資料與(s)中指定的BCD4位資料進行加法運算，將加法運算結果儲存到(d)中指定的元件中。



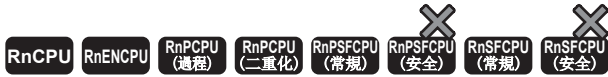
- 在加法運算結果超過了9999的情況下，位數上升將被忽略。在此情況下，SM700(進位標誌)不變為ON。



出錯

出錯代碼(SD0)	內容
3405H	當(s)的BCD資料為0~9999以外時。 當(d)的BCD資料為0~9999以外時。

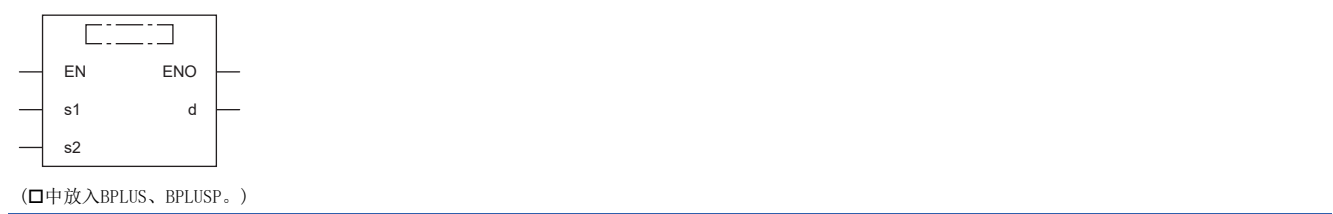
B+(P) [操作數為3個的情況下]



對指定的2個BCD4位資料進行加法運算。

梯形圖	ST
	ENO:=BPLUS (EN, s1, s2, d) ; ENO:=BPLUSP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
B+	
B+P	

設置數據

■內容、範圍、資料類型

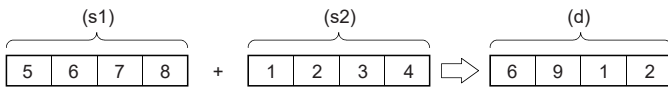
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被加法運算資料或儲存了被加法運算資料的元件	0~9999	BCD4位	ANY16
(s2)	加法運算資料或儲存了加法運算資料的元件	0~9999	BCD4位	ANY16
(d)	儲存運算結果的元件	—	BCD4位	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

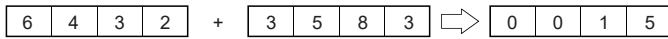
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

功能

- 將(s1)中指定的BCD4位資料與(s2)中指定的BCD4位資料進行加法運算，將加法運算結果儲存到(d)中指定的元件中。



- 在加法運算結果超過了9999的情況下，位數上升將被忽略。在此情況下，SM700(進位標誌)不變為ON。

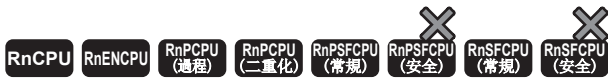


出錯

出錯代碼(SD0)	內容
3405H	當(s1)的BCD資料為0~9999以外時。
	當(s2)的BCD資料為0~9999以外時。

BCD4位減法運算

B-(P) [操作數為2個的情況下]



對指定的2個BCD4位資料進行減法運算。

梯形圖	ST
	不對應。 (☞ 257頁 B-(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 257頁 B-(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
B-	
B-P	

■設置數據

■內容、範圍、資料類型

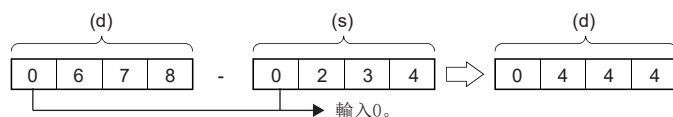
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	減法運算資料或儲存了減法運算資料的元件	0~9999	BCD4位	ANY16
(d)	儲存了被減法運算資料的元件	0~9999	BCD4位	ANY16

■可使用的元件

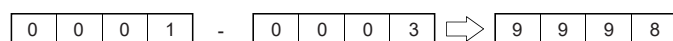
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

■功能

- 將(d)中指定的BCD4位資料與(s)中指定的BCD4位資料進行減法運算，將減法運算結果儲存到(d)中指定的元件中。



- 當減法運算結果發生了下溢時，其情況如下所示。在此情況下，SM700(進位標誌)不變為ON。



■出錯

出錯代碼(SD0)	內容
3405H	當(s)的BCD資料為0~9999以外時。
	當(d)的BCD資料為0~9999以外時。

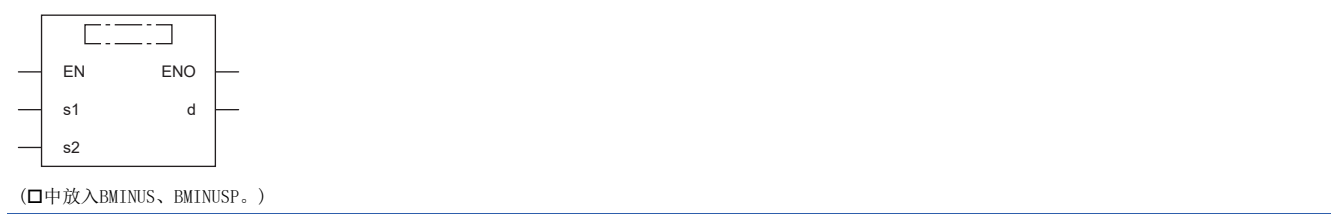
B-(P) [操作數為3個的情況下]



對指定的2個BCD4位資料進行減法運算。

梯形圖	ST
	ENO:=BMINUS(EN, s1, s2, d); ENO:=BMINUSP(EN, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
B-	
B-P	

設置數據

■內容、範圍、資料類型

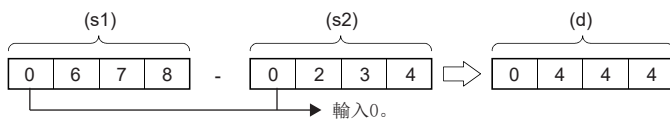
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被減法運算資料或儲存了被減法運算資料的元件	0~9999	BCD4位	ANY16
(s2)	減法運算資料或儲存了減法運算資料的元件	0~9999	BCD4位	ANY16
(d)	儲存運算結果的元件	—	BCD4位	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

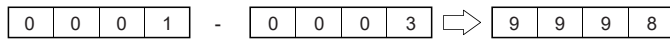
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

功能

- 將(s1)中指定的BCD4位資料與(s2)中指定的BCD4位資料進行減法運算，將減法運算結果儲存到(d)中指定的元件中。



- 當減法運算結果發生了下溢時，其情況如下所示。在此情況下，SM700(進位標誌)不變為ON。

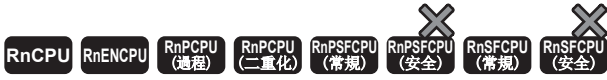


出錯

出錯代碼(SD0)	內容
3405H	當(s1)的BCD資料為0~9999以外時。
	當(s2)的BCD資料為0~9999以外時。

BCD8位加法運算

DB+(P) [操作數為2個的情況下]



對指定的2個BCD8位資料進行加法運算。

梯形圖	ST
	不對應。 (☞ 260頁 DB+(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 260頁 DB+(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
DB+	
DB+P	

■設置數據

■內容、範圍、資料類型

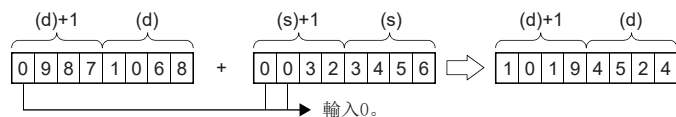
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	加法運算資料或儲存了加法運算資料的起始元件	0~99999999	BCD8位	ANY32
(d)	儲存了被加法運算資料的起始元件	0~99999999	BCD8位	ANY32

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\□(H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—	—

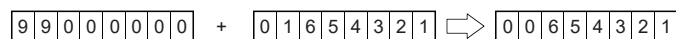
■功能

- 將(d)中指定的BCD8位資料與(s)中指定的BCD8位資料進行加法運算，將加法運算結果儲存到(d)中指定的元件中。



(d)+1、(s)+1: 高位4位
 (d)、(s): 低位4位

- 在加法運算結果超過了99999999的情況下，位數上升將被忽略。在此情況下，SM700(進位標誌)不變為ON。



■出錯

出錯代碼(SD0)	內容
3405H	當(s)的BCD資料為0~99999999以外時。
	當(d)的BCD資料為0~99999999以外時。

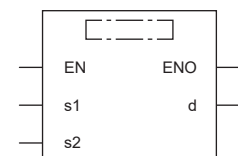
DB+(P) [操作數為3個的情況下]



對指定的2個BCD8位資料進行加法運算。

梯形圖	ST
	ENO:=DBPLUS (EN, s1, s2, d) ; ENO:=DBPLUSP (EN, s1, s2, d) ;

FBD/LD



(□中放入DBPLUS、DBPLUSP。)

■執行條件

指令	執行條件
DB+	
DB+P	

設置數據

■內容、範圍、資料類型

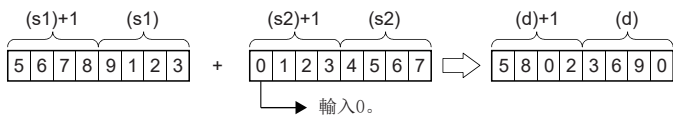
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被加法運算資料或儲存了被加法運算資料的起始元件	0~99999999	BCD8位	ANY32
(s2)	加法運算資料或儲存了加法運算資料的起始元件	0~99999999	BCD8位	ANY32
(d)	儲存運算結果的起始元件	—	BCD8位	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

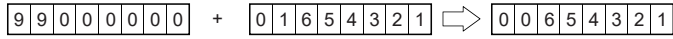
- 將(s1)中指定的BCD8位資料與(s2)中指定的BCD8位資料進行加法運算，將加法運算結果儲存到(d)中指定的元件中。



(d)+1、(s1)+1、(s2)+1: 高位4位

(d)、(s1)、(s2): 低位4位

- 在加法運算結果超過了99999999的情況下，位數上升將被忽略。在此情況下，SM700(進位標誌)不變為ON。

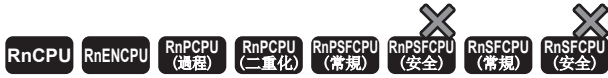


出錯

出錯代碼(SD0)	內容
3405H	當(s1)的BCD資料為0~99999999以外時。
	當(s2)的BCD資料為0~99999999以外時。

BCD8位減法運算

DB-(P) [操作數為2個的情況下]



對指定的2個BCD8位資料進行減法運算。

梯形圖	ST
	不對應。 (☞ 263頁 DB-(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 263頁 DB-(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
DB-	
DB-P	

■設置數據

■內容、範圍、資料類型

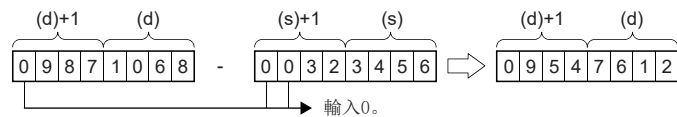
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	減法運算資料或儲存了減法運算資料的起始元件	0~99999999	BCD8位	ANY32
(d)	被減法運算資料或儲存了被減法運算資料的起始元件	0~99999999	BCD8位	ANY32

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—	—

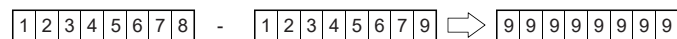
■功能

- 將(d)中指定的BCD8位資料與(s)中指定的BCD8位資料進行減法運算，將減法運算結果儲存到(d)中指定的元件中。



(d)+1、(s)+1: 高位4位
 (d)、(s): 低位4位

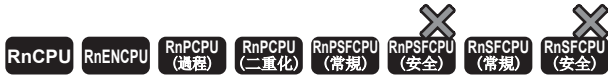
- 當減法運算結果發生了下溢時，其情況如下所示。在此情況下，SM700(進位標誌)不變為ON。



■出錯

出錯代碼(SD0)	內容
3405H	當(s)的BCD資料為0~99999999以外時。
	當(d)的BCD資料為0~99999999以外時。

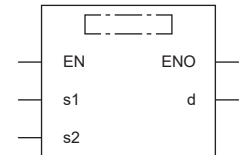
DB-(P) [操作數為3個的情況下]



對指定的2個BCD8位資料進行減法運算。

梯形圖	ST
	ENO:=DBMINUS (EN, s1, s2, d) ; ENO:=DBMINUSP (EN, s1, s2, d) ;

FBD/LD



(□中放入DBMINUS、DBMINUSP。)

■執行條件

指令	執行條件
DB-	
DB-P	

設置數據

■內容、範圍、資料類型

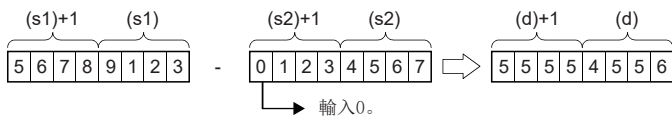
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被減法運算資料或儲存了被減法運算資料的起始元件	0~99999999	BCD8位	ANY32
(s2)	減法運算資料或儲存了減法運算資料的起始元件	0~99999999	BCD8位	ANY32
(d)	儲存運算結果的起始元件	—	BCD8位	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—

功能

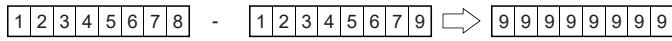
- 將(s1)中指定的BCD8位資料與(s2)中指定的BCD8位資料進行減法運算，將減法運算結果儲存到(d)中指定的元件中。



(d)+1、(s1)+1、(s2)+1: 高位4位

(d)、(s1)、(s2): 低位4位

- 當減法運算結果發生了下溢時，其情況如下所示。在此情況下，SM700(進位標誌)不變為0N。

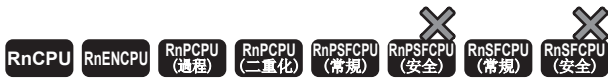


出錯

出錯代碼(SD0)	內容
3405H	當(s1)的BCD資料為0~99999999以外時。
	當(s2)的BCD資料為0~99999999以外時。

BCD4位乘法運算

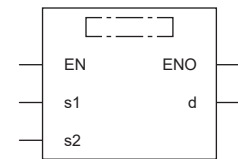
B*(P)



對指定的2個BCD4位資料進行乘法運算。

梯形圖	ST
	不對應。

FBD/LD



(□中放入BMULTI、BMULTIP。)

■執行條件

指令	執行條件
B*	
B*P	

設置數據

■內容、範圍、資料類型

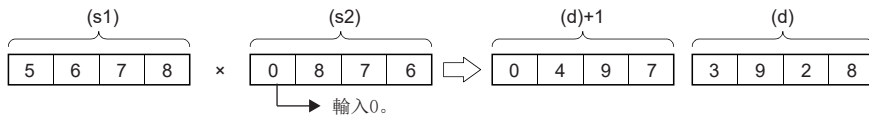
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被乘法運算資料或儲存了被乘法運算資料的元件	0~9999	BCD4位	ANY16
(s2)	乘法運算資料或儲存了乘法運算資料的元件	0~9999	BCD4位	ANY16
(d)	儲存運算結果的起始元件	—	BCD8位	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—

功能

- 將(s1)中指定的BCD4位資料與(s2)中指定的BCD4位資料進行乘法運算，將乘法運算結果儲存到(d)中指定的元件中。((d)+1為高位4位，(d)為低位4位)

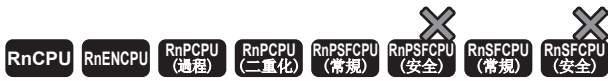


出錯

出錯代碼 (SD0)	內容
3405H	當 (s1) 的BCD資料為0~9999以外時。
	當 (s2) 的BCD資料為0~9999以外時。

BCD4位除法運算

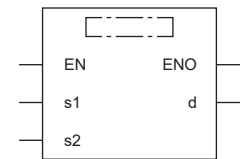
B/(P)



對指定的2個BCD4位資料進行除法運算。

梯形圖	ST
	不對應。

FBD/LD



(□中放入BDIVISION、BDIVISIONP。)

■執行條件

指令	執行條件
B/	
B/P	

設置數據

■內容、範圍、資料類型

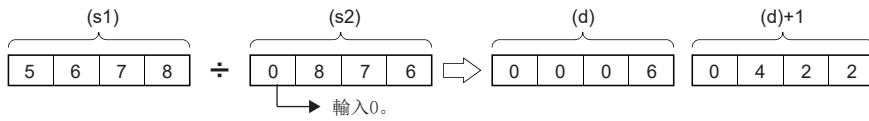
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被除法運算資料或儲存了被除法運算資料的元件	0~9999	BCD4位	ANY16
(s2)	除法運算資料或儲存了除法運算資料的元件	0~9999	BCD4位	ANY16
(d)	儲存運算結果的起始元件	—	BCD8位	ANY16_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 將(s1)中指定的BCD4位資料與(s2)中指定的BCD4位資料進行除法運算，將除法運算結果儲存到(d)中指定的元件中。



(d): 商

(d)+1: 餘數

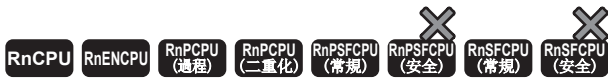
- 除法運算結果使用32位元儲存商及餘數。
- 商(BCD4位)...將被儲存到低位16位元中。
- 餘數(BCD4位)...將被儲存到高位16位元中。

出錯

出錯代碼 (SD0)	內容
3400H	當 (s2) 中指定的值 (除數) 為0時。
3405H	當 (s1) 的BCD資料為0~9999以外時。
	當 (s2) 的BCD資料為0~9999以外時。

BCD8位乘法運算

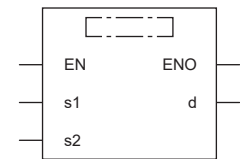
DB*(P)



對指定的2個BCD8位資料進行乘法運算。

梯形圖	ST
	不對應。

FBD/LD



(□中放入DBMULTI、DBMULTIP。)

■執行條件

指令	執行條件
DB*	
DB*P	

設置數據

■內容、範圍、資料類型

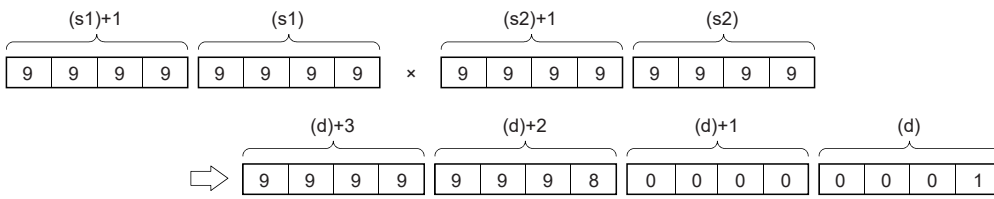
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被乘法運算資料或儲存了被乘法運算資料的起始元件	0~99999999	BCD8位	ANY32
(s2)	乘法運算資料或儲存了乘法運算資料的起始元件	0~99999999	BCD8位	ANY32
(d)	儲存運算結果的起始元件	—	BCD16位	ANY32_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\(\H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	—	○	—	—	○	—	○	—	—	—	—	—

功能

- 將 (s1) 中指定的BCD8位資料與 (s2) 中指定的BCD8位資料進行乘法運算，將乘法運算結果儲存到 (d) 中指定的元件中。



- 在 (d) 為位元件的情況下，乘法運算結果的低位8位 (低位32位元) 為止將成為對象，不能指定高位8位 (高位32位元)。

例

在 (d) 為位元元件情況下的乘法運算結果

- K1...低位1位 (b0~b3)
- K4...低位4位 (b0~b15)
- K8...低位8位 (b0~b31)

出錯

出錯代碼 (SD0)	內容
3405H	當 (s1) 的BCD資料為0~99999999以外時。
	當 (s2) 的BCD資料為0~99999999以外時。

BCD8位除法運算

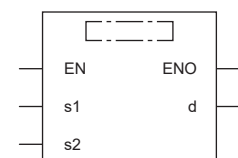
DB/ (P)



對指定的2個BCD8位資料進行除法運算。

梯形圖	ST
	不對應。

FBD/LD



(□中放入DBDIVISION、DBDIVISIONP。)

■執行條件

指令	執行條件
DB/	
DB/P	

設置數據

■內容、範圍、資料類型

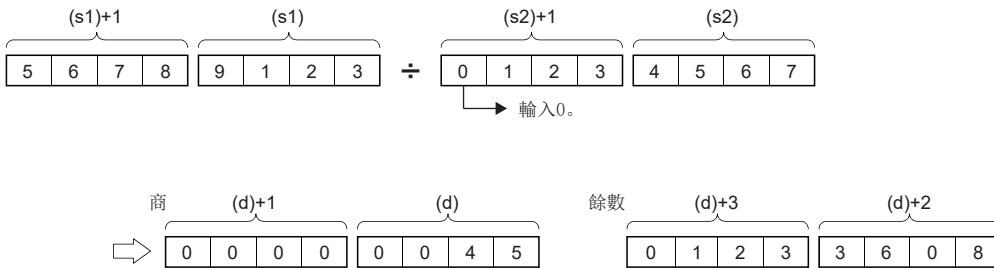
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被除法運算資料或儲存了被除法運算資料的起始元件	0~99999999	BCD8位	ANY32
(s2)	除法運算資料或儲存了除法運算資料的起始元件	0~99999999	BCD8位	ANY32
(d)	儲存運算結果的起始元件	—	BCD16位	ANY32_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	○	○	○	○	○	○	○	○	—	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	—	○	—	—	○	—	○	—	—	—	—

功能

- 將 (s1) 中指定的BCD8位資料與 (s2) 中指定的BCD8位資料進行除法運算，將除法運算結果儲存到 (d) 中指定的元件中。



(d)+1、(d)+3: 高位4位

(d)、(d)+2: 低位4位

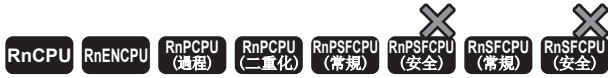
- 除法運算結果使用64位元儲存商及餘數。
- 商 (BCD8位) ... 被儲存到低位32位元中。
- 餘數 (BCD8位) ... 被儲存到高位32位元中。
- 在將 (d) 以位元元件進行了指定的情況下，除法運算結果的餘數將不被儲存。

出錯

出錯代碼 (SD0)	內容
3400H	當 (s2) 中指定的值 (除數) 為0時。
3405H	當 (s1) 的BCD資料為0~99999999以外時。
	當 (s2) 的BCD資料為0~99999999以外時。

BIN16位元塊資料加法運算

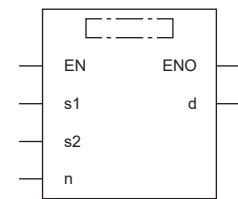
BK+(P) (_U)



對指定的2個BIN16位元資料塊進行加法運算。

梯形圖	ST*1	
	ENO:=BKPLUS (EN, s1, s2, n, d) ; ENO:=BKPLUS_U (EN, s1, s2, n, d) ;	ENO:=BKPLUS_U (EN, s1, s2, n, d) ; ENO:=BKPLUS_U (EN, s1, s2, n, d) ;

FBD/LD



(□中放入BKPLUS、BKPLUS_U、BKPLUS_U、BKPLUS_U。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
BK+ BK+_U	
BK+P BK+P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	BK+(P)	被加法運算資料或儲存了被加法運算資料的起始元件	-32768~32767	帶符號BIN16位元	ANY16_S*1
	BK+(P)_U		0~65535	無符號BIN16位元	ANY16_U*1
(s2)	BK+(P)	加法運算資料或儲存了加法運算資料的起始元件	-32768~32767	帶符號BIN16位元	ANY16_S*1
	BK+(P)_U		0~65535	無符號BIN16位元	ANY16_U*1
(d)	BK+(P)	儲存運算結果的起始元件	—	帶符號BIN16位元	ANY16_S*1
	BK+(P)_U		無符號BIN16位元	ANY16_U*1	
(n)	加法運算資料個數	0~65535	無符號BIN16位元	ANY16	
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

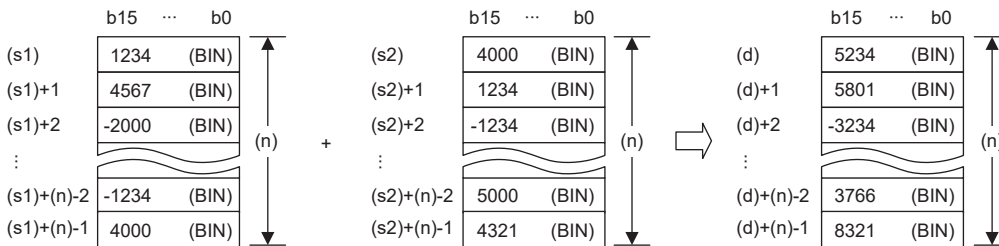
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	—	○	○	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

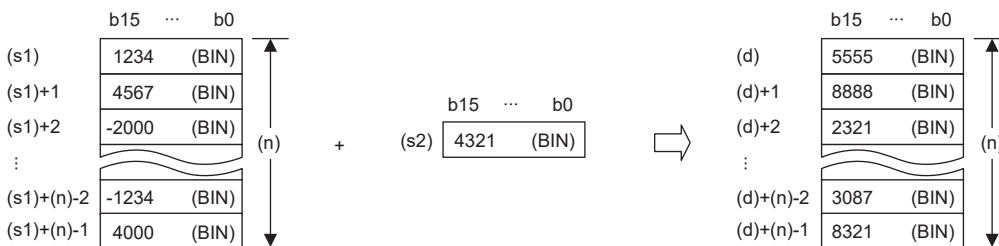
- 將(s1)中指定的元件開始的(n)點的BIN16位元資料與(s2)中指定的元件開始的(n)點的BIN16位元資料或常數進行加法運算，將加法運算結果儲存在(d)中指定的元件及其以後。
- 塊加法運算以16位元單位進行。

例

在(s2)中指定了元件的情況下(帶符號指定時)



在(s2)中指定了常數的情況下(帶符號指定時)



- 當運算結果發生了上溢時，其情況如下所示。在此情況下，SM700(進位標誌)不變為0N。

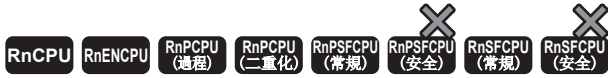
在指定了帶符號的情況下		在指定了無符號的情況下	
K32767 (H7FFF)	+ K2 (H0002)	⇒	K-32767 (H8001)
K-32767 (H8001)	+ K-2 (HFFFE)	⇒	K32767 (H7FFF)
K65535 (HFFFF)	+ K1 (H0001)	⇒	K0 (H0000)

出錯

出錯代碼(SD0)	內容
2821H	當(s1)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍重複時。 (s1)及(d)中指定了同一元件的情況下除外。)
	當(s2)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍重複時。 (s2)及(d)中指定了同一元件的情況下除外。)

BIN16位元塊資料減法運算

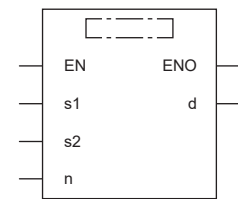
BK-(P) (_U)



對指定的2個BIN16位元資料塊進行減法運算。

梯形圖	ST*1	
	ENO:=BKMINUS (EN, s1, s2, n, d) ; ENO:=BKMINUSP (EN, s1, s2, n, d) ;	ENO:=BKMINUS_U (EN, s1, s2, n, d) ; ENO:=BKMINUSP_U (EN, s1, s2, n, d) ;

FBD/LD



(□中放入BKMINUS、BKMINUSP、BKMINUS_U、BKMINUSP_U。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
BK- BK-_U	
BK-P BK-P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	BK-(P)	-32768~32767	帶符號BIN16位元	ANY16_S*1
	BK-(P)_U	0~65535	無符號BIN16位元	ANY16_U*1
(s2)	BK-(P)	-32768~32767	帶符號BIN16位元	ANY16_S*1
	BK-(P)_U	0~65535	無符號BIN16位元	ANY16_U*1
(d)	BK-(P)	—	帶符號BIN16位元	ANY16_S*1
	BK-(P)_U	—	無符號BIN16位元	ANY16_U*1
(n)	減法運算資料個數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

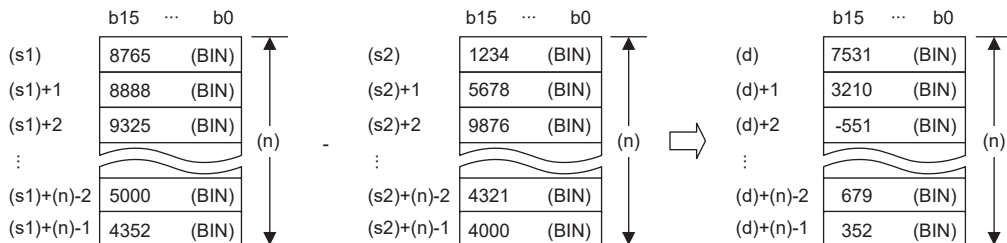
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—	—

功能

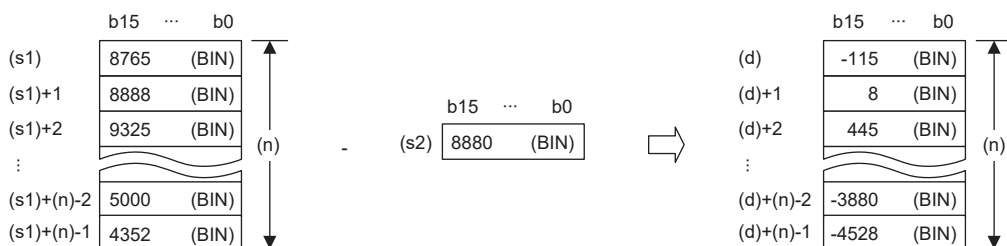
- 將(s1)中指定的元件開始的(n)點的BIN16位元資料與(s2)中指定的元件開始的(n)點的BIN16位元資料或常數進行減法運算，將減法運算結果儲存到(d)中指定的元件及其以後。
- 塊減法運算以16位元單位進行。

例

在(s2)中指定了元件的情況下



在(s2)中指定了常數的情況下



- 當運算結果發生了上溢時，其情況如下所示。在此情況下，SM700(進位標誌)不變為ON。

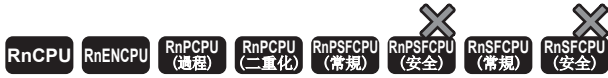
在指定了帶符號的情況下			在指定了無符號的情況下			
K-32768 (H8000)	-	K2 (H0002)	⇒	K32766 (H7FFE)	K0 (H0000)	
K32767 (H7FFF)	-	K-2 (HFFFE)	⇒	K-32767 (H8001)		K1 (H0001)
					⇒	K65535 (HFFFF)

出錯

出錯代碼(SD0)	內容
2821H	當(s1)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍重複時。 (s1)及(d)中指定了同一元件的情況下除外。)
	當(s2)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍重複時。 (s2)及(d)中指定了同一元件的情況下除外。)

BIN32位元塊資料加法運算

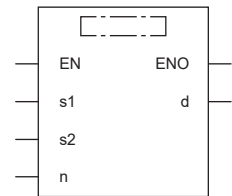
DBK+ (P) (_U)



對指定的2個BIN32位元資料塊進行加法運算。

梯形圖	ST*1	
	ENO:=DBKPLUS (EN, s1, s2, n, d) ; ENO:=DBKPLUSP (EN, s1, s2, n, d) ;	ENO:=DBKPLUS_U (EN, s1, s2, n, d) ; ENO:=DBKPLUSP_U (EN, s1, s2, n, d) ;

FBD/LD



(□中放入DBKPLUS、DBKPLUSP、DBKPLUS_U、DBKPLUSP_U。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DBK+ DBK+_U	
DBK+P DBK+P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	DBK+ (P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S*1
	DBK+ (P)_U		無符號BIN32位元	ANY32_U*1
(s2)	DBK+ (P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S*1
	DBK+ (P)_U		無符號BIN32位元	ANY32_U*1
(d)	DBK+ (P)	—	帶符號BIN32位元	ANY32_S*1
	DBK+ (P)_U		無符號BIN32位元	ANY32_U*1
(n)	加法運算資料個數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

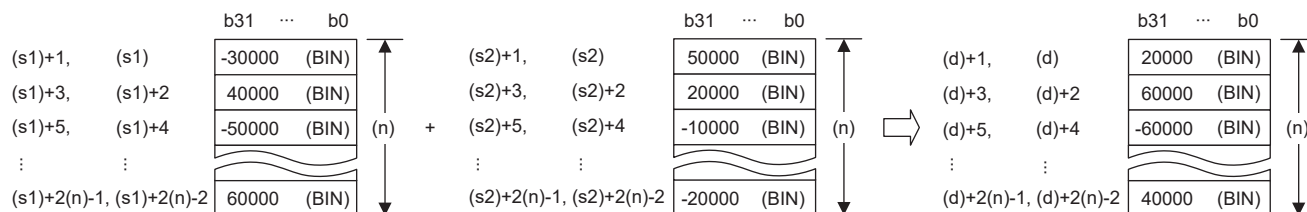
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—	—

功能

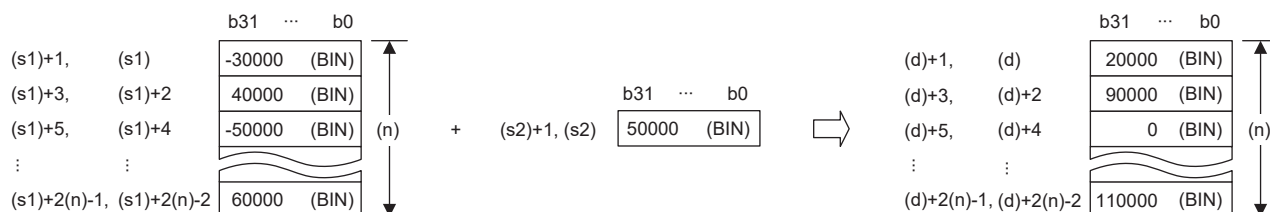
- 將(s1)中指定的元件開始的(n)點的BIN32位元資料與(s2)中指定的元件開始的(n)點的BIN32位元資料或常數進行加法運算，將加法運算結果儲存到(d)中指定的元件及其以後。
- 塊加法運算以32位元單位進行。

例

在(s2)中指定了元件的情況下(帶符號指定時)



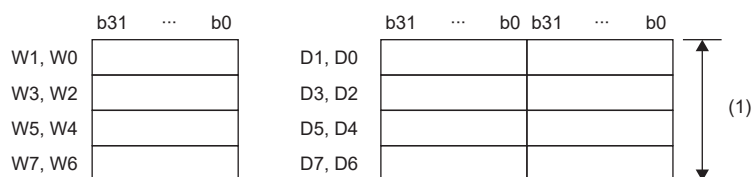
在(s2)中指定了常數的情況下(帶符號指定時)



- 在(s1)或(s2)與(d)以同一元件(完全一致)指定的情況下，可以運算。但是，(s1)或(s2)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍有部分一致(重複)的情況下，將變為出錯狀態。

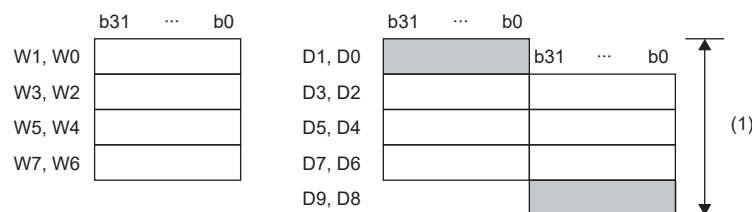
例

在(s2)與(d)開始的4點的元件完全一致的情況下



(1) 由於完全一致，因此可以運算。

在(s2)、(d)開始的4點的元件有部分一致的情況下



(1) 由於部分一致，因此變為運算出錯。

- 在(n)中指定的值為0的情況下將變為無處理。
- 當運算結果發生了上溢時，其情況如下所示。在此情況下，SM700(進位標誌)不變為ON。

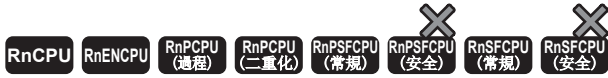
在指定了帶符號的情況下		在指定了無符號的情況下	
K2147483647 (H7FFFFFFF)	+ K2 (H00000002) →	K-2147483647 (H80000001)	K4294967295 (HFFFFFFF)
K-2147483647 (H80000001)	+ K-2 (HFFFFFFFE) →	K2147483647 (H7FFFFFFF)	K1 (H00000001) →
		K0 (H00000000)	

出錯

出錯代碼 (SD0)	內容
2821H	當 (s1) 開始的 (n) 點的元件範圍與 (d) 開始的 (n) 點的元件範圍重複時。 (s1) 及 (d) 中指定了同一元件的情況下除外。)
	當 (s2) 開始的 (n) 點的元件範圍與 (d) 開始的 (n) 點的元件範圍重複時。 (s2) 及 (d) 中指定了同一元件的情況下除外。)

BIN32位元塊資料減法運算

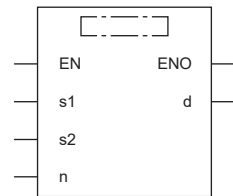
DBK-(P) (_U)



對指定的2個BIN32位元資料塊進行減法運算。

梯形圖	ST*1	
	ENO:=DBKMINUS (EN, s1, s2, n, d) ; ENO:=DBKMINUSP (EN, s1, s2, n, d) ;	ENO:=DBKMINUS_U (EN, s1, s2, n, d) ; ENO:=DBKMINUSP_U (EN, s1, s2, n, d) ;

FBD/LD



(□中放入DBKMINUS、DBKMINUSP、DBKMINUS_U、DBKMINUSP_U。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DBK- DBK-_U	
DBK-P DBK-P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	DBK-(P) DBK-(P)_U	被減法運算資料或儲存了被減法運算資料的起始元件	-2147483648~2147483647 0~4294967295	帶符號BIN32位元 無符號BIN32位元	ANY32_S*1 ANY32_U*1
(s2)	DBK-(P) DBK-(P)_U	減法運算資料或儲存了減法運算資料的起始元件	-2147483648~2147483647 0~4294967295	帶符號BIN32位元 無符號BIN32位元	ANY32_S*1 ANY32_U*1
(d)	DBK-(P) DBK-(P)_U	儲存運算結果的起始元件	—	帶符號BIN32位元 無符號BIN32位元	ANY32_S*1 ANY32_U*1
(n)		減法運算資料個數	0~65535	無符號BIN16位元	ANY16
EN		執行條件	—	位元	BOOL
ENO		執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

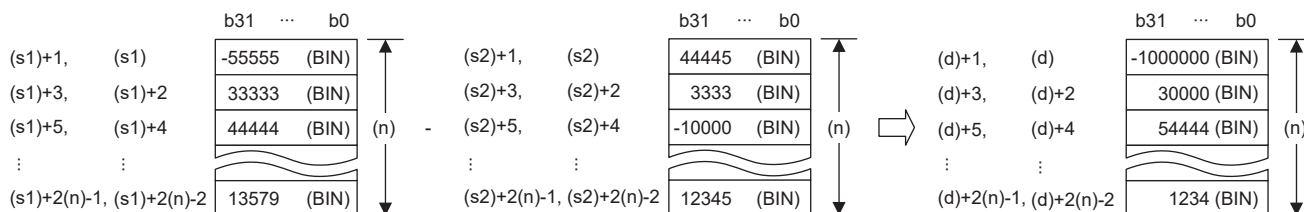
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

功能

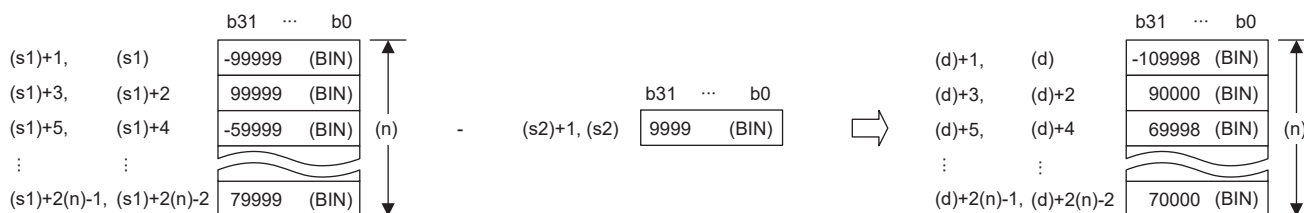
- 將(s1)中指定的元件開始的(n)點的BIN32位元資料與(s2)中指定的元件開始的(n)點的BIN32位元資料或常數進行減法運算，將減法運算結果儲存到(d)中指定的元件及其以後。
- 塊減法運算以32位元單位進行。

例

在(s2)中指定了元件的情況下(帶符號指定時)



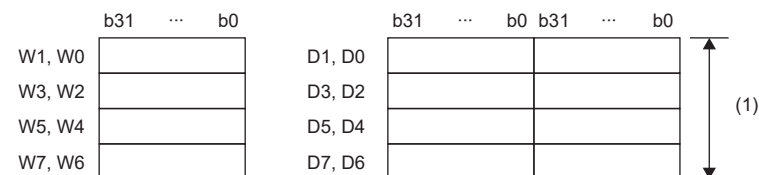
在(s2)中指定了常數的情況下(帶符號指定時)



- 在(s1)或(s2)與(d)以同一元件(完全一致)指定的情況下，可以運算。但是，(s1)或(s2)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍有部分一致(重複)的情況下，將變為出錯狀態。

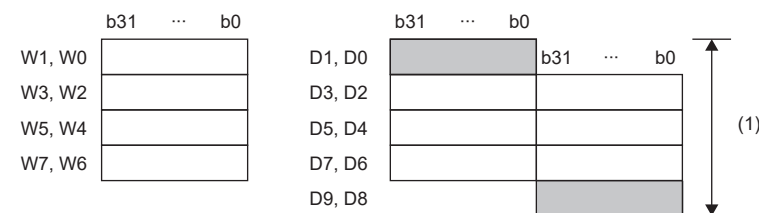
例

在(s2)與(d)開始的4點的元件完全一致的情況下



(1) 由於完全一致，因此可以運算。

在(s2)、(d)開始的4點的元件有部分一致的情況下



(1) 由於部分一致，因此變為運算出錯。

- 在(n)中指定的值為0的情況下將變為無處理。
- 當運算結果發生了上溢時，其情況如下所示。在此情況下，SM700(進位標誌)不變為ON。

在指定了帶符號的情況下		在指定了無符號的情況下	
K2147483647 (H7FFFFFFF)	- K-2 (HFFFFFFFE)	→ K-2147483647 (H80000001)	
K-2147483647 (H80000001)	- K2 (H00000002)	→ K2147483647 (H7FFFFFFF)	
			K0 (H00000000) - K1 (H00000001) → K4294967295 (HFFFFFFF)

出錯

出錯代碼 (SD0)	內容
2821H	當 (s1) 開始的 (n) 點的元件範圍與 (d) 開始的 (n) 點的元件範圍重複時。 (s1) 及 (d) 中指定了同一元件的情況下除外。)
	當 (s2) 開始的 (n) 點的元件範圍與 (d) 開始的 (n) 點的元件範圍重複時。 (s2) 及 (d) 中指定了同一元件的情況下除外。)

16位元BIN資料遞增

INC(P) (_U)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的BIN16位元資料進行+1。

梯形圖	ST	
	ENO:=INC(EN, d); ENO:=INCP(EN, d);	ENO:=INC_U(EN, d); ENO:=INCP_U(EN, d);

FBD/LD

執行條件

指令	執行條件
INC INC_U	
INCP INCP_U	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(d)	INC(P)	進行+1的元件	-32768~32767	帶符號BIN16位元	ANY16_S
	INC(P)_U		0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

可使用的元件

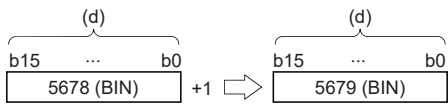
操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
(d)	○	○	○	○	○	—	—	○	—	—	—	—

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	○	○	—

功能

- 對 (d) 中指定的元件 (BIN16 位元資料) 進行 +1。



- 在 (d) 中指定的元件的內容為 32767 時執行了 INC (P) 指令的情況下，-32768 將被儲存到 (d) 中指定的元件中。(在指定帶符號的情況下)
- 在 (d) 中指定的元件內容為 65535 時執行了 INC (P)_U 指令的情況下，0 將被儲存到 (d) 中指定的元件中。(在指定無符號的情況下)

出錯

沒有運算出錯。

16位元BIN資料遞減

DEC(P) (_U)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的BIN16位元資料進行-1。

梯形圖	ST	
	ENO:=DEC(EN, d); ENO:=DECP(EN, d);	ENO:=DEC_U(EN, d); ENO:=DECP_U(EN, d);

FBD/LD

■執行條件

指令	執行條件
DEC DEC_U	
DECP DECP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(d)	DEC(P)	進行-1的元件	-32768~32767	帶符號BIN16位元	ANY16_S
	DEC(P)_U		0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

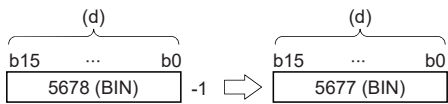
操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	○	○	—

功能

- 對(d)中指定的元件(BIN16位元資料)進行-1。



- 在(d)中指定的元件內容為-32768時執行了DEC(P)指令的情況下，32767將被儲存到(d)中指定的元件中。(在指定帶符號的情況下)
- 在(d)中指定的元件內容為0時執行了DEC(P)_U指令的情況下，65535將被儲存到(d)中指定的元件中。(在指定無符號的情況下)

出錯

沒有運算出錯。

32位元BIN資料遞增

DINC(P) (_U)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的BIN32位元資料進行+1。

梯形圖	ST	
	ENO:=DINC(EN, d); ENO:=DINCP(EN, d);	ENO:=DINC_U(EN, d); ENO:=DINCP_U(EN, d);

FBD/LD

■執行條件

指令	執行條件
DINC DINC_U	
DINCP DINCP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(d)	DINC(P)	進行+1的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DINC(P)_U		0~4294967295	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

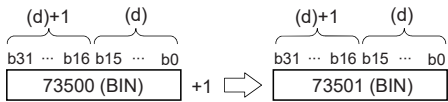
操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
(d)	○	○	○	○	○	○	○	○	—	—	—	—

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	○	○	—

功能

- 對 (d) 中指定的元件內容 (BIN32位元資料) 進行+1。



- 在 (d) 中指定的元件內容為2147483647時執行了DINC(P) 指令的情況下，-2147483648將被儲存到 (d) 中指定的元件中。(在指定帶符號的情況下)
- 在 (d) 中指定的元件內容為4294967295時執行了DINC(P)_U指令的情況下，0將被儲存到 (d) 中指定的元件中。(在指定無符號的情況下)

出錯

沒有運算出錯。

32位元BIN資料遞減

DDEC(P) (_U)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的BIN32位元資料進行-1。

梯形圖	ST	
	ENO:=DDEC(EN, d); ENO:=DDECP(EN, d);	ENO:=DDEC_U(EN, d); ENO:=DDECP_U(EN, d);

FBD/LD

■執行條件

指令	執行條件
DDEC DDEC_U	
DDECP DDECP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(d)	DDEC(P)	進行-1的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DDEC(P)_U		0~4294967295	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

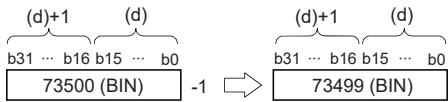
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	\$	
(d)	○	○	○	○	○	○	○	○	—	—	—	—

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元		字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B		SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(d)	○		○	—

功能

- 對(d)中指定的元件內容(BIN32位元資料)進行-1。



- 在(d)中指定的元件內容為-2147483648時執行了DDEC(P)指令的情況下，2147483647將被儲存到(d)中指定的元件中。(在指定帶符號的情況下)
- 當(d)中指定的元件內容為0時執行了DDEC(P)指令時，-1將被儲存到(d)中指定的元件中。(當帶符號指定時)
- 在(d)中指定的元件內容為0時執行了DDEC(P)_U指令的情況下，4294967295將被儲存到(d)中指定的元件中。(在指定無符號的情況下)

出錯

沒有運算出錯。

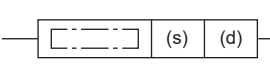
6.3 邏輯運算指令

16位元資料邏輯且

WAND(P) [操作數為2個的情況下]



RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN16位元資料的邏輯且。

梯形圖	ST
	不對應。 (☞ 293頁 WAND(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 293頁 WAND(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
WAND	
WANDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存進行邏輯且的資料或儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存邏輯且結果的元件	-32768~32767	帶符號BIN16位元	ANY16

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

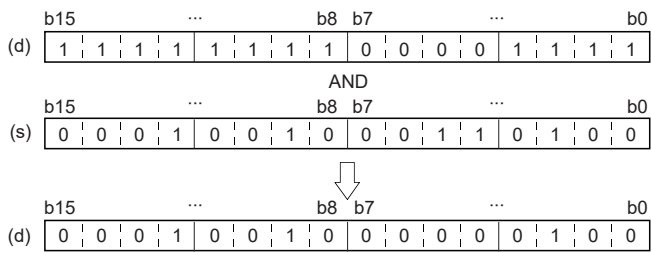
操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 對(d)中指定的元件的BIN16位元資料與(s)中指定的元件的BIN16位元資料的各個位元進行邏輯且的運算，將結果儲存到(d)中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

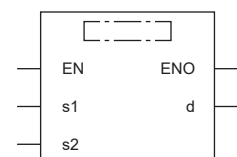
WAND(P) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN16位元資料的邏輯且。

梯形圖	ST
	ENO:=WAND (EN, s1, s2, d) ; ENO:=WANDP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
WAND	
WANDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存進行邏輯且的資料或儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(s2)	儲存進行邏輯且的資料或儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存邏輯且結果的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

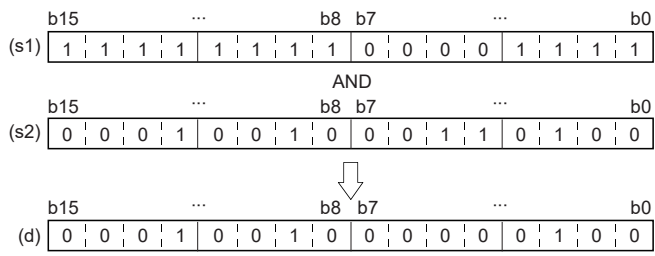
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	○	○	○	○	○	—	—	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 對(s1)中指定的元件的BIN16位元資料與(s2)中指定的元件的BIN16位元資料的各個位元進行邏輯且的運算，將結果儲存在(d)中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

32位元資料邏輯且

DAND(P) [操作數為2個的情況下]

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的2個BIN32位元資料進行邏輯且。

梯形圖	ST
	不對應。 (☞ 297頁 DAND(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 297頁 DAND(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
DAND	
DANDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行邏輯且的資料或儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存邏輯且結果的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

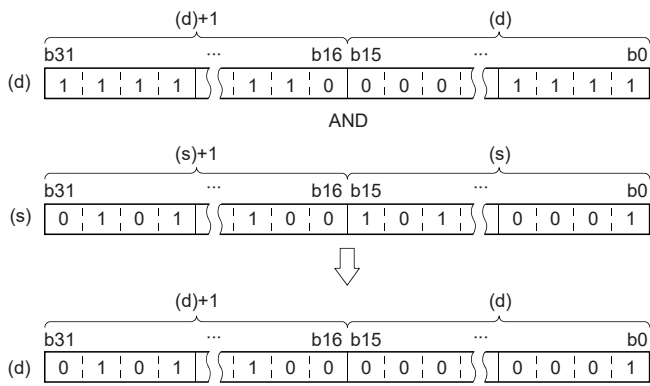
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 對 (d) 中指定的元件的 BIN32 位元資料與 (s) 中指定的元件的 BIN32 位元資料的各個位元進行邏輯且的運算，將結果儲存到 (d) 中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

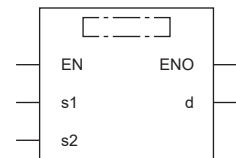
DAND(P) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFPCPU (常規) RnSFPCPU (安全)

對指定的2個BIN32位元資料進行邏輯且。

梯形圖	ST
	ENO:=DAND (EN, s1, s2, d) ; ENO:=DANDP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
DAND	
DANDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	進行邏輯且的資料或儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(s2)	進行邏輯且的資料或儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存邏輯且結果的起始元件	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

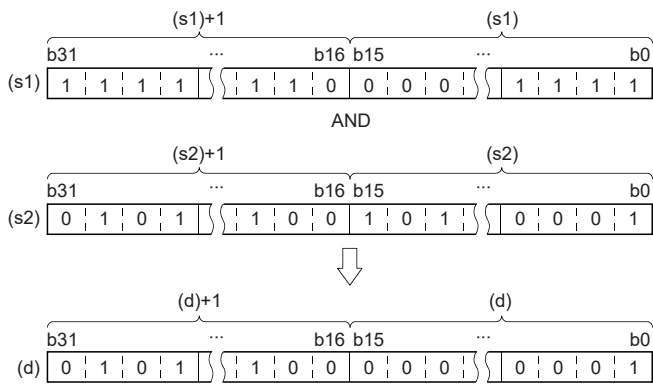
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 對 (s1) 中指定的元件的BIN32位元資料與 (s2) 中指定的元件的BIN32位元資料的各個位進行邏輯且的運算，將結果儲存到 (d) 中指定的元件中。



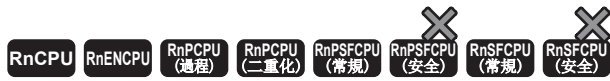
- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

16位元塊資料邏輯且

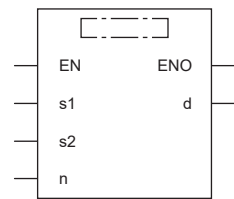
BKAND (P)



對指定的2個BIN16位元資料塊進行邏輯且。

梯形圖	ST
	ENO:=BKAND (EN, s1, s2, n, d) ; ENO:=BKANDP (EN, s1, s2, n, d) ;

FBD/LD



■執行條件

指令	執行條件
BKAND	
BKANDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	進行邏輯且的資料或儲存了進行邏輯且的資料的起始元件	—	帶符號BIN16位元	ANY16*1
(s2)	進行邏輯且的資料或儲存了資料的起始元件	-32768~32767	帶符號BIN16位元	ANY16*1
(d)	儲存邏輯且結果的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	運算資料個數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

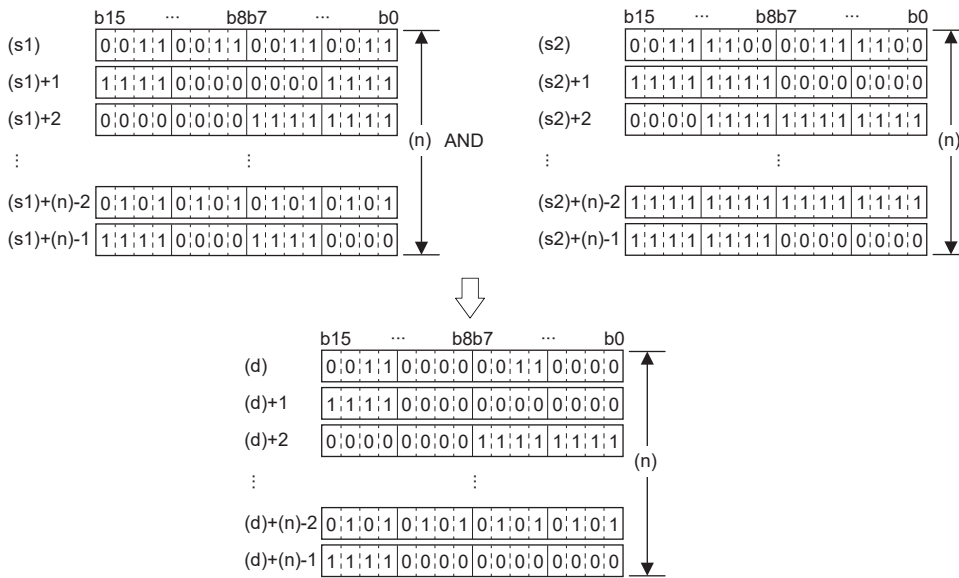
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)*1	—	—	○	—	—	—	○	—	—	—	—	—
(s2)*1	—	—	○	—	—	—	○	○	—	—	—	—
(d)*1	—	—	○	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—	—

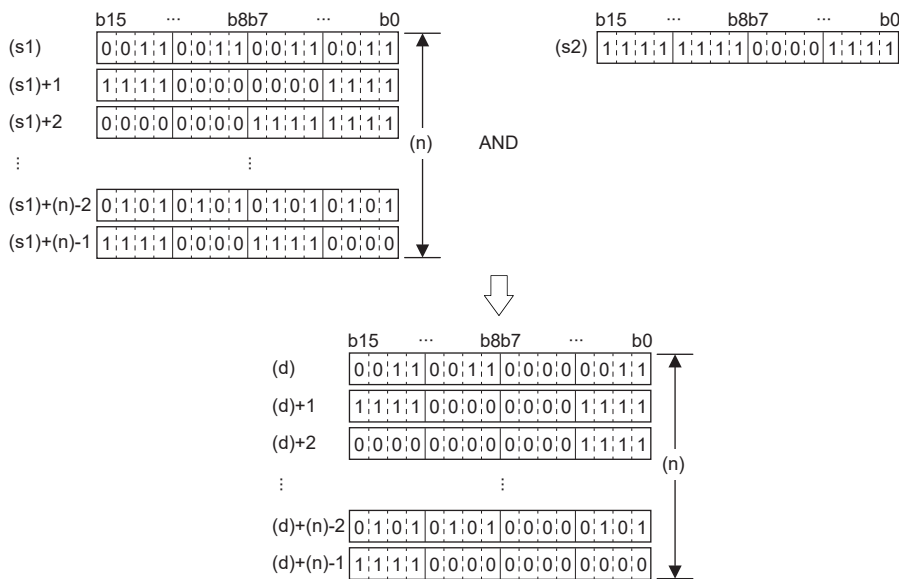
*1 (s1) 與 (d) 或 (s2) 與 (d) 可以指定同一元件編號。

功能

- 對(s1)中指定的元件開始的(n)點的內容與(s2)中指定的元件開始的(n)點的內容進行邏輯且的運算，將結果儲存到(d)中指定的元件及其以後。



- (s2)中可以指定-32768~32767(帶符號BIN16位元)的常數。



出錯

出錯代碼(SD0)	內容
2821H	當(s1)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍的一部分重複時。 (s1)及(d)中指定了同一元件的情況下除外。
	當(s2)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍的一部分重複時。 (s2)及(d)中指定了同一元件的情況下除外。

16位元資料邏輯或

WOR(P) [操作數為2個的情況下]

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN16位元資料的邏輯或。

梯形圖	ST
	不對應。 (☞ 303頁 WOR(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 303頁 WOR(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
WOR	
WORP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行邏輯或的資料或儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存邏輯或結果的元件	-32768~32767	帶符號BIN16位元	ANY16

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

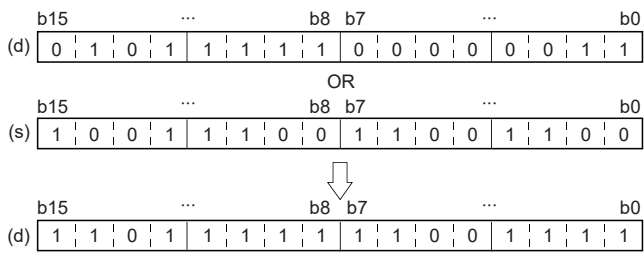
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 對(d)中指定的元件的BIN16位元資料與(s)中指定的元件的BIN16位元資料的各個位元進行邏輯或的運算，將結果儲存到(d)中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

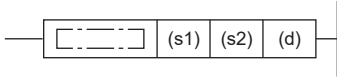
出錯

沒有運算出錯。

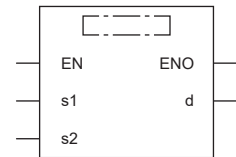
WOR(P) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)



進行指定的2個BIN16位元資料的邏輯或。

梯形圖	ST
	ENO:=WOR (EN, s1, s2, d) ; ENO:=WORP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
WOR	
WORP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	進行邏輯或的資料或儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(s2)	進行邏輯或的資料或儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存邏輯或結果的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

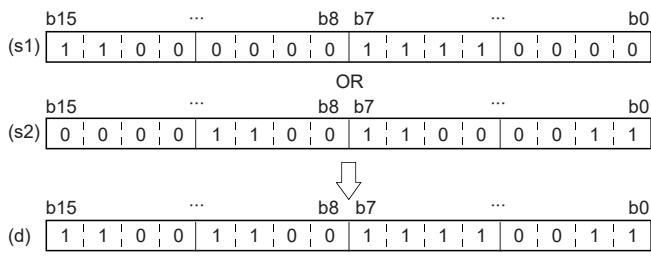
操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 對(s1)中指定的元件的BIN16位元資料與(s2)中指定的元件的BIN16位元資料的各個位元進行邏輯或的運算，將結果儲存到(d)中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

32位元資料邏輯或

DOR(P) [操作數為2個的情況下]

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN32位元資料的邏輯或。

梯形圖	ST
	不對應。 (☞ 307頁 DOR(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 307頁 DOR(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
DOR	
DORP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行邏輯或的資料或儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存邏輯或結果的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

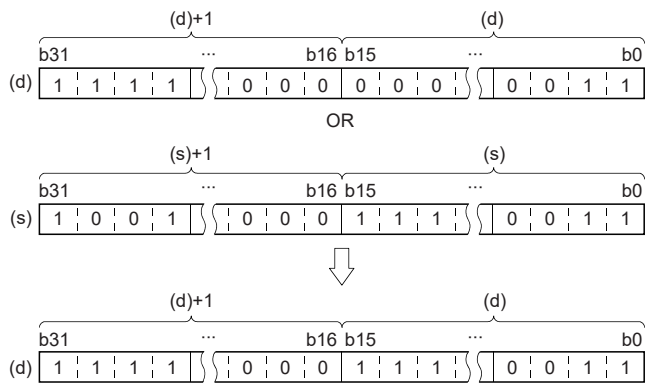
操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 對 (d) 中指定的元件的BIN32位元資料與 (s) 中指定的元件的BIN32位元資料的各個位元進行邏輯或的運算，將結果儲存到 (d) 中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

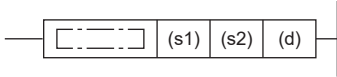
出錯

沒有運算出錯。

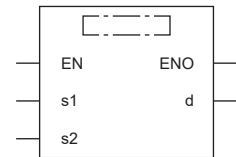
DOR(P) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN32位元資料的邏輯或。

梯形圖	ST
	ENO:=DOR (EN, s1, s2, d) ; ENO:=DORP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
DOR	
DORP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	進行邏輯或的資料或儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(s2)	進行邏輯或的資料或儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存邏輯或結果的起始元件	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

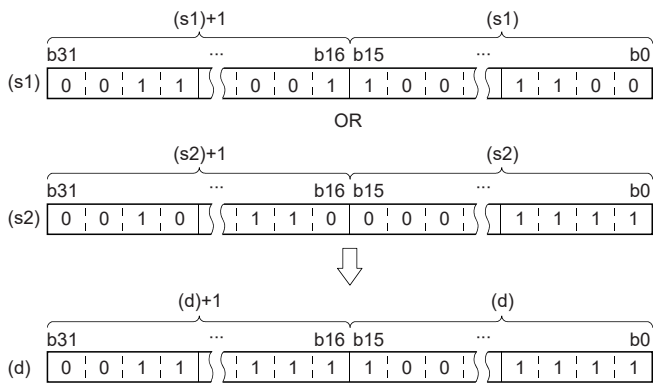
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 對(s1)中指定的元件的BIN32位元資料與(s2)中指定的元件的BIN32位元資料的各個位進行邏輯或的運算，將結果儲存到(d)中指定的元件中。



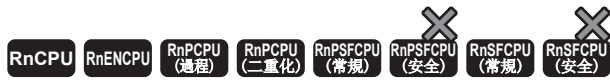
- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

16位元塊資料邏輯或

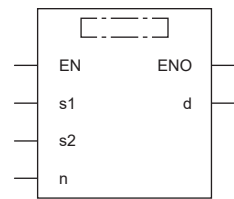
BKOR (P)



進行指定的2個BIN16位元資料塊的邏輯或。

梯形圖	ST
	ENO:=BKOR (EN, s1, s2, n, d) ; ENO:=BKORP (EN, s1, s2, n, d) ;

FBD/LD



執行條件

指令	執行條件
BKOR	
BKORP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	進行邏輯或的資料或儲存了進行邏輯或的資料的起始元件	—	帶符號BIN16位元	ANY16*1
(s2)	進行邏輯或的資料或儲存了資料的起始元件	-32768~32767	帶符號BIN16位元	ANY16*1
(d)	儲存邏輯或結果的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	運算資料個數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

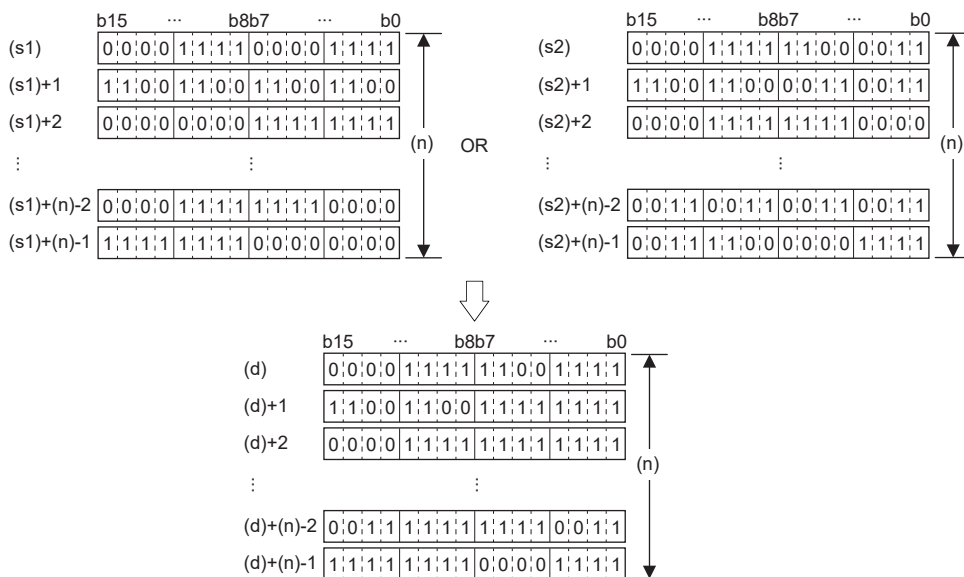
可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)*1	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)*1	—	—	○	—	—	—	—	○	○	—	—	—	
(d)*1	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

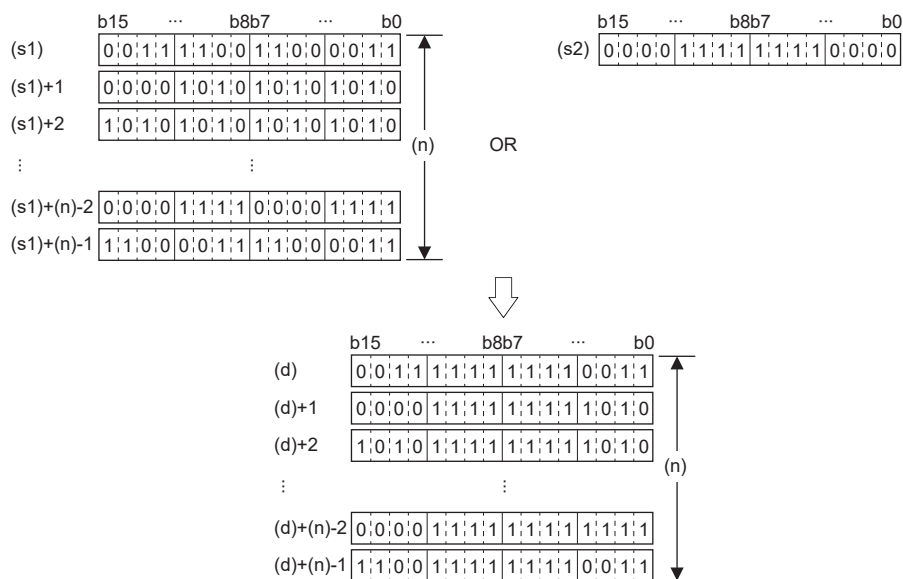
*1 (s1) 與 (d) 或 (s2) 與 (d) 可以指定同一元件編號。

功能

- 對(s1)中指定的元件開始的(n)點的內容與(s2)中指定的元件開始的(n)點的內容進行邏輯或的運算，將結果儲存到(d)中指定的元件及其以後。



- (s2)中可以指定-32768~32767(帶符號BIN16位元)的常數。



出錯

出錯代碼(SD0)	內容
2821H	當(s1)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍的一部分重複時。 (s1)及(d)中指定了同一元件的情況下除外。
	當(s2)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍的一部分重複時。 (s2)及(d)中指定了同一元件的情況下除外。

16位元資料排他邏輯或

WXOR(P) [操作數為2個的情況下]

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN16位元資料的排他邏輯或。

梯形圖	ST
	不對應。 (☞ 313頁 WXOR(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 313頁 WXOR(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
WXOR	
WXORP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行排他邏輯或的資料或者儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存排他邏輯或的結果的元件	-32768~32767	帶符號BIN16位元	ANY16

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

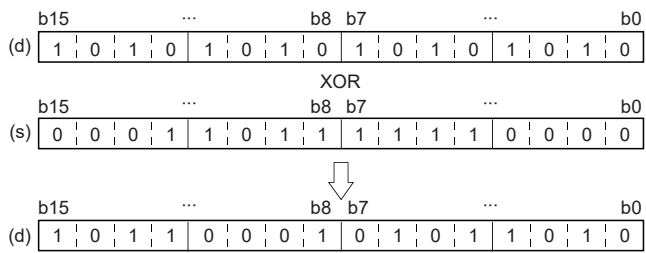
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 對 (d) 中指定的元件的BIN16位元資料與 (s) 中指定的元件的BIN16位元資料的各個位元進行排他邏輯或運算，將結果儲存到 (d) 中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

WXOR(P) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN16位元資料的排他邏輯或。

梯形圖	ST
	ENO:=WXOR (EN, s1, s2, d) ; ENO:=WXORP (EN, s1, s2, d) ;

FBD/LD

■執行條件

指令	執行條件
WXOR	
WXORP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	進行排他邏輯或的資料或者儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(s2)	進行排他邏輯或的資料或者儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存排他邏輯或的結果的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

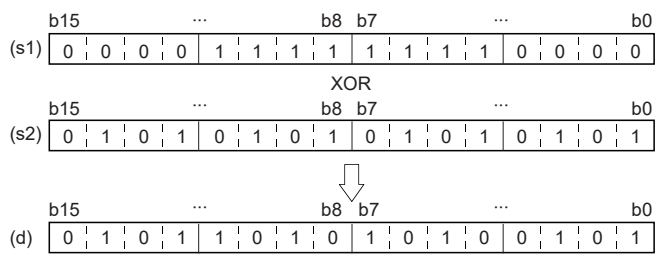
操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 對(s1)中指定的元件的BIN16位元資料與(s2)中指定的元件的BIN16位元資料的各個位元進行排他邏輯或運算，將結果儲存到(d)中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

32位元資料排他邏輯或

DXOR(P) [操作數為2個的情況下]

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN32位元資料的排他邏輯或。

梯形圖	ST
	不對應。 (☞ 317頁 DXOR(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 317頁 DXOR(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
DXOR	
DXORP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行排他邏輯或的資料或者儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存排他邏輯或的結果的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

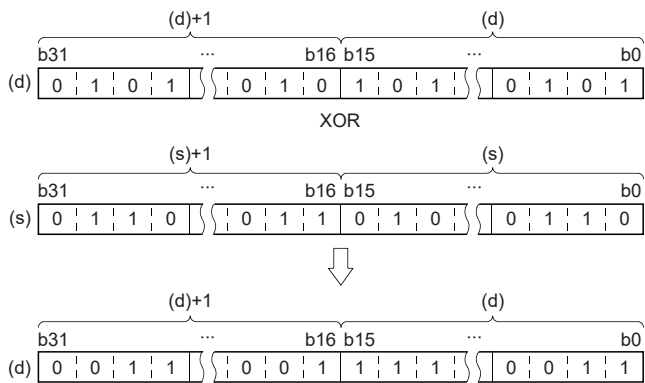
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	\$	
(s)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 對 (d) 中指定的元件的BIN32位元資料與 (s) 中指定的元件的BIN32位元資料的各個位元進行排他邏輯或運算，將結果儲存到 (d) 中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

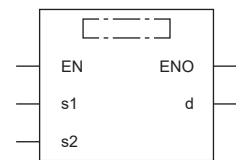
DXOR(P) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN32位元資料的排他邏輯或。

梯形圖	ST
	ENO:=DXOR (EN, s1, s2, d) ; ENO:=DXORP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
DXOR	
DXORP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	進行排他邏輯或的資料或者儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(s2)	進行排他邏輯或的資料或者儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存排他邏輯或的結果的起始元件	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

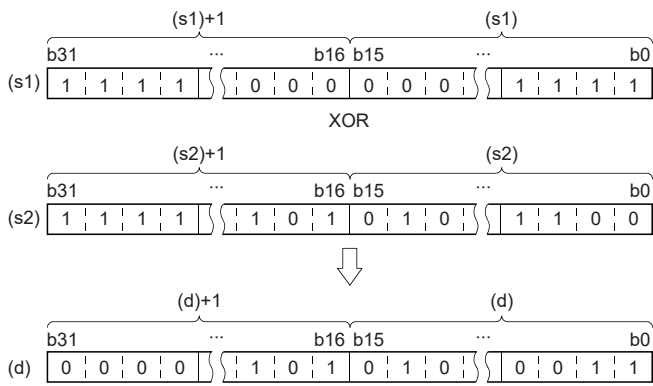
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 對 (s1) 中指定的元件的BIN32位元資料與 (s2) 中指定的元件的BIN32位元資料的各個位進行排他邏輯或運算，將結果儲存在 (d) 中指定的元件中。



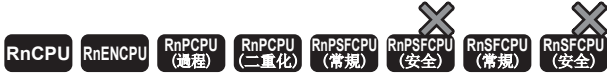
- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

16位元塊資料排他邏輯或

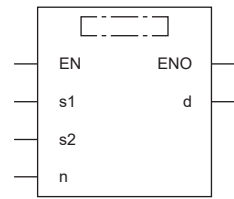
BKXOR (P)



進行指定的2個BIN16位元資料塊的排他邏輯或。

梯形圖	ST
	ENO:=BKXOR (EN, s1, s2, n, d) ; ENO:=BKXORP (EN, s1, s2, n, d) ;

FBD/LD



■執行條件

指令	執行條件
BKXOR	
BKXORP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存邏輯運算資料的起始元件	—	帶符號BIN16位元	ANY16*1
(s2)	邏輯運算資料或者儲存了邏輯運算資料的起始元件	-32768~32767	帶符號BIN16位元	ANY16*1
(d)	儲存運算結果的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	運算資料個數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

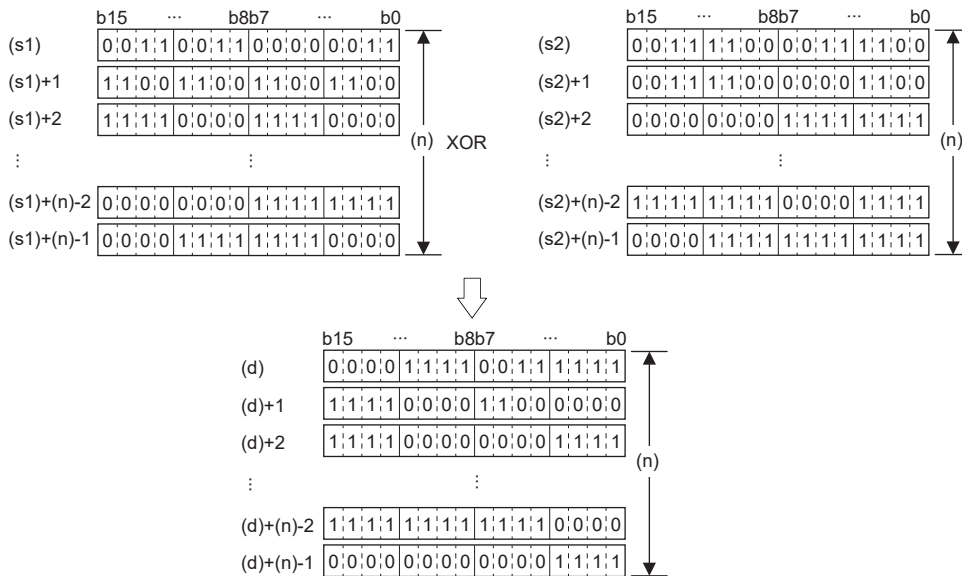
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)*1	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)*1	—	—	○	—	—	—	—	○	○	—	—	—	—
(d)*1	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

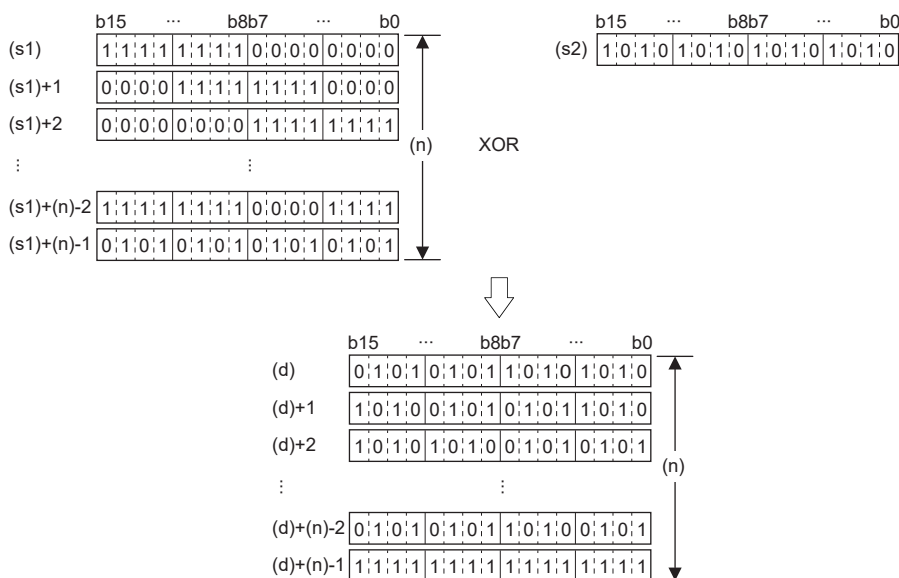
*1 (s1) 與 (d) 或 (s2) 與 (d) 可以指定同一元件編號。

功能

- 對(s1)中指定的元件開始的(n)點的內容與(s2)中指定的元件開始的(n)點的內容進行排他邏輯或運算，將結果儲存到(d)中指定的元件及其以後。



- (s2)中可以指定-32768~32767(帶符號BIN16位元)的常數。



出錯

出錯代碼(SD0)	內容
2821H	當(s1)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍的一部分重複時。 (s1)及(d)中指定了同一元件的情況下除外。)
	當(s2)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍的一部分重複時。 (s2)及(d)中指定了同一元件的情況下除外。)

16位元資料否定排他邏輯或

WXNR(P) [操作數為2個的情況下]

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN16位元資料的否定排他邏輯或。

梯形圖	ST
	不對應。 (☞ 323頁 WXNR(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 323頁 WXNR(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
WXNR	
WXNRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行否定排他邏輯或的資料或者儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存否定排他邏輯或的結果的元件	-32768~32767	帶符號BIN16位元	ANY16

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 對 (d) 中指定的元件的BIN16位元資料與 (s) 中指定的元件的BIN16位元資料進行否定排他邏輯或運算，將結果儲存到 (d) 中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

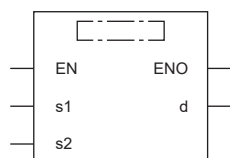
WXNR (P) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN16位元資料的否定排他邏輯或。

梯形圖	ST
	ENO:=WXNR (EN, s1, s2, d) ; ENO:=WXNRP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
WXNR	
WXNRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)、(s2)	進行否定排他邏輯或運算的資料或者儲存了資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存否定排他邏輯或的結果的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

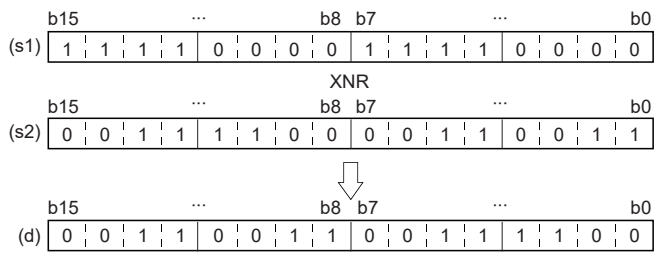
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 對(s1)中指定的元件的BIN16位元資料與(s2)中指定的元件的BIN16位元資料進行否定排他邏輯或運算，將結果儲存到(d)中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

32位元資料否定排他邏輯或

DXNR (P) [操作數為2個的情況下]

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN32位元資料的否定排他邏輯或。

梯形圖	ST
	不對應。 (☞ 327頁 DXNR (P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 327頁 DXNR (P) [操作數為3個的情況下])

■執行條件

指令	執行條件
DXNR	
DXNRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行否定排他邏輯或的資料或者儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存否定排他邏輯或的結果的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

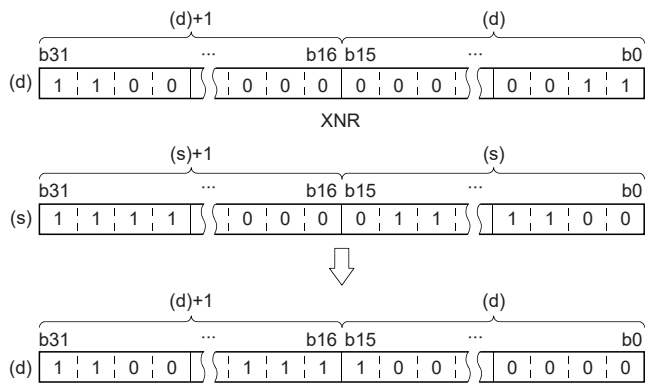
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	\$	
(s)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 對 (d) 中指定的元件的BIN32位元資料與 (s) 中指定的元件的BIN32位元資料進行否定排他邏輯或運算，將結果儲存到 (d) 中指定的元件中。



- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

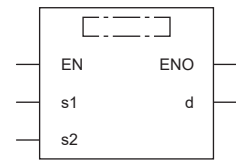
DXNR (P) [操作數為3個的情況下]

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

進行指定的2個BIN32位元資料的否定排他邏輯或。

梯形圖	ST
	ENO:=DXNR (EN, s1, s2, d) ; ENO:=DXNRP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
DXNR	
DXNRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)、(s2)	進行否定排他邏輯或運算的資料或者儲存了資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存否定排他邏輯或的結果的起始元件	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

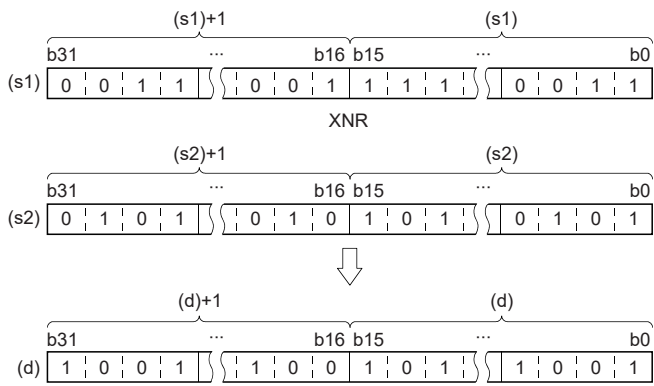
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s1)	○	○	○	○	○	○	○	○	○	—	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s1)	○	○	○
(s2)	○	○	○
(d)	○	○	—

功能

- 對(s1)中指定的元件的BIN32位元資料與(s2)中指定的元件的BIN32位元資料進行否定排他邏輯或運算，將結果儲存到(d)中指定的元件中。



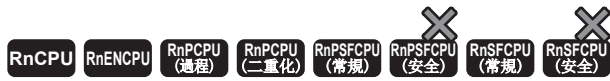
- 在位元元件的情況下，位指定的點數以後的位元元件將作為0進行運算。

出錯

沒有運算出錯。

16位元塊資料否定排他邏輯或

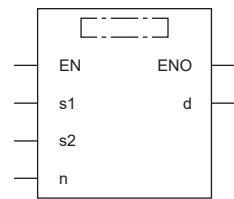
BKXNR (P)



進行指定的2個BIN16位元資料塊的否定排他邏輯或。

梯形圖	ST
	ENO:=BKXNR (EN, s1, s2, n, d) ; ENO:=BKXNRP (EN, s1, s2, n, d) ;

FBD/LD



執行條件

指令	執行條件
BKXNR	
BKXNRP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存邏輯運算資料的起始元件	—	帶符號BIN16位元	ANY16*1
(s2)	邏輯運算資料或者儲存了邏輯運算資料的起始元件	-32768~32767	帶符號BIN16位元	ANY16*1
(d)	儲存運算結果的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	運算資料個數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

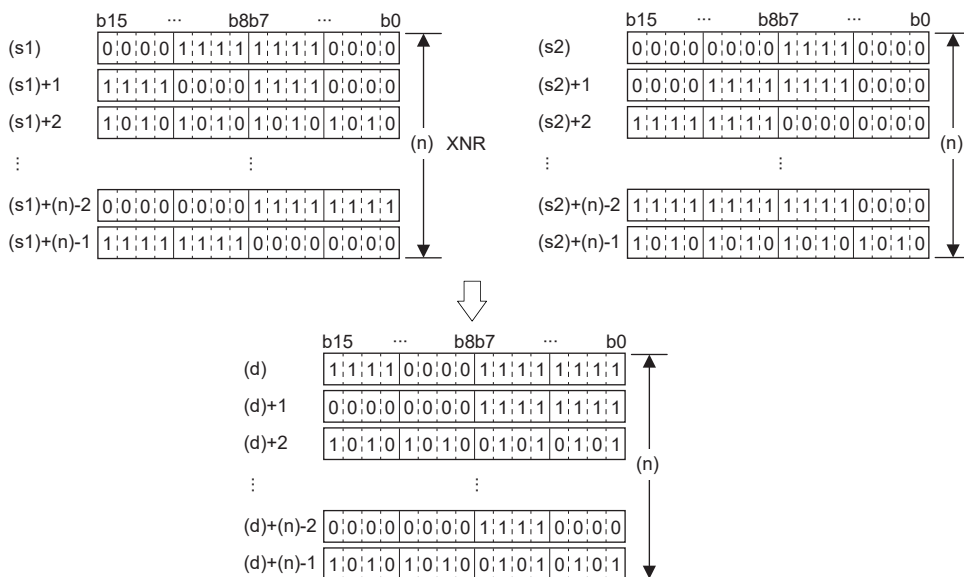
可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)*1	—	—	○	—	—	—	○	—	—	—	—	—
(s2)*1	—	—	○	—	—	—	○	○	—	—	—	—
(d)*1	—	—	○	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—	—

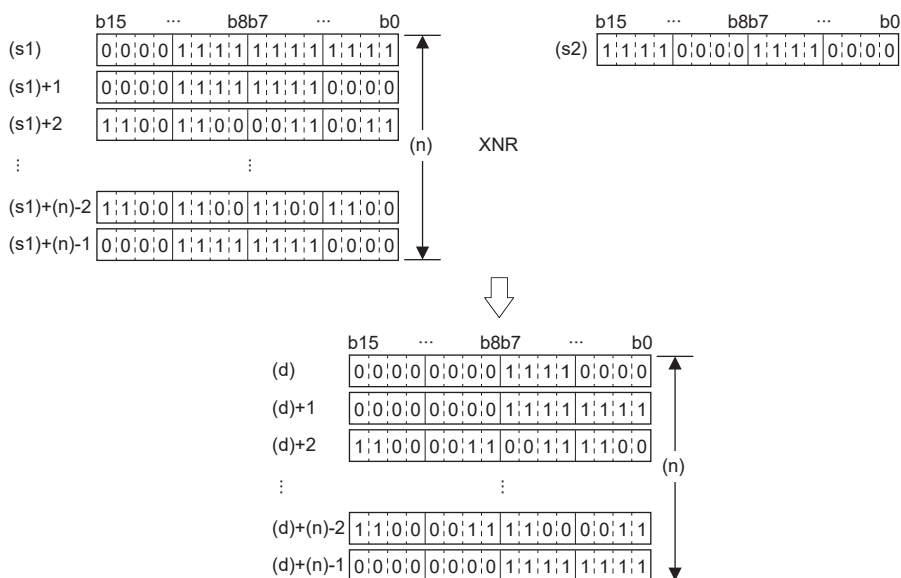
*1 (s1)與(d)或(s2)與(d)可以指定同一元件編號。

功能

- 對(s1)中指定的元件開始的(n)點的內容與(s2)中指定的元件開始的(n)點的內容進行否定排他邏輯或運算，將結果儲存到(d)中指定的元件以後。



- (s2)中可以指定-32768~32767(帶符號BIN16位元)的常數。



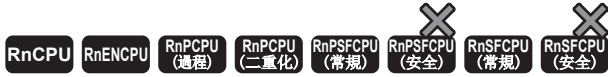
出錯

出錯代碼(SD0)	內容
2821H	當(s1)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍的一部分重複時。 (s1)及(d)中指定了同一元件的情況下除外。)
	當(s2)開始的(n)點的元件範圍與(d)開始的(n)點的元件範圍的一部分重複時。 (s2)及(d)中指定了同一元件的情況下除外。)

6.4 位元處理指令

字元元件的位設定

BSET (P)



對指定的字元元件內的第n位元進行(1)。

梯形圖	ST
	ENO:=BSET (EN, n, d) ; ENO:=BSETP (EN, n, d) ;

FBD/LD

■執行條件

指令	執行條件
BSET	
BSETP	

設置數據

■內容、範圍、資料類型

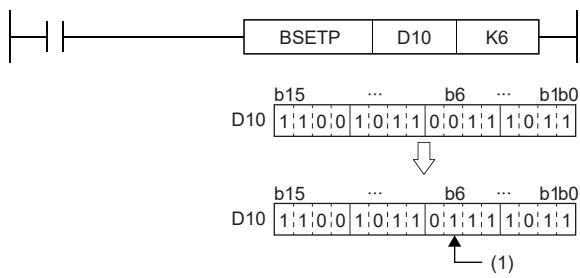
操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	進行位設定的元件	—	帶符號BIN16位元	ANY16
(n)	進行位元設定的位元的位置	0~15	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 對(d)中指定的字元元件的第(n)位元進行設定(1)。



(1) 設定(1)D10的第6位元。

- 在(n)中超過了15的情況下，以低位4位元的資料執行。

出錯

沒有運算出錯。

字元元件的位元重設

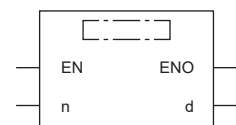
BRST (P)



對指定的字元元件內的第n位元進行重設(0)。

梯形圖	ST
	ENO:=BRST (EN, n, d) ; ENO:=BRSTP (EN, n, d) ;

FBD/LD



■執行條件

指令	執行條件
BRST	
BRSTP	

設置數據

■內容、範圍、資料類型

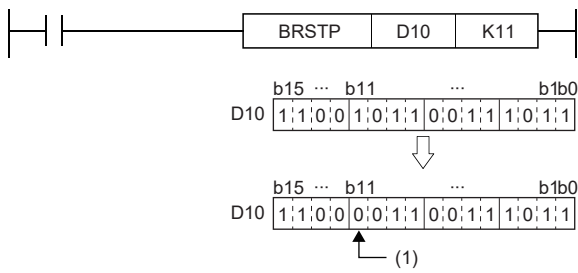
操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	進行位重設的元件	—	帶符號BIN16位元	ANY16
(n)	進行位重設的位元的位置	0~15	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(d)	○	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 對(d)中指定的字元元件的第(n)位元進行重設(0)。



(1) 重設(0)D10的第11位元。

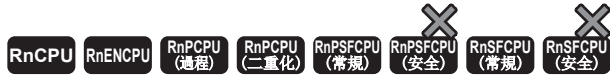
- 在(n)中超過了15的情況下，以低位4位元的資料執行。

出錯

沒有運算出錯。

16位元測試

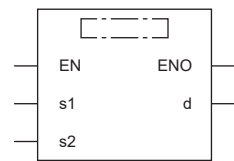
TEST (P)



對指定的字元元件內的第n位元進行提取。

梯形圖	ST
	ENO:=TEST (EN, s1, s2, d) ; ENO:=TESTP (EN, s1, s2, d) ;

FBD/LD



執行條件

指令	執行條件
TEST	
TESTP	

設置數據

內容、範圍、資料類型

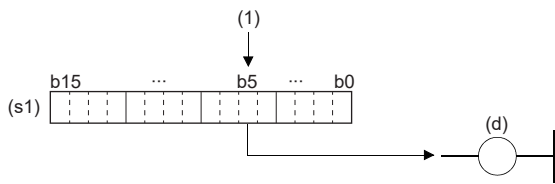
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	儲存了要提取的位元資料的元件編號	—	帶符號BIN16位元	ANY16
(s2)	要提取的位元資料的位置	0~15	無符號BIN16位元	ANY16
(d)	儲存已提取的位元資料的位元元件編號	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	○	○	○	○	—	—	○	—	—	—	—	
(s2)	—	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	—	—	—	○	—	—	—	—	

功能

- 從(s1)中指定的字元元件開始，提取(s2)中指定的位置的位元資料後，寫入到(d)中指定的位元元件中。



(1): (s2) 位元 ((s2)=5 的情況下)

- 對於(d)中指定的位元元件，相應位元為“0”時OFF、為“1”時ON。
- 在(s2)中，指定1字元資料的各個位元的位置(0~15)。在(s2)中指定了16及其以上的情況下， $(s2) \div 16$ 的餘數值將變為位元的位置。

例

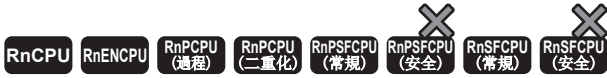
當(s2)=18時，因為 $18 \div 16 = 1$ 餘2，因此變為b2的資料。

出錯

沒有運算出錯。

32位元測試

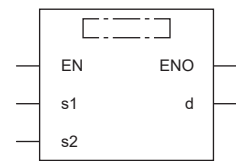
DTEST (P)



對指定的2字元元件內的第n位元進行提取。

梯形圖	ST
	ENO:=DTEST (EN, s1, s2, d) ; ENO:=DTESTP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
DTEST	
DTESTP	

設置數據

■內容、範圍、資料類型

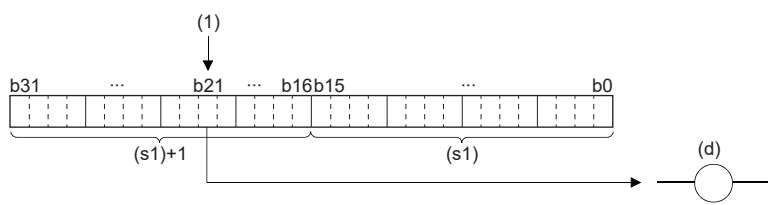
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	儲存了要提取的位元資料的元件編號	—	帶符號BIN32位元	ANY32
(s2)	要提取的位元資料的位置	0~31	無符號BIN16位元	ANY16
(d)	儲存已提取的位元資料的位元元件編號	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	○	○	○	○	○	○	○	—	—	—	—	—
(s2)	—	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	—	—	—	○	—	—	—	—	—

功能

- 從(s1)中指定的雙字元元件內，提取(s2)中指定的位置的位元資料後，寫入到(d)中指定的位元元件中。



(1): (s2)位((s2)=21的情況下)

- 對於(d)中指定的位元元件，相應位元為“0”時OFF、為“1”時ON。
- 在(s2)中，指定2字元資料的各個位元的位置(0~31)。在(s2)中指定了32及其以上的情況下， $(s2) \div 32$ 的餘數值將變為位元的位置。

例

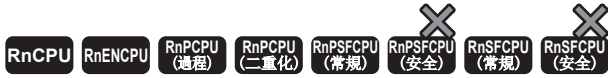
當(s2)=34時，因為 $34 \div 32 = 1$ 餘2，因此變為b2的資料。

出錯

沒有運算出錯。

位元元件的批量重設

BKRST (P)



對從指定的位元元件開始的n點的位元元件進行重設。

梯形圖	ST
	ENO:=BKRST (EN, n, d) ; ENO:=BKRSTP (EN, n, d) ;

FBD/LD



■執行條件

指令	執行條件
BKRST	
BKRSTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	進行重設的起始元件	—	位元	ANY_BOOL
(n)	進行重設的元件數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(d)	○	—	○	—	—	○	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 從(d)中指定的位元元件開始，對(n)點的位元元件進行重設。
- 位元元件的重設狀態如下所示。

元件	狀態
報警器(F)	<ul style="list-style-type: none">• 將從(d)中指定的報警器(F)編號開始的(n)點設為OFF。• 將變為OFF的報警器編號從SD64～SD79中刪除後，向前填充對齊。• 將SD64～SD79中儲存的報警器數儲存到SD63中。
定時器(T)、累計定時器(ST)、計數器(C)、超長定時器(LT)、超長累計定時器(LST)、超長計數器(LC)	<ul style="list-style-type: none">• 從(d)中指定的定時器(T)、累計定時器(ST)、計數器(C)、超長定時器(LT)、超長累計定時器(LST)、超長計數器(LC)編號中，將(n)點的當前值設為0、線圈觸點設為OFF。
上述以外的位元元件	<ul style="list-style-type: none">• 將(d)中指定的元件開始的(n)點的線圈、觸點設為OFF。

- 在指定的元件為OFF的情況下，元件的狀態不變化。

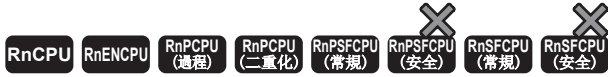
出錯

沒有運算出錯。

6.5 資料移位指令

BIN16位元資料的n位元右移

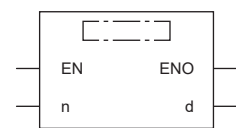
SFR(P)



將指定的元件的BIN16位元資料向右移動(n)位元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFR(EN, n, d); ENO:=SFRP(EN, n, d);

FBD/LD



■執行條件

指令	執行條件
SFR	
SFRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的元件	—	帶符號BIN16位元	ANY16
(n)	移位數	0~15	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	○*1	○	○	○	○	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—

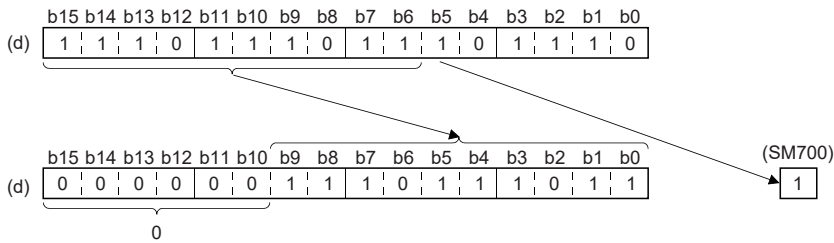
*1 不能使用FX、FY。

功能

- 將(d)中指定的元件的BIN16位元資料向右移動(n)位元。
- 從最高位開始的(n)位元將變為0。
- SM700中，將儲存移動範圍的右邊一個的1位元。

例

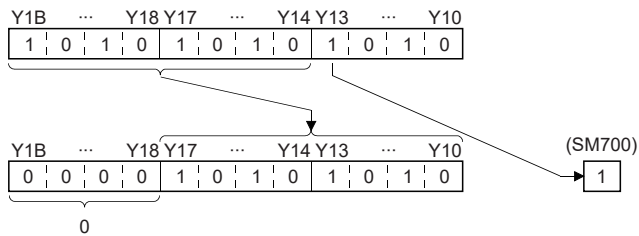
在(n)=6的情況下



- 在(d)中指定了位元元件的情況下，將以位指定中指定的元件範圍進行右移。

例

在(n)=4的情況下



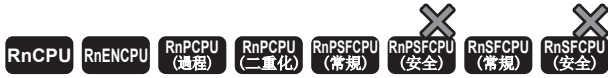
- 實際移位的位元數將變為 $(n) \div (\text{位指定中指定的點數})$ 的餘數。例如， $(n)=15$ 、(位指定中指定的點數)=8位元時，因為 $15 \div 8 = 1$ 餘7，因此移動7位元。
- (n) 的指定範圍為0~15。在 (n) 中指定了16或其以上的值的情況下，將以 $(n) \div 16$ 的餘數值進行右移。例如， $(n)=18$ 時，因為 $18 \div 16 = 1$ 餘2，因此右移2位元。

出錯

沒有運算出錯。

BIN16位元資料的n位元左移

SFL(P)



將指定的元件的BIN16位元資料向左移動(n)位元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFL(EN, n, d); ENO:=SFLP(EN, n, d);

FBD/LD

■執行條件

指令	執行條件
SFL	
SFLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的元件	—	帶符號BIN16位元	ANY16
(n)	移位數	0~15	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(d)	○*1	○	○	○	○	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—

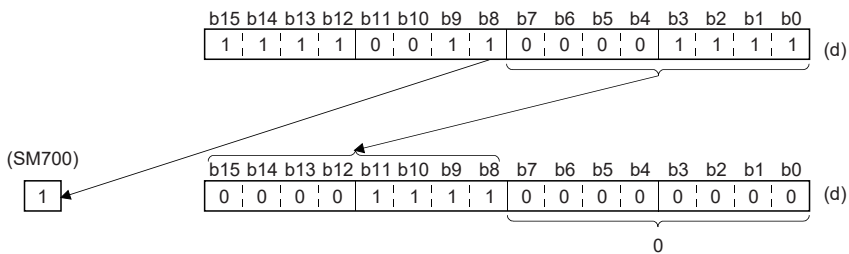
*1 不能使用FX、FY。

功能

- 將(d)中指定的元件的BIN16位元資料向左移動(n)位元。
- 從最低位元開始的(n)位元將變為0。
- SM700中，將儲存移動範圍的左邊一個的1位元。

例

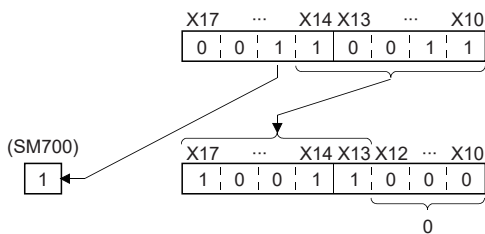
在(n)=8的情況下



- 在(d)中指定了位元元件的情況下，將以位指定中指定的元件範圍進行左移。

例

在(n)=5的情況下



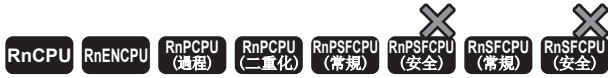
- 實際移位的位元數將變為 $(n) \div (\text{位指定中指定的點數})$ 的餘數。例如， $(n)=15$ 、 $(\text{位指定中指定的點數})=8$ 位元時，因為 $15 \div 8 = 1$ 餘 7，因此移動7位元。
- (n) 的指定範圍為0~15。在 (n) 中指定了16或其以上的值的情況下，將以 $(n) \div 16$ 的餘數值進行左移。例如， $(n)=18$ 時，因為 $18 \div 16 = 1$ 餘 2，因此左移2位元。

出錯

沒有運算出錯。

n位元資料的1位元右移

BSFR (P)



將從指定的元件開始的 (n) 點向右移動1位元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=BSFR (EN, n, d) ; ENO:=BSFRP (EN, n, d) ;

FBD/LD

■執行條件

指令	執行條件
BSFR	
BSFRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	移位的元件	—	位元	ANY_BOOL
(n)	移位的資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

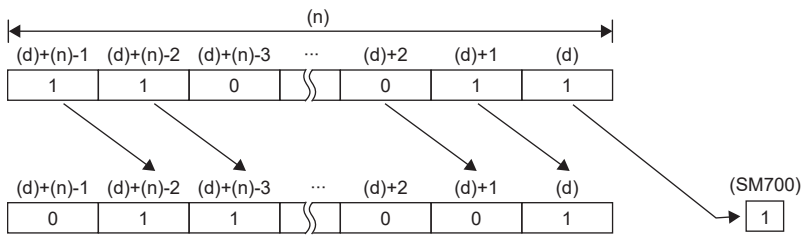
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(d)	○*1	—	○	—	—	—	○	—	—	—	—	—
(n)	○*2	○	○	○	○	—	○	○	—	—	—	—

*1 不能使用T、C、ST。

*2 不能使用FX、FY。

功能

- 將從(d)中指定的元件開始的(n)點向右移動1位元。
- 從最高位開始的1位元將變為0。
- SM700中，將儲存移動範圍的右邊一個的1位元。



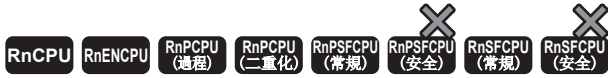
- (n)中指定的值為0時將變為無處理。

出錯

沒有運算出錯。

n位元資料的1位元左移

BSFL (P)



將從指定的元件開始的 (n) 點向左移動1位元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=BSFL (EN, n, d) ; ENO:=BSFLP (EN, n, d) ;

FBD/LD

■執行條件

指令	執行條件
BSFL	
BSFLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	移位的元件	—	位元	ANY_BOOL
(n)	移位的資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

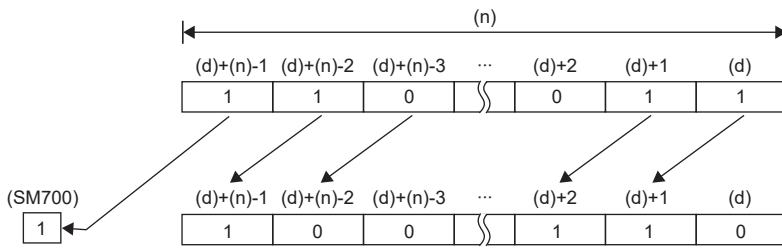
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(d)	○*1	—	○	—	—	—	○	—	—	—	—
(n)	○*2	○	○	○	○	—	○	○	—	—	—

*1 不能使用T、C、ST。

*2 不能使用FX、FY。

功能

- 將從(d)中指定的元件開始的(n)點向左移動1位元。
- 從最低位開始的1位元將變為0。
- SM700中，將儲存移動範圍的左邊一個的1位元。



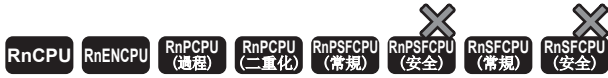
- (n) 中指定的值為0時將變為無處理。

出錯

沒有運算出錯。

n字元資料的1字元右移

DSFR (P)



將從指定的元件開始的 (n) 點向右移動1字元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=DSFR (EN, n, d) ; ENO:=DSFRP (EN, n, d) ;

FBD/LD

■執行條件

指令	執行條件
DSFR	
DSFRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	移位的元件	—	字元	ANY16
(n)	移位的資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

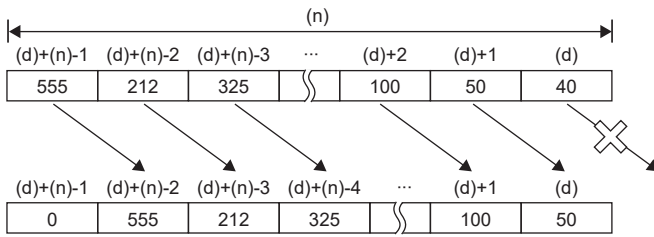
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—

*1 不能使用FX、FY。

功能

- 將(d)中指定的元件開始的(n)點的資料向右移動1字元。
- 最高位開始的1字元將變為0。



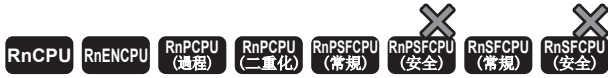
- (n)中指定的值為0時將變為無處理。

出錯

沒有運算出錯。

n字元資料的1字元左移

DSFL (P)



將指定的元件開始的(n)點向左移動1字元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=DSFL(EN, n, d); ENO:=DSFLP(EN, n, d);

FBD/LD

■執行條件

指令	執行條件
DSFL	
DSFLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的元件	—	字元	ANY16
(n)	移位的資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

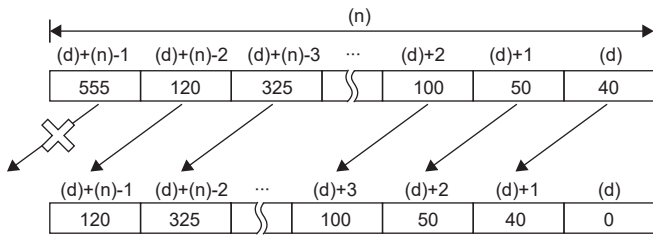
■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—

*1 不能使用FX、FY。

功能

- 將(d)中指定的元件開始的(n)點的資料向左移動1字元。
- 從最低位開始的1字元將變為0。

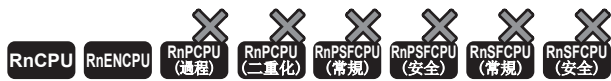


出錯

沒有運算出錯。

n雙字元資料的1雙字元右移

DDSF_R(P)

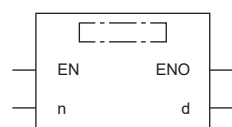


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始將n雙字元向右移1雙字元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=DDSF _R (EN, n, d); ENO:=DDSF _R P(EN, n, d);

FBD/LD



■執行條件

指令	執行條件
DDSF _R	
DDSF _R P	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	雙字元	ANY32
(n)	移位的資料數	0~65535	無符號BIN16位元	ANY16

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(d)	—	—	○	—	—	—	○	—	—	—	—	—
(n)	○*1	○	○	○	○	—	○	○	—	—	—	—

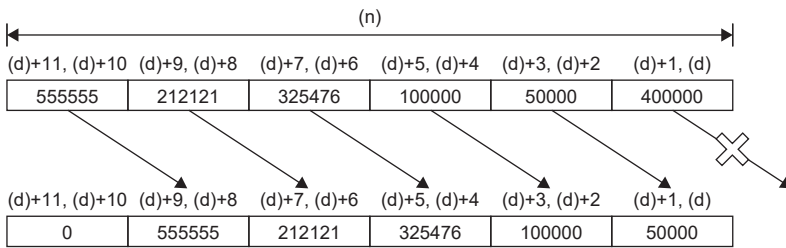
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始將(n)雙字元向右移1雙字元。
- 從最高位開始的1雙字元將變為0。
- 在(n)中指定的值為0的情況下將變為無處理。

例

在(n)=6的情況下

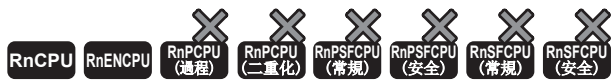


出錯

沒有運算出錯。

n雙字元資料的1雙字元左移

DDSFL (P)

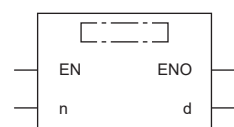


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始將(n)雙字元向左移1雙字元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=DDSFL(EN, n, d); ENO:=DDSFLP(EN, n, d);

FBD/LD



■執行條件

指令	執行條件
DDSFL	
DDSFLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	雙字元	ANY32
(n)	移位的資料數	0~65535	無符號BIN16位元	ANY16

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\(\H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	

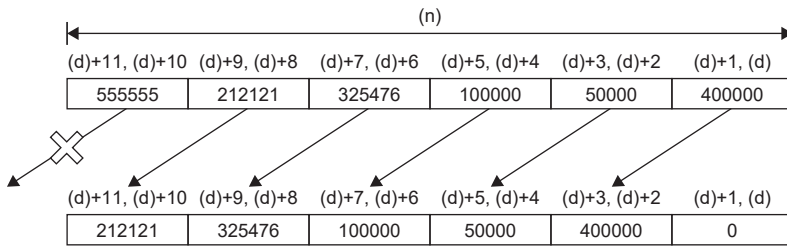
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始將(n)雙字元向左移1雙字元。
- 從最低位開始的1雙字元將變為0。
- 在(n)中指定的值為0的情況下將變為無處理。

例

在(n)=6的情況下

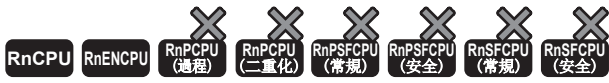


出錯

沒有運算出錯。

單精度實數n點的1點右移

ESFR (P)

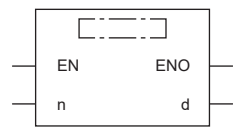


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始將單精度實數(n)點的資料向右移1點。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=ESFR (EN, n, d) ; ENO:=ESFRP (EN, n, d) ;

FBD/LD



■執行條件

指令	執行條件
ESFR	
ESFRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	單精度實數	ANYREAL_32
(n)	移位的資料數	0~65535	無符號BIN16位元	ANY16

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	

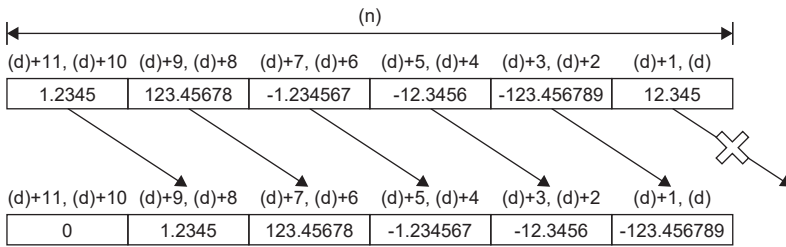
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始將單精度實數(n)點向右移1點。
- 從最高位開始的1單精度實數將變為0。
- 在(n)中指定的值為0的情況下將變為無處理。

例

在(n)=6的情況下

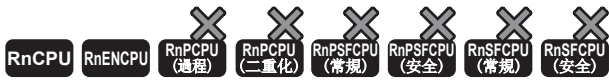


出錯

沒有運算出錯。

單精度實數n點的1點左移

ESFL (P)



- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始將單精度實數(n)點向左移1點。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=ESFL(EN, n, d); ENO:=ESFLP(EN, n, d);

FBD/LD

■執行條件

指令	執行條件
ESFL	
ESFLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	單精度實數	ANYREAL_32
(n)	移位的資料數	0~65535	無符號BIN16位元	ANY16

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	

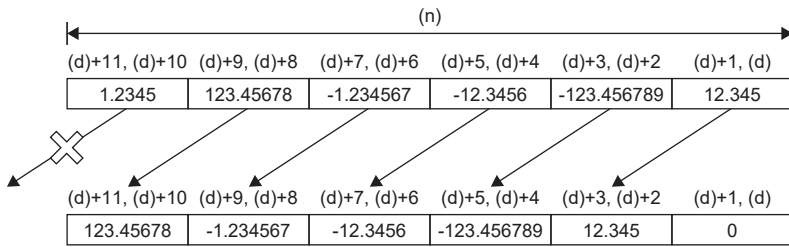
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始將單精度實數(n)點向左移1點。
- 從最低位開始的1單精度實數將變為0。
- 在(n)中指定的值為0的情況下將變為無處理。

例

在(n)=6的情況下

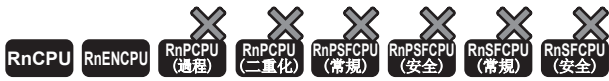


出錯

沒有運算出錯。

雙精度實數n點的1點右移

EDSFR (P)

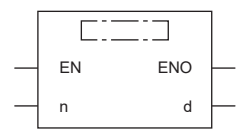


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始將雙精度實數(n)點的資料向右移1點。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=EDSFR(EN, n, d); ENO:=EDSFRP(EN, n, d);

FBD/LD



■執行條件

指令	執行條件
EDSFR	
EDSFRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	雙精度實數	ANYREAL_64
(n)	移位的資料數	0~65535	無符號BIN16位元	ANY16

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	

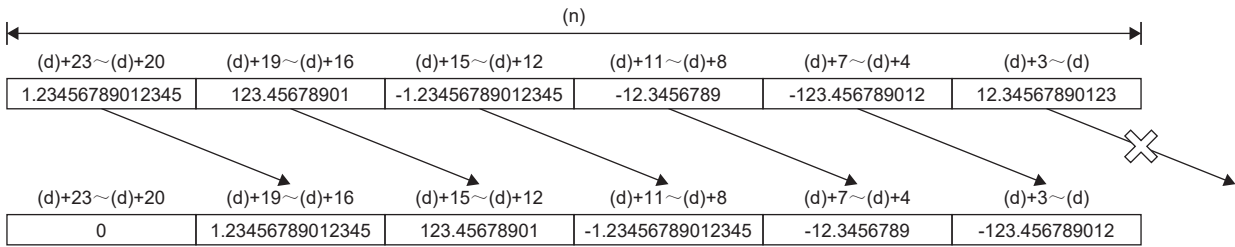
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始將雙精度實數(n)點向右移1點。
- 從最高位開始的1雙精度實數將變為0。
- 在(n)中指定的值為0的情況下將變為無處理。

例

在(n)=6的情況下

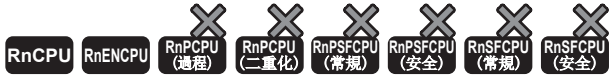


出錯

沒有運算出錯。

雙精度實數n點的1點左移

EDSFL (P)

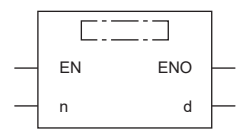


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始將雙精度實數(n)點向左移1點。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=EDSFL (EN, n, d) ; ENO:=EDSFLP (EN, n, d) ;

FBD/LD



■執行條件

指令	執行條件
EDSFL	
EDSFLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	雙精度實數	ANYREAL_64
(n)	移位的資料數	0~65535	無符號BIN16位元	ANY16

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	

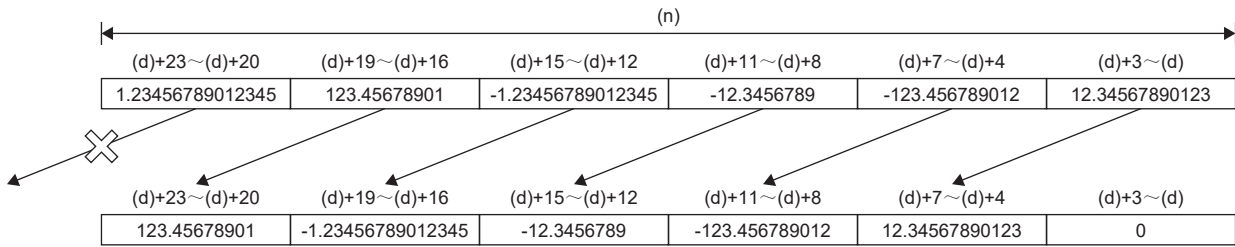
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始將雙精度實數(n)點向左移1點。
- 從最低位開始的1雙精度實數將變為0。
- 在(n)中指定的值為0的情況下將變為無處理。

例

在(n)=6的情況下



出錯

沒有運算出錯。

n位元資料的n位元右移

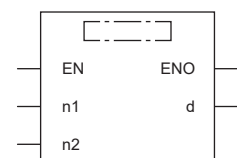
SFTBR (P)



從指定的元件開始在 (n1) 位元的資料範圍內向右移動n2位元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFTBR (EN, n1, n2, d) ; ENO:=SFTBRP (EN, n1, n2, d) ;

FBD/LD



執行條件

指令	執行條件
SFTBR	
SFTBRP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的元件	—	位元	ANY_BOOL
(n1)	移位的資料數	0~64	無符號BIN16位元	ANY16
(n2)	移位數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(d)	○*1	—	○	—	—	—	○	—	—	—	—
(n1)	○*2	○	○	○	○	—	○	○	—	—	—
(n2)	○*2	○	○	○	○	—	○	○	—	—	—

*1 不能使用T、C、ST。

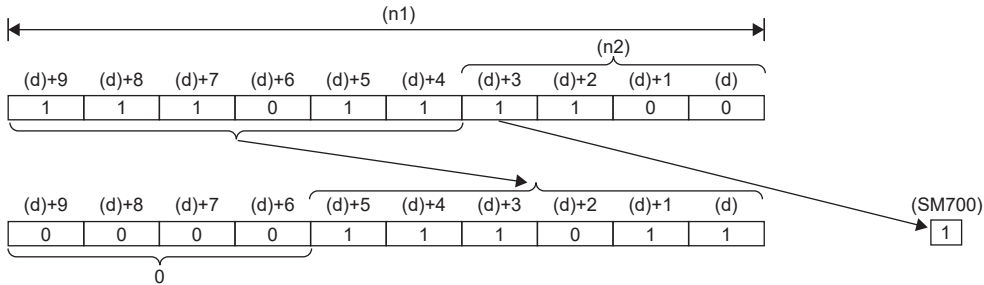
*2 不能使用FX、FY。

功能

- 從在(d)中指定的元件開始在(n1)位元的資料範圍內向右移動(n2)位元。
- SM700中，將儲存移動範圍的右邊一個的1位元。

例

在(n1)=10、(n2)=4的情況下



- (n1)、(n2)按(n1)>(n2)的方式指定。在(n1)≤(n2)的情況下，將以(n2)÷(n1)的餘數值進行移位。但是，(n2)÷(n1)的餘數值為0時，將變為無處理。
- (n1)的設定範圍為1~64。
- 從最高位開始的(n2)位元將變為0。在(n1)<(n2)的情況下，(n2)÷(n1)的餘數值部分將變為0。
- 當(n1)或(n2)中指定的值為0時，將變為無處理。

出錯

出錯代碼(SD0)	內容
3405H	當(n1)超出0~64的範圍時。

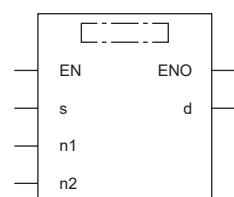
SFTR (P)



- R00CPU、R01CPU、R02CPU沒有版本的限制。
 - RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 從指定的元件開始在(n1)位的資料範圍內向右移動(n2)位。移位後的空餘資料將儲存指定資料。

梯形圖	ST
	ENO:=SFTR (EN, s, n1, n2, d); ENO:=SFTRP (EN, s, n1, n2, d);

FBD/LD



執行條件

指令	執行條件
SFTR	
SFTRP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存至移位後空餘資料的起始元件	—	位元	ANY_BOOL
(d)	移位的起始元件	—	位元	ANY_BOOL
(n1)*1	移位的資料數	0~65535	無符號BIN16位元	ANY16_U
(n2)*1	移位數	0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 應設定成 $(n2) \leq (n1)$ 。

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	—	○*1	—	—	—	—	○	○*2	—	—	—	—
(d)	○	—	○*1	—	—	—	—	○	—	—	—	—	—
(n1)	○*3	○	○	○	○	—	—	○	○	—	—	—	—
(n2)	○*3	○	○	○	○	—	—	○	○	—	—	—	—

*1 不能使用T、ST、C、FD。

*2 只能使用0或1。

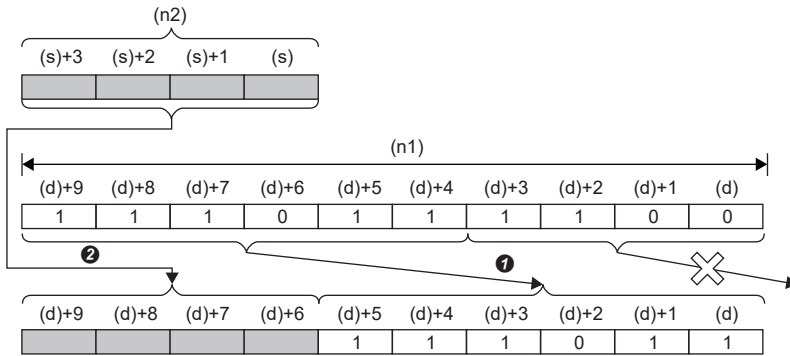
*3 不能使用FX、FY。

功能

- 從在(d)中指定的元件開始在(n1)位元的資料範圍內向右移動(n2)位元。移位後，空餘資料部分將儲存(s)開始的(n2)點。
- 在(s)中指定了常數0時，移位後的最高位開始的(n2)點將變為0。
- 在(s)中指定了常數1時，移位後的最高位開始的(n2)點將變為1。
- (n2)中指定的值為0的情況下，將變為無處理。

例

在(n1)=10、(n2)=4的情況下



① (n2)位元，向右移位

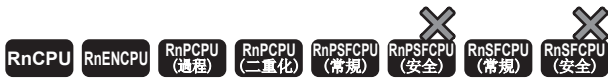
② 複製

出錯

出錯代碼 (SD0)	內容
2821H	當(s)開始的(n2)點的範圍與(d)開始的(n1)點的範圍重複時。
3405H	在(s)的常數指定時，指定了0或1以外的常數。 (n1)、(n2)中指定的值為(n1) < (n2)時。

n位資料的n位元左移

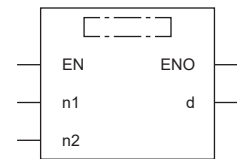
SFTBL (P)



從指定的元件開始在 (n1) 位元的資料範圍內向左移動 (n2) 位元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFTBL (EN, n1, n2, d) ; ENO:=SFTBLP (EN, n1, n2, d) ;

FBD/LD



執行條件

指令	執行條件
SFTBL	
SFTBLP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的元件	—	位元	ANY_BOOL
(n1)	移位的資料數	0~64	無符號BIN16位元	ANY16
(n2)	移位數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(d)	○*1	—	○	—	—	—	○	—	—	—	—
(n1)	○*2	○	○	○	○	—	○	○	—	—	—
(n2)	○*2	○	○	○	○	—	○	○	—	—	—

*1 不能使用T、C、ST。

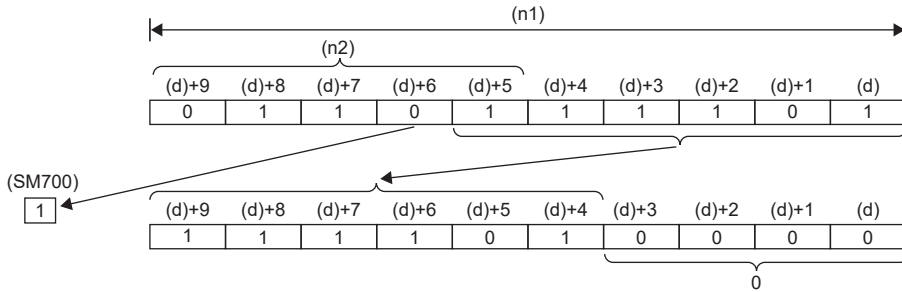
*2 不能使用FX、FY。

功能

- 從在(d)中指定的元件開始在(n1)位元的資料範圍內向左移動(n2)位元。
- SM700中，將儲存移動範圍的左邊一個的1位元。

例

在(n1)=10、(n2)=4的情況下

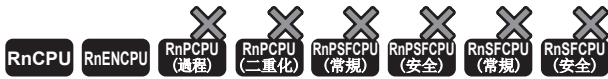


- (n1)、(n2)按(n1)>(n2)的方式指定。在(n1)≤(n2)的情況下，將以(n2)÷(n1)的餘數值進行移位。但是，(n2)÷(n1)的餘數值為0時，將變為無處理。
- (n1)的設定範圍為1~64。
- 從最低位開始的(n2)位將變為0。在(n1)<(n2)的情況下，(n2)÷(n1)的餘數值部分將變為0。
- 當(n1)或(n2)中指定的值為0時，將變為無處理。

出錯

出錯代碼(SD0)	內容
3405H	當(n1)超出0~64的範圍時。

SFTL (P)

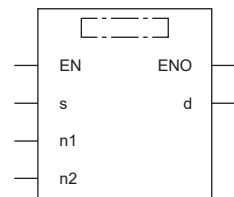


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

從指定的元件開始在(n1)位元的資料範圍內向左移動(n2)位元。移位後的空餘資料將儲存指定資料。

梯形圖	ST
	ENO:=SFTL (EN, s, n1, n2, d); ENO:=SFTLP (EN, s, n1, n2, d);

FBD/LD



■執行條件

指令	執行條件
SFTL	
SFTLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存至移位後空餘資料的起始元件	—	位元	ANY_BOOL
(d)	移位的起始元件	—	位元	ANY_BOOL
(n1)*1	移位的資料數	0~65535	無符號BIN16位元	ANY16_U
(n2)*1	移位數	0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 應設定成 $(n2) \leq (n1)$ 。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	—	○*1	—	—	—	—	○	○*2	—	—	—	—
(d)	○	—	○*1	—	—	—	—	○	—	—	—	—	—
(n1)	○*3	○	○	○	○	—	—	○	○	—	—	—	—
(n2)	○*3	○	○	○	○	—	—	○	○	—	—	—	—

*1 不能使用T、ST、C、FD。

*2 只能使用0或1。

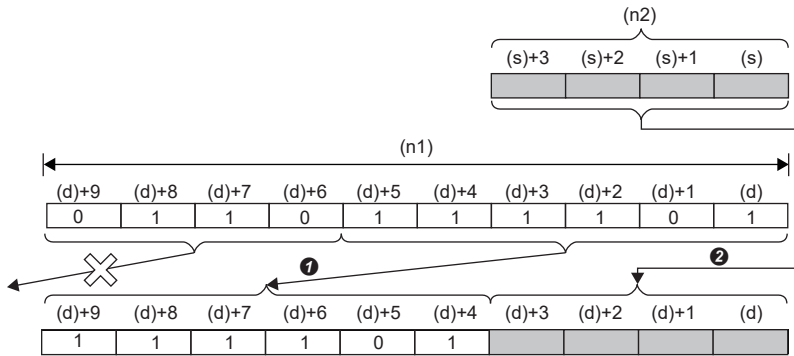
*3 不能使用FX、FY。

功能

- 從在(d)中指定的元件開始在(n1)位元的資料範圍內向左移動(n2)位元。移位後，空餘資料部分將儲存(s)開始的(n2)點。
- 在(s)中指定了常數0時，移位後的最低位開始(n2)點將變為0。
- 在(s)中指定了常數1時，移位後的最低位開始(n2)點將變為1。
- (n2)中指定的值為0的情況下，將變為無處理。

例

在(n1)=10、(n2)=4的情況下



① (n2) 位，向左移位

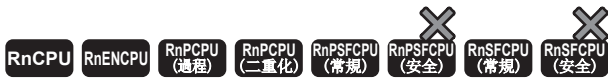
② 複製

出錯

出錯代碼 (SD0)	內容
2821H	當 (s) 開始的 (n2) 點的範圍與 (d) 開始的 (n1) 點的範圍重複時。
3405H	在 (s) 的常數指定時，指定了 0 或 1 以外的常數。 (n1)、(n2) 中指定的值為 (n1) < (n2) 時。

n字資料的n字元右移

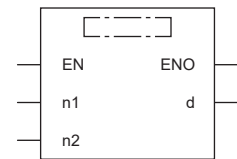
SFTWR (P)



從指定的元件開始在 (n1) 字元的資料範圍內向右移動 (n2) 字元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFTWR (EN, n1, n2, d) ; ENO:=SFTWRP (EN, n1, n2, d) ;

FBD/LD



■執行條件

指令	執行條件
SFTWR	
SFTWRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的元件	—	字元	ANY16
(n1)	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)	移位數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(d)	—	—	○	—	—	—	○	—	—	—	—	—
(n1)	○*1	○	○	○	○	—	○	○	—	—	—	—
(n2)	○*1	○	○	○	○	—	○	○	—	—	—	—

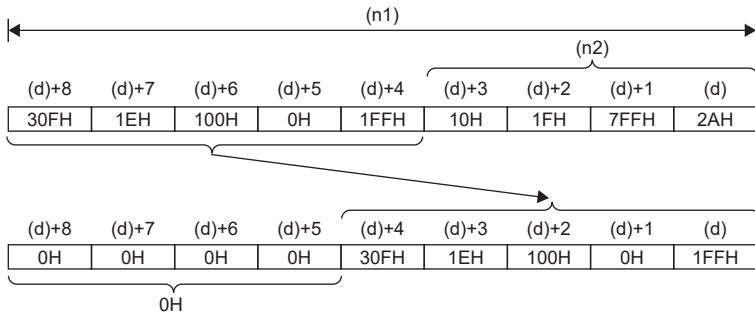
*1 不能使用FX、FY。

功能

- 從在(d)中指定的元件開始在(n1)字元的資料範圍內向右移動(n2)字元。
- 最高位開始的(n2)字元將變為0H。
- 當(n1)或(n2)中指定的值為0H時，將變為無處理。
- 在 $(n1) \leq (n2)$ 的情況下，(d)中指定的元件開始的(n1)字元的資料將全部變為0H。

例

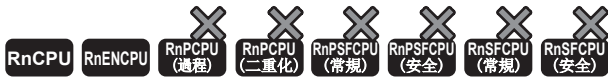
在 $(n1)=9$ 、 $(n2)=4$ 的情況下



出錯

沒有運算出錯。

WSFR (P)

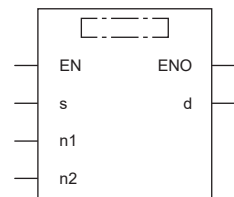


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

從指定的元件開始在(n1)字元的資料範圍內向右移動(n2)字元。移位後的空餘資料將儲存指定資料。

梯形圖	ST
	<pre>ENO:=WSFR (EN, s, n1, n2, d); ENO:=WSFRP (EN, s, n1, n2, d);</pre>

FBD/LD



■執行條件

指令	執行條件
WSFR	
WSFRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存至移位後空餘資料的起始元件	—	無符號BIN16位元	ANY16
(d)	移位的起始元件	—	字元	ANY16
(n1)*1	移位的資料數	0~65535	無符號BIN16位元	ANY16_U
(n2)*1	移位數	0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 應設定成 $(n2) \leq (n1)$ 。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(s)	○*1	—	○	—	—	—	○	○	—	—	—
(d)	○*1	—	○	—	—	—	○	—	—	—	—
(n1)	○*1	○	○	○	○	—	○	○	—	—	—
(n2)	○*1	○	○	○	○	—	○	○	—	—	—

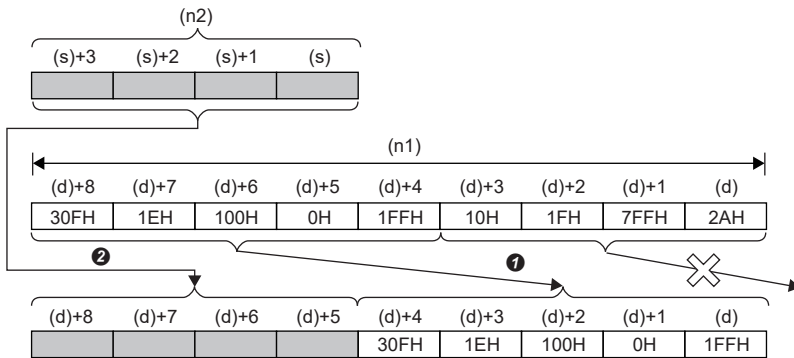
*1 不能使用FX、FY。

功能

- 從在(d)中指定的元件開始在(n1)字元的資料範圍內向右移動(n2)字元。移位後，空餘資料部分將儲存(s)開始的(n2)點。
- 在(s)中指定了常數時，移位後的最高位開始的(n2)點將變為指定的值。
- (n2)中指定的值為0的情況下，將變為無處理。

例

在(n1)=9、(n2)=4的情況下



① (n2) 字元，向右移位

② 複製

出錯

出錯代碼 (SD0)	內容
2821H	當(s)開始的(n2)點的範圍與(d)開始的(n1)點的範圍重複時。
3405H	(n1)、(n2)中指定的值為(n1) < (n2)時。

注意事項

以位元的位數指定來指定(s)和(d)時，應使(s)的位數與(d)的位數一致。

n字元資料的n字元左移

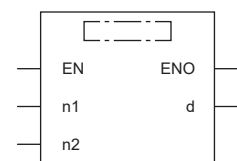
SFTWL (P)



從指定的元件開始在 (n1) 字元的資料範圍內向左移動 (n2) 字元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFTWL (EN, n1, n2, d) ; ENO:=SFTWLP (EN, n1, n2, d) ;

FBD/LD



執行條件

指令	執行條件
SFTWL	
SFTWLP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的元件	—	字元	ANY16
(n1)	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)	移位數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(d)	—	—	○	—	—	—	○	—	—	—	—	—
(n1)	○*1	○	○	○	○	—	○	○	—	—	—	—
(n2)	○*1	○	○	○	○	—	○	○	—	—	—	—

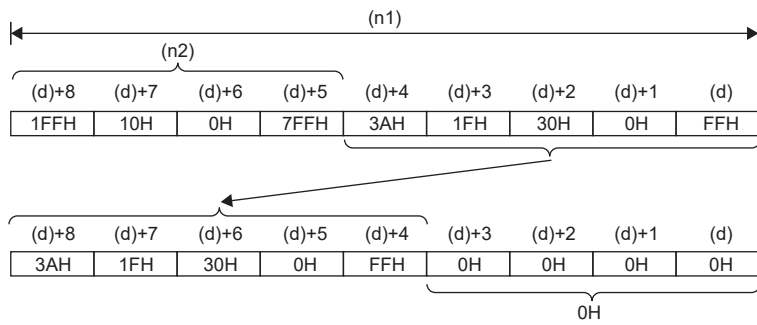
*1 不能使用FX、FY。

功能

- 從在(d)中指定的元件開始在(n1)字元的資料範圍內左右移動(n2)字元。
- 最低位開始的(n2)字元將變為0H。
- 當(n1)或(n2)中指定的值為0H時，將變為無處理。
- 在 $(n1) \leq (n2)$ 的情況下，(d)中指定的元件開始的(n1)字元的資料將全部變為0H。

例

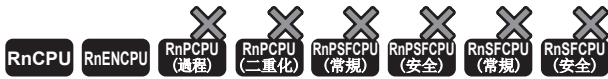
在 $(n1)=9$ 、 $(n2)=4$ 的情況下



出錯

沒有運算出錯。

WSFL (P)

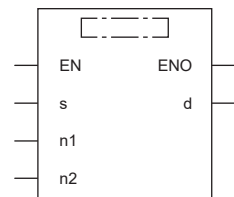


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

從指定的元件開始在(n1)字元的資料範圍內向左移動(n2)字元。移位後的空餘資料將儲存指定資料。

梯形圖	ST
	ENO:=WSFL (EN, s, n1, n2, d); ENO:=WSFLP (EN, s, n1, n2, d);

FBD/LD



■執行條件

指令	執行條件
WSFL	
WSFLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存至移位後空餘資料的起始元件	—	無符號BIN16位元	ANY16
(d)	移位的起始元件	—	字元	ANY16
(n1)*1	移位的資料數	0~65535	無符號BIN16位元	ANY16_U
(n2)*1	移位數	0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 應設定成 $(n2) \leq (n1)$ 。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s)	○*1	—	○	—	—	—	—	○	○	—	—	—
(d)	○*1	—	○	—	—	—	—	○	—	—	—	—
(n1)	○*1	○	○	○	○	—	—	○	○	—	—	—
(n2)	○*1	○	○	○	○	—	—	○	○	—	—	—

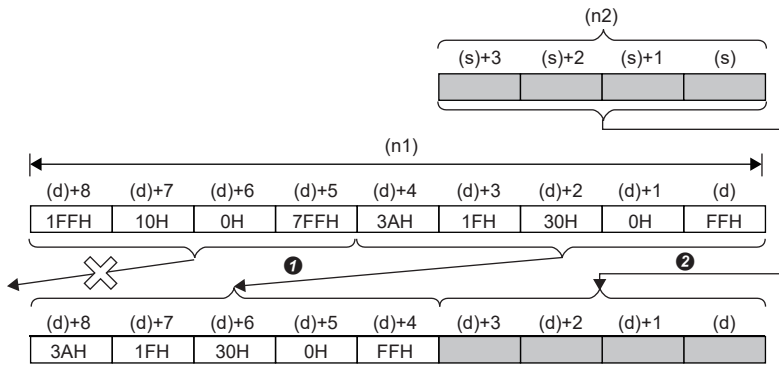
*1 不能使用FX、FY。

功能

- 從在(d)中指定的元件開始在(n1)字元的資料範圍內左右移動(n2)字元。移位後，空餘資料部分將儲存(s)開始的(n2)點。
- 在(s)中指定了常數時，移位後的最低位開始的(n2)點將變為指定的值。
- (n2)中指定的值為0的情況下，將變為無處理。

例

在(n1)=9、(n2)=4的情況下



- ① (n2) 字元，向左移位
- ② 複製

出錯

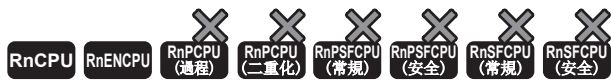
出錯代碼 (SD0)	內容
2821H	當 (s) 開始的 (n2) 點的範圍與 (d) 開始的 (n1) 點的範圍重複時。
3405H	(n1)、(n2) 中指定的值為 (n1) < (n2) 時。

注意事項

以位元的位數指定來指定 (s) 和 (d) 時，應使 (s) 的位數與 (d) 的位數一致。

n雙字元資料的n雙字元右移

SFTDWR (P)

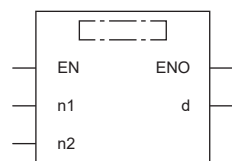


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在(n1)雙字元的資料範圍內向右移(n2)雙字元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFTDWR (EN, n1, n2, d); ENO:=SFTDWRP (EN, n1, n2, d);

FBD/LD



■執行條件

指令	執行條件
SFTDWR	
SFTDWRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	雙字元	ANY32
(n1)	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)	移位數	0~65535	無符號BIN16位元	ANY16

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	—	—	○	—	—	—	○	—	—	—	—	
(n1)	○*1	○	○	○	○	—	○	○	—	—	—	
(n2)	○*1	○	○	○	○	—	○	○	—	—	—	

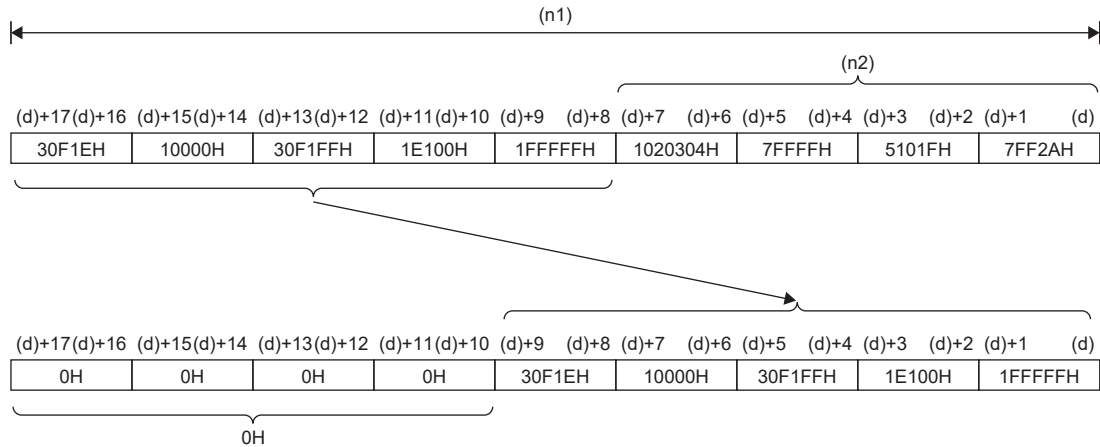
*1 不能使用FX、FY。

功能

- 從 (d) 中指定的元件開始在 (n1) 雙字元的資料範圍內向右移 (n2) 雙字元。
- 最高位開始的 (n2) 雙字元將變為 0H。
- 當 (n1) 或 (n2) 中指定的值為 0 時，將變為無處理。
- 在 $(n1) \leq (n2)$ 的情況下，(d) 中指定的元件開始的 (n1) 雙字元的資料將全部變為 0H。

例

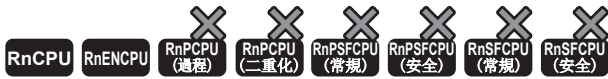
在 $(n1)=9$ 、 $(n2)=4$ 的情況下



出錯

沒有運算出錯。

DWSFTR (P)

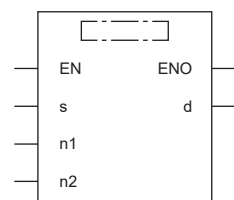


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在(n1)雙字元的資料範圍內向右移(n2)雙字元。移位後的空餘資料將儲存指定資料。

梯形圖	ST
	ENO:=DWSFTR (EN, s, n1, n2, d); ENO:=DWSFTRP (EN, s, n1, n2, d);

FBD/LD



■執行條件

指令	執行條件
DWSFTR	
DWSFTRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存至移位後空餘資料的起始元件	—	雙字元	ANY32
(d)	移位的起始元件	—	雙字元	ANY32
(n1) ^{*1}	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2) ^{*1}	移位數	0~65535	無符號BIN16位元	ANY16

*1 應設定成(n2)≤(n1)。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○ ^{*1}	—	○	—	—	—	—	○	○	—	—	—	
(d)	○ ^{*1}	—	○	—	—	—	—	○	—	—	—	—	
(n1)	○ ^{*1}	○	○	○	○	—	—	○	○	—	—	—	
(n2)	○ ^{*1}	○	○	○	○	—	—	○	○	—	—	—	

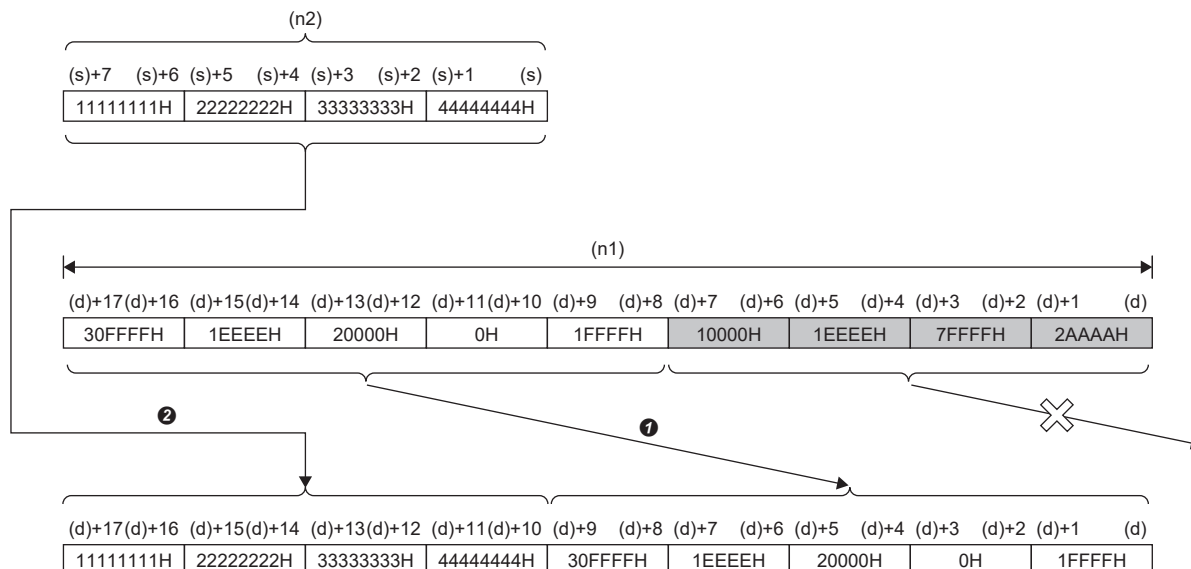
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始在(n1)雙字元的資料範圍內向右移(n2)雙字元。移位後，空餘資料部分將儲存(s)開始的(n2)點。
- 在(s)中指定了常數時，移位後的最高位開始的(n2)點將變為指定的值。
- (n2)中指定的值為0的情況下，將變為無處理。

例

在(n1)=9、(n2)=4的情況下



❶ (n2) 雙字元，向右移位

❷ 複製

出錯

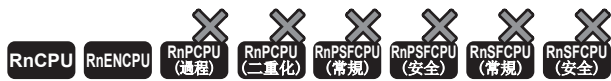
出錯代碼 (SD0)	內容
2821H	當(s)開始的(n2)點的範圍與(d)開始的(n1)點的範圍重複時。
3405H	(n1)、(n2)中指定的值為(n1) < (n2)時。

注意事項

以位元的位數指定來指定(s)和(d)時，應使(s)的位數與(d)的位數一致。

n雙字元資料的n雙字元左移

SFTDWL (P)



- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在(n1)雙字元的資料範圍內向左移(n2)雙字元。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFTDWL(EN, n1, n2, d); ENO:=SFTDWLP(EN, n1, n2, d);

FBD/LD

執行條件

指令	執行條件
SFTDWL	
SFTDWLP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	雙字元	ANY32
(n1)	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)	移位數	0~65535	無符號BIN16位元	ANY16

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(d)	—	—	○	—	—	—	○	—	—	—	—	—
(n1)	○*1	○	○	○	○	—	○	○	—	—	—	—
(n2)	○*1	○	○	○	○	—	○	○	—	—	—	—

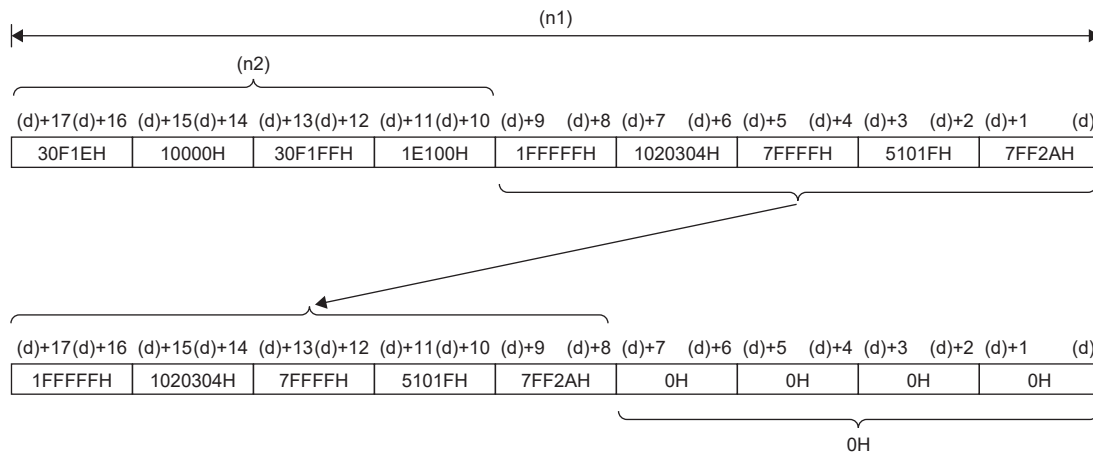
*1 不能使用FX、FY。

功能

- 從 (d) 中指定的元件開始在 (n1) 雙字元的資料範圍內向左移 (n2) 雙字元。
- 最低位開始的 (n2) 雙字元將變為 0H。
- 當 (n1) 或 (n2) 中指定的值為 0 時，將變為無處理。
- 在 $(n1) \leq (n2)$ 的情況下，(d) 中指定的元件開始的 (n1) 雙字元的資料將全部變為 0H。

例

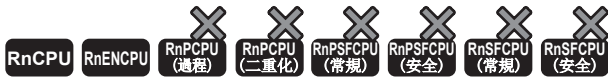
在 $(n1)=9$ 、 $(n2)=4$ 的情況下



出錯

沒有運算出錯。

DWSFTL (P)

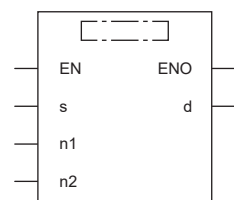


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在(n1)雙字元的資料範圍內向左移(n2)雙字元。移位後的空餘資料將儲存指定資料。

梯形圖	ST
	ENO:=DWSFTL(EN, s, n1, n2, d); ENO:=DWSFTLP(EN, s, n1, n2, d);

FBD/LD



■執行條件

指令	執行條件
DWSFTL	
DWSFTLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存至移位後空餘資料的起始元件	—	雙字元	ANY32
(d)	移位的起始元件	—	雙字元	ANY32
(n1)*1	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)*1	移位數	0~65535	無符號BIN16位元	ANY16

*1 應設定成(n2)≤(n1)。

■可使用的元件

操作數	位元		字元				雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ	K、H		E	\$			
(s)	○*1	—	○	—	—	—	○	○	○	—	—	—	—	
(d)	○*1	—	○	—	—	—	○	○	○	—	—	—	—	
(n1)	○*1	○	○	○	○	—	○	○	○	○	—	—	—	
(n2)	○*1	○	○	○	○	—	○	○	○	○	—	—	—	

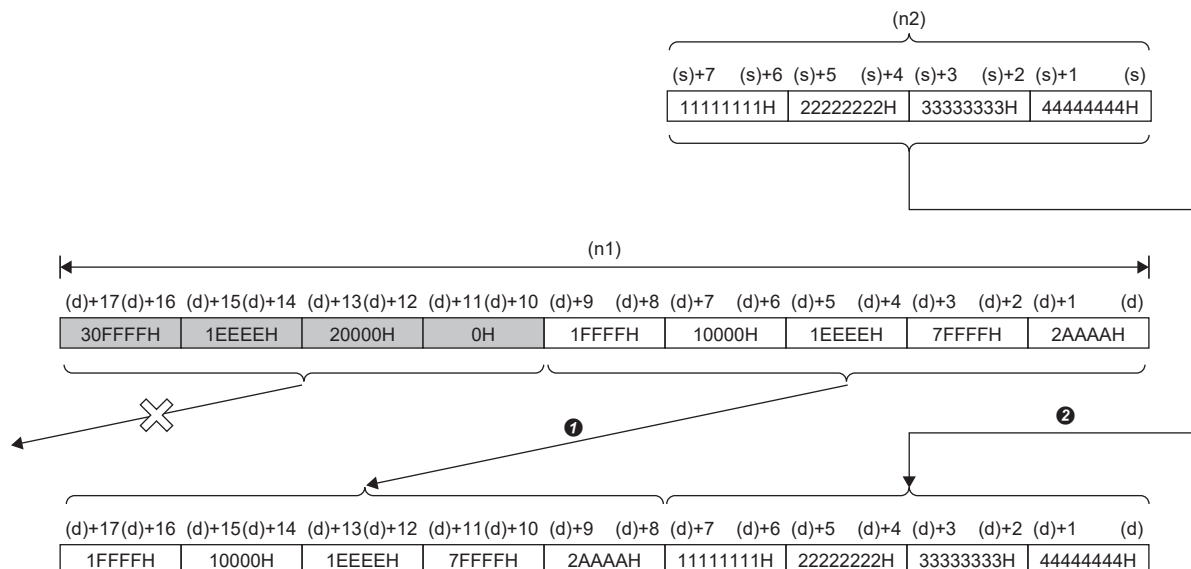
*1 不能使用FX、FY。

功能

- 從 (d) 中指定的元件開始在 (n1) 雙字元的資料範圍內向左移 (n2) 雙字元。移位後，空餘資料部分將儲存 (s) 開始的 (n2) 點。
- 在 (s) 中指定了常數時，移位後的最低位開始的 (n2) 點將變為指定的值。
- (n2) 中指定的值為 0 的情況下，將變為無處理。

例

在 (n1)=9、(n2)=4 的情況下



❶ (n2) 雙字元，向左移位

❷ 複製

出錯

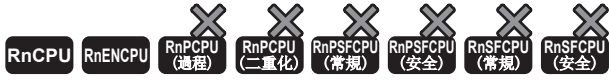
出錯代碼 (SD0)	內容
2821H	當 (s) 開始的 (n2) 點的範圍與 (d) 開始的 (n1) 點的範圍重複時。
3405H	(n1)、(n2) 中指定的值為 (n1) < (n2) 時。

注意事項

以位元的位數指定來指定 (s) 和 (d) 時，應使 (s) 的位數與 (d) 的位數一致。

單精度實數n點的n點右移

SFTER (P)



- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在單精度實數(n1)點的資料範圍內向右移(n2)點。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFTER(EN, n1, n2, d); ENO:=SFTERP(EN, n1, n2, d);

FBD/LD

■執行條件

指令	執行條件
SFTER	
SFTERP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	單精度實數	ANYREAL_32
(n1)	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)	移位數	0~65535	無符號BIN16位元	ANY16

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(d)	—	—	○	—	—	—	○	—	—	—	—
(n1)	○*1	○	○	○	○	—	○	○	—	—	—
(n2)	○*1	○	○	○	○	—	○	○	—	—	—

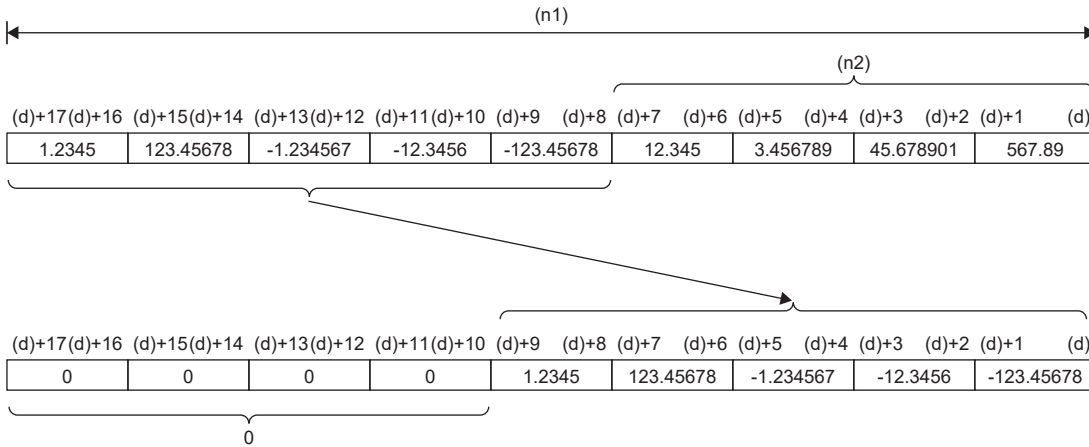
*1 不能使用FX、FY。

功能

- 從 (d) 中指定的元件開始在單精度實數 (n1) 點的資料範圍內向右移 (n2) 點。
- 最高位開始的 (n2) 點將變為0。
- 當 (n1) 或 (n2) 中指定的值為0時，將變為無處理。
- 在 $(n1) \leq (n2)$ 的情況下，(d) 中指定的元件開始的 (n1) 點的資料將全部變為0。

例

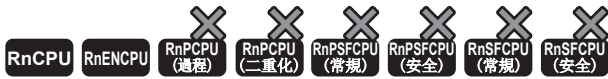
在 $(n1)=9$ 、 $(n2)=4$ 的情況下



出錯

沒有運算出錯。

ESFTR (P)

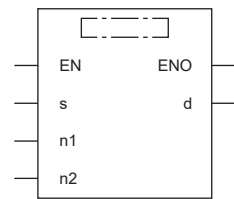


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在單精度實數(n1)點的資料範圍內向右移(n2)點。移位後的空餘資料將儲存指定資料。

梯形圖	ST
	ENO:=ESFTR(EN, s, n1, n2, d); ENO:=ESFTRP(EN, s, n1, n2, d);

FBD/LD



■執行條件

指令	執行條件
ESFTR	
ESFTRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存至移位後空餘資料的起始元件	—	單精度實數	ANYREAL_32
(d)	移位的起始元件	—	單精度實數	ANYREAL_32
(n1)*1	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)*1	移位數	0~65535	無符號BIN16位元	ANY16

*1 應設定成 $(n2) \leq (n1)$ 。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n1)	○*1	○	○	○	○	—	—	○	○	—	—	—	—
(n2)	○*1	○	○	○	○	—	—	○	○	—	—	—	—

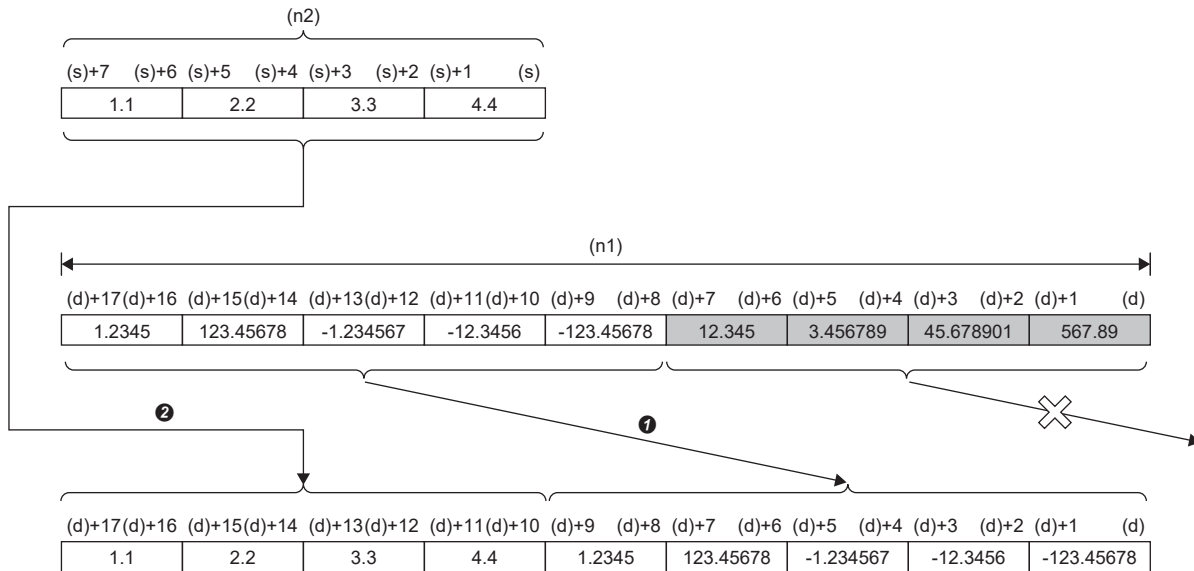
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始在單精度實數(n1)點的資料範圍內向右移(n2)點。移位後，空餘資料部分將儲存(s)開始的(n2)點。
- 在(s)中指定了常數時，移位後的最高位開始的(n2)點將變為指定的值。
- (n2)中指定的值為0的情況下，將變為無處理。

例

在(n1)=9、(n2)=4的情況下



① (n2)點，向右移位

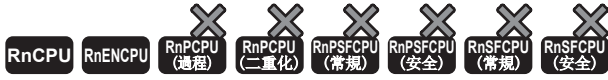
② 複製

出錯

出錯代碼 (SD0)	內容
2821H	當(s)開始的(n2)點的範圍與(d)開始的(n1)點的範圍重複時。
3405H	(n1)、(n2)中指定的值為(n1) < (n2)時。

單精度實數n點的n點左移

SFTEL (P)

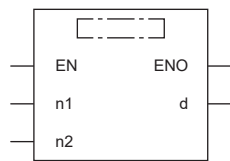


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在單精度實數(n1)點的資料範圍內向左移(n2)點。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFTEL (EN, n1, n2, d) ; ENO:=SFTELP (EN, n1, n2, d) ;

FBD/LD



■執行條件

指令	執行條件
SFTEL	
SFTELP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	單精度實數	ANYREAL_32
(n1)	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)	移位數	0~65535	無符號BIN16位元	ANY16

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(d)	—	—	○	—	—	—	○	—	—	—	—	—
(n1)	○*1	○	○	○	○	—	○	○	—	—	—	—
(n2)	○*1	○	○	○	○	—	○	○	—	—	—	—

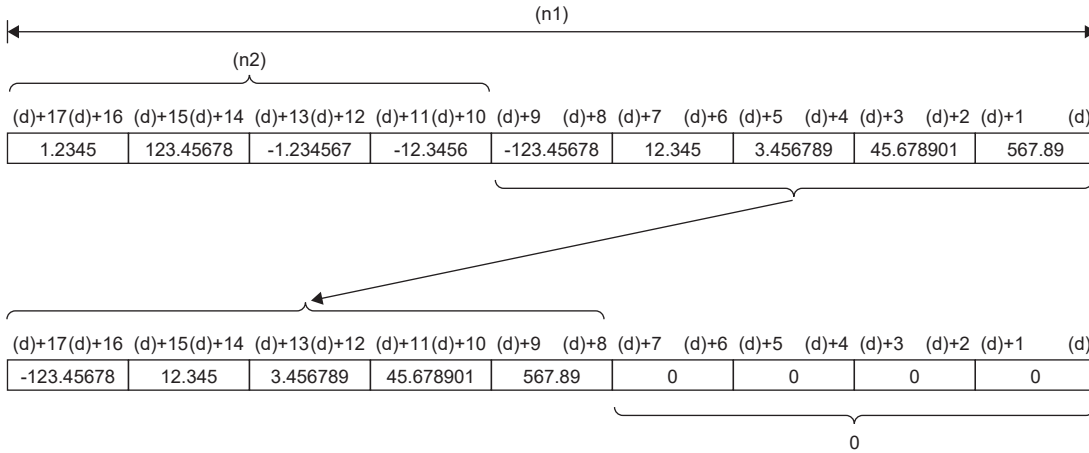
*1 不能使用FX、FY。

功能

- 從 (d) 中指定的元件開始在單精度實數 (n1) 點的資料範圍內向左移 (n2) 點。
- 最低位開始的 (n2) 點將變為 0。
- 當 (n1) 或 (n2) 中指定的值為 0 時，將變為無處理。
- 在 $(n1) \leq (n2)$ 的情況下，(d) 中指定的元件開始的 (n1) 點的資料將全部變為 0。

例

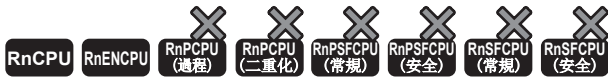
在 $(n1)=9$ 、 $(n2)=4$ 的情況下



出錯

沒有運算出錯。

ESFTL (P)

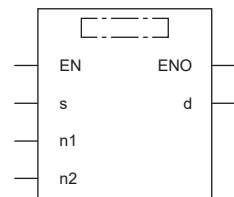


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在單精度實數(n1)點的資料範圍內向左移(n2)點。移位後的空餘資料將儲存指定資料。

梯形圖	ST
	ENO:=ESFTL (EN, s, n1, n2, d); ENO:=ESFTLP (EN, s, n1, n2, d);

FBD/LD



■執行條件

指令	執行條件
ESFTL	
ESFTLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存至移位後空餘資料的起始元件	—	單精度實數	ANYREAL_32
(d)	移位的起始元件	—	單精度實數	ANYREAL_32
(n1)*1	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)*1	移位數	0~65535	無符號BIN16位元	ANY16

*1 應設定成(n2)≤(n1)。

■可使用的元件

操作數	位元		字元				雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ	K、H		E	\$			
(s)	—	—	○	—	—	—	—	○	—	○	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—	
(n1)	○*1	○	○	○	—	—	○	○	—	—	—	—	—	
(n2)	○*1	○	○	○	—	—	○	○	—	—	—	—	—	

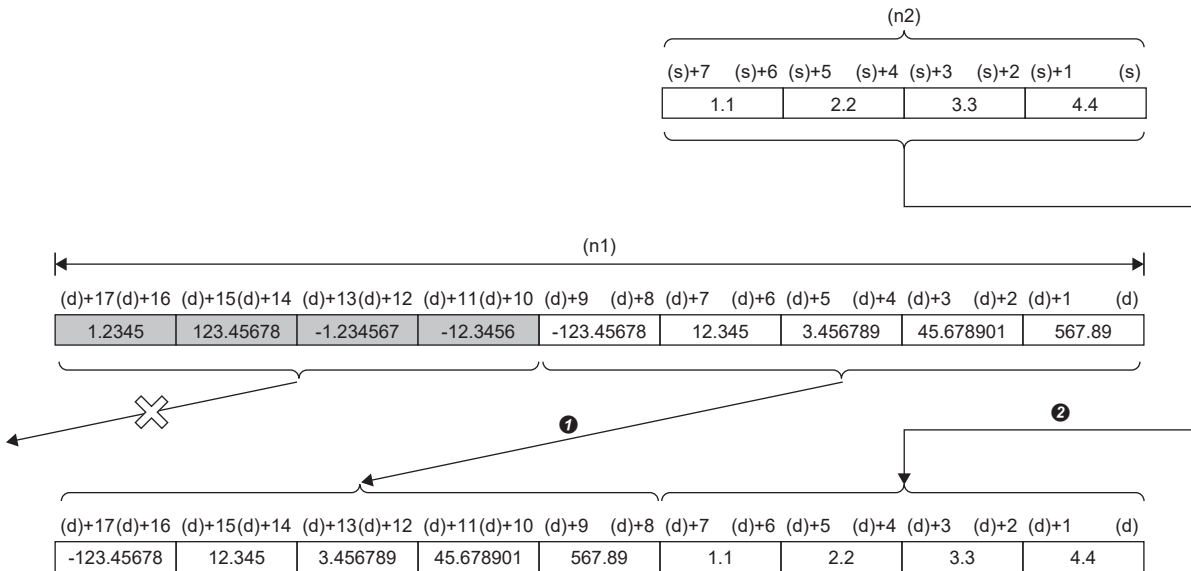
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始在單精度實數(n1)點的資料範圍內向左移(n2)點。移位後，空餘資料部分將儲存(s)開始的(n2)點。
- 在(s)中指定了常數時，移位後的最低位開始的(n2)點將變為指定的值。
- (n2)中指定的值為0的情況下，將變為無處理。

例

在(n1)=9、(n2)=4的情況下



❶ (n2)點，向左移位

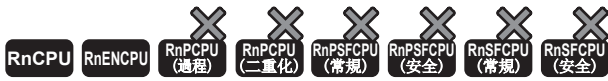
❷ 複製

出錯

出錯代碼 (SD0)	內容
2821H	當(s)開始的(n2)點的範圍與(d)開始的(n1)點的範圍重複時。
3405H	(n1)、(n2)中指定的值為(n1)<(n2)時。

雙精度實數n點的n點右移

SFTEDR (P)



- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在雙精度實數(n1)點的資料範圍內向右移(n2)點。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFTEDR (EN, n1, n2, d); ENO:=SFTEDRP (EN, n1, n2, d);

FBD/LD

執行條件

指令	執行條件
SFTEDR	
SFTEDRP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	雙精度實數	ANYREAL_64
(n1)	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)	移位數	0~65535	無符號BIN16位元	ANY16

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(d)	—	—	○	—	—	—	○	—	—	—	—
(n1)	○*1	○	○	○	○	—	○	○	—	—	—
(n2)	○*1	○	○	○	○	—	○	○	—	—	—

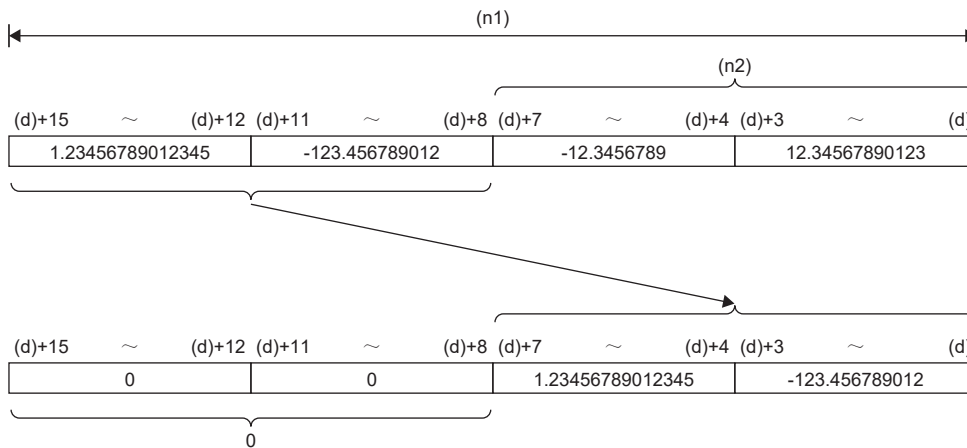
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始在雙精度實數(n1)點的資料範圍內向右移(n2)點。
- 最高位開始的(n2)點將變為0。
- 當(n1)或(n2)中指定的值為0時，將變為無處理。
- 在 $(n1) \leq (n2)$ 的情況下，(d)中指定的元件開始的(n1)點的資料將全部變為0。

例

在 $(n1)=4$ 、 $(n2)=2$ 的情況下



出錯

沒有運算出錯。

EDSFTR (P)

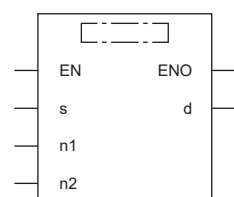


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在雙精度實數(n1)點的資料範圍內向右移(n2)點。移位後的空餘資料將儲存指定資料。

梯形圖	ST
	ENO:=EDSFTR (EN, s, n1, n2, d); ENO:=EDSFTRP (EN, s, n1, n2, d);

FBD/LD



■執行條件

指令	執行條件
EDSFTR	
EDSFTRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存至移位後空餘資料的起始元件	—	雙精度實數	ANYREAL_64
(d)	移位的起始元件	—	雙精度實數	ANYREAL_64
(n1)*1	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)*1	移位數	0~65535	無符號BIN16位元	ANY16

*1 應設定成(n2)≤(n1)。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n1)	○*1	○	○	○	○	—	—	○	○	—	—	—	—
(n2)	○*1	○	○	○	○	—	—	○	○	—	—	—	—

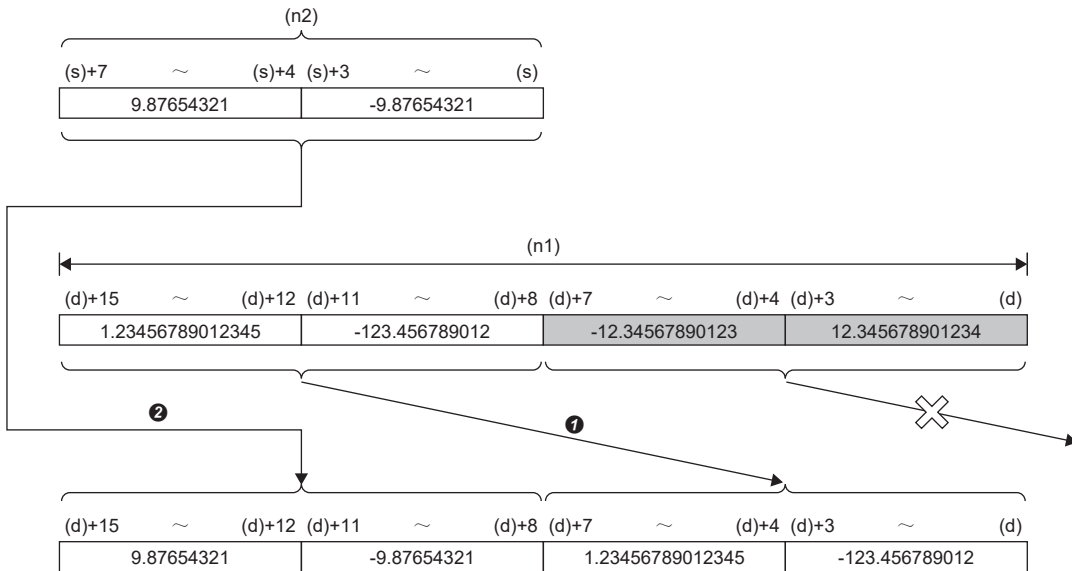
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始在雙精度實數(n1)點的資料範圍內向右移(n2)點。移位後，空餘資料部分將儲存(s)開始的(n2)點。
- 在(s)中指定了常數時，移位後的最高位開始的(n2)點將變為指定的值。
- (n2)中指定的值為0的情況下，將變為無處理。

例

在(n1)=4、(n2)=2的情況下



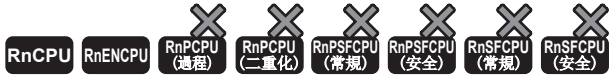
- ① (n2)點，向右移位
- ②複製

出錯

出錯代碼 (SD0)	內容
2821H	當(s)開始的(n2)點的範圍與(d)開始的(n1)點的範圍重複時。
3405H	(n1)、(n2)中指定的值為(n1)<(n2)時。

雙精度實數n點的n點左移

SFTEDL (P)



- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在雙精度實數(n1)點的資料範圍內向左移(n2)點。移位後的空餘資料將儲存0。

梯形圖	ST
	ENO:=SFTEDL(EN, n1, n2, d); ENO:=SFTEDLP(EN, n1, n2, d);

FBD/LD

■執行條件

指令	執行條件
SFTEDL	
SFTEDLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	移位的起始元件	—	雙精度實數	ANYREAL_64
(n1)	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)	移位數	0~65535	無符號BIN16位元	ANY16

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(d)	—	—	○	—	—	—	○	—	—	—	—	—
(n1)	○*1	○	○	○	○	—	○	○	—	—	—	—
(n2)	○*1	○	○	○	○	—	○	○	—	—	—	—

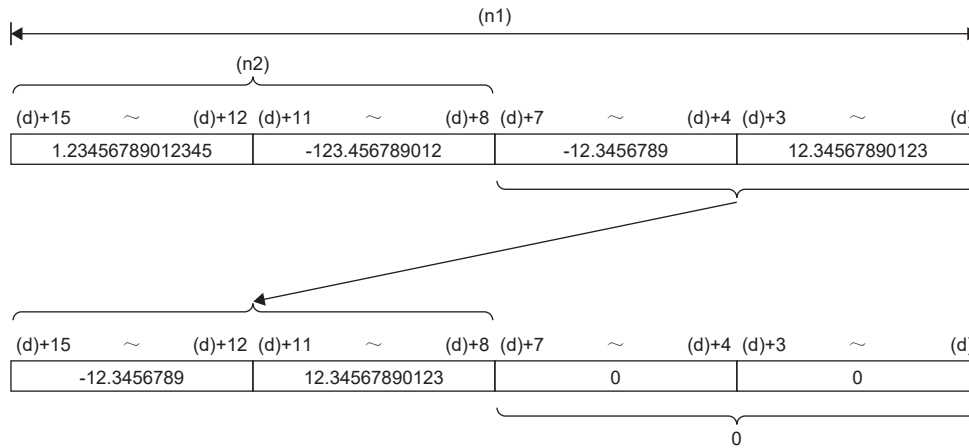
*1 不能使用FX、FY。

功能

- 從 (d) 中指定的元件開始在雙精度實數 (n1) 點的資料範圍內向左移 (n2) 點。
- 最低位開始的 (n2) 點將變為 0。
- 當 (n1) 或 (n2) 中指定的值為 0 時，將變為無處理。
- 在 $(n1) \leq (n2)$ 的情況下，(d) 中指定的元件開始的 (n1) 點的資料將全部變為 0。

例

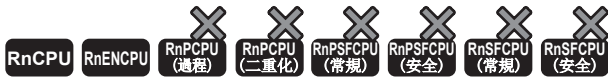
在 $(n1)=4$ 、 $(n2)=2$ 的情況下



出錯

沒有運算出錯。

EDSFTL (P)

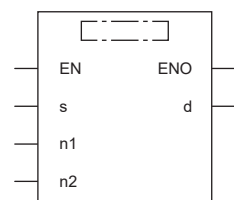


- R00CPU、R01CPU、R02CPU的韌體版本為“06”及其以後的版本可使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

從指定的元件開始在雙精度實數(n1)點的資料範圍內向左移(n2)點。移位後的空餘資料將儲存指定資料。

梯形圖	ST
	ENO:=EDSFTL(EN, s, n1, n2, d); ENO:=EDSFTLP(EN, s, n1, n2, d);

FBD/LD



■執行條件

指令	執行條件
EDSFTL	
EDSFTLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存至移位後空餘資料的起始元件	—	雙精度實數	ANYREAL_64
(d)	移位的起始元件	—	雙精度實數	ANYREAL_64
(n1)*1	移位的資料數	0~65535	無符號BIN16位元	ANY16
(n2)*1	移位數	0~65535	無符號BIN16位元	ANY16

*1 應設定成(n2)≤(n1)。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\(\H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	○	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n1)	○*1	○	○	○	○	—	—	○	○	—	—	—
(n2)	○*1	○	○	○	○	—	—	○	○	—	—	—

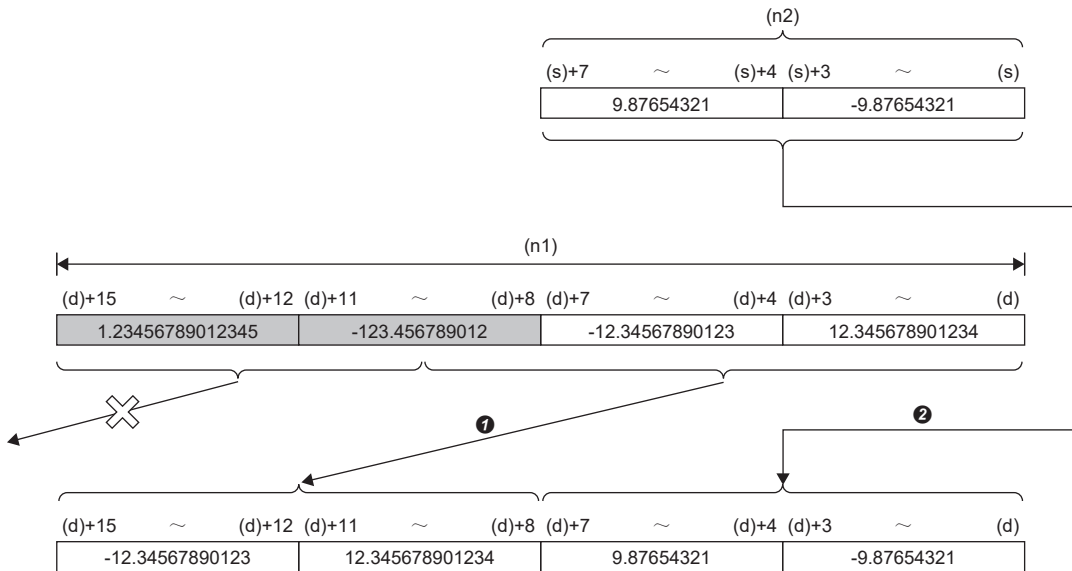
*1 不能使用FX、FY。

功能

- 從(d)中指定的元件開始在雙精度實數(n1)點的資料範圍內向左移(n2)點。移位後，空餘資料部分將儲存(s)開始的(n2)點。
- 在(s)中指定了常數時，移位後的最低位開始的(n2)點將變為指定的值。
- (n2)中指定的值為0的情況下，將變為無處理。

例

在(n1)=4、(n2)=2的情況下



① (n2)點，向左移位

②複製

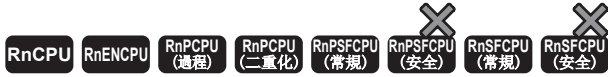
出錯

出錯代碼 (SD0)	內容
2821H	當(s)開始的(n2)點的範圍與(d)開始的(n1)點的範圍重複時。
3405H	(n1)、(n2)中指定的值為(n1) < (n2)時。

6.6 資料轉換指令

BIN資料→BCD4位轉換

BCD(P)



將指定的BIN16位元資料轉換為BCD4位資料。

梯形圖	ST
	ENO:=BCD(EN, s, d); ENO:=BCDP(EN, s, d);

FBD/LD



■執行條件

指令	執行條件
BCD	
BCDP	

設置數據

■內容、範圍、資料類型

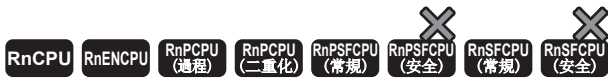
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BIN資料或儲存了BIN資料的元件	0~9999	帶符號BIN16位元	ANY16
(d)	儲存BCD資料的元件	—	BCD4位	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	○	○	○	○	—	—	○	○	—	—	—	—
(d)	—	○	○	○	○	—	—	○	—	—	—	—	—

BIN資料→BCD8位轉換

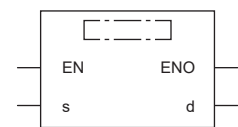
DBCD (P)



將指定的BIN32位元資料轉換為BCD8位資料。

梯形圖	ST
	ENO:=DBCD (EN, s, d) ; ENO:=DBC DP (EN, s, d) ;

FBD/LD



■執行條件

指令	執行條件
DBCD	
DBC DP	

設置數據

■內容、範圍、資料類型

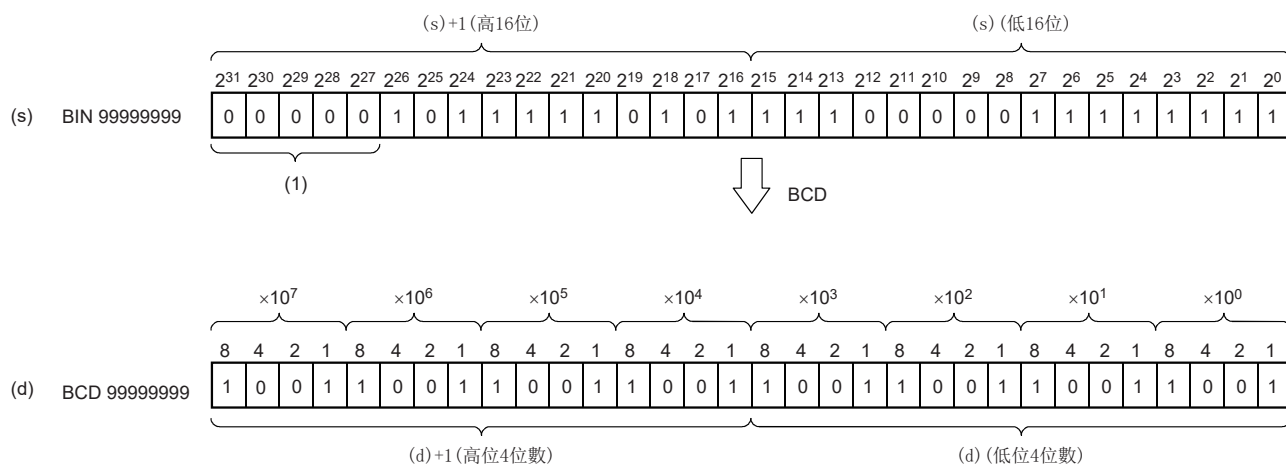
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	BIN資料或儲存了BIN資料的起始元件	0~99999999	帶符號BIN32位元	ANY32
(d)	儲存BCD資料的起始元件	—	BCD8位	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—

功能

- 將(s)中指定的元件的BIN32位元資料(0~99999999)轉換為BCD8位資料後，儲存到(d)中指定的元件中。



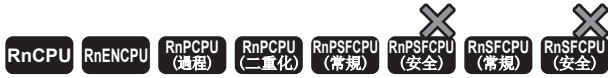
(1): 應設定為0。(高位5位元)

出錯

出錯代碼 (SD0)	內容
3401H	(s)的資料超出0~99999999的範圍時。

BCD4位→BIN16位元資料轉換

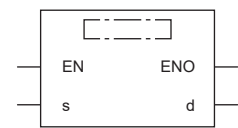
BIN(P)



將指定的BCD4位資料轉換為BIN16位元資料。

梯形圖	ST
	ENO:=BIN(EN, s, d); ENO:=BINP(EN, s, d);

FBD/LD



執行條件

指令	執行條件
BIN	
BINP	

設置數據

內容、範圍、資料類型

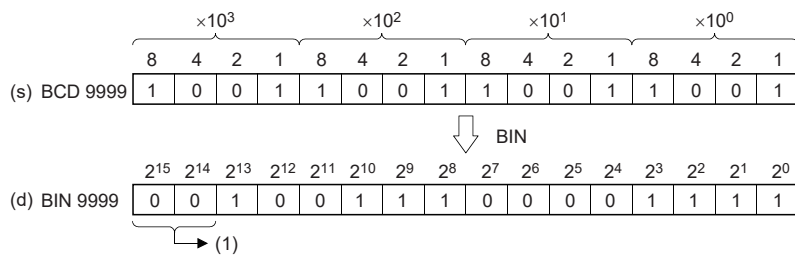
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BCD資料或儲存了BCD資料的元件	0~9999	BCD4位	ANY16
(d)	儲存BIN資料的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元				雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ	K		H	E	\$		
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—	

功能

- 將(s)中指定的元件的BCD4位資料(0~9999)轉換為BIN16位元資料後，儲存到(d)中指定的元件中。



(1): 一定變為0。

出錯

出錯代碼 (SD0)	內容
3401H	(s) 的各位中有0~9以外的值時。 ^{*1}

- *1 透過預先將SM754 (BIN、DBIN指令出錯禁止標誌) 設為ON, 可以變為不檢測本出錯。
此外, 在設定了超出範圍的數值的情況下, 與SM754的ON/OFF無關, 指令將不被執行。
此外, BIN (P) 指令的情況下, 與出錯的有無無關, 在將指令(執行條件)設為OFF→ON之前, 不執行下一個運算。

BCD8位→BIN32位元資料轉換

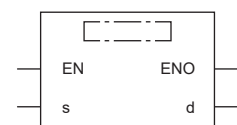
DBIN (P)



將指定的BCD8位資料轉換為BIN32位元資料。

梯形圖	ST
	ENO:=DBIN (EN, s, d) ; ENO:=DBINP (EN, s, d) ;

FBD/LD



■執行條件

指令	執行條件
DBIN	
DBINP	

設置數據

■內容、範圍、資料類型

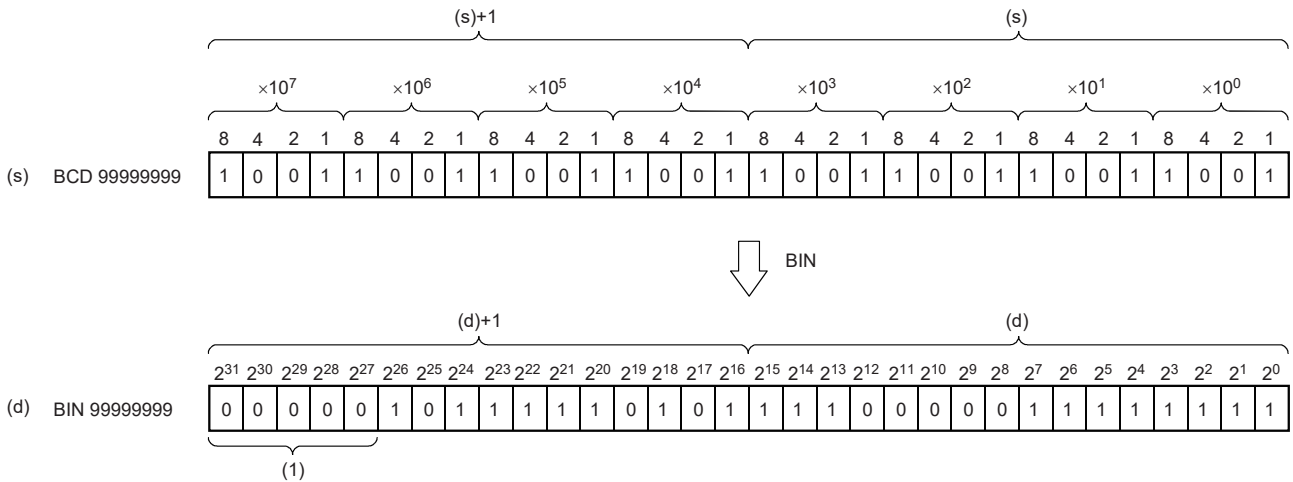
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BCD資料或儲存了BCD資料的起始元件	0~99999999	BCD8位	ANY32
(d)	儲存BIN資料的起始元件	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 將(s)中指定的元件的BCD8位資料(0~99999999)轉換為BIN32位元資料後，儲存在(d)中指定的元件中。



(1): 一定變為0。

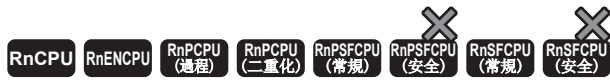
出錯

出錯代碼 (SD0)	內容
3401H	(s)的各位中有0~9以外的值時。*1

- *1 透過預先將SM754 (BIN、DBIN指令出錯禁止標誌)設為ON，可以變為不檢測本出錯。
 此外，在設定了超出範圍的數值的情況下，與SM754的ON/OFF無關，指令將不被執行。
 此外，在DBIN(P)指令的情況下，與出錯的有無無關，在將指令(執行條件)設為OFF→ON之前，不執行下一個運算。

單精度實數→帶符號BIN16位元資料

FLT2INT (P)



將指定的單精度實數轉換為帶符號BIN16位元資料。

梯形圖	ST*1
	ENO:=FLT2INT (EN, s, d) ; ENO:=FLT2INTP (EN, s, d) ;

FBD/LD

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
FLT2INT	
FLT2INTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	單精度實數或儲存了單精度實數的起始元件	-32768~32767	單精度實數	ANYREAL_32
(d)	儲存BIN資料的元件	—	帶符號BIN16位元	ANY16_S
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

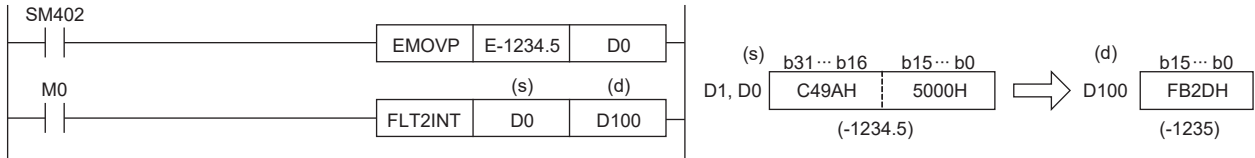
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

功能

- 將(s)中指定的單精度實數轉換為帶符號BIN16位元資料後，儲存到(d)中指定的元件中。
- 對於轉換後的資料，將變為(s)中指定的單精度實數的小數點以下第1位被四捨五入後的值。
- 在透過工程工具設定輸入值的情況下，有可能會發生化整誤差。關於透過工程工具設定輸入值時的注意事項，請參閱下述內容。

☞ 45頁 注意事項

在下述程式示例中，M0變為ON時，將D0～D1中儲存的單精度實數轉換為帶符號BIN16位元資料後，儲存到D100中。

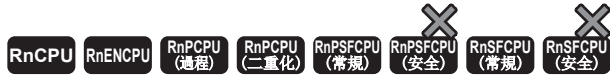


出錯

出錯代碼 (SD0)	內容
3401H	(s) 中設定的單精度實數超出了-32768~32767的範圍時。
3402H	當 (s) 中設定了特殊數時。 <ul style="list-style-type: none"> • 設定的單精度實數不在下述範圍內。 $0, 2^{-126} \leq (s) < 2^{128}$ • 設定的元件、標籤的內容為-0、非正規化數、非數、±∞時。

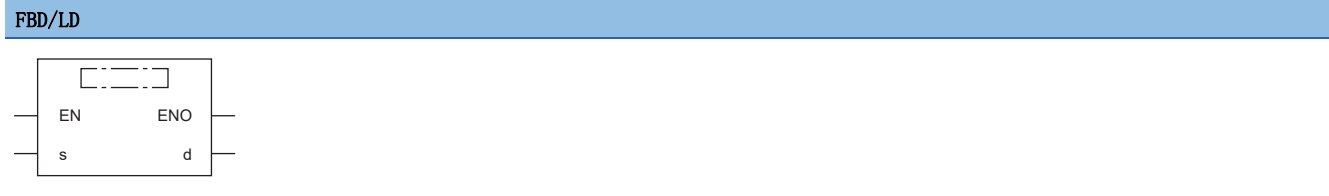
單精度實數→無符號BIN16位元資料

FLT2UINT (P)



將指定的單精度實數轉換為無符號BIN16位元資料。

梯形圖	ST*1
	ENO:=FLT2UINT (EN, s, d) ; ENO:=FLT2UINTP (EN, s, d) ;



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
FLT2UINT	
FLT2UINTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	單精度實數或儲存了單精度實數的起始元件	0~65535	單精度實數	ANYREAL_32
(d)	儲存BIN資料的元件	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

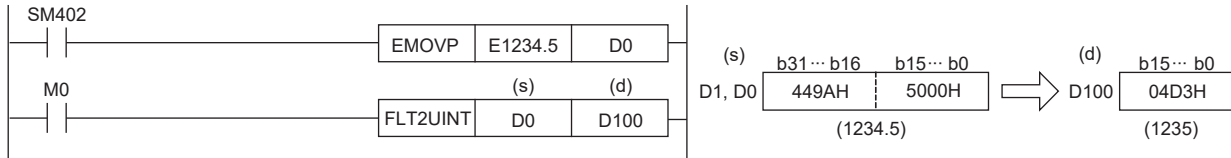
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

功能

- 將(s)中指定的單精度實數轉換為無符號BIN16位元資料後，儲存到(d)中指定的元件中。
- 對於轉換後的資料，將變為(s)中指定的單精度實數的小數點以下第1位被四捨五入後的值。
- 在透過工程工具設定輸入值的情況下，有可能會發生化整誤差。關於透過工程工具設定輸入值時的注意事項，請參閱下述內容。

☞ 45頁 注意事項

在下述程式示例中，當M0變為ON時，將D0～D1中儲存的單精度實數轉換為無符號BIN16位元資料後，儲存到D100中。

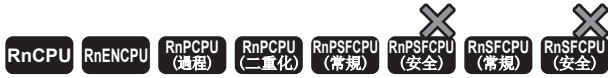


出錯

出錯代碼 (SD0)	內容
3401H	(s) 中設定的單精度實數超出了0~65535的範圍時。
3402H	當 (s) 中設定了特殊數時。 <ul style="list-style-type: none"> • 設定的單精度實數不在下述範圍內。 $0, 2^{-126} \leq (s) < 2^{128}$ • 設定的元件、標籤的內容為-0、非正規化數、非數、±∞時。

單精度實數→帶符號BIN32位元資料

FLT2DINT (P)



將指定的單精度實數轉換為帶符號BIN32位元資料。

梯形圖	ST*1
	ENO:=FLT2DINT (EN, s, d) ; ENO:=FLT2DINTP (EN, s, d) ;

FBD/LD

*1 在版本為“1.035M”及其以後的工程工具中對應。

執行條件

指令	執行條件
FLT2DINT	
FLT2DINTP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	單精度實數或儲存了單精度實數的起始元件	-2147483648~2147483647	單精度實數	ANYREAL_32
(d)	儲存BIN資料的起始元件	—	帶符號BIN32位元	ANY32_S
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

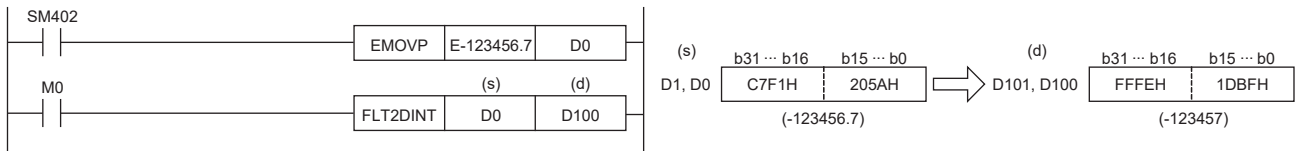
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—	
(d)	○	○	○	○	○	○	○	○	—	—	—	—	

功能

- 將(s)中指定的單精度實數轉換為帶符號BIN32位元資料後，儲存到(d)中指定的元件中。
- 對於轉換後的資料，將變為(s)中指定的單精度實數的小數點以下第1位被四捨五入後的值。
- 在透過工程工具設定輸入值的情況下，有可能會發生化整誤差。關於透過工程工具設定輸入值時的注意事項，請參閱下述內容。

☞ 45頁 注意事項

在下述程式示例中，M0變為ON時，將D0～D1中儲存的單精度實數轉換為帶符號BIN32位元資料後，儲存到D100～D101中。

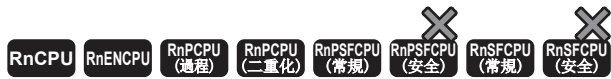


出錯

出錯代碼(SD0)	內容
3401H	(s)中設定的單精度實數超出了-2147483648~2147483647的範圍時。
3402H	當(s)中設定了特殊數時。 <ul style="list-style-type: none"> 設定的單精度實數不在下述範圍內。 $0, 2^{-126} \leq s < 2^{128}$ 設定的元件、標籤的內容為-0、非正規化數、非數、±∞時。

單精度實數→無符號BIN32位元資料

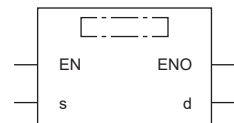
FLT2UDINT (P)



將指定的單精度實數轉換為無符號BIN32位元資料。

梯形圖	ST*1
	ENO:=FLT2UDINT (EN, s, d) ; ENO:=FLT2UDINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
FLT2UDINT	
FLT2UDINTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	單精度實數或儲存了單精度實數的起始元件	0~4294967295	單精度實數	ANYREAL_32
(d)	儲存BIN資料的起始元件	—	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

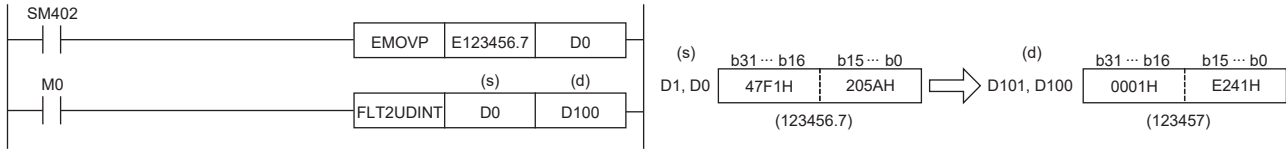
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—	
(d)	○	○	○	○	○	○	○	○	—	—	—	—	

功能

- 將(s)中指定的單精度實數轉換為無符號BIN32位元資料後，儲存到(d)中指定的元件中。
- 對於轉換後的資料，將變為(s)中指定的單精度實數的小數點以下第1位被四捨五入後的值。
- 在透過工程工具設定輸入值的情況下，有可能會發生化整誤差。關於透過工程工具設定輸入值時的注意事項，請參閱下述內容。

☞ 45頁 注意事項

在下述程式示例中，M0變為ON時，將D0～D1中儲存的單精度實數轉換為無符號BIN32位元資料後，儲存到D100～D101中。



出錯

出錯代碼 (SD0)	內容
3401H	(s)中設定的單精度實數超出了0~4294967295的範圍時。
3402H	當(s)中設定了特殊數時。 <ul style="list-style-type: none"> 設定的單精度實數不在下述範圍內。 $0, 2^{-126} \leq (s) < 2^{128}$ 設定的元件、標籤的內容為-0、非正規化數、非數、±∞時。

雙精度實數→帶符號BIN16位元資料

DBL2INT (P)



將指定的雙精度實數轉換為帶符號BIN16位元資料。

梯形圖	ST*1
	ENO:=DBL2INT (EN, s, d) ; ENO:=DBL2INTP (EN, s, d) ;

FBD/LD

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DBL2INT	
DBL2INTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	雙精度實數或儲存了雙精度實數的起始元件	-32768~32767	雙精度實數	ANYREAL_64
(d)	儲存BIN資料的元件	—	帶符號BIN16位元	ANY16_S
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

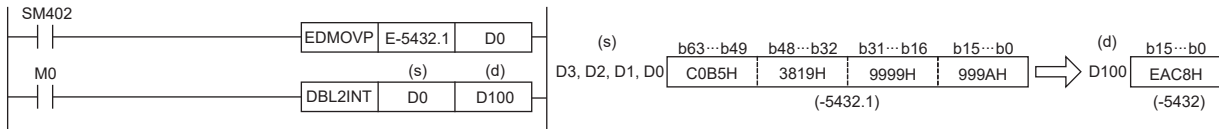
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—	
(d)	○	—	○	—	○	—	—	○	—	—	—	—	

功能

- 將(s)中指定的雙精度實數轉換為帶符號BIN16位元資料後，儲存到(d)中指定的元件中。
- 對於轉換後的資料，將變為(s)中指定的雙精度實數的小數點以下第1位被四捨五入後的值。
- 在透過工程工具設定輸入值的情況下，有可能會發生化整誤差。關於透過工程工具設定輸入值時的注意事項，請參閱下述內容。

☞ 45頁 注意事項

在下述程式示例中，M0變為ON時，將D0～D3中儲存的雙精度實數轉換為帶符號BIN16位元資料後，儲存到D100中。

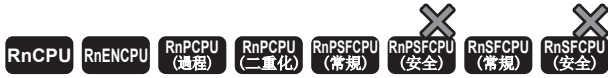


出錯

出錯代碼 (SD0)	內容
3401H	(s) 中設定的雙精度實數超出了-32768~32767的範圍時。
3402H	當(s)中設定了特殊數時。 <ul style="list-style-type: none"> 設定的雙精度實數不在下述範圍內。 $0, 2^{-1022} \leq (s) < 2^{1024}$ 設定的元件、標籤的內容為-0、非正規化數、非數、±∞時。

雙精度實數→無符號BIN16位元資料

DBL2UINT (P)



將指定的雙精度實數轉換為無符號BIN16位元資料。

梯形圖	ST*1
	ENO:=DBL2UINT (EN, s, d) ; ENO:=DBL2UINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DBL2UINT	
DBL2UINTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	雙精度實數或儲存了雙精度實數的起始元件	0~65535	雙精度實數	ANYREAL_64
(d)	儲存BIN資料的元件	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

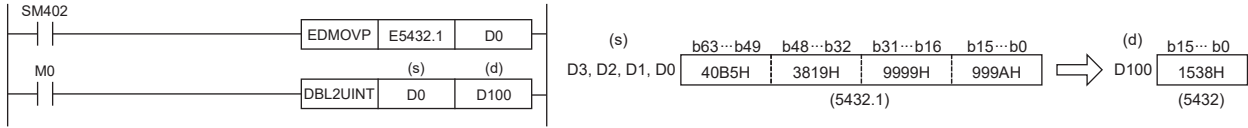
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—	
(d)	○	—	○	—	○	—	—	○	—	—	—	—	

功能

- 將(s)中指定的雙精度實數轉換為無符號BIN16位元資料後，儲存到(d)中指定的元件中。
- 對於轉換後的資料，將變為(s)中指定的雙精度實數的小數點以下第1位被四捨五入後的值。
- 在透過工程工具設定輸入值的情況下，有可能會發生化整誤差。關於透過工程工具設定輸入值時的注意事項，請參閱下述內容。

☞ 45頁 注意事項

在下述程式示例中，當M0變為ON時，將D0～D3中儲存的雙精度實數轉換為無符號BIN16位元資料後，儲存到D100中。



出錯

出錯代碼(SD0)	內容
3401H	(s)中設定的雙精度實數超出了0~65535的範圍時。
3402H	當(s)中設定了特殊數時。 <ul style="list-style-type: none"> 設定的雙精度實數不在下述範圍內。 $0, 2^{-1022} \leq s < 2^{1024}$ 設定的元件、標籤的內容為-0、非正規化數、非數、±∞時。

雙精度實數→帶符號BIN32位元資料

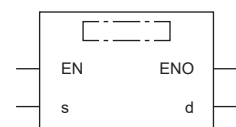
DBL2DINT (P)



將指定的雙精度實數轉換為帶符號BIN32位元資料。

梯形圖	ST*1
	ENO:=DBL2DINT (EN, s, d) ; ENO:=DBL2DINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DBL2DINT	
DBL2DINTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	雙精度實數或儲存了雙精度實數的起始元件	-2147483648~2147483647	雙精度實數	ANYREAL_64
(d)	儲存BIN資料的起始元件	—	帶符號BIN32位元	ANY32_S
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

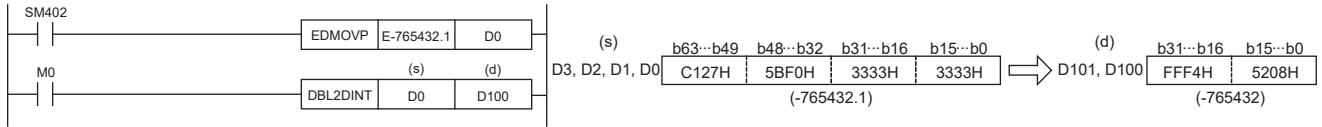
操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	○	—	○	—	○	○	○	○	—	—	—	—

功能

- 將(s)中指定的雙精度實數轉換為帶符號BIN32位元資料後，儲存到(d)中指定的元件中。
- 對於轉換後的資料，將變為(s)中指定的雙精度實數的小數點以下第1位被四捨五入後的值。
- 在透過工程工具設定輸入值的情況下，有可能會發生化整誤差。關於透過工程工具設定輸入值時的注意事項，請參閱下述內容。

☞ 45頁 注意事項

在下述程式示例中，M0變為ON時，將D0～D3中儲存的雙精度實數轉換為帶符號BIN32位元資料後，儲存到D100～D101中。



出錯

出錯代碼(SD0)	內容
3401H	(s)中設定的雙精度實數超出了-2147483648~2147483647的範圍時。
3402H	當(s)中設定了特殊數時。 <ul style="list-style-type: none"> 設定的雙精度實數不在下述範圍內。 $0, 2^{-1022} \leq s < 2^{1024}$ 設定的元件、標籤的內容為-0、非正規化數、非數、±∞時。

雙精度實數→無符號BIN32位元資料

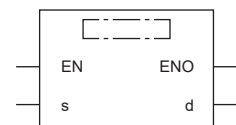
DBL2UDINT (P)



將指定的雙精度實數轉換為無符號BIN32位元資料。

梯形圖	ST*1
	ENO:=DBL2UDINT (EN, s, d) ; ENO:=DBL2UDINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DBL2UDINT	
DBL2UDINTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	雙精度實數或儲存了雙精度實數的起始元件	0~4294967295	雙精度實數	ANYREAL_64
(d)	儲存BIN資料的起始元件	—	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

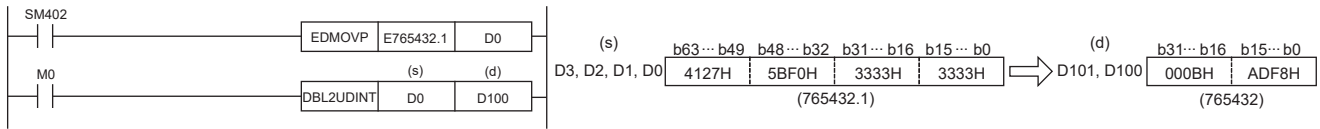
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—	
(d)	○	—	○	—	○	○	○	○	—	—	—	—	

功能

- 將(s)中指定的雙精度實數轉換為無符號BIN32位元資料後，儲存在(d)中指定的元件中。
- 對於轉換後的資料，將變為(s)中指定的雙精度實數的小數點以下第1位被四捨五入後的值。
- 在透過工程工具設定輸入值的情況下，有可能會發生化整誤差。關於透過工程工具設定輸入值時的注意事項，請參閱下述內容。

☞ 45頁 注意事項

在下述程式示例中，當M0變為ON時，將D0～D3中儲存的雙精度實數轉換為無符號BIN32位元資料後，儲存在D100～D101中。

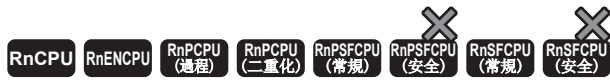


出錯

出錯代碼(SD0)	內容
3401H	(s)中設定的雙精度實數超出了0~4294967295的範圍時。
3402H	當(s)中設定了特殊數時。 <ul style="list-style-type: none"> 設定的雙精度實數不在下述範圍內。 $0, 2^{-1022} \leq s < 2^{1024}$ 設定的元件、標籤的內容為-0、非正規化數、非數、±∞時。

帶符號BIN16位元資料→無符號BIN16位元資料轉換

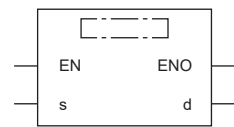
INT2UINT (P)



將指定的帶符號BIN16位元資料轉換為無符號BIN16位元資料。

梯形圖	ST*1
	ENO:=INT2UINT (EN, s, d) ; ENO:=INT2UINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

執行條件

指令	執行條件
INT2UINT	
INT2UINTP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BIN資料或儲存了BIN資料的標籤	-32768~32767	帶符號BIN16位元	ANY16_S
(d)	儲存BIN資料的標籤	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		Z	雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□		LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

要點

INT2UINT (P) 指令在透過標籤進行程式時，是在指令的操作數中可指定的資料類型與被指定的標籤的資料類型一致的情況下使用。
 在透過元件進行程式的情況下，無需使用INT2UINT (P) 指令。

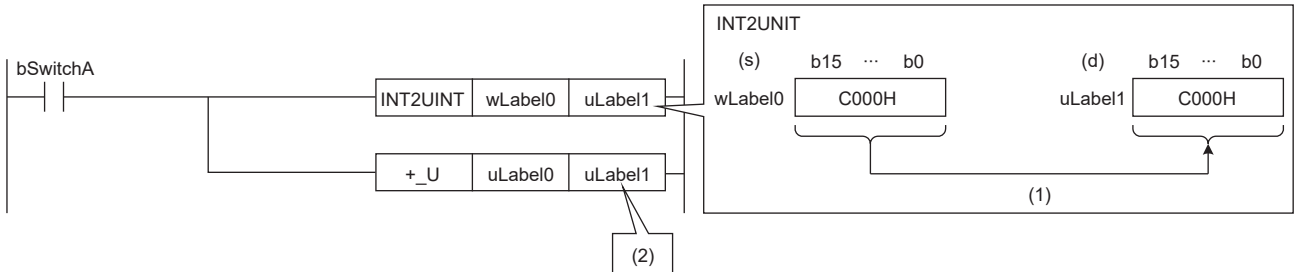
功能

- 將(s)中指定的帶符號BIN16位元資料(ANY16_S)轉換為無符號BIN16位元資料(ANY16_U)後，儲存在(d)中指定的標籤中。
- INT2UINT(P)指令的使用示例如下所示。

例

因為+_U指令需要在操作數中指定ANY16_U，因此執行+_U指令之前應使用INT2UINT指令，將ANY16_S的wLabel0轉換為ANY16_U的uLabel1。

將wLabel0作為值直接儲存在uLabel1中。



bSwitchA: 位元

wLabel0: 字元[帶符號]

uLabel0、uLabel1: 字元[無符號]/位元串[16位元]

(1): 直接儲存。

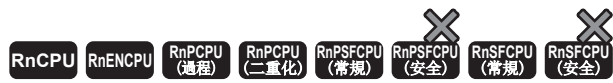
(2): 根據+_U指令的操作數的資料類型進行運算。

出錯

沒有運算出錯。

帶符號BIN16位元資料→帶符號BIN32位元資料轉換

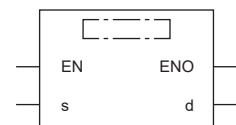
INT2DINT (P)



將指定的帶符號BIN16位元資料轉換為帶符號BIN32位元資料。

梯形圖	ST*1
	ENO:=INT2DINT (EN, s, d) ; ENO:=INT2DINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
INT2DINT	
INT2DINTP	

設置數據

■內容、範圍、資料類型

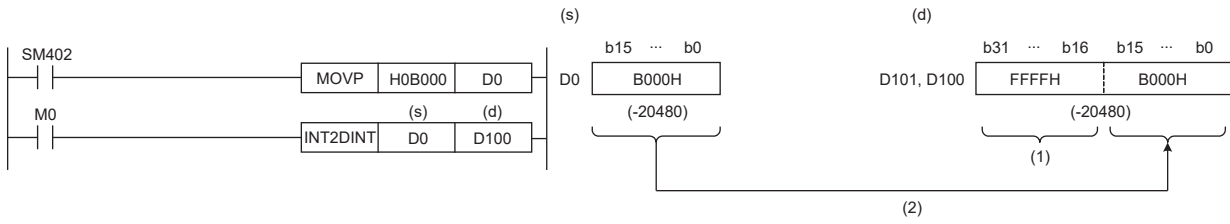
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BIN資料或儲存了BIN資料的元件	-32768~32767	帶符號BIN16位元	ANY16_S
(d)	儲存BIN資料的起始元件	—	帶符號BIN32位元	ANY32_S
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 將(s)中指定的帶符號BIN16位元資料轉換為帶符號BIN32位元資料後，儲存到(d)中指定的元件中。
- 在下述程式示例中，M0變為ON時，將D0中儲存的帶符號BIN16位元資料轉換為帶符號BIN32位元資料後，儲存到D100~D101中。



(1): 輸入轉換前的最高位的值。

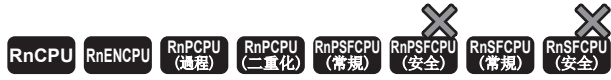
(2): 儲存至低位16位元

出錯

沒有運算出錯。

帶符號BIN16位元資料→無符號BIN32位元資料轉換

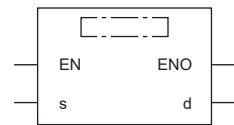
INT2UDINT (P)



將指定的帶符號BIN16位元資料轉換為無符號BIN32位元資料。

梯形圖	ST*1
	ENO:=INT2UDINT (EN, s, d) ; ENO:=INT2UDINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
INT2UDINT	
INT2UDINTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	BIN資料或儲存了BIN資料的元件	-32768~32767	帶符號BIN16位元	ANY16_S
(d)	儲存BIN資料的起始元件	—	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

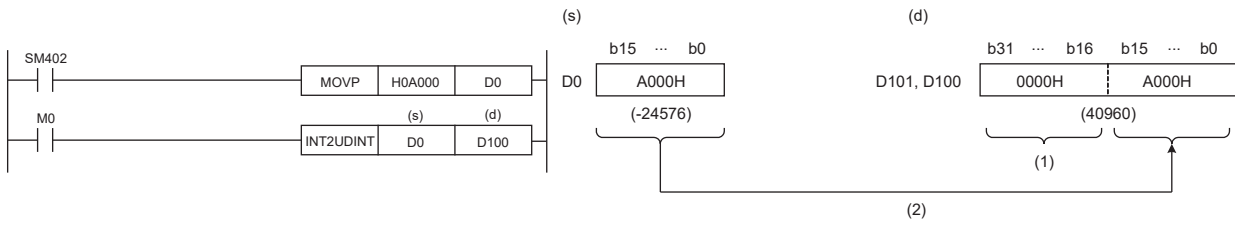
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 將(s)中指定的帶符號BIN16位元資料轉換為無符號BIN32位元資料後，儲存到(d)中指定的元件中。

在下述程式示例中，當M0變為ON時，將D0中儲存的帶符號BIN16位元資料轉換為無符號BIN32位元資料後，儲存到D100~D101中。



(1): 儲存0

(2): 儲存至低位16位元

出錯

沒有運算出錯。

無符號BIN16位元資料→帶符號BIN16位元資料轉換

UINT2INT (P)



將指定的無符號BIN16位元資料轉換就為帶符號BIN16位元資料。

梯形圖	ST*1
	ENO:=UINT2INT (EN, s, d) ; ENO:=UINT2INTP (EN, s, d) ;

FBD/LD

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
UINT2INT	
UINT2INTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	BIN資料或儲存了BIN資料的標籤	0~65535	無符號BIN16位元	ANY16_U
(d)	儲存BIN資料的標籤	—	帶符號BIN16位元	ANY16_S
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

要點

UINT2INT (P) 指令在透過標籤進行程式時，是在指令的操作數中可指定的資料類型與指定的標籤的資料類型一致的情況下使用。
 在透過元件進行程式的情況下，無需使用UINT2INT (P) 指令。

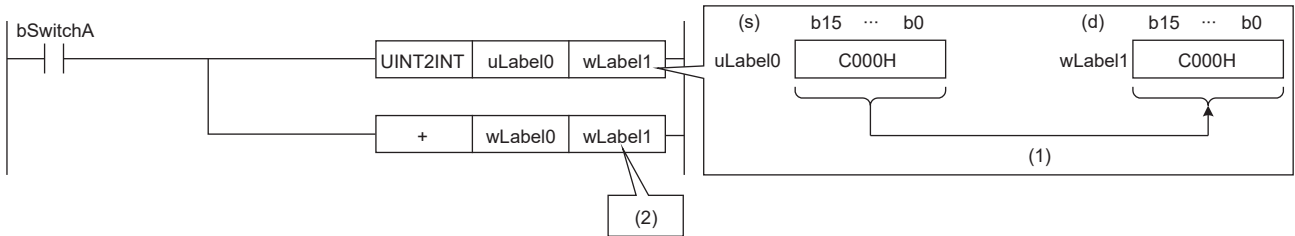
功能

- 將(s)中指定的無符號BIN16位元資料(ANY16_U)轉換為帶符號BIN16位元資料(ANY16_S)後，儲存在(d)中指定的標籤中。
- UINT2INT(P)指令的使用示例如下所示。

例

因為+指令需要在操作數中指定ANY16_S，因此執行+指令之前應使用UINT2INT指令，將ANY16_U的uLabel0轉換為ANY16_S的wLabel1。

將uLabel0作為值直接儲存在wLabel1中。



bSwitchA: 位元

wLabel0, wLabel1: 字元[帶符號]

uLabel0: 字元[無符號]/位元串[16位元]

(1): 直接儲存。

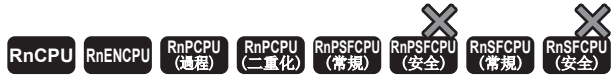
(2): 根據+指令的操作數的資料類型進行運算。

出錯

沒有運算出錯。

無符號BIN16位元資料→帶符號BIN32位元資料轉換

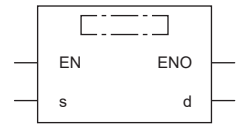
UINT2DINT (P)



將指定的無符號BIN16位元資料轉換為帶符號BIN32位元資料。

梯形圖	ST*1
	ENO:=UINT2DINT (EN, s, d) ; ENO:=UINT2DINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
UINT2DINT	
UINT2DINTP	

設置數據

■內容、範圍、資料類型

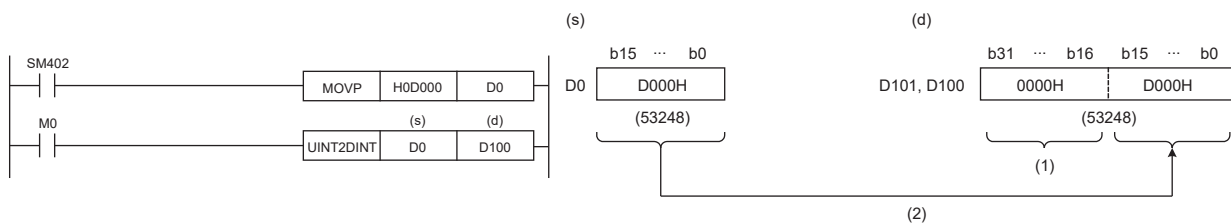
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	BIN資料或儲存了BIN資料的元件	0~65535	無符號BIN16位元	ANY16_U
(d)	儲存BIN資料的起始元件	—	帶符號BIN32位元	ANY32_S
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 將(s)中指定的無符號BIN16位元資料轉換為帶符號BIN32位元資料後，儲存到(d)中指定的元件中。
- 在下述程式示例中，M0變為ON時，將D0中儲存的無符號BIN16位元資料轉換為帶符號BIN32位元資料後，儲存到D100~D101中。



(1): 儲存0

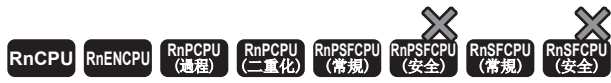
(2): 儲存至低位16位元

出錯

沒有運算出錯。

無符號BIN16位元資料→無符號BIN32位元資料轉換

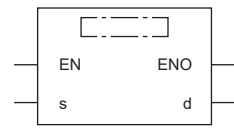
UINT2UDINT (P)



將指定的無符號BIN16位元資料轉換為無符號BIN32位元資料。

梯形圖	ST*1
	ENO:=UINT2UDINT (EN, s, d) ; ENO:=UINT2UDINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
UINT2UDINT	
UINT2UDINTP	

設置數據

■內容、範圍、資料類型

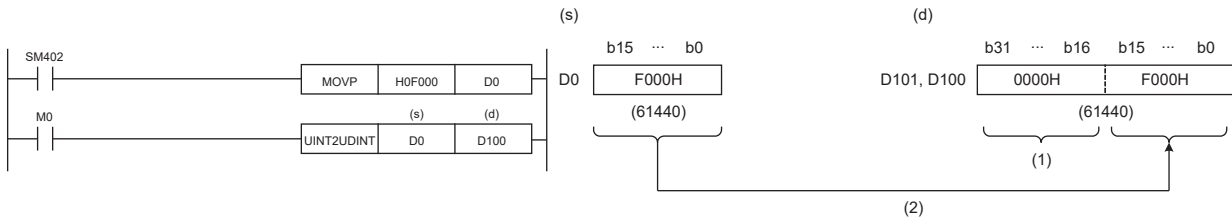
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	BIN資料或儲存了BIN資料的元件	0~65535	無符號BIN16位元	ANY16_U
(d)	儲存BIN資料的起始元件	—	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	○	○	○	○	—	—	—	—	

功能

- 將(s)中指定的無符號BIN16位元資料轉換為無符號BIN32位元資料後，儲存到(d)中指定的元件中。
- 在下述程式示例中，當M0變為ON時，將D0中儲存的無符號BIN16位元資料轉換無符號BIN32位元資料後，儲存到D100~D101中。



(1): 儲存0

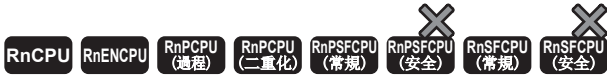
(2): 儲存至低位16位元

出錯

沒有運算出錯。

帶符號BIN32位元資料→帶符號BIN16位元資料轉換

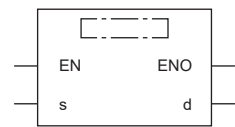
DINT2INT (P)



將指定的帶符號BIN32位元資料轉換為帶符號BIN16位元資料。

梯形圖	ST*1
	ENO:=DINT2INT (EN, s, d) ; ENO:=DINT2INTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DINT2INT	
DINT2INTP	

設置數據

■內容、範圍、資料類型

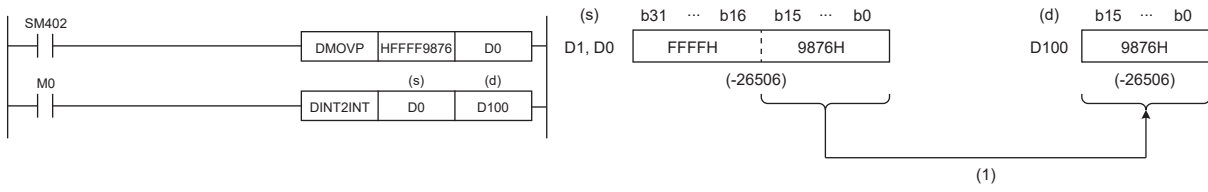
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BIN資料或儲存了BIN資料的起始元件	-32768~32767	帶符號BIN32位元	ANY32_S
(d)	儲存BIN資料的元件	—	帶符號BIN16位元	ANY16_S
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- 將(s)中指定的帶符號BIN32位元資料轉換為帶符號BIN16位元資料轉換後，儲存到(d)中指定的元件中。
- 在下述程式示例中，當M0變為ON時，將D0～D1中儲存的帶符號BIN32位元資料轉換為帶符號BIN16位元資料後，儲存到D100中。



(1): 儲存至低位16位元

出錯

出錯代碼 (SD0)	內容
3401H	當(s)中設定的帶符號BIN32位元資料超出了-32768~32767的範圍時。

帶符號BIN32位元資料→無符號BIN16位元資料轉換

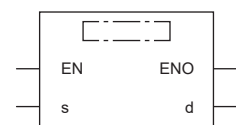
DINT2UINT (P)



將指定的帶符號BIN32位元資料轉換為無符號BIN16位元資料。

梯形圖	ST*1
	ENO:=DINT2UINT (EN, s, d) ; ENO:=DINT2UINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DINT2UINT	
DINT2UINTP	

設置數據

■內容、範圍、資料類型

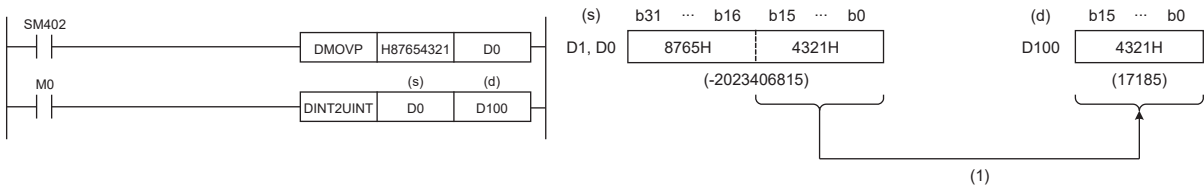
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BIN資料或儲存了BIN資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
(d)	儲存BIN資料的元件	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- 將(s)中指定的帶符號BIN32位元資料轉換為無符號BIN16位元資料後，儲存到(d)中指定的元件中。
- 在下述程式示例中，當M0變為ON時，將D0～D1中儲存的帶符號BIN32位元資料轉換為無符號BIN16位元資料後，儲存到D100中。



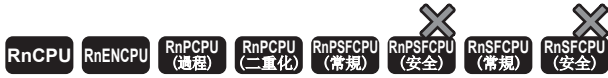
(1): 儲存至低位16位元

出錯

沒有運算出錯。

帶符號BIN32位元資料→無符號BIN32位元資料轉換

DINT2UDINT (P)



將指定的帶符號BIN32位元資料轉換為無符號BIN32位元資料。

梯形圖	ST*1
	ENO:=DINT2UDINT (EN, s, d); ENO:=DINT2UDINTP (EN, s, d);

FBD/LD

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DINT2UDINT	
DINT2UDINTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BIN資料或儲存了BIN資料的標籤	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
(d)	儲存BIN資料的標籤	—	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

要點

DINT2UDINT (P) 指令在透過標籤進行程式的情況下，是在指令的操作數中可指定的資料類型與指定標籤的資料類型一致的情況下而使用。

在透過元件進行程式的情況下，無需使用DINT2UDINT (P) 指令。

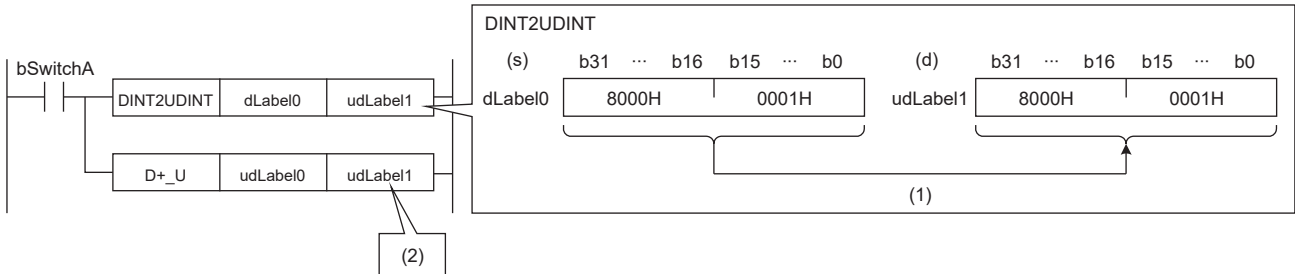
功能

- 將(s)中指定的帶符號BIN32位元資料(ANY32_S)轉換為無符號BIN32位元資料(ANY32_U)後，儲存到(d)中指定的標籤中。
- DINT2UDINT(P)指令的使用示例如下所示。

例

因為D+_U指令需要在操作數中指定ANY32_U，因此執行D+_U指令之前應使用DINT2UDINT指令，將ANY32_S的dLabel0轉換為ANY32_U的udLabel1。

將dLabel0作為值直接儲存到udLabel1中。



bSwitchA: 位元

dLabel0: 雙字元[帶符號]

udLabel0、udLabel1: 雙字元[無符號]/位元串[32位元]

(1): 直接儲存。

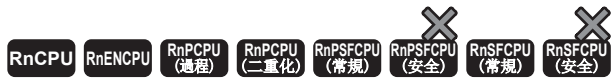
(2): 根據D+_U指令的操作數的資料類型進行運算。

出錯

沒有運算出錯。

無符號BIN32位元資料→帶符號BIN16位元資料轉換

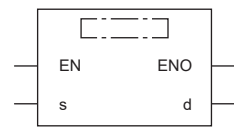
UDINT2INT (P)



將指定的無符號BIN32位元資料轉換為帶符號BIN16位元資料。

梯形圖	ST*1
	ENO:=UDINT2INT (EN, s, d) ; ENO:=UDINT2INTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
UDINT2INT	
UDINT2INTP	

設置數據

■內容、範圍、資料類型

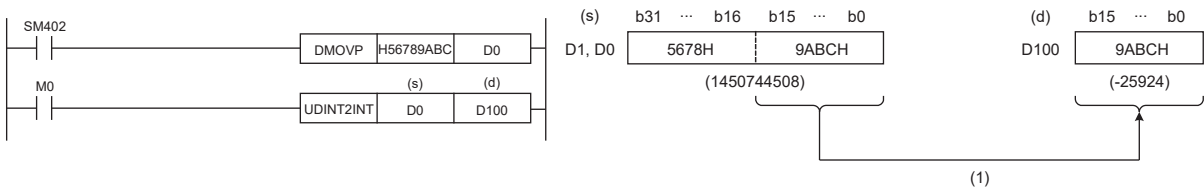
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BIN資料或儲存了BIN資料的起始元件	0~4294967295	無符號BIN32位元	ANY32_U
(d)	儲存BIN資料的元件	—	帶符號BIN16位元	ANY16_S
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 將(s)中指定的無符號BIN32位元資料轉換為帶符號BIN16位元資料後，儲存到(d)中指定的元件中。
- 在下述程式示例中，當M0變為ON時，將D0~D1中儲存的無符號BIN32位元資料轉換為帶符號BIN16位元資料後，儲存到D100中。



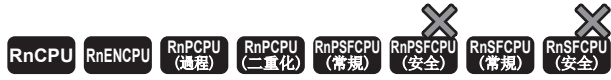
(1): 儲存至低位16位元

出錯

沒有運算出錯。

無符號BIN32位元資料→無符號BIN16位元資料轉換

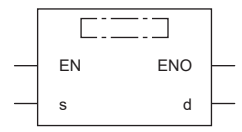
UDINT2UINT (P)



將指定的無符號BIN32位元資料轉換為無符號BIN16位元資料。

梯形圖	ST*1
	ENO:=UDINT2UINT (EN, s, d) ; ENO:=UDINT2UINTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
UDINT2UINT	
UDINT2UINTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BIN資料或儲存了BIN資料的起始元件	0~4294967295	無符號BIN32位元	ANY32_U
(d)	儲存BIN資料的元件	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

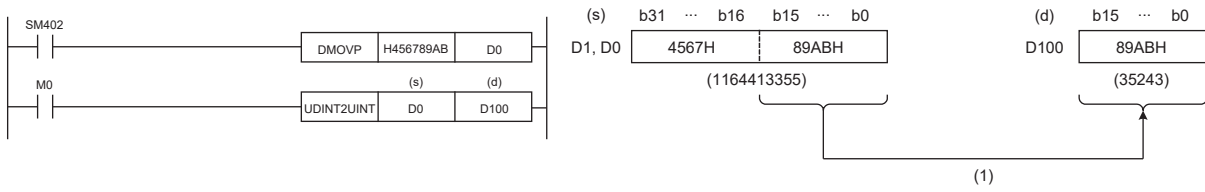
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- 將(s)中指定的無符號BIN32位元資料轉換為無符號BIN16位元資料後，儲存到(d)中指定的元件中。

在下述程式示例中，當M0變為ON時，將D0~D1中儲存的無符號BIN32位元資料轉換為無符號BIN16位元資料後，儲存到D100中。



(1): 儲存至低位16位元

出錯

沒有運算出錯。

無符號BIN32位元資料→帶符號BIN32位元資料轉換

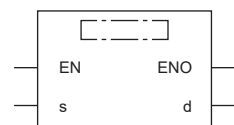
UDINT2DINT (P)



將指定的無符號BIN32位元資料轉換為帶符號BIN32位元資料。

梯形圖	ST*1
	ENO:=UDINT2DINT (EN, s, d); ENO:=UDINT2DINTP (EN, s, d);

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
UDINT2DINT	
UDINT2DINTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	BIN資料或儲存了BIN資料的標籤	0~4294967295	無符號BIN32位元	ANY32_U
(d)	儲存BIN資料的標籤	—	帶符號BIN32位元	ANY32_S
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

要點

UDINT2DINT (P) 指令在透過標籤進行程式的情況下，是在指令的操作數中可指定的資料類型與指定的標籤的資料類型一致的情況下使用。
 在透過元件進行程式的情況下，無需使用UDINT2DINT (P) 指令。

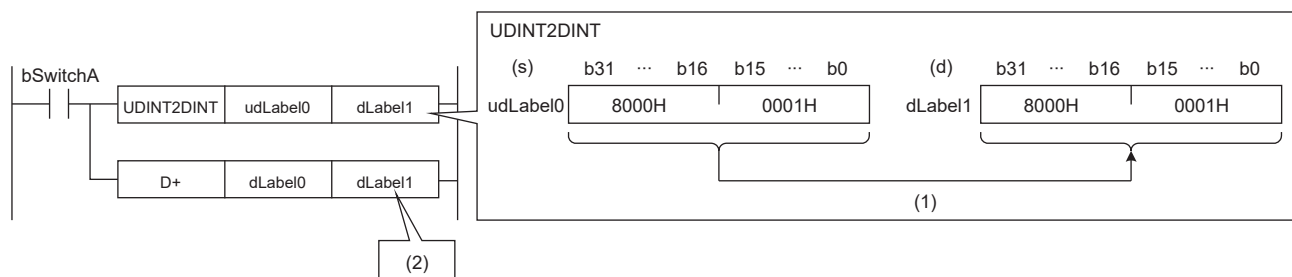
功能

- 將(s)中指定的無符號BIN32位元資料(ANY32_U)轉換為帶符號BIN32位元資料(ANY32_S)後，儲存到(d)中指定的標籤中。
- UDINT2DINT(P)指令的使用示例如下所示。

例

因為D+指令需要在操作數中指定ANY32_S，因此執行D+指令之前應使用UDINT2DINT指令，將ANY32_U的udLabel0轉換為ANY32_S的dLabel1。

將udLabel0作為值直接儲存到dLabel1中。



bSwitchA: 位元

dLabel0, dLabel1: 雙字元[帶符號]

udLabel0: 雙字元[無符號]/位元串[32位元]

(1): 直接儲存。

(2): 根據D+指令的操作數的資料類型進行運算。

出錯

沒有運算出錯。

BIN16位元資料→BIN格雷碼資料轉換

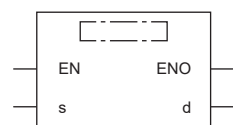
GRY(P) (_U)



將指定的BIN16位元資料轉換為BIN16位元格雷碼資料。

梯形圖	ST	
	ENO:=GRY(EN, s, d); ENO:=GRYP(EN, s, d);	ENO:=GRY_U(EN, s, d); ENO:=GRYP_U(EN, s, d);

FBD/LD



■執行條件

指令	執行條件
GRY GRY_U	
GRYP GRYP_U	

設置數據

■內容、範圍、資料類型

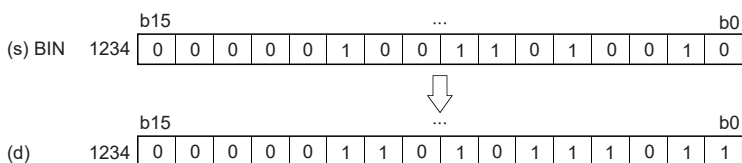
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	GRY(P)	0~32767	帶符號BIN16位元	ANY16_S
	GRY(P)_U	0~65535	無符號BIN16位元	ANY16_U
(d)	GRY(P)	—	帶符號BIN16位元	ANY16_S
	GRY(P)_U	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

功能

- 將(s)中指定的BIN16位元資料轉換為BIN16位元格雷碼資料後，儲存到(d)中指定的元件中。

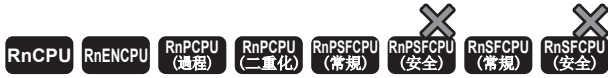


出錯

出錯代碼 (SD0)	內容
3401H	使用GRY (P) 指令時，(s)中指定的值超出0~32767的範圍時。

BIN32位元資料→格雷碼資料轉換

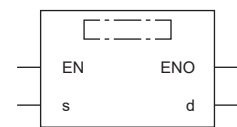
DGRY(P) (_U)



將指定的BIN32位元資料轉換為BIN32位元格雷碼資料。

梯形圖	ST	
	ENO:=DGRY(EN, s, d); ENO:=DGRYP(EN, s, d);	ENO:=DGRY_U(EN, s, d); ENO:=DGRYP_U(EN, s, d);

FBD/LD



■執行條件

指令	執行條件
DGRY DGRY_U	
DGRYP DGRYP_U	

設置數據

■內容、範圍、資料類型

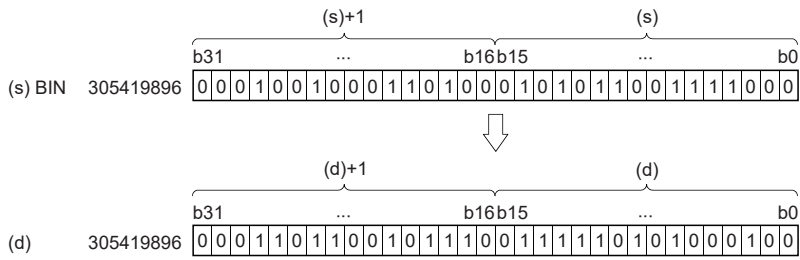
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	DGRY(P) DGRY(P)_U	BIN資料或儲存了BIN資料的起始元件 0~2147483647 0~4294967295	帶符號BIN32位元 無符號BIN32位元	ANY32_S ANY32_U
(d)	DGRY(P) DGRY(P)_U	儲存轉換後的格雷碼資料的起始元件 —	帶符號BIN32位元 無符號BIN32位元	ANY32_S ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

功能

- 將 (s) 中指定的BIN32位元資料轉換為BIN32位元格雷碼資料後，儲存到 (d) 中指定的元件中。



(s)+1: 高位16位元

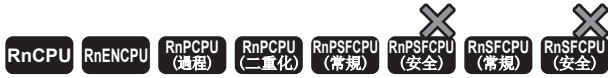
(s): 低位16位元

出錯

出錯代碼 (SD0)	內容
3401H	使用DGRY(P)指令時，(s) 中指定的值超出了0~2147483647的範圍時。

BIN16位元格雷碼資料→BIN16位元資料轉換

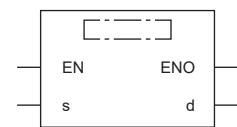
GBIN(P) (_U)



將指定的BIN16位元格雷碼資料轉換為BIN16位元資料。

梯形圖	ST	
	ENO:=GBIN(EN, s, d); ENO:=GBINP(EN, s, d);	ENO:=GBIN_U(EN, s, d); ENO:=GBINP_U(EN, s, d);

FBD/LD



■執行條件

指令	執行條件
GBIN GBIN_U	
GBINP GBINP_U	

設置數據

■內容、範圍、資料類型

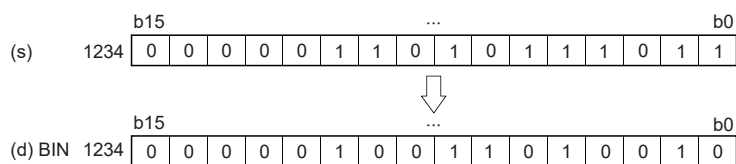
操作數	內容	範圍	資料類型	資料類型(標籤)	
(s)	GBIN(P)	格雷碼資料或儲存了格雷碼資料的元件	0~32767	帶符號BIN16位元	ANY16_S
	GBIN(P)_U		0~65535	無符號BIN16位元	ANY16_U
(d)	GBIN(P)	儲存轉換後的BIN資料的元件	—	帶符號BIN16位元	ANY16_S
	GBIN(P)_U			無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

功能

- 將(s)中指定的元件中儲存的BIN16位元格雷碼資料轉換為BIN16位元資料後，儲存到(d)中指定的元件中。



出錯

出錯代碼 (SD0)	內容
3401H	使用GBIN(P)指令時，(s)中指定的值超出了0~32767的範圍時。

BIN32位元格雷碼資料→BIN32位元資料轉換

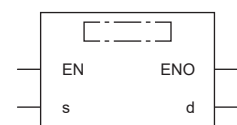
DGBIN(P) (_U)



將指定的BIN32位元格雷碼資料轉換為BIN32位元資料。

梯形圖	ST	
	ENO:=DGBIN(EN, s, d); ENO:=DGBINP(EN, s, d);	ENO:=DGBIN_U(EN, s, d); ENO:=DGBINP_U(EN, s, d);

FBD/LD



■執行條件

指令	執行條件
DGBIN DGBIN_U	
DGBINP DGBINP_U	

設置數據

■內容、範圍、資料類型

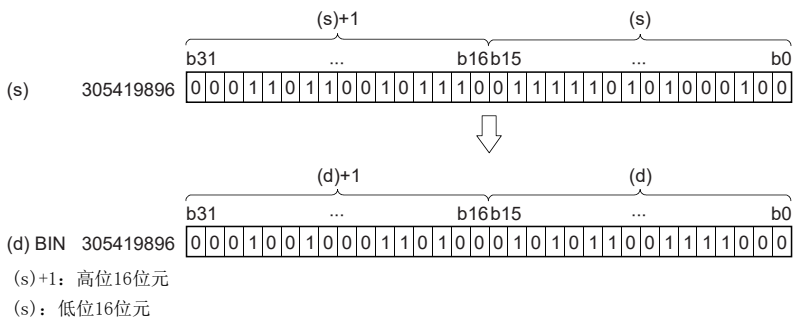
操作數	內容	範圍	資料類型	資料類型(標籤)	
(s)	DGBIN(P)	格雷碼資料或儲存了格雷碼資料的起始元件	0~2147483647	帶符號BIN32位元	ANY32_S
	DGBIN(P)_U		0~4294967295	無符號BIN32位元	ANY32_U
(d)	DGBIN(P)	儲存轉換後的BIN資料的起始元件	—	帶符號BIN32位元	ANY32_S
	DGBIN(P)_U		—	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 將 (s) 中指定的元件中儲存的BIN32位元格雷碼資料轉換為BIN32位元資料後，儲存到 (d) 中指定的元件中。

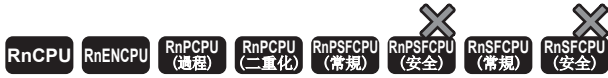


出錯

出錯代碼 (SD0)	內容
3401H	使用DGBIN(P)指令時，(s)中指定的值超出了0~2147483647的範圍時。

BIN16位元資料塊→BCD4位資料轉換塊

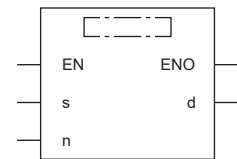
BKBCD (P)



對指定的元件開始的 (n) 點的BIN資料 (0~9999) 進行BCD轉換。

梯形圖	ST
	ENO:=BKBCD (EN, s, n, d) ; ENO:=BKBCDP (EN, s, n, d) ;

FBD/LD



執行條件

指令	執行條件
BKBCD	
BKBCDP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存了BIN資料的起始元件	—	帶符號BIN16位元	ANY16*1
(d)	儲存轉換後的BCD資料的起始元件	—	BCD4位	ANY16*1
(n)	變數資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

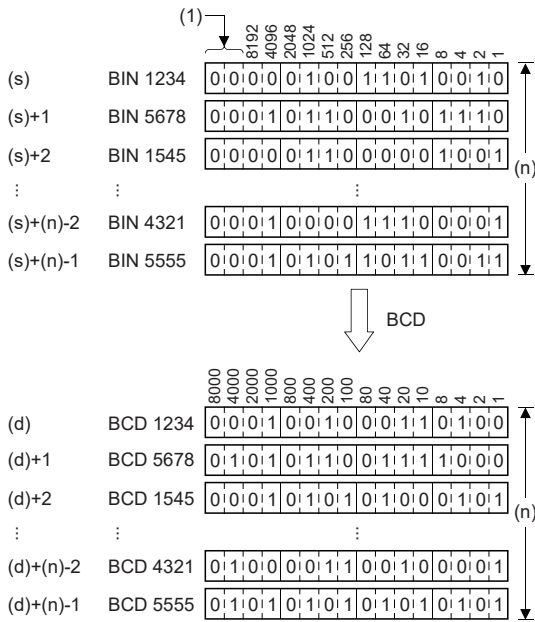
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 對從(s)中指定的元件開始的(n)點的BIN16位元資料(0~9999)進行BCD轉換後，儲存到(d)中指定的元件及其以後。



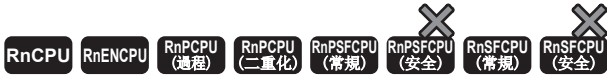
(1): 應設定為0。

出錯

出錯代碼 (SD0)	內容
2821H	當(s)、(d)的元件重複時。
3401H	當(s)的元件開始的(n)點的資料超出0~9999的範圍時。

塊BCD4位資料→塊BIN16位元轉換資料

BKBIN(P)



對指定的元件開始的(n)點的BCD資料(0~9999)進行BIN轉換。

梯形圖	ST
	ENO:=BKBIN(EN, s, n, d); ENO:=BKBINP(EN, s, n, d);

FBD/LD



■執行條件

指令	執行條件
BKBIN	
BKBINP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存了BCD資料的起始元件	—	BCD4位	ANY16*1
(d)	儲存轉換後的BIN資料的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	變數資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

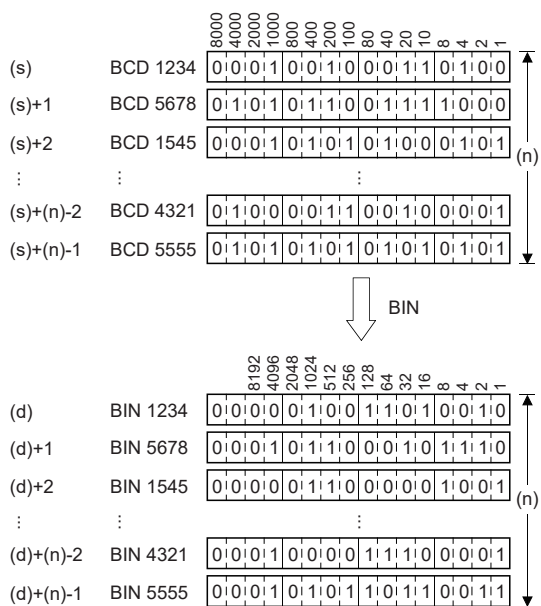
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 對從(s)中指定的元件開始的(n)點的BCD資料(0~9999)進行BIN16位元轉換後，儲存到(d)中指定的元件及其以後。

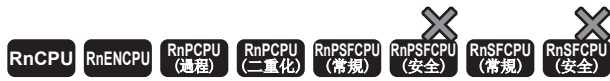


出錯

出錯代碼 (SD0)	內容
2821H	當 (s)、(d) 的元件重複時。
3401H	當 (s) 的元件開始的 (n) 點的資料超出 0~9999 的範圍時。

10進制ASCII資料→BIN16位元資料轉換

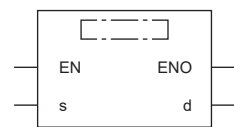
DABIN(P) (_U)



將10進制ASCII資料轉換為BIN16位元資料。

梯形圖	ST	
	ENO:=DABIN(EN, s, d); ENO:=DABINP(EN, s, d);	ENO:=DABIN_U(EN, s, d); ENO:=DABINP_U(EN, s, d);

FBD/LD



■執行條件

指令	執行條件
DABIN DABIN_U	
DABINP DABINP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	轉換為BIN值的ASCII資料、或儲存了ASCII資料的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	DABIN(P)	—	帶符號BIN16位元	ANY16_S
	DABIN(P)_U		無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

功能

- 將 (s) 中指定的元件編號及其以後中儲存的10進制ASCII資料轉換為BIN16位元資料後，儲存到 (d) 中指定的元件中。
- 根據SM705 (轉換位數切換) 的狀態，(s) 中設定的10進制ASCII資料的設定方法有所不同。

SM705*1的狀態	(s) 的設定方法	參閱目標
OFF	以固定位數(符號+數值部分5位)進行設定	466頁 SM705 (轉換位數切換) 為OFF的情況下的 (s) 的設定方法
ON	以任意的位數(最大: 符號+數值部分5位)進行設定	467頁 SM705 (轉換位數切換) 為ON的情況下的 (s) 的設定方法

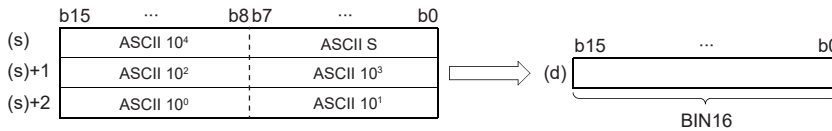
*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■SM705 (轉換位數切換) 為OFF的情況下的 (s) 的設定方法

(s) ~ (s)+2中，將10進制ASCII資料以固定位數進行設定。



ASCII S: 符號的ASCII碼

ASCII 10⁴: 萬位的ASCII碼

ASCII 10³: 千位的ASCII碼

ASCII 10²: 百位的ASCII碼

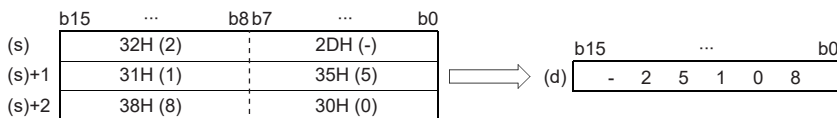
ASCII 10¹: 十位的ASCII碼

ASCII 10⁰: 個位的ASCII碼

- (s) ~ (s)+2中指定的ASCII資料，在DABIN(P)指令的情況下落在-32768~32767的範圍內，DABIN(P)_U指令的情況下落在0~65535的範圍內。
- (s)+3及其以後的資料將被忽略。
- 在符號資料中，轉換的資料為正時設定20H(空格)、為負時設定2DH(-)。(在設定20H、2DH以外的情況下，將被作為正的資料處理。)
- 各位中設定的ASCII碼的範圍為30H~39H。
- 各位中設定的ASCII碼為20H或00H時，將作為30H處理。

例

DABIN(P)指令中，(s)中設定“-25108”的情況下



■SM705 (轉換位數切換) 為ON的情況下的 (s) 的設定方法

(s) 中，設定任意的位數 (00H (NULL代碼) 為止) 的10進制ASCII資料。但是，整數部分為最大位數 (5位) 的情況下，無需00H (NULL代碼) 的設定。

(s) 的設定方法如下所示。

(s) 中設定的值	(s) ~ (s)+2 的內容	(s) 中設定的值	(s) ~ (s)+2 的內容																		
<ul style="list-style-type: none"> 0 正數 (數值部分1位) 	<ul style="list-style-type: none"> (s) 的高位字節中設定00H。 (s)+1 及其以後將被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td>(s)+1</td> <td>00H ... ASCII 10⁰</td> </tr> <tr> <td>(s)+2</td> <td></td> </tr> </table>	(s)	b15 ... b8 b7 ... b0	(s)+1	00H ... ASCII 10 ⁰	(s)+2		<ul style="list-style-type: none"> 正數 (數值部分2位) 負數 (數值部分1位) 	<ul style="list-style-type: none"> (s)+1 的低位字節中設定00H。 (s)+1 的高位字節及其以後將被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td>(s)</td> <td>ASCII 10⁰ ... ASCII 10¹ / 2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td></td> </tr> <tr> <td>(s)+2</td> <td>00H</td> </tr> </table>	(s)	b15 ... b8 b7 ... b0	(s)	ASCII 10 ⁰ ... ASCII 10 ¹ / 2DH (-)	(s)+1		(s)+2	00H				
(s)	b15 ... b8 b7 ... b0																				
(s)+1	00H ... ASCII 10 ⁰																				
(s)+2																					
(s)	b15 ... b8 b7 ... b0																				
(s)	ASCII 10 ⁰ ... ASCII 10 ¹ / 2DH (-)																				
(s)+1																					
(s)+2	00H																				
⋮																					
<ul style="list-style-type: none"> 負數 (數值部分4位) 	<ul style="list-style-type: none"> (s)+2 的高位字節中設定00H。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td>(s)</td> <td>ASCII 10³ ... 2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10¹ ... ASCII 10²</td> </tr> <tr> <td>(s)+2</td> <td>00H ... ASCII 10⁰</td> </tr> </table>	(s)	b15 ... b8 b7 ... b0	(s)	ASCII 10 ³ ... 2DH (-)	(s)+1	ASCII 10 ¹ ... ASCII 10 ²	(s)+2	00H ... ASCII 10 ⁰	<ul style="list-style-type: none"> 正數 (數值部分5位) 	<ul style="list-style-type: none"> (s)+2 的高位字節及其以後將被忽略。由於為最大位數，不需要00H的設定。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td>(s)</td> <td>ASCII 10³ ... ASCII 10⁴</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10¹ ... ASCII 10²</td> </tr> <tr> <td>(s)+2</td> <td></td> </tr> <tr> <td>(s)+2</td> <td>ASCII 10⁰</td> </tr> </table>	(s)	b15 ... b8 b7 ... b0	(s)	ASCII 10 ³ ... ASCII 10 ⁴	(s)+1	ASCII 10 ¹ ... ASCII 10 ²	(s)+2		(s)+2	ASCII 10 ⁰
(s)	b15 ... b8 b7 ... b0																				
(s)	ASCII 10 ³ ... 2DH (-)																				
(s)+1	ASCII 10 ¹ ... ASCII 10 ²																				
(s)+2	00H ... ASCII 10 ⁰																				
(s)	b15 ... b8 b7 ... b0																				
(s)	ASCII 10 ³ ... ASCII 10 ⁴																				
(s)+1	ASCII 10 ¹ ... ASCII 10 ²																				
(s)+2																					
(s)+2	ASCII 10 ⁰																				
<ul style="list-style-type: none"> 負數 (數值部分5位) 	<ul style="list-style-type: none"> (s)+3 及其以後將被忽略。由於為最大位數，不需要00H的設定。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td>(s)</td> <td>ASCII 10⁴ ... 2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10² ... ASCII 10³</td> </tr> <tr> <td>(s)+2</td> <td>ASCII 10⁰ ... ASCII 10¹</td> </tr> </table>	(s)	b15 ... b8 b7 ... b0	(s)	ASCII 10 ⁴ ... 2DH (-)	(s)+1	ASCII 10 ² ... ASCII 10 ³	(s)+2	ASCII 10 ⁰ ... ASCII 10 ¹	ASCII 10 ⁰ : 個位的ASCII碼 ASCII 10 ¹ : 十位的ASCII碼 ⋮ ASCII 10 ⁴ : 萬位的ASCII碼											
(s)	b15 ... b8 b7 ... b0																				
(s)	ASCII 10 ⁴ ... 2DH (-)																				
(s)+1	ASCII 10 ² ... ASCII 10 ³																				
(s)+2	ASCII 10 ⁰ ... ASCII 10 ¹																				

- (s) ~ (s)+2 中指定的ASCII資料，在DABIN (P) 指令的情況下落在-32768 ~ 32767 的範圍內，DABIN (P)_U 指令的情況下落在0 ~ 65535 的範圍內。
- 轉換資料為負數的情況下，在 (s)+0 的低位字節中作為符號資料設定2DH (-)。轉換資料為0或正數的情況下不設定符號資料，設定最高位位的ASCII碼。
- 各位中設定的ASCII碼的範圍為30H ~ 39H。
- 正數且數值部分5位的情況下，(s)+2 的高位字節及其以後的資料將被忽略。負數且數值部分5位的情況下，(s)+3 及其以後的資料將被忽略。
- 各位中設定的ASCII碼為20H時，將作為30H處理。00H時，作為10進制ASCII資料的終端進行處理。
- 下述的情況下，(d) 將儲存0。
 - 第1個字元為00H (NULL)
 - 第1個字元為2DH (-)，且第2個字元為00H (NULL)

出錯

出錯代碼 (SD0)	內容
3401H	當 (s) ~ (s)+2 中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none"> 第1個字元的ASCII碼超出2DH、30H ~ 39H、20H、00H 的範圍。*1 第2個字元及其以後的ASCII碼超出30H ~ 39H、20H、00H 的範圍。 使用DABIN (P) 指令時，ASCII資料為-32768 ~ 32767 以外。 使用DABIN (P)_U 指令時，ASCII資料為0 ~ 65535 以外。

*1 SM705 (轉換位數切換) 為OFF 的情況下，無論第一個字元的ASCII碼設定為何，皆不會變為出錯。

10進制ASCII資料→BIN32位元資料轉換

DDABIN(P) (_U)



將10進制ASCII資料轉換為BIN32位元資料。

梯形圖	ST	
	ENO:=DDABIN(EN, s, d); ENO:=DDABINP(EN, s, d);	ENO:=DDABIN_U(EN, s, d); ENO:=DDABINP_U(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
DDABIN DDABIN_U	
DDABINP DDABINP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	轉換為BIN值的ASCII資料、或儲存了ASCII資料的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	DDABIN(P)	—	帶符號BIN32位元	ANY32_S
	DDABIN(P)_U		無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—	
(d)	○	○	○	○	○	○	○	○	—	—	—	—	

功能

- 將 (s) 中指定的元件編號及其以後中儲存的10進制ASCII資料轉換為BIN32位元資料後，儲存到 (d) 中指定的元件編號中。
- 根據SM705 (轉換位數切換) 的狀態，(s) 中設定的10進制ASCII資料的設定方法有所不同。

SM705*1的狀態	(s)的設定方法	參閱目標
OFF	以固定位數(符號+數值部分10位)進行設定	469頁 SM705 (轉換位數切換) 為OFF的情況下的 (s) 的設定方法
ON	以任意的位數(最大: 符號+數值部分10位)進行設定	470頁 SM705 (轉換位數切換) 為ON的情況下的 (s) 的設定方法

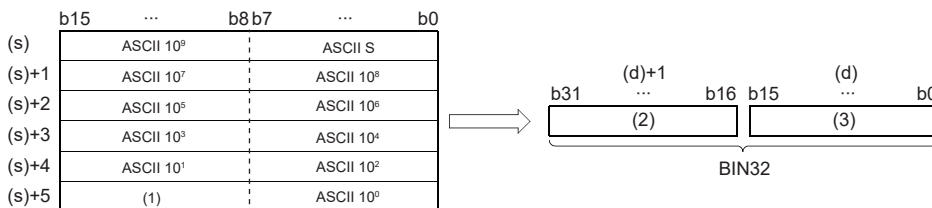
*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■SM705 (轉換位數切換) 為OFF的情況下的 (s) 的設定方法

(s) ~ (s)+5中，將10進制ASCII資料以固定位數進行設定。



ASCII S: 符號的ASCII碼

ASCII 10⁰: 個位的ASCII碼

ASCII 10¹: 十位的ASCII碼

ASCII 10²: 百位的ASCII碼

ASCII 10³: 千位的ASCII碼

ASCII 10⁴: 萬位的ASCII碼

ASCII 10⁵: 十萬位的ASCII碼

ASCII 10⁶: 百萬位的ASCII碼

ASCII 10⁷: 千萬位的ASCII碼

ASCII 10⁸: 億位的ASCII碼

ASCII 10⁹: 十億位的ASCII碼

(1): 忽略。

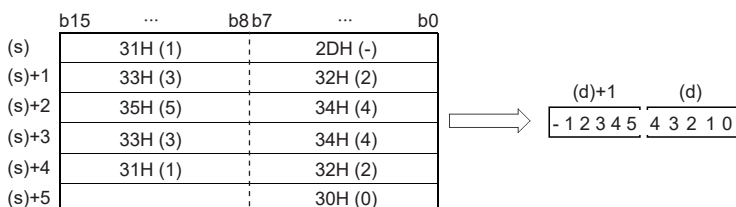
(2): 高位16位元

(3): 低位16位元

- (s) ~ (s)+5中指定的ASCII資料，在DDABIN(P)指令的情況下落在-2147483648~2147483647的範圍內，DDABIN(P)_U指令的情況下落在0~4294967295的範圍內。此外，(s)+5的高位字節、(s)+6及其以後中儲存的資料將被忽略。
- 在符號資料中，轉換的資料為正時設定20H、為負時設定2DH。(在設定20H、2DH以外的情況下，將被作為正的資料處理。)
- 各位中設定的ASCII碼的範圍為30H~39H。
- 各位中設定的ASCII碼為20H或00H時，將作為30H處理。

例

DDABIN(P)指令中，(s)中設定“-1234543210”的情況下



■SM705 (轉換位數切換) 為ON的情況下的 (s) 的設定方法

(s) 中，設定任意的位數 (00H (NULL代碼) 為止) 的10進制ASCII資料。但是，整數部分為最大位數 (10位) 的情況下，無需 00H (NULL代碼) 的設定。

(s) 的設定方法如下所示。

(s) 中設定的值	(s) ~ (s)+5 的內容	(s) 中設定的值	(s) ~ (s)+5 的內容																														
<ul style="list-style-type: none"> 0 正數 (數值部分1位) 	<ul style="list-style-type: none"> (s) 的高位字節中設定00H。 (s)+1 及其以後將被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td></td> <td>00H ... ASCII 10⁰</td> </tr> <tr> <td>(s)+1</td> <td></td> </tr> <tr> <td>(s)+2</td> <td></td> </tr> <tr> <td>(s)+3</td> <td></td> </tr> <tr> <td>(s)+4</td> <td></td> </tr> <tr> <td>(s)+5</td> <td></td> </tr> </table>	(s)	b15 ... b8 b7 ... b0		00H ... ASCII 10 ⁰	(s)+1		(s)+2		(s)+3		(s)+4		(s)+5		<ul style="list-style-type: none"> 正數 (數值部分2位) 負數 (數值部分1位) 	<ul style="list-style-type: none"> (s)+1 的低位字節中設定00H。 (s)+1 的高位字節及其以後將被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td></td> <td>ASCII 10⁰ ... ASCII 10¹ / 2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td></td> </tr> <tr> <td></td> <td>00H</td> </tr> <tr> <td>(s)+2</td> <td></td> </tr> <tr> <td>(s)+3</td> <td></td> </tr> <tr> <td>(s)+4</td> <td></td> </tr> <tr> <td>(s)+5</td> <td></td> </tr> </table>	(s)	b15 ... b8 b7 ... b0		ASCII 10 ⁰ ... ASCII 10 ¹ / 2DH (-)	(s)+1			00H	(s)+2		(s)+3		(s)+4		(s)+5	
(s)	b15 ... b8 b7 ... b0																																
	00H ... ASCII 10 ⁰																																
(s)+1																																	
(s)+2																																	
(s)+3																																	
(s)+4																																	
(s)+5																																	
(s)	b15 ... b8 b7 ... b0																																
	ASCII 10 ⁰ ... ASCII 10 ¹ / 2DH (-)																																
(s)+1																																	
	00H																																
(s)+2																																	
(s)+3																																	
(s)+4																																	
(s)+5																																	
<ul style="list-style-type: none"> 正數 (數值部分9位) 負數 (數值部分8位) 	<ul style="list-style-type: none"> (s)+4 的高位字節中設定00H。 (s)+5 之後將被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td></td> <td>ASCII 10⁷ ... ASCII 10⁸ / 2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10⁵ ... ASCII 10⁶</td> </tr> <tr> <td>(s)+2</td> <td>ASCII 10³ ... ASCII 10⁴</td> </tr> <tr> <td>(s)+3</td> <td>ASCII 10¹ ... ASCII 10²</td> </tr> <tr> <td>(s)+4</td> <td>00H ... ASCII 10⁰</td> </tr> <tr> <td>(s)+5</td> <td></td> </tr> </table>	(s)	b15 ... b8 b7 ... b0		ASCII 10 ⁷ ... ASCII 10 ⁸ / 2DH (-)	(s)+1	ASCII 10 ⁵ ... ASCII 10 ⁶	(s)+2	ASCII 10 ³ ... ASCII 10 ⁴	(s)+3	ASCII 10 ¹ ... ASCII 10 ²	(s)+4	00H ... ASCII 10 ⁰	(s)+5		<ul style="list-style-type: none"> 負數 (數值部分9位) 	<ul style="list-style-type: none"> (s)+5 的低位字節中設定00H。 (s)+5 的高位字節將被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td></td> <td>ASCII 10⁸ ... 2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10⁶ ... ASCII 10⁷</td> </tr> <tr> <td>(s)+2</td> <td>ASCII 10⁴ ... ASCII 10⁵</td> </tr> <tr> <td>(s)+3</td> <td>ASCII 10² ... ASCII 10³</td> </tr> <tr> <td>(s)+4</td> <td>ASCII 10⁰ ... ASCII 10¹</td> </tr> <tr> <td>(s)+5</td> <td></td> </tr> <tr> <td></td> <td>00H</td> </tr> </table>	(s)	b15 ... b8 b7 ... b0		ASCII 10 ⁸ ... 2DH (-)	(s)+1	ASCII 10 ⁶ ... ASCII 10 ⁷	(s)+2	ASCII 10 ⁴ ... ASCII 10 ⁵	(s)+3	ASCII 10 ² ... ASCII 10 ³	(s)+4	ASCII 10 ⁰ ... ASCII 10 ¹	(s)+5			00H
(s)	b15 ... b8 b7 ... b0																																
	ASCII 10 ⁷ ... ASCII 10 ⁸ / 2DH (-)																																
(s)+1	ASCII 10 ⁵ ... ASCII 10 ⁶																																
(s)+2	ASCII 10 ³ ... ASCII 10 ⁴																																
(s)+3	ASCII 10 ¹ ... ASCII 10 ²																																
(s)+4	00H ... ASCII 10 ⁰																																
(s)+5																																	
(s)	b15 ... b8 b7 ... b0																																
	ASCII 10 ⁸ ... 2DH (-)																																
(s)+1	ASCII 10 ⁶ ... ASCII 10 ⁷																																
(s)+2	ASCII 10 ⁴ ... ASCII 10 ⁵																																
(s)+3	ASCII 10 ² ... ASCII 10 ³																																
(s)+4	ASCII 10 ⁰ ... ASCII 10 ¹																																
(s)+5																																	
	00H																																
<ul style="list-style-type: none"> 正數 (數值部分10位) 	<ul style="list-style-type: none"> (s)+5 之後將被忽略。由於為最大位數，不需要00H 的設定。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td></td> <td>ASCII 10⁸ ... ASCII 10⁹</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10⁶ ... ASCII 10⁷</td> </tr> <tr> <td>(s)+2</td> <td>ASCII 10⁴ ... ASCII 10⁵</td> </tr> <tr> <td>(s)+3</td> <td>ASCII 10² ... ASCII 10³</td> </tr> <tr> <td>(s)+4</td> <td>ASCII 10⁰ ... ASCII 10¹</td> </tr> <tr> <td>(s)+5</td> <td></td> </tr> </table>	(s)	b15 ... b8 b7 ... b0		ASCII 10 ⁸ ... ASCII 10 ⁹	(s)+1	ASCII 10 ⁶ ... ASCII 10 ⁷	(s)+2	ASCII 10 ⁴ ... ASCII 10 ⁵	(s)+3	ASCII 10 ² ... ASCII 10 ³	(s)+4	ASCII 10 ⁰ ... ASCII 10 ¹	(s)+5		<ul style="list-style-type: none"> 負數 (數值部分10位) 	<ul style="list-style-type: none"> (s)+5 的高位字節將被忽略。由於為最大位數，不需要00H 的設定。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td></td> <td>ASCII 10⁹ ... 2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10⁷ ... ASCII 10⁸</td> </tr> <tr> <td>(s)+2</td> <td>ASCII 10⁵ ... ASCII 10⁶</td> </tr> <tr> <td>(s)+3</td> <td>ASCII 10³ ... ASCII 10⁴</td> </tr> <tr> <td>(s)+4</td> <td>ASCII 10¹ ... ASCII 10²</td> </tr> <tr> <td>(s)+5</td> <td></td> </tr> <tr> <td></td> <td>ASCII 10⁰</td> </tr> </table>	(s)	b15 ... b8 b7 ... b0		ASCII 10 ⁹ ... 2DH (-)	(s)+1	ASCII 10 ⁷ ... ASCII 10 ⁸	(s)+2	ASCII 10 ⁵ ... ASCII 10 ⁶	(s)+3	ASCII 10 ³ ... ASCII 10 ⁴	(s)+4	ASCII 10 ¹ ... ASCII 10 ²	(s)+5			ASCII 10 ⁰
(s)	b15 ... b8 b7 ... b0																																
	ASCII 10 ⁸ ... ASCII 10 ⁹																																
(s)+1	ASCII 10 ⁶ ... ASCII 10 ⁷																																
(s)+2	ASCII 10 ⁴ ... ASCII 10 ⁵																																
(s)+3	ASCII 10 ² ... ASCII 10 ³																																
(s)+4	ASCII 10 ⁰ ... ASCII 10 ¹																																
(s)+5																																	
(s)	b15 ... b8 b7 ... b0																																
	ASCII 10 ⁹ ... 2DH (-)																																
(s)+1	ASCII 10 ⁷ ... ASCII 10 ⁸																																
(s)+2	ASCII 10 ⁵ ... ASCII 10 ⁶																																
(s)+3	ASCII 10 ³ ... ASCII 10 ⁴																																
(s)+4	ASCII 10 ¹ ... ASCII 10 ²																																
(s)+5																																	
	ASCII 10 ⁰																																

ASCII 10⁰: 個位的ASCII碼

ASCII 10¹: 十位的ASCII碼

⋮

ASCII 10⁹: 十億位的ASCII碼

- (s) ~ (s)+5 中指定的ASCII資料，在 DDABIN (P) 指令的情況下落在 -2147483648 ~ 2147483647 的範圍內，DDABIN (P)_U 指令的情況下落在 0 ~ 4294967295 的範圍內。
- 轉換資料為負數的情況下，在 (s)+0 的低位字節中作為符號資料設定 2DH (-)。轉換資料為 0 或正數的情況下不設定符號資料，設定最高位位的ASCII碼。
- 各位中設定的ASCII碼的範圍為 30H ~ 39H。
- 正數且數值部分10位的情況下，(s)+5 及其以後儲存的資料將被忽略。負數且數值部分10位的情況下，(s)+5 的高位字節及其以後儲存的資料將被忽略。
- 各位中設定的ASCII碼為 20H 時，將作為 30H 處理。00H 時，作為 10 進制ASCII資料的終端進行處理。
- 下述的情況下，(d) 將儲存 0。
 - 第1個字元為 00H (NULL)
 - 第1個字元為 2DH (-)，且第2個字元為 00H (NULL)

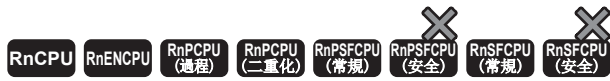
出錯

出錯代碼 (SD0)	內容
3401H	當 (s) ~ (s)+5 中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none"> 第1個字元的ASCII碼超出 2DH、30H ~ 39H、20H、00H 的範圍。*1 第2個字元及其以後的ASCII碼超出 30H ~ 39H、20H、00H 的範圍。 使用 DDABIN (P) 指令時，ASCII資料為 -2147483648 ~ 2147483647 以外。 使用 DDABIN (P)_U 指令時，ASCII資料為 0 ~ 4294967295 以外。

*1 SM705 (轉換位數切換) 為 OFF 的情況下，無論第一個字元的ASCII碼設定為何，皆不會變為出錯。

16進制ASCII資料→BIN16位元資料轉換

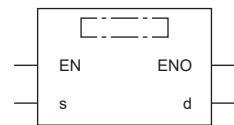
HABIN(P)



將16進制ASCII資料轉換為BIN16位元資料。

梯形圖	ST
	ENO:=HABIN(EN, s, d); ENO:=HABINP(EN, s, d);

FBD/LD



執行條件

指令	執行條件
HABIN	
HABINP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	轉換為BIN值的ASCII資料、或儲存了ASCII資料的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存轉換結果的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 將 (s) 中指定的元件編號及其以後中儲存的16進制ASCII資料轉換為BIN16位元資料後，儲存到 (d) 中指定的元件中。
- 根據SM705 (轉換位數切換) 的狀態，(s) 中設定的16進制ASCII資料的設定方法有所不同。

SM705*1的狀態	(s)的設定方法	參閱目標
OFF	以固定位數(4位)進行設定	472頁 SM705 (轉換位數切換) 為OFF的情況下的 (s) 的設定方法
ON	以任意的位數(最大4位)進行設定	472頁 SM705 (轉換位數切換) 為ON的情況下的 (s) 的設定方法

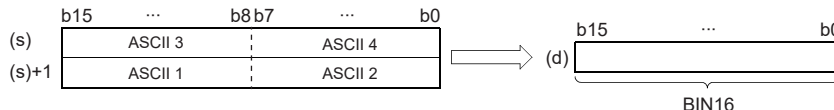
*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■SM705 (轉換位數切換) 為OFF的情況下的 (s) 的設定方法

- (s) ~ (s)+1 中，將16進制ASCII資料以4位固定進行設定。

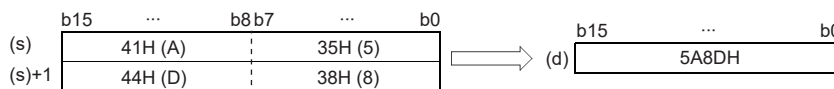


ASCII □: 第 □ 位的ASCII碼

- (s) ~ (s)+1 中指定的ASCII資料為0000H~FFFFH的範圍內。
- (s)+2及其以後的資料將被忽略。
- 各位中設定的ASCII碼為30H~39H及41H~46H的範圍內。

例

(s) 中指定了5A8DH的情況下



■SM705 (轉換位數切換) 為ON的情況下的 (s) 的設定方法

(s) 中，設定任意的位數 (00H (NULL代碼) 為止) 的16進制ASCII資料。但是，最大位數 (4位) 的情況下，無需00H (NULL代碼) 的設定。

(s) 的設定方法如下所示。

(s) 中設定的值	(s) ~ (s)+1 的內容	(s) 中設定的值	(s) ~ (s)+1 的內容																														
• 0H~FH	<ul style="list-style-type: none"> (s)+0 的高位字節中設定00H。 (s)+1 及其以後將被忽略。 <div style="text-align: center;"> <table border="1"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(s)</td><td>00H</td><td></td><td>ASCII 1</td><td></td> </tr> <tr> <td>(s)+1</td><td colspan="4" style="background-color: #cccccc;"></td> </tr> </table> </div>	b15	...	b8 b7	...	b0	(s)	00H		ASCII 1		(s)+1					• 10H~FFH	<ul style="list-style-type: none"> (s)+1 的低位字節中設定00H。 (s)+1 的高位字節及其以後將被忽略。 <div style="text-align: center;"> <table border="1"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(s)</td><td>ASCII 1</td><td></td><td>ASCII 2</td><td></td> </tr> <tr> <td>(s)+1</td><td colspan="4" style="background-color: #cccccc;"></td> </tr> </table> </div>	b15	...	b8 b7	...	b0	(s)	ASCII 1		ASCII 2		(s)+1				
b15	...	b8 b7	...	b0																													
(s)	00H		ASCII 1																														
(s)+1																																	
b15	...	b8 b7	...	b0																													
(s)	ASCII 1		ASCII 2																														
(s)+1																																	
• 100H~FFFH	<ul style="list-style-type: none"> (s)+1 的高位字節中設定00H。 (s)+2 及其以後將被忽略。 <div style="text-align: center;"> <table border="1"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(s)</td><td>ASCII 2</td><td></td><td>ASCII 3</td><td></td> </tr> <tr> <td>(s)+1</td><td>00H</td><td></td><td>ASCII 1</td><td></td> </tr> </table> </div>	b15	...	b8 b7	...	b0	(s)	ASCII 2		ASCII 3		(s)+1	00H		ASCII 1		• 1000H~FFFFH	<ul style="list-style-type: none"> (s)+2 及其以後將被忽略。 <div style="text-align: center;"> <table border="1"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(s)</td><td>ASCII 3</td><td></td><td>ASCII 4</td><td></td> </tr> <tr> <td>(s)+1</td><td>ASCII 1</td><td></td><td>ASCII 2</td><td></td> </tr> </table> </div>	b15	...	b8 b7	...	b0	(s)	ASCII 3		ASCII 4		(s)+1	ASCII 1		ASCII 2	
b15	...	b8 b7	...	b0																													
(s)	ASCII 2		ASCII 3																														
(s)+1	00H		ASCII 1																														
b15	...	b8 b7	...	b0																													
(s)	ASCII 3		ASCII 4																														
(s)+1	ASCII 1		ASCII 2																														

ASCII □: 第 □ 位的ASCII碼

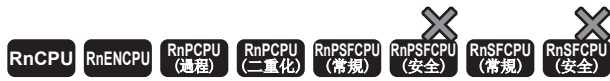
- (s) ~ (s)+1 中指定的ASCII資料為0000H~FFFFH的範圍內。
- (s)+2及其以後的資料將被忽略。
- 各位中設定的ASCII碼為30H~39H及41H~46H的範圍內。
- 各位中設定的ASCII碼為00H時，將作為16進制ASCII資料的終端進行處理。

出錯

出錯代碼(SD0)	內容
3401H	當 (s) ~ (s)+1 中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none"> 各位的ASCII碼為30H~39H、41H~46H以外。

16進制ASCII資料→BIN32位元資料轉換

DHABIN(P)



將16進制ASCII資料轉換為BIN32位元資料。

梯形圖	ST
	ENO:=DHABIN(EN, s, d); ENO:=DHABINP(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
DHABIN	
DHABINP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	轉換為BIN值的ASCII資料、或儲存了ASCII資料的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存轉換結果的起始元件	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 將 (s) 中指定的元件編號及其以後中儲存的16進制ASCII資料轉換為BIN32位元資料後，儲存到 (d) 中指定的元件編號中。
- 根據SM705 (轉換位數切換) 的狀態，(s) 中設定的16進制ASCII資料的設定方法有所不同。

SM705*1的狀態	(s) 的設定方法	參閱目標
OFF	以固定位數(8位)進行設定	474頁 SM705 (轉換位數切換) 為OFF的情況下的 (s) 的設定方法
ON	以任意的位數(最大8位)進行設定	475頁 SM705 (轉換位數切換) 為ON的情況下的 (s) 的設定方法

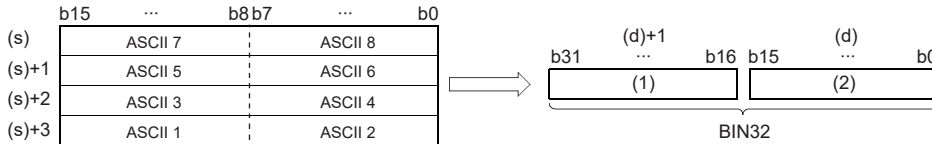
*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■SM705 (轉換位數切換) 為OFF的情況下的 (s) 的設定方法

- (s) ~ (s)+3中，將16進制ASCII資料以8位固定進行設定。



ASCII □: 第 □ 位的ASCII碼

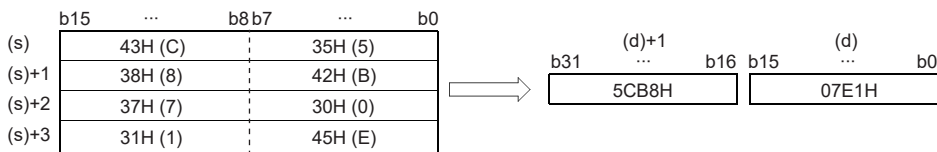
(1): 高位16位元

(2): 低位16位元

- (s) ~ (s)+3中指定的ASCII資料為00000000H~FFFFFFFH的範圍內。
- (s)+4及其以後的資料將被忽略。
- 各位中設定的ASCII碼為30H~39H及41H~46H的範圍內。

例

在 (s) 中指定了5CB807E1H的情況下



■SM705 (轉換位數切換) 為0N的情況下的 (s) 的設定方法

(s) 中，設定任意的位數 (00H (NULL代碼) 為止) 的16進制ASCII資料。但是，最大位數 (8位) 的情況下，無需00H (NULL代碼) 的設定。

(s) 的設定方法如下所示。

(s) 中設定的值	(s) ~ (s)+3 的內容	(s) 中設定的值	(s) ~ (s)+3 的內容																																																		
• 0H~FH	<ul style="list-style-type: none"> (s) 的高位字節中設定00H。 (s)+1及其以後將被忽略。 <table border="1"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(s)</td><td>00H</td><td>:</td><td>ASCII 1</td><td></td> </tr> <tr> <td>(s)+1</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>(s)+2</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>(s)+3</td><td></td><td></td><td></td><td></td> </tr> </table>	b15	...	b8 b7	...	b0	(s)	00H	:	ASCII 1		(s)+1					(s)+2					(s)+3					• 10H~FFH	<ul style="list-style-type: none"> (s)+1的低位字節中設定00H。 (s)+1的高位字節及其以後將被忽略。 <table border="1"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(s)</td><td>ASCII 1</td><td>:</td><td>ASCII 2</td><td></td> </tr> <tr> <td>(s)+1</td><td></td><td></td><td>00H</td><td></td> </tr> <tr> <td>(s)+2</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>(s)+3</td><td></td><td></td><td></td><td></td> </tr> </table>	b15	...	b8 b7	...	b0	(s)	ASCII 1	:	ASCII 2		(s)+1			00H		(s)+2					(s)+3				
b15	...	b8 b7	...	b0																																																	
(s)	00H	:	ASCII 1																																																		
(s)+1																																																					
(s)+2																																																					
(s)+3																																																					
b15	...	b8 b7	...	b0																																																	
(s)	ASCII 1	:	ASCII 2																																																		
(s)+1			00H																																																		
(s)+2																																																					
(s)+3																																																					
⋮																																																					
• 1000000H~FFFFFFFH	<ul style="list-style-type: none"> (s)+3的高位字節中設定00H。 (s)+4及其以後將被忽略。 <table border="1"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(s)</td><td>ASCII 6</td><td>:</td><td>ASCII 7</td><td></td> </tr> <tr> <td>(s)+1</td><td>ASCII 4</td><td>:</td><td>ASCII 5</td><td></td> </tr> <tr> <td>(s)+2</td><td>ASCII 2</td><td>:</td><td>ASCII 3</td><td></td> </tr> <tr> <td>(s)+3</td><td>00H</td><td>:</td><td>ASCII 1</td><td></td> </tr> </table>	b15	...	b8 b7	...	b0	(s)	ASCII 6	:	ASCII 7		(s)+1	ASCII 4	:	ASCII 5		(s)+2	ASCII 2	:	ASCII 3		(s)+3	00H	:	ASCII 1		• 10000000H~FFFFFFFH	<ul style="list-style-type: none"> (s)+4及其以後將被忽略。 <table border="1"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(s)</td><td>ASCII 7</td><td>:</td><td>ASCII 8</td><td></td> </tr> <tr> <td>(s)+1</td><td>ASCII 5</td><td>:</td><td>ASCII 6</td><td></td> </tr> <tr> <td>(s)+2</td><td>ASCII 3</td><td>:</td><td>ASCII 4</td><td></td> </tr> <tr> <td>(s)+3</td><td>ASCII 1</td><td>:</td><td>ASCII 2</td><td></td> </tr> </table>	b15	...	b8 b7	...	b0	(s)	ASCII 7	:	ASCII 8		(s)+1	ASCII 5	:	ASCII 6		(s)+2	ASCII 3	:	ASCII 4		(s)+3	ASCII 1	:	ASCII 2	
b15	...	b8 b7	...	b0																																																	
(s)	ASCII 6	:	ASCII 7																																																		
(s)+1	ASCII 4	:	ASCII 5																																																		
(s)+2	ASCII 2	:	ASCII 3																																																		
(s)+3	00H	:	ASCII 1																																																		
b15	...	b8 b7	...	b0																																																	
(s)	ASCII 7	:	ASCII 8																																																		
(s)+1	ASCII 5	:	ASCII 6																																																		
(s)+2	ASCII 3	:	ASCII 4																																																		
(s)+3	ASCII 1	:	ASCII 2																																																		

ASCII □: 第 □位的ASCII碼

- (s) ~ (s)+3中指定的ASCII資料為00000000H~FFFFFFFH的範圍內。
- (s)+4及其以後的資料將被忽略。
- 各位中設定的ASCII碼為30H~39H及41H~46H的範圍內。
- 各位中設定的ASCII碼為00H時，將作為16進制ASCII資料的終端進行處理。

出錯

出錯代碼 (SD0)	內容
3401H	當 (s) ~ (s)+3 中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none"> • 各位的ASCII碼為30H~39H、41H~46H以外。

10進制ASCII資料→BCD4位資料轉換

DABCD (P)



將10進制ASCII資料轉換為BCD4位資料。

梯形圖	ST
	ENO:=DABCD (EN, s, d) ; ENO:=DABCDP (EN, s, d)

FBD/LD

■執行條件

指令	執行條件
DABCD	
DABCDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	轉換為BCD值的ASCII資料、或儲存了ASCII資料的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存轉換結果的元件	—	BCD4位	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 將(s)中指定的元件編號及其以後中儲存的10進制ASCII資料轉換為BCD4位資料後，儲存到(d)中指定的元件中。
- 根據SM705(轉換位數切換)的狀態，(s)中設定的10進制ASCII資料的設定方法有所不同。

SM705*1的狀態	(s)的設定方法	參閱目標
OFF	以固定位數(4位)進行設定	477頁 SM705(轉換位數切換)為OFF的情況下的(s)的設定方法
ON	以任意的位數(最大4位)進行設定	478頁 SM705(轉換位數切換)為ON的情況下的(s)的設定方法

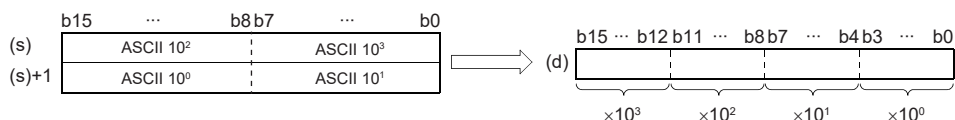
*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■SM705(轉換位數切換)為OFF的情況下的(s)的設定方法

- (s)~(s)+1中，將10進制ASCII資料以4位固定進行設定。



ASCII 10³: 千位的ASCII碼

ASCII 10²: 百位的ASCII碼

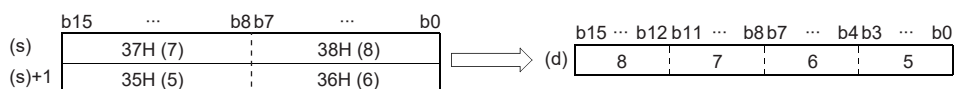
ASCII 10¹: 十位的ASCII碼

ASCII 10⁰: 個位的ASCII碼

- (s)~(s)+1中指定的ASCII資料為0~9999的範圍內。
- (s)+2及其以後的資料將被忽略。
- 各位中設定的ASCII碼的範圍為30H~39H。
- 各位中設定的ASCII碼為20H或00H時，將作為30H處理。

例

在(s)中指定了8765的情況下



■SM705 (轉換位數切換) 為0N的情況下的 (s) 的設定方法

(s) 中，設定任意的位數 (00H (NULL代碼) 為止) 的10進制ASCII資料。但是，最大位數 (4位) 的情況下，無需00H (NULL代碼) 的設定。

(s) 的設定方法如下所示。

(s) 中設定的值	(s) ~ (s)+1 的內容	(s) 中設定的值	(s) ~ (s)+1 的內容																														
<ul style="list-style-type: none"> 0~9 	<ul style="list-style-type: none"> (s)+0 的高位字節中設定00H。 (s)+1 及其以後將被忽略。 <table border="1"> <tr> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td>00H</td> <td>ASCII 10⁰</td> <td></td> <td></td> </tr> <tr> <td>(s)+1</td> <td colspan="4" style="background-color: #cccccc;"></td> </tr> </table>	b15	...	b8 b7	...	b0	(s)	00H	ASCII 10 ⁰			(s)+1					<ul style="list-style-type: none"> 10~99 	<ul style="list-style-type: none"> (s)+1 的低位字節中設定00H。 (s)+1 的高位字節及其以後將被忽略。 <table border="1"> <tr> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td>ASCII 10⁰</td> <td>ASCII 10¹</td> <td></td> <td></td> </tr> <tr> <td>(s)+1</td> <td colspan="2" style="background-color: #cccccc;"></td> <td>00H</td> <td></td> </tr> </table>	b15	...	b8 b7	...	b0	(s)	ASCII 10 ⁰	ASCII 10 ¹			(s)+1			00H	
b15	...	b8 b7	...	b0																													
(s)	00H	ASCII 10 ⁰																															
(s)+1																																	
b15	...	b8 b7	...	b0																													
(s)	ASCII 10 ⁰	ASCII 10 ¹																															
(s)+1			00H																														
<ul style="list-style-type: none"> 100~999 	<ul style="list-style-type: none"> (s)+1 的高位字節中設定00H。 (s)+2 及其以後將被忽略。 <table border="1"> <tr> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td>ASCII 10¹</td> <td>ASCII 10²</td> <td></td> <td></td> </tr> <tr> <td>(s)+1</td> <td>00H</td> <td>ASCII 10⁰</td> <td></td> <td></td> </tr> </table>	b15	...	b8 b7	...	b0	(s)	ASCII 10 ¹	ASCII 10 ²			(s)+1	00H	ASCII 10 ⁰			<ul style="list-style-type: none"> 1000~9999 	<ul style="list-style-type: none"> (s)+2 及其以後將被忽略。 <table border="1"> <tr> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td>ASCII 10²</td> <td>ASCII 10³</td> <td></td> <td></td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10⁰</td> <td>ASCII 10¹</td> <td></td> <td></td> </tr> </table>	b15	...	b8 b7	...	b0	(s)	ASCII 10 ²	ASCII 10 ³			(s)+1	ASCII 10 ⁰	ASCII 10 ¹		
b15	...	b8 b7	...	b0																													
(s)	ASCII 10 ¹	ASCII 10 ²																															
(s)+1	00H	ASCII 10 ⁰																															
b15	...	b8 b7	...	b0																													
(s)	ASCII 10 ²	ASCII 10 ³																															
(s)+1	ASCII 10 ⁰	ASCII 10 ¹																															

ASCII 10³: 千位的ASCII碼

ASCII 10²: 百位的ASCII碼

ASCII 10¹: 十位的ASCII碼

ASCII 10⁰: 個位的ASCII碼

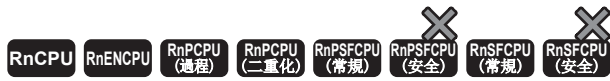
- (s) ~ (s)+1 中指定的ASCII資料為0~9999的範圍內。
- (s)+2 及其以後的資料將被忽略。
- 各位中設定的ASCII碼的範圍為30H~39H。
- 各位中設定的ASCII碼為20H時，將作為30H處理。00H時，作為10進制ASCII資料的終端進行處理。

出錯

出錯代碼 (SD0)	內容
3401H	當 (s) 中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none"> 資料中有0~9以外的字元。

10進制ASCII資料→BCD8位資料轉換

DDABCD (P)



將10進制ASCII資料轉換為BCD8位資料。

梯形圖	ST
	ENO:=DDABCD (EN, s, d) ; ENO:=DDABCDP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
DDABCD	
DDABCDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	轉換為BCD值的ASCII資料、或儲存了ASCII資料的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存轉換結果的起始元件	—	BCD8位	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 將(s)中指定的元件編號及其以後中儲存的10進制ASCII資料轉換為BCD8位資料後，儲存到(d)中指定的元件編號中。
- 根據SM705(轉換位數切換)的狀態，(s)中設定的10進制ASCII資料的設定方法有所不同。

SM705*1的狀態	(s)的設定方法	參閱目標
OFF	以固定位數(8位)進行設定	480頁 SM705(轉換位數切換)為OFF的情況下的(s)的設定方法
ON	以任意的位數(最大8位)進行設定	481頁 SM705(轉換位數切換)為ON的情況下的(s)的設定方法

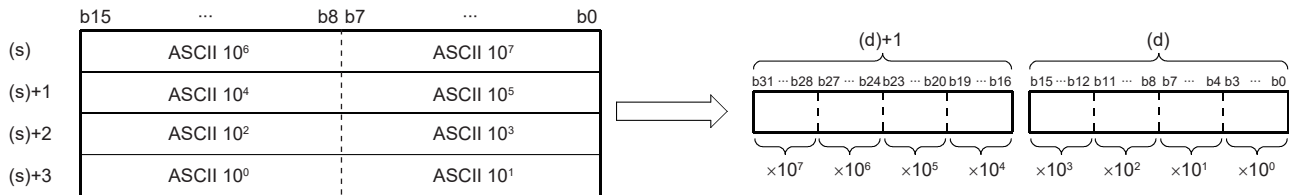
*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■SM705(轉換位數切換)為OFF的情況下的(s)的設定方法

- (s)~(s)+3中，將10進制ASCII資料以8位固定進行設定。



ASCII 10⁷: 千萬位的ASCII碼

ASCII 10⁶: 百萬位的ASCII碼

ASCII 10⁵: 十萬位的ASCII碼

ASCII 10⁴: 萬位的ASCII碼

ASCII 10³: 千位的ASCII碼

ASCII 10²: 百位的ASCII碼

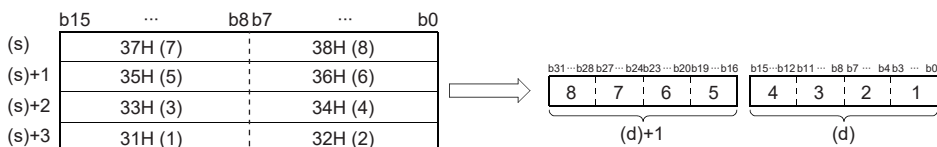
ASCII 10¹: 十位的ASCII碼

ASCII 10⁰: 個位的ASCII碼

- (s)~(s)+3中指定的ASCII資料為0~99999999的範圍內。
- (s)+4及其以後的資料將被忽略。
- 各位中設定的ASCII碼的範圍為30H~39H。
- 各位中設定的ASCII碼為20H或00H時，將作為30H處理。

例

在(s)中指定了87654321的情況下



■SM705 (轉換位數切換) 為0N的情況下的 (s) 的設定方法

(s) 中，設定任意的位數 (00H (NULL代碼) 為止) 的10進制ASCII資料。但是，最大位數 (8位) 的情況下，無需00H (NULL代碼) 的設定。

(s) 的設定方法如下所示。

(s) 中設定的值	(s) ~ (s)+3 的內容	(s) 中設定的值	(s) ~ (s)+3 的內容																																																												
• 0~9	<ul style="list-style-type: none"> (s) 的高位字節中設定00H。 (s)+1 及其以後將被忽略。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td colspan="2">00H</td> <td>:</td> <td colspan="2">ASCII 10⁰</td> </tr> <tr> <td>(s)+1</td> <td colspan="5">[Greyed out]</td> </tr> <tr> <td>(s)+2</td> <td colspan="5">[Greyed out]</td> </tr> <tr> <td>(s)+3</td> <td colspan="5">[Greyed out]</td> </tr> </table>		b15	...	b8 b7	...	b0	(s)	00H		:	ASCII 10 ⁰		(s)+1	[Greyed out]					(s)+2	[Greyed out]					(s)+3	[Greyed out]					• 10~99	<ul style="list-style-type: none"> (s)+1 的低位字節中設定00H。 (s)+1 的高位字節及其以後將被忽略。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁰</td> <td>:</td> <td colspan="2">ASCII 10¹</td> </tr> <tr> <td>(s)+1</td> <td colspan="2">[Greyed out]</td> <td>:</td> <td colspan="2">00H</td> </tr> <tr> <td>(s)+2</td> <td colspan="5">[Greyed out]</td> </tr> <tr> <td>(s)+3</td> <td colspan="5">[Greyed out]</td> </tr> </table>		b15	...	b8 b7	...	b0	(s)	ASCII 10 ⁰		:	ASCII 10 ¹		(s)+1	[Greyed out]		:	00H		(s)+2	[Greyed out]					(s)+3	[Greyed out]				
	b15	...	b8 b7	...	b0																																																										
(s)	00H		:	ASCII 10 ⁰																																																											
(s)+1	[Greyed out]																																																														
(s)+2	[Greyed out]																																																														
(s)+3	[Greyed out]																																																														
	b15	...	b8 b7	...	b0																																																										
(s)	ASCII 10 ⁰		:	ASCII 10 ¹																																																											
(s)+1	[Greyed out]		:	00H																																																											
(s)+2	[Greyed out]																																																														
(s)+3	[Greyed out]																																																														
⋮																																																															
• 1000000 ~ 9999999	<ul style="list-style-type: none"> (s)+3 的高位字節中設定00H。 (s)+4 及其以後將被忽略。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁵</td> <td>:</td> <td colspan="2">ASCII 10⁶</td> </tr> <tr> <td>(s)+1</td> <td colspan="2">ASCII 10³</td> <td>:</td> <td colspan="2">ASCII 10⁴</td> </tr> <tr> <td>(s)+2</td> <td colspan="2">ASCII 10¹</td> <td>:</td> <td colspan="2">ASCII 10²</td> </tr> <tr> <td>(s)+3</td> <td colspan="2">00H</td> <td>:</td> <td colspan="2">ASCII 10⁰</td> </tr> </table>		b15	...	b8 b7	...	b0	(s)	ASCII 10 ⁵		:	ASCII 10 ⁶		(s)+1	ASCII 10 ³		:	ASCII 10 ⁴		(s)+2	ASCII 10 ¹		:	ASCII 10 ²		(s)+3	00H		:	ASCII 10 ⁰		• 10000000 ~ 99999999	<ul style="list-style-type: none"> (s)+4 及其以後將被忽略。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁶</td> <td>:</td> <td colspan="2">ASCII 10⁷</td> </tr> <tr> <td>(s)+1</td> <td colspan="2">ASCII 10⁴</td> <td>:</td> <td colspan="2">ASCII 10⁵</td> </tr> <tr> <td>(s)+2</td> <td colspan="2">ASCII 10²</td> <td>:</td> <td colspan="2">ASCII 10³</td> </tr> <tr> <td>(s)+3</td> <td colspan="2">ASCII 10⁰</td> <td>:</td> <td colspan="2">ASCII 10¹</td> </tr> </table>		b15	...	b8 b7	...	b0	(s)	ASCII 10 ⁶		:	ASCII 10 ⁷		(s)+1	ASCII 10 ⁴		:	ASCII 10 ⁵		(s)+2	ASCII 10 ²		:	ASCII 10 ³		(s)+3	ASCII 10 ⁰		:	ASCII 10 ¹	
	b15	...	b8 b7	...	b0																																																										
(s)	ASCII 10 ⁵		:	ASCII 10 ⁶																																																											
(s)+1	ASCII 10 ³		:	ASCII 10 ⁴																																																											
(s)+2	ASCII 10 ¹		:	ASCII 10 ²																																																											
(s)+3	00H		:	ASCII 10 ⁰																																																											
	b15	...	b8 b7	...	b0																																																										
(s)	ASCII 10 ⁶		:	ASCII 10 ⁷																																																											
(s)+1	ASCII 10 ⁴		:	ASCII 10 ⁵																																																											
(s)+2	ASCII 10 ²		:	ASCII 10 ³																																																											
(s)+3	ASCII 10 ⁰		:	ASCII 10 ¹																																																											

ASCII 10⁷: 千萬位的ASCII碼

ASCII 10⁶: 百萬位的ASCII碼

ASCII 10⁵: 十萬位的ASCII碼

ASCII 10⁴: 萬位的ASCII碼

ASCII 10³: 千位的ASCII碼

ASCII 10²: 百位的ASCII碼

ASCII 10¹: 十位的ASCII碼

ASCII 10⁰: 個位的ASCII碼

- (s) ~ (s)+3 中指定的ASCII資料為0~99999999的範圍內。
- (s)+4 及其以後的資料將被忽略。
- 各位中設定的ASCII碼的範圍為30H~39H。
- 各位中設定的ASCII碼為20H時，將作為30H處理。00H時，作為10進制ASCII資料的終端進行處理。

出錯

出錯代碼 (SD0)	內容
3401H	當 (s) 中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none"> • 資料中有0~9以外的字元。

10進制字元串→BIN16位元資料轉換

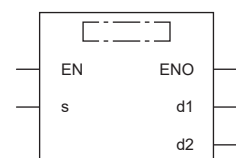
VAL(P) (_U)



將字元串轉換為BIN16位元資料。

梯形圖	ST	
	ENO:=VAL(EN, s, d1, d2); ENO:=VALP(EN, s, d1, d2);	ENO:=VAL_U(EN, s, d1, d2); ENO:=VALP_U(EN, s, d1, d2);

FBD/LD



■執行條件

指令	執行條件
VAL VAL_U	
VALP VALP_U	

設置數據

■內容、範圍、資料類型

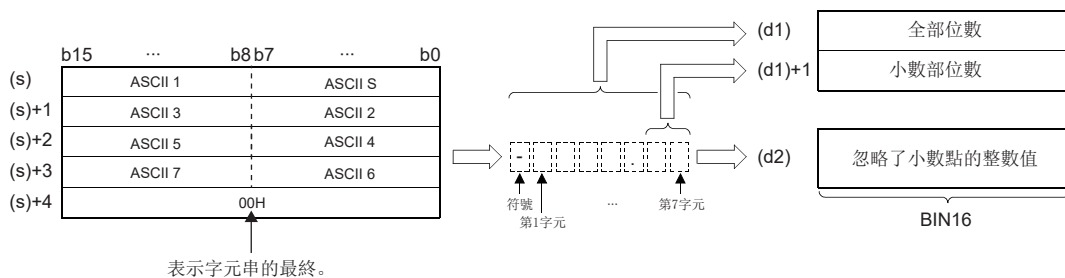
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	轉換為BIN資料的字元串或儲存了字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d1)	VAL(P)	—	帶符號BIN16位元	ANY16_S_ARRAY (要素數: 2)
	VAL(P)_U		無符號BIN16位元	ANY16_U_ARRAY (要素數: 2)
(d2)	VAL(P)	—	帶符號BIN16位元	ANY16_S
	VAL(P)_U		無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s)	—	—	○	—	—	—	○	—	—	○	—	
(d1)	○	—	○	—	—	—	○	—	—	—	—	
(d2)	○	○	○	○	○	—	○	—	—	—	—	

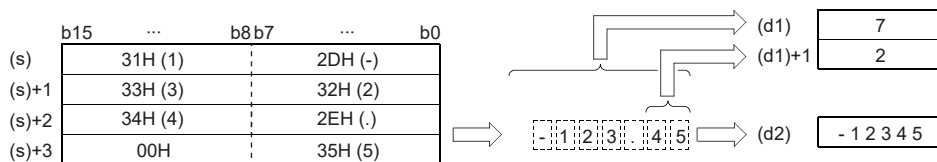
功能

- 將 (s) 中指定的元件編號及其以後中儲存的字元串轉換為BIN16位元資料後，將位數儲存在 (d1) 中、將BIN資料儲存在 (d2) 中。在字元串→BIN轉換中，將 (s) 中指定的元件編號開始至儲存了00H的元件編號為止的資料作為字元串處理。



例

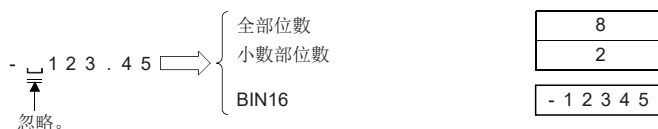
在 (s) 及其以後指定了 “-123.45” 的字元串的情況下 (指定了帶符號的情況下)



- (s) 中指定的字元串的全部字元數為2~8字元。
- (s) 中指定的字元串中，小數部分的字元數為0~5字元。但是，指定時不應超過“全部位數-3”。
- 對於可轉換為BIN值的數值的字元串的範圍，在忽略小數點的值指定帶符號的情況下其範圍為-32768~32767、指定無符號的情況下為0~65535。此外，對於符號及小數點除外的數值的字元串，只能在30H~39H的範圍內指定。(忽略小數點的值...“-12345.6”的情況下，將變為“-123456”。)
- 表示正數值的情況下在符號中設定20H、表示負數值的情況下設定2DH。
- 在小數點中設定2EH。
- (d1) 中儲存的全部位數儲存表示數值的所有字元 (包括符號、小數點) 的數。(d1)+1 中儲存的小數部位數儲存表示2EH(.) 及其以後的小數部分的字元數。對於 (d2) 中儲存的BIN16位元資料，將忽略小數點的字元串轉換為BIN值後儲存。
- 在 (s) 中指定的字元串中，在與符號、首個0以外的數值之間，存在有20H (空格) 或30H (0) 的情況下，將忽略20H、30H而轉換為BIN值。

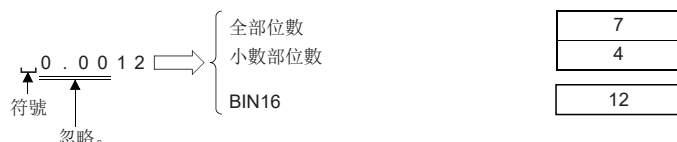
例

在與符號、首個0以外的數值之間存在有20H的情況下 (指定了帶符號的情況下)



例

在與符號、首個0以外的數值之間存在有30H的情況下

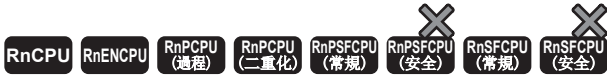


出錯

出錯代碼 (SD0)	內容
2820H	在 (s) 中指定的元件編號開始，至相應元件的最終元件編號為止之間，未設定00H時。
3401H	當 (s) 中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none">• 字元數為2~8以外。• 小數部字元數為0~5以外。• 全部字元數與小數部字元數的關係超出了下述範圍。 全部字元數-3≥小數部字元數• 使用VAL(P)指令時，符號中設定了20H、2DH以外的ASCII碼。• 使用VAL(P)_U指令時，符號中設定了20H以外的ASCII碼。• 各數位的位數中設定了30H~39H及2EH(小數點)以外的ASCII碼。• 設定了多個小數點。 當轉換的BIN值的值超過了各指令中可轉換的範圍時。
3405H	當 (s) 中指定的字元串的字元數超過了16383字元時。

10進制字元串→BIN32位元資料轉換

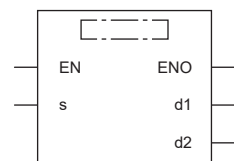
DVAL (P) (_U)



字元串轉換為BIN32位元資料。

梯形圖	ST	
	ENO:=DVAL (EN, s, d1, d2) ; ENO:=DVALP (EN, s, d1, d2) ;	ENO:=DVAL_U (EN, s, d1, d2) ; ENO:=DVALP_U (EN, s, d1, d2) ;

FBD/LD



■執行條件

指令	執行條件
DVAL DVAL_U	
DVALP DVALP_U	

設置數據

■內容、範圍、資料類型

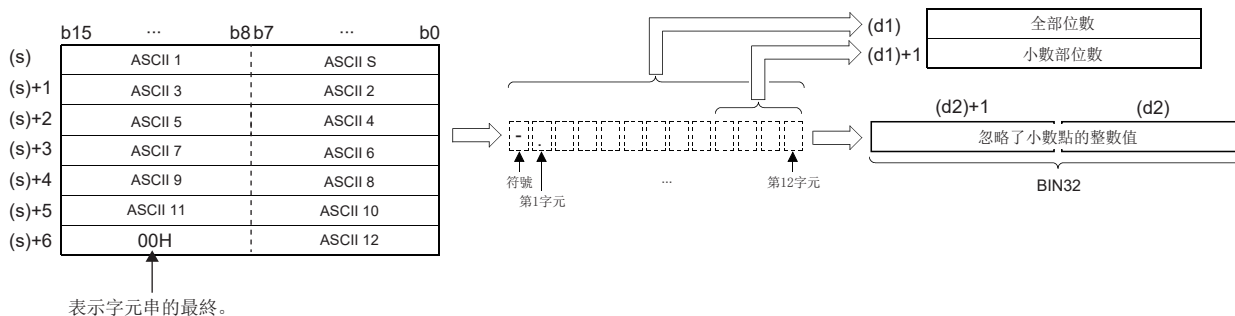
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	轉換為BIN資料的字元串或儲存了字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d1)	DVAL (P)	—	帶符號BIN16位元	ANY16_S_ARRAY (要素數: 2)
	DVAL (P)_U		無符號BIN16位元	ANY16_U_ARRAY (要素數: 2)
(d2)	DVAL (P)	—	帶符號BIN32位元	ANY32_S
	DVAL (P)_U		無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	—	○	—	—	○	—
(d1)	○	—	○	—	—	—	○	—	—	—	—
(d2)	○	○	○	○	○	○	○	—	—	—	—

功能

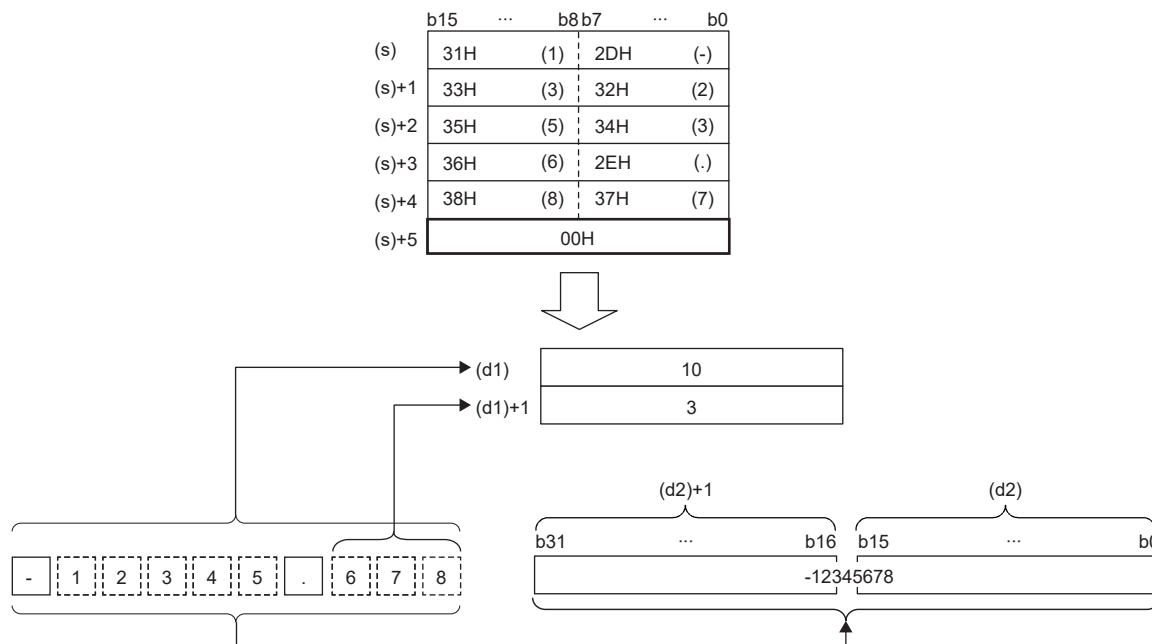
- 將 (s) 中指定的元件編號及其以後中儲存的字元串轉換為BIN32位元資料後，將位數儲存在 (d1) 中、將BIN資料儲存在 (d2) 中。在字元串→BIN轉換中，將 (s) 中指定的元件編號開始至儲存了00H的元件編號為止的資料作為字元串處理。



ASCII S: 符號的ASCII碼
ASCII □: 第 □ 字元的ASCII碼

例

在 (s) 及其以後指定了 “-12345.678” 的字元串的情況下 (指定了帶符號的情況下)

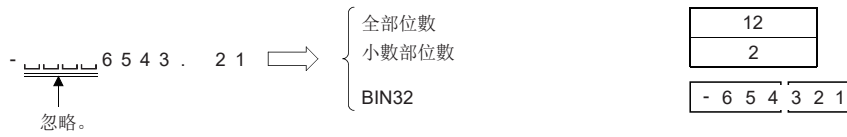


- (s) 中指定的字元串的全部字元數為2~13字元。
- (s) 中指定的字元串中，小數部分的字元數為0~10字元。但是，不應超過“全部位數-3”。
- 可轉換為BIN值的數值的字元串的範圍如下所示。此外，對於符號及小數點除外的數值的字元串，只能在30H~39H的範圍內指定。(忽略小數點的值...“-12345.6”的情況下，將變為“-123456”。)
 - 在忽略小數點的值指定帶符號的情況下：-2147483648~2147483647
 - 在忽略小數點的值指定無符號的情況下：0~4294967295
- 表示正數值的情況下在符號中設定20H、表示負數值的情況下設定2DH。
- 在小數點中設定2EH。
- (d1) 中儲存的全部位數儲存表示數值的所有字元 (包括符號、小數點) 的數。(d1)+1 中儲存的小數部位數儲存表示2EH(.) 及其以後的小數部分的字元數。對於 (d2) 中儲存的BIN32位元資料，將忽略了小數點的字元串轉換為BIN值後儲存。

- 在(s)中指定的字元串中，在與符號、首個0以外的數值之間，存在有20H(空格)或30H(0)的情況下，將忽略20H、30H而轉換為BIN值。

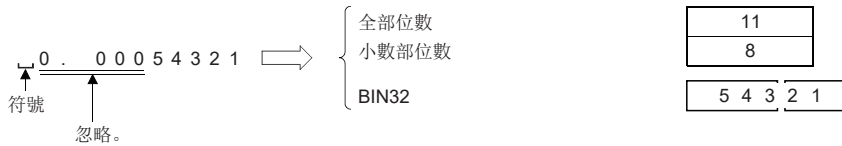
例

在與符號、首個0以外的數值之間存在有20H的情況下(指定了帶符號的情況下)



例

在與符號、首個“0”以外的數值之間存在有30H的情況下

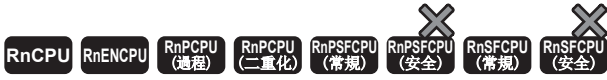


出錯

出錯代碼(SD0)	內容
2820H	在(s)中指定的元件編號開始，至相應元件的最終元件編號為止之間，未設定00H時。
3401H	<p>當(s)中輸入了不能轉換的不正確資料時。</p> <ul style="list-style-type: none"> 字元串的字元數為2~13以外。 字元串的小數部字元數為0~10以外。 全部字元數與小數部字元數的關係超出了下述範圍。 <p>全部字元數-3≥小數部字元數</p> <ul style="list-style-type: none"> 使用DVAL(P)指令時，符號中設定了20H、2DH以外的ASCII碼。 使用DVAL(P)_U指令時，符號中設定了20H以外的ASCII碼。 各數位的位數中設定了30H~39H及2EH(小數點)以外的ASCII碼。 設定了多個小數點。 <p>當轉換的BIN值的值超過了各指令中可轉換的範圍時。</p>
3405H	當(s)中指定的字元串的字元數超過了16383字元時。

16進制ASCII→16進制資料BIN轉換

ASC2INT (P)



將16進制ASCII資料轉換為BIN值。

梯形圖	ST
	ENO:=ASC2INT (EN, s, n, d) ; ENO:=ASC2INTP (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
ASC2INT	
ASC2INTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	儲存了轉換為BIN資料的字元串的起始元件	—	字元串	ANYSTRING_SINGLE*1
(d)	儲存轉換後的BIN資料的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	儲存的字元數	0~16383	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

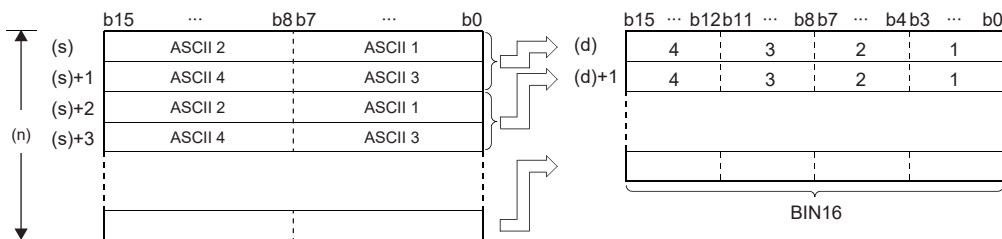
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 將 (s) 中指定的元件編號及其以後，(n) 中指定的字元數中儲存的16進制ASCII資料轉換為BIN值後，儲存到 (d) 中指定的元件編號及其以後。

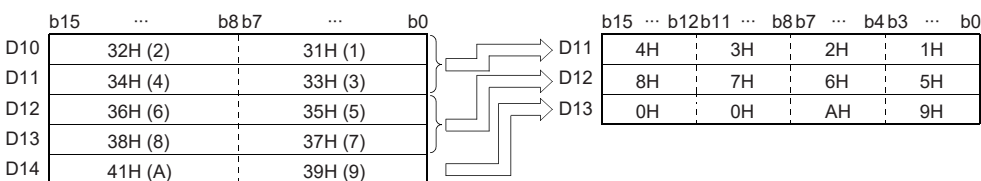


ASCII □: 第 □ 位的ASCII碼

1~4: 第1~4位

(n): (n) 中被指定的字元數分

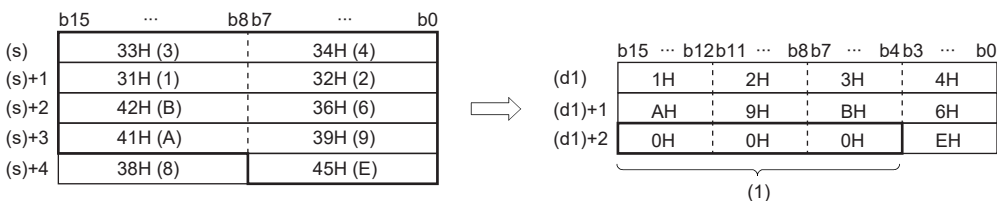
- (n) 中設定字元數時，(s) 中指定的字元串的範圍及 (d) 中指定的儲存BIN資料的元件的範圍將自動確定。
- 即使儲存要轉換的ASCII資料的元件範圍與儲存轉換後的BIN資料的元件範圍重複的情況下，也將作為正常處理。



- (n) 中指定的字元數不是4的倍數的情況下，在儲存轉換後的BIN值的元件編號中，將在最終元件編號的指定字元數以後的位數中自動儲存0H。

例

在 (n) 的字元數為9的情況下



(1): 自動儲存0H。

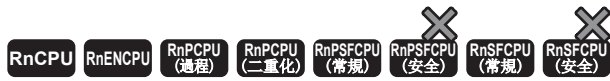
- 在 (n) 中指定的字元數為“0”的情況下，不進行轉換處理。
- (s) 中可指定的ASCII碼為30H~39H、41H~46H的範圍內。

出錯

出錯代碼 (SD0)	內容
3401H	(s) 中設定了16進制數值的字元串以外的字元 (30H~39H、41H~46H以外的ASCII碼)。
3405H	(n) 中輸入了超出可指定範圍的資料時。 • 指定的字元數超出了0~16383的範圍。

單精度實數→BCD分解

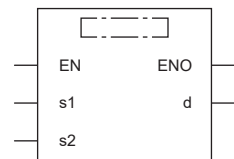
EMOD (P)



將單精度實數資料分解為BCD型浮點格式。

梯形圖	ST
	ENO:=EMOD(EN, s1, s2, d); ENO:=EMODP(EN, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
EMOD	
EMODP	

設置數據

■內容、範圍、資料類型

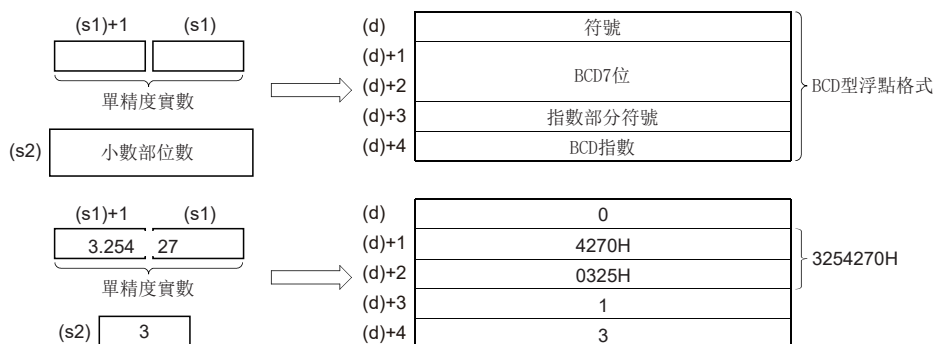
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	單精度實數資料或儲存了單精度實數資料的起始元件	$0, 2^{-126} \leq (s1) < 2^{128}$	單精度實數	ANYREAL_32
(s2)	小數位資料	0~7	帶符號BIN16位元	ANY16
(d)	儲存BCD分解後的資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 5)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	○	○	○	○	○	—	○	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 基於(s2)中指定的元件內所儲存的小數位數，將(s1)中指定的元件中儲存的單精度實數資料分解為BCD型浮點格式，儲存在(d)中指定的元件編號及其以後。

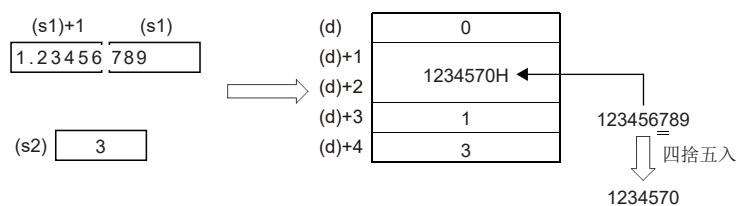


- (d)的符號及(d)+3的指數部符號中，為正時儲存0、為負時儲存1。
- (d)+4的BCD指數中儲存0~38。
- 對於(s2)指定(s1)的單精度實數資料的小數位。在上圖的示例中，變為以下情況。

3.25427

 (s2)=3

- (d)+1、(d)+2中儲存的BCD的有效位為將第7位進行了四捨五入後的6位。



- (s2)的小數位中可設定0~7。
- 在透過工程工具設定輸入值的情況下，有可能會發生化整誤差。關於透過工程工具設定輸入值時的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3401H	(s2)中指定的小數位數超出了0~7的範圍時。
3402H	(s1)中設定的元件、標籤的內容為-0、非正規化數、非數、±∞時。

BIN16位元資料2的補數(符號反轉)

NEG(P)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對BIN16位元元件的符號進行反轉。

梯形圖	ST
	ENO:=NEG(EN, d); ENO:=NEGP(EN, d);

FBD/LD



■執行條件

指令	執行條件
NEG	
NEGP	

設置數據

■內容、範圍、資料類型

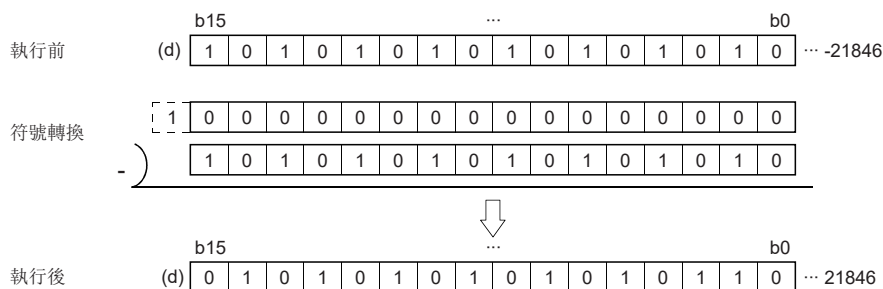
操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存進行2的補數的資料的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 對(d)中指定的BIN16位元元件的符號進行反轉後，儲存到(d)中指定的元件中。
- 用於對正負的符號進行反轉。



出錯

沒有運算出錯。

BIN32位元資料2的補數(符號反轉)

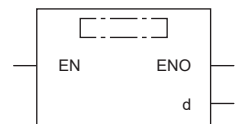
DNEG (P)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對BIN32位元元件的符號進行反轉。

梯形圖	ST
	ENO:=DNEG (EN, d) ; ENO:=DNEGP (EN, d) ;

FBD/LD



■執行條件

指令	執行條件
DNEG	
DNEGP	

設置數據

■內容、範圍、資料類型

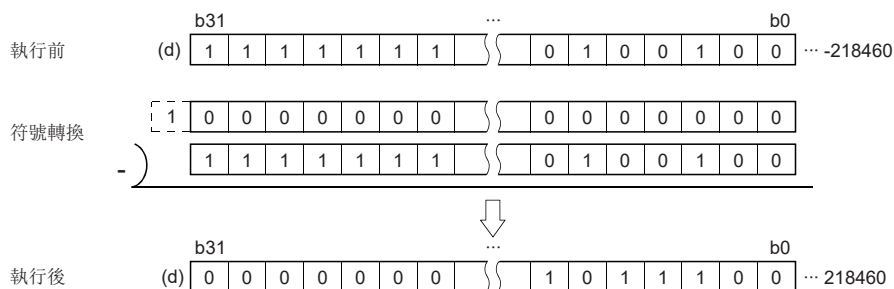
操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存進行2的補數的資料的起始元件	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(d)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 對(d)中指定的BIN32位元元件的符號進行反轉後，儲存到(d)中指定的元件中。
- 用於對正負的符號進行反轉。



出錯

沒有運算出錯。

8→256位元解碼

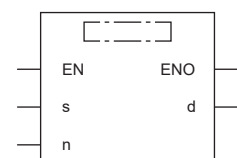
DECO (P)



對指定的元件的低位 (n) 位元進行解碼。

梯形圖	ST
	ENO:=DECO (EN, s, n, d) ; ENO:=DECOP (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
DECO	
DECOP	

設置數據

■內容、範圍、資料類型

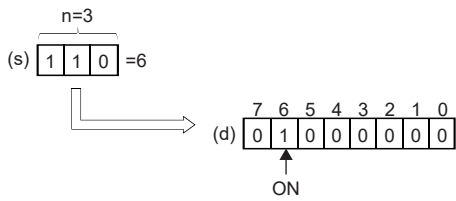
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	解碼資料或儲存了解碼資料的元件編號	—	位元/帶符號BIN16位元	ANY_ELEMENTARY
(d)	儲存解碼結果的元件	—	位元/字元	ANY_ELEMENTARY
(n)	有效位元長	1~8	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 將(s)的低位(n)位元中指定的BIN值對應的(d)的位元的位置設為ON。



- (n)可指定1~8。
- 當(n)=0時將變為無處理，(d)中指定的元件的內容不變化。
- 位元件作為1位元處理，字元件作為16位元處理。

出錯

出錯代碼 (SD0)	內容
3401H	(n) 為0~8以外時。

256→8位編碼

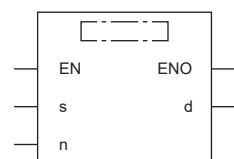
ENCO (P)



對2的(n)次方位的資料進行編碼。

梯形圖	ST
	ENO:=ENCO (EN, s, n, d) ; ENO:=ENCOP (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
ENCO	
ENCOP	

設置數據

■內容、範圍、資料類型

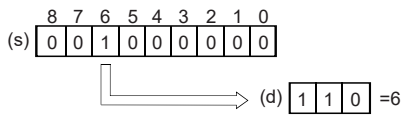
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存編碼資料的元件	—	位元/字元	ANY_ELEMENTARY
(d)	儲存編碼結果的元件編號	—	位元/帶符號BIN16位元	ANY_ELEMENTARY
(n)	有效位元長	1~8	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	—	○	—	—	—	—	○	—	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 將從(s)的 $2^{(n)}$ 位的資料變為1的位元所對應的BIN值儲存到(d)中。



- (n)可指定1~8。
- 當(n)=0時將變為無處理，(d)中指定的元件的內容不變化。
- 位元件作為1位元處理，字元件作為16位元處理。
- 多個位為1時以高位的位元的位置進行處理。

出錯

出錯代碼(SD0)	內容
3401H	(n)為0~8以外時。
	從(s)開始的 $2^{(n)}$ 位的資料全部為0時。

7段解碼

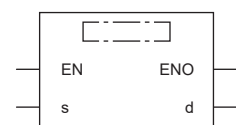
SEG (P)



將元件的低位4位元中指定的0~F的資料解碼為7段顯示資料。

梯形圖	ST
	ENO:=SEG (EN, s, d) ; ENO:=SEGP (EN, s, d) ;

FBD/LD



■執行條件

指令	執行條件
SEG	
SEGP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	解碼資料或儲存了解碼資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存解碼結果的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

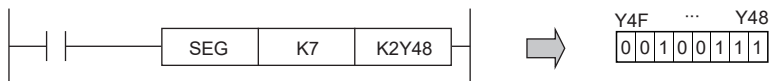
操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 將(s)的低位4位元中指定的0~F的資料解碼為7段顯示資料後，儲存到(d)中。
- 位元元件時(d)表示儲存7段顯示資料的起始元件，字元元件中儲存的元件編號如下所示。

例

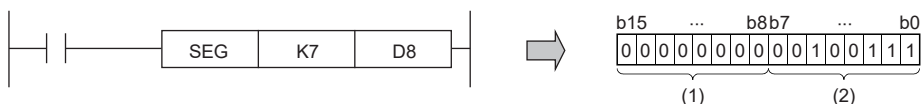
在位元元件的情況下



Y48~Y4F在輸出下一個資料之前不變化。

例

在字元元件的情況下



- (1): 高位8位元一定變為0。
- (2): 向低位8位元儲存7段顯示資料。

- 7段解碼表如下所示。

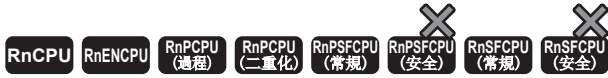
(s)		7段的構成	(d)								顯示資料
16進制數	位元模式		b7	b6	b5	b4	b3	b2	b1	b0	
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		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

出錯

沒有運算出錯。

BIN16位元資料的4位元分離

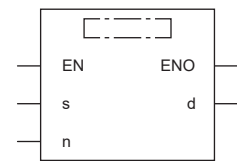
DIS (P)



將指定的BIN16位元資料的低位 (n) 位的資料儲存到指定的元件中。

梯形圖	ST
	ENO:=DIS (EN, s, n, d) ; ENO:=DISP (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
DIS	
DISP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	儲存分離的資料的元件	—	帶符號BIN16位元	ANY16
(d)	儲存分離後資料的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	分離數	1~4	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

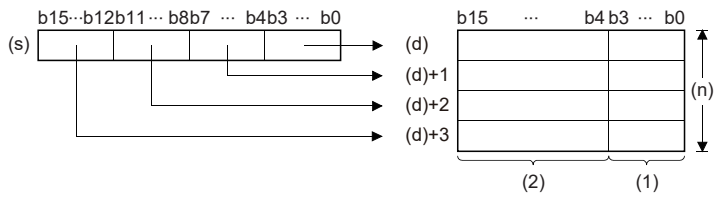
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 將 (s) 中指定的 BIN16 位元資料的低位 (n) 位 (1 位數 4 位元) 的資料儲存到 (d) 中指定的元件開始的 (n) 點的低位 4 位元中。



- (1) 儲存區
(2) 0 被儲存。

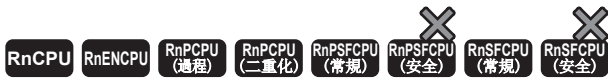
- (d) 中指定的元件開始的 (n) 點的高位 12 位元將變為 0。
- (n) 可指定 1~4。
- 當 (n)=0 時將變為無處理，(d) 的元件開始的 (n) 點的內容不變化。

出錯

出錯代碼 (SD0)	內容
3401H	(n) 為 0~4 以外時。

BIN16位元資料的4位合併

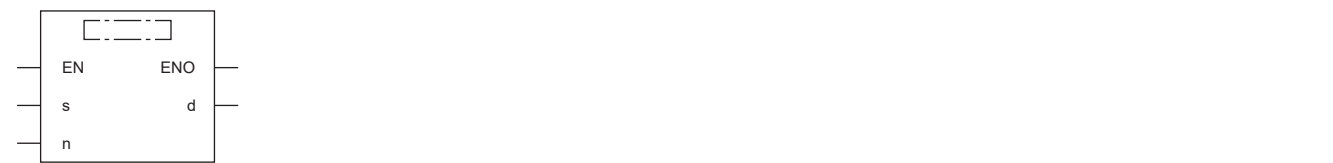
UNI (P)



將指定的元件開始的 (n) 點的BIN16位元資料的低位4位元合併到16位元元件中。

梯形圖	ST
	ENO:=UNI (EN, s, n, d) ; ENO:=UNIP (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
UNI	
UNIP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	儲存合併的資料的起始元件	—	帶符號BIN16位元	ANY16*1
(d)	儲存合併後的資料的元件	—	帶符號BIN16位元	ANY16
(n)	合併數	1~4	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

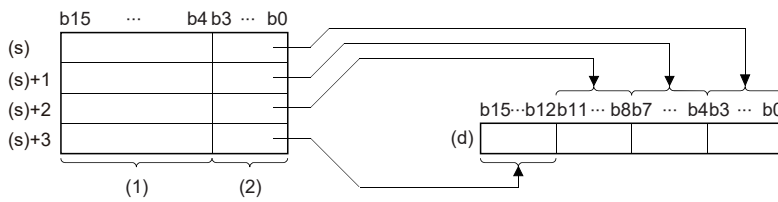
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 將(s)中指定的元件開始的(n)點的BIN16位元資料的低位4位元，合併到(d)中指定的BIN16位元元件中。



(1): 忽略。

(2): 合併資料

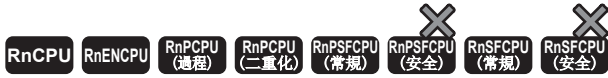
- (d)中指定的元件的高位(4-n)的位的位元將變為0。
- (n)可指定1~4。
- 當(n)=0時將變為無處理，(d)的元件的內容不變化。

出錯

出錯代碼(SD0)	內容
3401H	(n)為0~4以外時。

任意資料的位分離

NDIS (P)



將指定資料的各位逐個分離為任意的位。

梯形圖	ST
	ENO:=NDIS (EN, s1, s2, d) ; ENO:=NDISP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
NDIS	
NDISP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	儲存分離的資料的起始元件	—	帶符號BIN16位元	ANY16*1
(d)	儲存分離後的資料的起始元件	—	帶符號BIN16位元	ANY16*1
(s2)	儲存分離單位的起始元件	—	帶符號BIN16位元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

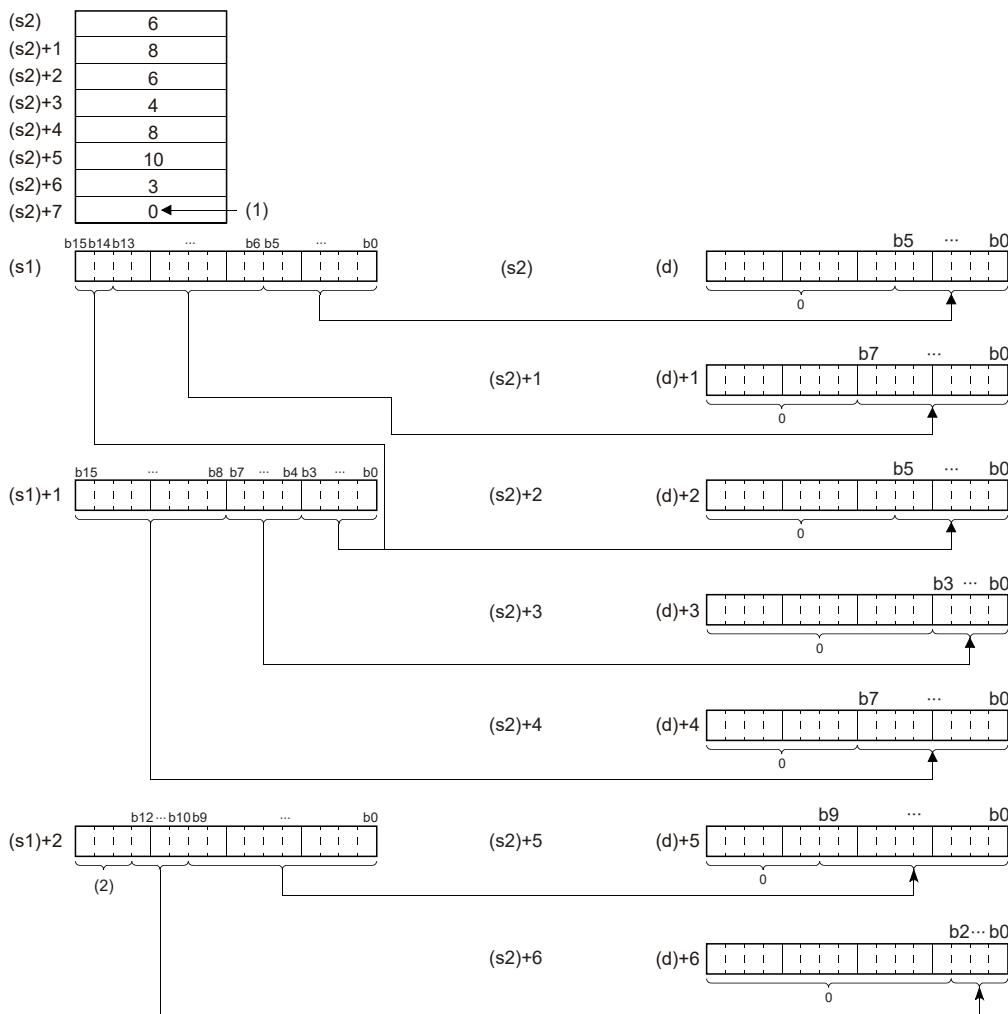
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—

功能

- 將(s1)中指定的元件編號及其以後儲存的資料的各位元按(s2)中指定的位元分離後，儲存到(d)中指定的元件編號及其以後。



(s2) ~ (s2)+6: (s2) ~ (s2)+6的指定位元分

(1): 設定結束指定

(2): 忽略。

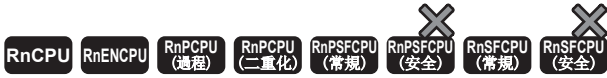
- (s2)中指定的分離位數可在1~16位元的範圍內指定。
- 將從透過(s2)指定的元件編號開始至儲存了“0”的元件編號為止作為分離位元數處理。
- 應避免分離的資料的元件範圍((s1)~(s1)的結束範圍)與儲存分離後的資料的元件範圍((d)~(d)的結束範圍)重複。在重複的情況下，可能無法獲得正確的運算結果。
- (s1)、(s2)、(d)中指定的元件編號應避免重複。

出錯

出錯代碼 (SD0)	內容
2821H	當(s1)、(s2)的元件重複時。
	當(s1)、(d)的元件重複時。
	當(s2)、(d)的元件重複時。
3401H	當(s2)中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none"> 指定的分離位數指定超出了1~16位元的設定範圍。 從指定的元件開始，至超出元件/標籤記憶體的各設定區域為止之間，未設定0。

任意資料的位元合併

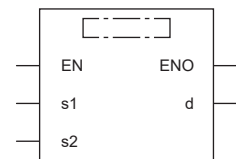
NUNI (P)



將指定資料的各位元逐個合併為任意的位元。

梯形圖	ST
	ENO:=NUNI (EN, s1, s2, d) ; ENO:=NUNIP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
NUNI	
NUNIP	

■設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	儲存合併的資料的起始元件	—	帶符號BIN16位元	ANY16*1
(d)	儲存合併後的資料的起始元件	—	帶符號BIN16位元	ANY16*1
(s2)	儲存合併單位的起始元件	—	帶符號BIN16位元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

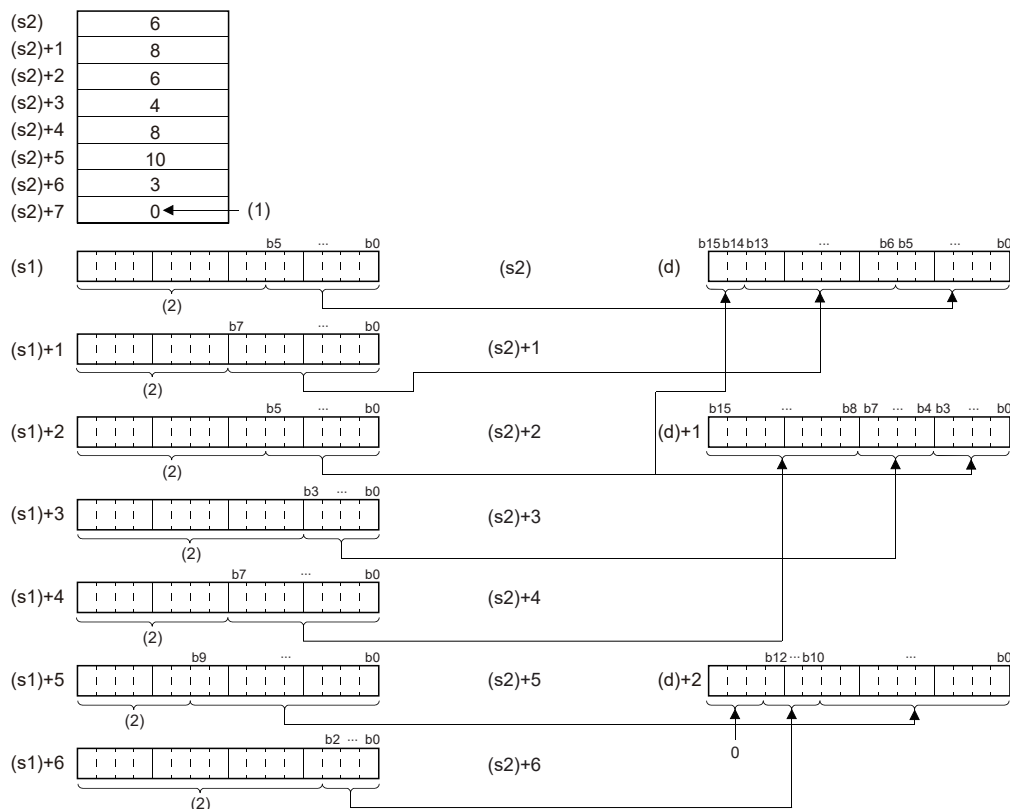
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—

功能

- 將(s1)中指定的元件編號及其以後中儲存的資料的各位元按(s2)中指定的位元合併後，儲存到(d)中指定的元件編號及其以後。



(s2) ~ (s2)+6: (s2) ~ (s2)+6的指定位元分

(1): 設定結束指定

(2): 忽略。

- (s2)中指定的合併位元數可以在1~16位元的範圍內指定。
- 將從透過(s2)指定的元件編號開始至儲存了“0”的元件編號為止作為合併位元數處理。
- 合併的資料的元件範圍((s1)~(s1)的結束範圍)與儲存合併後的資料的元件範圍((d)~(d)的結束範圍)應避免重複。在重複的情況下，可能無法獲得正確的運算結果。
- (s1)、(s2)、(d)中指定的元件編號應避免重複。

出錯

出錯代碼(SD0)	內容
2821H	當(s1)、(s2)的元件重複時。
	當(s1)、(d)的元件重複時。
	當(s2)、(d)的元件重複時。
3401H	當(s2)中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none"> 指定的合併位元數指定超出了1~16位元的設定範圍。 從指定的元件開始，至超出元件/標籤記憶體各設定區域為止之間，未設定0。

字節單位資料分離

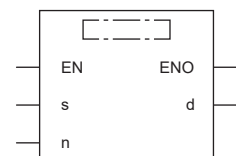
WTOB (P)



將BIN16位元資料分離為 (n) 字節。

梯形圖	ST
	ENO:=WTOB (EN, s, n, d) ; ENO:=WTOBP (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
WTOB	
WTOBP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	儲存以字節單位分離的資料的起始元件	—	帶符號BIN16位元	ANY16*1
(d)	儲存以字節單位分離的結果的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	分離的字節資料的個數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

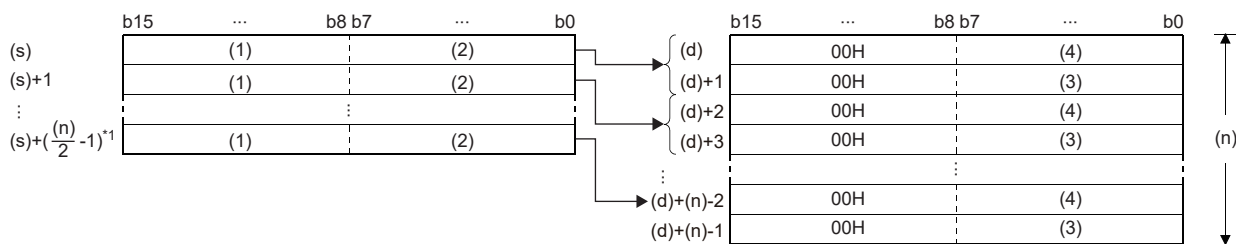
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 將(s)中指定的元件編號及其以後中儲存的BIN16位元資料分離為(n)字節後，儲存到(d)中指定的元件編號以後。

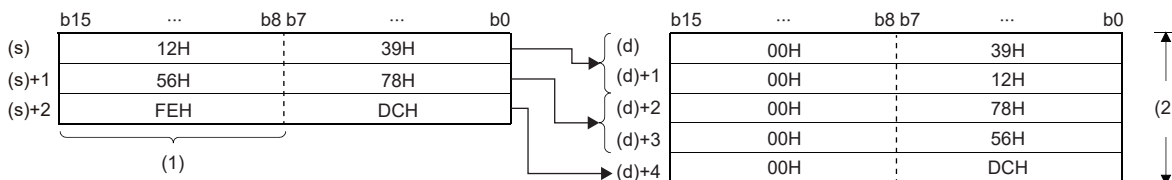


- (1): 高位字節
- (2): 低位字節
- (3): 高位字節的資料
- (4): 低位字節的資料

*1 小數點以下進位。

例

在(n)=5的情況下，將(s)~(s)+2的低位8位元為止的資料儲存到(d)~(d)+4中。

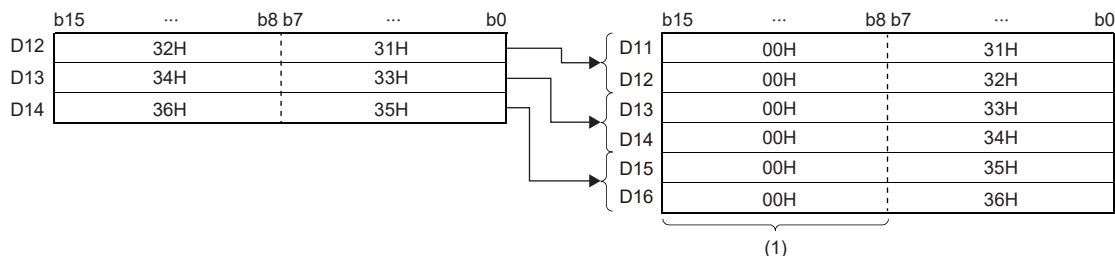


- (1): (n)=5時，忽略。
- (2): (n)=5時

- 透過在(n)中設定字節數，(s)中指定的BIN16位元資料的範圍及儲存(d)中指定的字節資料的元件的範圍將被自動確定。
- 在(n)中指定的字節數為0的情況下，不進行處理。
- (d)中指定的字節資料儲存元件的高位8位元中將被自動儲存00H。

例

在將D12~D14儲存到D11~D16的低位8位元中的情況下



- (1): 自動儲存00H。

- 即使儲存了分離的資料的元件範圍與儲存分離後資料的元件範圍重複的情況下，也將作為正常處理。

儲存了分離的資料的元件範圍	儲存分離後資料的元件範圍
(s)~(s)+(n/2-1)	(d)+0~(d)+(n)-1

出錯

沒有運算出錯。

字節單位資料合併

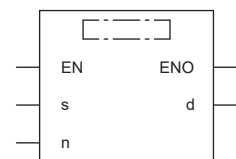
BTOW (P)



將BIN16位元資料的低位8位元合併為字元單位。

梯形圖	ST
	ENO:=BTOW (EN, s, n, d) ; ENO:=BTOWP (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
BTOW	
BTOWP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	儲存了以字節單位合併的資料的起始元件	—	帶符號BIN16位元	ANY16*1
(d)	儲存以字節單位合併的結果的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	合併的字節資料的個數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

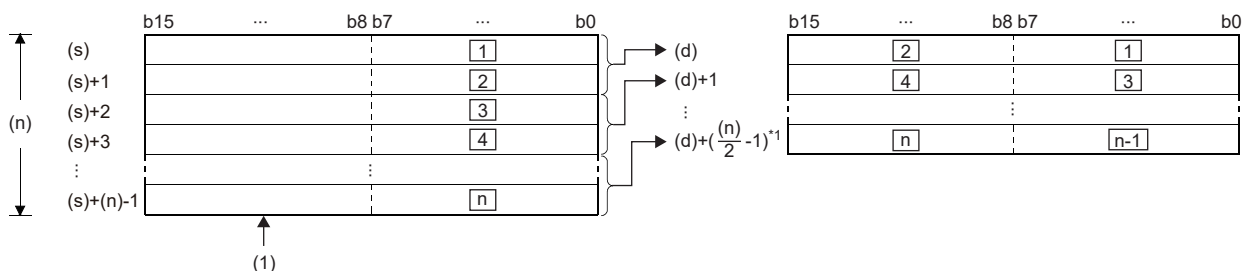
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 將 (s) 中指定的元件編號及其以後中儲存的 (n) 字的BIN16位元資料的低位8位元合併為字單位後，儲存到 (d) 中指定的元件編號以後。
- (s) 中指定的元件編號及其以後中儲存的 (n) 字的資料的高位8位元將被忽略。此外，在 (n) 為奇數的情況下，儲存了第 (n) 字節的資料的元件的高位8位元中將儲存0。



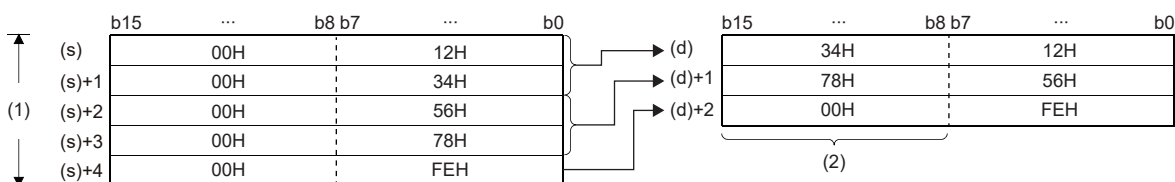
□: 第□字節的資料

(1) 忽略高位字節。

*1 小數點以下進位。

例

在 (n)=5 的情況下，將 (s) ~ (s)+4 的低位8位元為止的資料合併後儲存到 (d) ~ (d)+2 中。



(1) (n)=5時

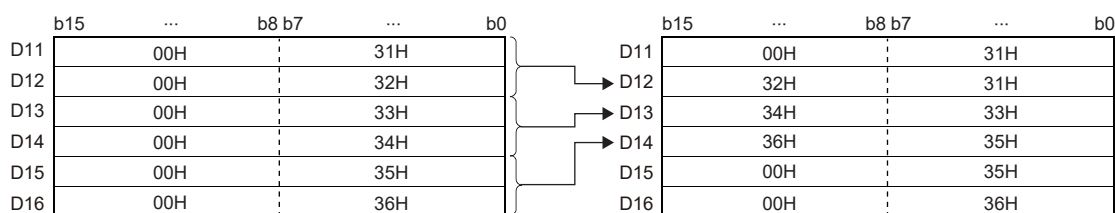
(2) 00H被儲存。

- 透過在 (n) 中設定字節數，(s) 中指定的字節資料的範圍及儲存 (d) 中指定的合併資料的元件的範圍將被自動確定。
- 在 (n) 中指定的字節數為0的情況下，不進行處理。
- (s) 中指定的字節資料儲存元件的高位8位元將被忽略、低位8位元將成為對象。
- 即使儲存有合併的資料的元件範圍與儲存合併後的資料的元件範圍重複的情況下，也將被作為正常處理。

儲存有合併的資料的元件範圍	儲存合併後資料的元件範圍
(s)+0 ~ (s)+(n)-1	(d) ~ (d) + $(\frac{n}{2} - 1)$

例

在將D11~D16的低位8位元儲存到D12~D14中的情況下



出錯

沒有運算出錯。

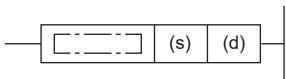
6.7 資料傳送指令

BIN16位元資料傳送

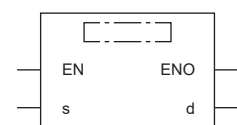
MOV (P)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

傳送指定的元件的BIN16位元資料。

梯形圖	ST
	ENO:=MOV (EN, s, d); ENO:=MOV (EN, s, d);

FBD/LD



■執行條件

指令	執行條件
MOV	
MOV P	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	傳送源資料或儲存了資料的元件編號	-32768~32767	帶符號BIN16位元	ANY16
(d)	傳送目標的元件編號	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

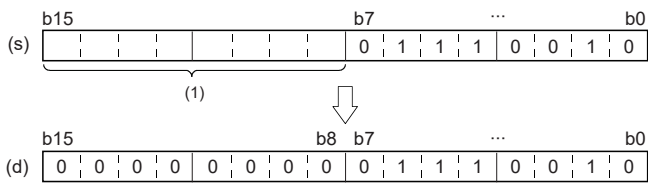
操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 將(s)中指定的BIN16位元資料傳送到(d)中指定的元件。
- 在(s)為進行了位指定的位元元件的情況下，位指定中指定的位元將成為傳送的對象。傳送對象以外的位元中將被作為指定了0而進行傳送。



(1): (s)不足16位元時，將當作指定了0而進行傳送。

出錯

沒有運算出錯。

BIN32位元資料傳送

DMOV (P)

RnCPU RnENCPU RnPCPU (通稱) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

傳送指定的元件的BIN32位元資料。

梯形圖	ST
	ENO:=DMOV (EN, s, d) ; ENO:=DMOVP (EN, s, d)

FBD/LD

■執行條件

指令	執行條件
DMOV	
DMOVP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	傳送源資料或儲存了資料的元件編號	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	傳送目標的元件編號	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

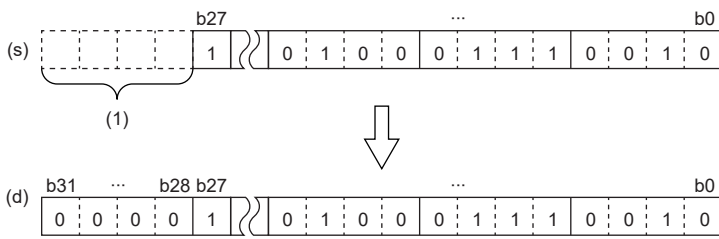
操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 將(s)中指定的BIN32位元資料傳送到(d)中指定的元件。
- 在(s)為進行了位指定的位元元件的情況下，位指定中指定的位元將成為傳送的對象。傳送對象以外的位元中將被作為指定了0而進行傳送。



(1): (s)不足32位元時，傳送指定為0的部分。

出錯

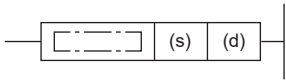
沒有運算出錯。

BIN16位元資料否定傳送

CML (P)

RnCPU RnENCPU RnPCPU (通稱) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)


對指定的BIN16位元資料執行逐位反轉後進行傳送。

梯形圖	ST
	ENO:=CML (EN, s, d); ENO:=CMLP (EN, s, d);

FBD/LD



■執行條件

指令	執行條件
CML	
CMLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	反轉的資料或儲存了資料的元件編號	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存反轉結果的元件編號	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

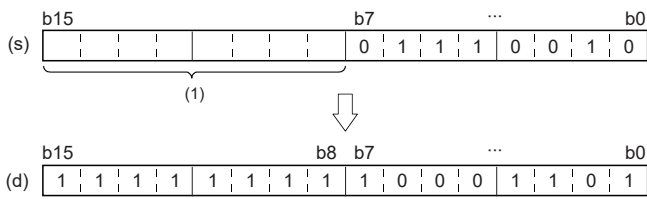
操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 將(s)中指定的BIN16位元資料進行逐位反轉，將其結果傳送到(d)中指定的元件中。
- 在(s)為位指定的位元元件的情況下，位指定中指定的位元將成為對象。在對象以外的位元中，反轉指定為0的部分後進行傳送。



(1): (s)不足16位元時，反轉指定為0的部分。

出錯

沒有運算出錯。

BIN32位元資料否定傳送

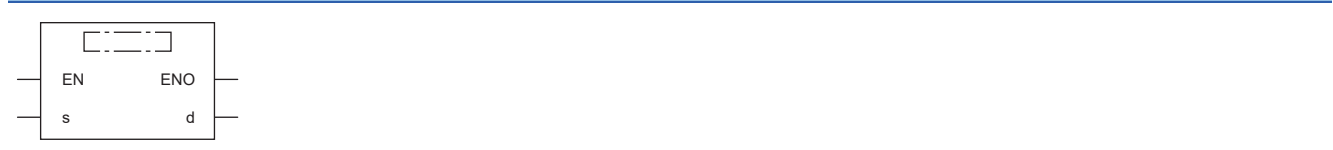
DCML (P)



對指定的BIN32位元資料執行逐位反轉後進行傳送。

梯形圖	ST
	ENO:=DCML (EN, s, d) ; ENO:=DCMLP (EN, s, d) ;

FBD/LD



■執行條件

指令	執行條件
DCML	
DCMLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	反轉的資料或儲存了資料的元件編號	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存反轉結果的元件編號	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

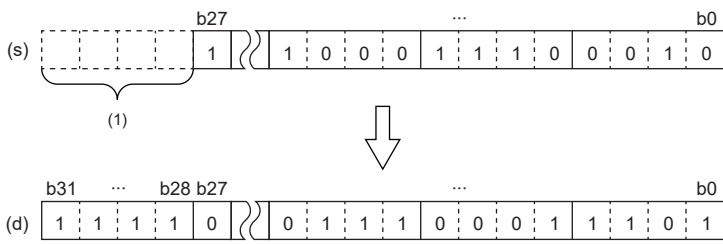
操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	○
(d)	○	○	—

功能

- 對(s)中指定的BIN32位元資料進行逐位反轉後，將其結果傳送到(d)中指定的元件。
- 在(s)為位指定的位元元件的情況下，位指定中指定的位元將成為對象。在對象以外的位元中，反轉指定為0的部分後進行傳送。



(1): (s)不足32位元時，反轉指定為0的部分。

出錯

沒有運算出錯。

位移動

SMOV (P)

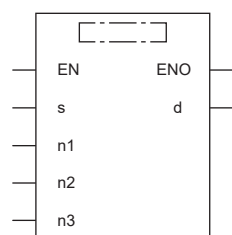


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

以位為單位(4位元)分配/合成資料。

梯形圖	ST
	ENO:=SMOV (EN, s, n1, n2, n3, d) ; ENO:=SMOVP (EN, s, n1, n2, n3, d) ;

FBD/LD



■執行條件

指令	執行條件
SMOV	
SMOVP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存指定位的資料的元件	—	帶符號BIN16位元	ANY16
(n1)*1	儲存了移動的資料的起始元件	1~4	無符號BIN16位元	ANY16_U
(n2)*1	移動的位的個數	1~4	無符號BIN16位元	ANY16_U
(d)	儲存位移動的資料的元件	—	帶符號BIN16位元	ANY16
(n3)*1	移動目標的起始位位置	1~4	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 應設定成 $(n2) \leq (n1)$ 、 $(n2) \leq (n3)$ 。

■可使用的元件

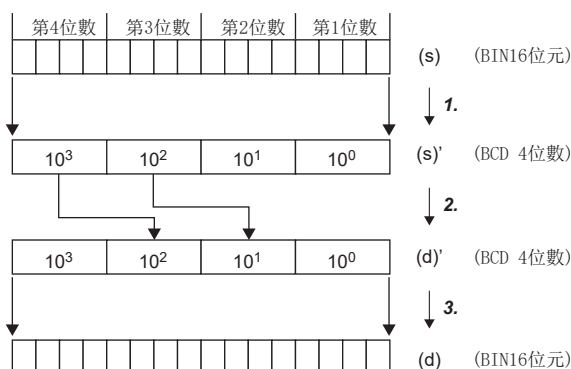
操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○*1	○	○	○	○	—	—	○	—	—	—	—
(n1)	○*1	○	○	○	○	—	—	○	○	—	—	—
(n2)	○*1	○	○	○	○	—	—	○	○	—	—	—
(d)	○*1	○	○	○	○	—	—	○	—	—	—	—
(n3)	○*1	○	○	○	○	—	—	○	○	—	—	—

*1 不能使用FX、FY。

功能

• SM773 (SMOV指令 BCD更改禁止標誌) 為OFF的情況下，將 (s) 與 (d) 的內容轉換 (0000~9999) 為4位的BCD，將第 (n1) 位的低位 (n2) 位以 (d)' 的第 (n3) 位為起始傳送 (合成) 後，轉換為BIN，並儲存至 (d)。

(n1)=4、(n2)=2、(n3)=3時



1. 轉換

將 (s) 由BIN→BCD進行轉換。

2. 位移動

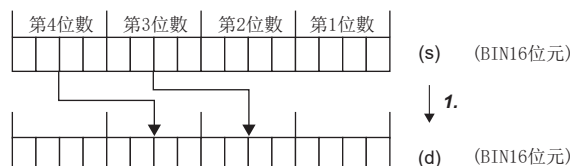
將第 (n1) 位的低位 (n2) 位數傳送 (合成) 至以 (d)' 的第 (n3) 位為起始的 (n2) 位。如左圖所示，從 (s)' 傳送時，(d)' 的第1位、第4位不受影響。

3. 轉換

將合成後的資料 (BCD) 轉換為BIN，儲存至 (d)。

• SM773為ON時，不進行 BIN→BCD轉換，以4位元單位進行位移動。

(n1)=4、(n2)=2、(n3)=3時



1. 位移動

傳送 (合成) 第 (n1)×4 位的低位 (n2)×4 位元分、(d) 的 (n3)×4 位元的地位 (n2)×4 位元分。如左圖所示，從 (s) 傳送時，(d) 的第1位、第4位不受影響。

• 在 (n1) 中使用了位元的位指定時，從起始位元元件進行指定位數×4位元分的位指定。

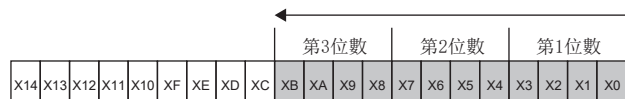
例

從X0開始指定3位×4位元的情況下

```
[ SMOV K3X0 K3 K2 K2X100 K2 ]
```

(1) 指定常數

(2) 起始位元元件



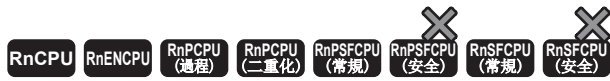
變為從起始位元元件X0指定位數3×4位元=12位元的位指定。

出錯

出錯代碼 (SD0)	內容
3405H	當 (n1)、(n2)、(n3) 中的任意一個在 1~4 以外時。
	當 SM773 為 OFF 時，(s)、(d) 中的任意一個在 0~9999 以外時。
	當 (n2) 大於 (n1) 或 (n3) 時。

1位元資料反轉傳送

CMLB (P)



對指定的位元資料進行反轉後傳送。

梯形圖	ST
	ENO:=CMLB (EN, s, d) ; ENO:=CMLBP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
CMLB	
CMLBP	

■設置數據

■內容、範圍、資料類型

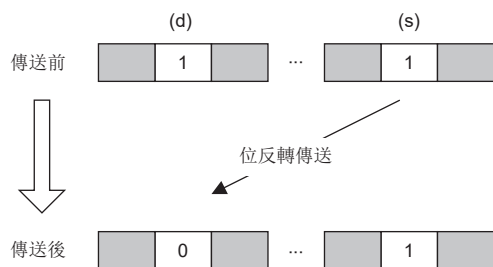
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	反轉的資料或儲存了資料的元件編號	—	位元	ANY_BOOL
(d)	傳送目標的元件編號	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	○	○	—	—	○	—	○	—	—	—	—
(d)	○	○	○	—	—	○	—	○	—	—	—	—

■功能

- 對(s)中指定的位元資料進行反轉，將其結果儲存到(d)中指定的元件。



出錯

沒有運算出錯。

BIN16位元資料塊傳送(16位元)

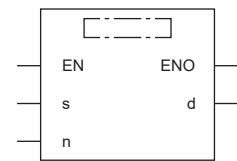
BMOV (P)

RnCPU RnENCPU RnPCPU (通稱) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將從指定的元件開始的(n)點(0~65535)的BIN16位元資料進行批量傳送。

梯形圖	ST
	ENO:=BMOV(EN, s, n, d); ENO:=BMOV(EN, s, n, d);

FBD/LD



■執行條件

指令	執行條件
BMOV	
BMOV P	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存了傳送資料的起始元件	—	帶符號BIN16位元	ANY16*1*2
(d)	傳送目標的起始元件	—	帶符號BIN16位元	ANY16*1*2
(n)	傳送數	0~65535	無符號BIN16位元	ANY16*2
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

*2 在版本為“1.030G”以後的工程工具中，INT型/WORD型以外的資料類型也可使用。(不包含BOOL型/POINTER型/STRUCT型(結構體))(可使用的資料類型與GX Works2相同。)此外，將選項設定的“執行指令引數的資料類型檢查”選擇“是”後，則可使用的資料類型將僅剩INT型/WORD型(ANY16的範圍)。

關於選項設定，請參閱下述手冊。

GX Works3 操作手冊

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

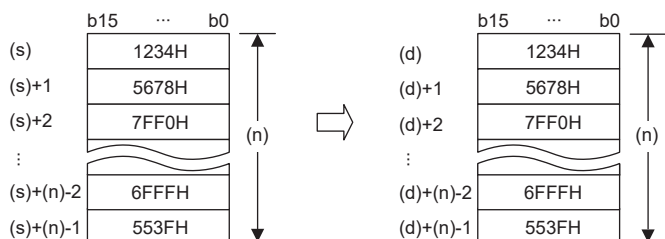
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	—	○	—	○	—	—	—	—	
(d)	○	○	○	○	—	○	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	
(s)	○	○	—
(d)	○	○	—
(n)	○	○	○

功能

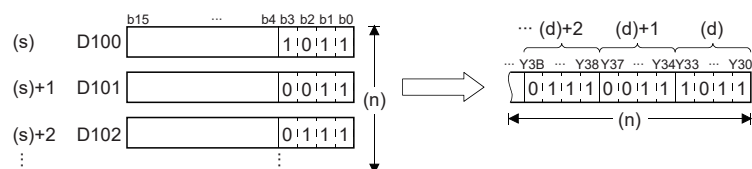
- 將(s)中指定的元件開始的(n)點的BIN16位元資料批量傳送到(d)中指定的元件。



- 傳送源與傳送目標元件重複的情況下也可進行傳送。向元件編號的小編號向傳送的情況下從(s)開始傳送，向元件編號的大編號方向傳送的情況下從(s)+(n)-1開始傳送。
- 在(s)為字元元件而(d)為位元元件的情況下，對於字元元件，位元元件的位指定中指定的位元數將成為對象。

例

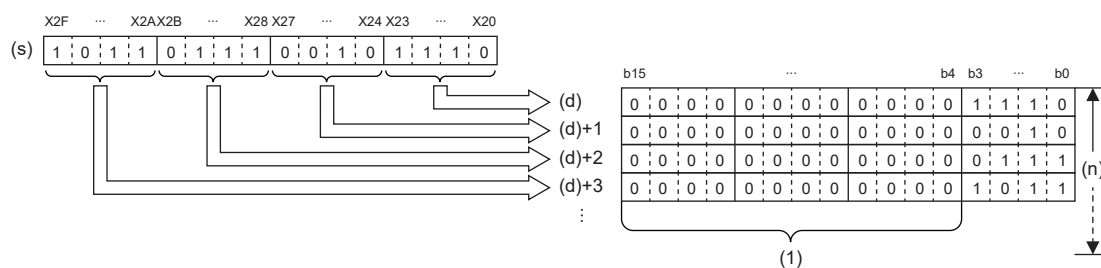
在(d)中指定K1Y30的情況下，(s)中指定的字元元件的低位4位元將成為對象。



- (s)為位指定的位元元件，在(d)為字元元件的情況下，對於字元元件，位元元件的位指定中指定的位數將成為對象。

例

在(s)中指定了K1X20的情況下，將傳送到(d)的字元元件的低位4位元中。



(1) 0被儲存。

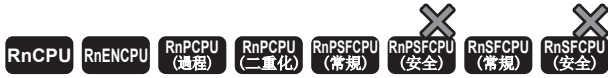
- 在(s)、(d)兩方均為位元元件的情況下，必須將(s)、(d)的位數設定為相同。
- 在(s)、(d)中使用直接連結元件、模組存取元件、CPU緩衝記憶體存取元件的情況下，只應指定在(s)或(d)中的一方。但是，對於I/O No. 指定中未指定變址修飾的本機CPU模組的CPU緩衝記憶體存取元件(U3E0\G□)，可透過(s)、(d)的兩方指定。

出錯

出錯代碼(SD0)	內容
3420H	(s)、(d)的兩方中指定了直接連結元件、模組存取元件、CPU緩衝記憶體存取元件時。但是，指定I/O No. 指定中未指定變址修飾的本機CPU模組的CPU緩衝記憶體存取元件(U3E0\G□)時除外。

BIN16位元資料塊傳送(32位元)

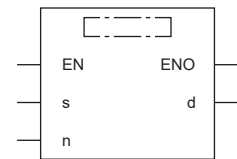
BMOV L(P)



將從指定的元件開始的(n)點(1~4294967295)的BIN16位元資料進行批量傳送。

梯形圖	ST
	ENO:=BMOV L(EN, s, n, d); ENO:=BMOVLP(EN, s, n, d);

FBD/LD



執行條件

指令	執行條件
BMOV L	
BMOVLP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存了傳送資料的起始元件	—	帶符號BIN16位元	ANY16*1
(d)	傳送目標的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	傳送數	0~4294967295	無符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

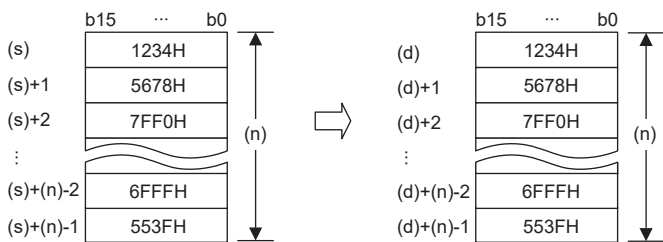
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\(\H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	—	○	—	○	—	—	—	—	—
(d)	○	○	○	○	—	○	—	○	—	—	—	—	—
(n)	○	○	○	○	○	○	○	○	○	—	—	—	—

功能

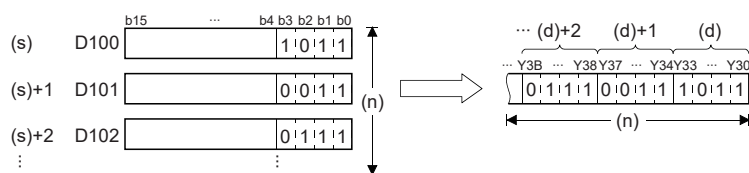
- 將從(s)中指定的元件開始的(n)點的BIN16位元資料，批量傳送到(d)中指定的元件開始的(n)點。



- 傳送源與傳送目標元件重複的情況下也可進行傳送。向元件編號的小編號向傳送的情況下從(s)開始傳送，向元件編號的大編號方向傳送的情況下從(s)+(n)-1開始傳送。
- 在(s)為字元元件而(d)為位元元件的情況下，對於字元元件，位元元件的位指定中指定的位元數將成為對象。

例

在(d)中指定K1Y30的情況下，(s)中指定的字元元件的低位4位元將成為對象。



- 在(s)、(d)兩方均為位元元件的情況下，必須將(s)、(d)的位數設定為相同。
- 在(s)、(d)中使用直接連結元件、模組存取元件、CPU緩衝記憶體存取元件的情況下，只應指定在(s)或(d)中的一方。但是，對於I/O No. 指定中未指定變址修飾的本機CPU模組的CPU緩衝記憶體存取元件(U3E0\G□)，可透過(s)、(d)的兩方指定。

出錯

出錯代碼(SD0)	內容
3420H	(s)、(d)的兩方中指定了直接連結元件、模組存取元件、CPU緩衝記憶體存取元件時。 但是，指定I/O No. 指定中未指定變址修飾的本機CPU模組的CPU緩衝記憶體存取元件(U3E0\G□)時除外。

同一BIN16位元資料塊傳送(16位元)

FMOV (P)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將BIN16位元資料傳送到指定的元件開始的(n)點(0~65535)中。

梯形圖	ST
	ENO:=FMOV (EN, s, n, d); ENO:=FMOV (EN, s, n, d);

FBD/LD



■執行條件

指令	執行條件
FMOV	
FMOVP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	傳送的資料或儲存了傳送資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	傳送目標的起始元件	—	帶符號BIN16位元	ANY16*1*2
(n)	傳送數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

- *1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。
- *2 在版本為“1.030G”以後的工程工具中，INT型/WORD型以外的資料類型也可使用。(不包含BOOL型/POINTER型/STRUCT型(結構體))(可使用的資料類型與GX Works2相同。)此外，將選項設定的“執行指令引數的資料類型檢查”選擇“是”後，則可使用的資料類型將僅剩INT型/WORD型(ANY16的範圍)。
關於選項設定，請參閱下述手冊。
 GX Works3 操作手冊
- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

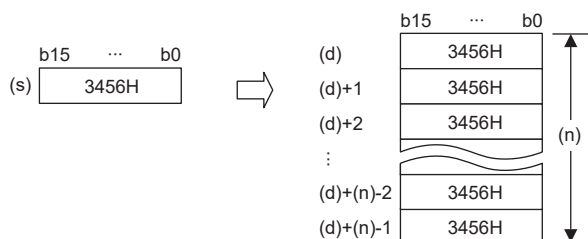
操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	—
(d)	○	○	—
(n)	○	○	○

功能

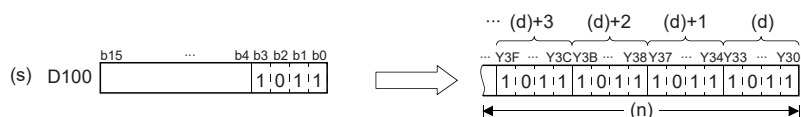
- 將與(s)中指定的元件的BIN16位元資料相同的資料，以(n)點傳送到(d)中指定的元件中。



- 在(s)為字元元件而(d)為位元元件的情況下，對於字元元件，位元元件的位指定中指定的位元數將成為對象。

例

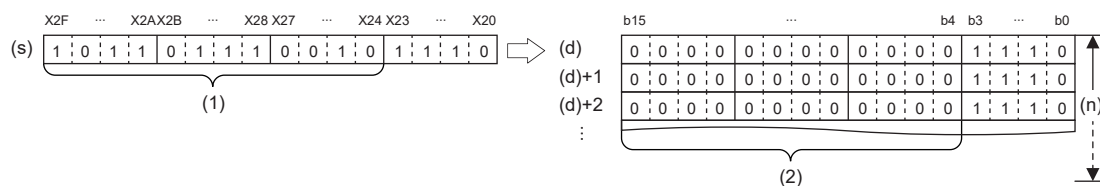
在(d)中指定K1Y30的情況下，(s)中指定的字元元件的低4位元將成為對象。



- (s)為位指定的位元元件，在(d)為字元元件的情況下，對於字元元件，位元元件的位指定中指定的位數將成為對象。

例

在(s)中指定了K1X20的情況下，將傳送到(d)的字元元件的低4位元中。



- 忽略。
- 0被儲存。

- 在(s)、(d)兩方均為位元元件的情況下，必須將(s)、(d)的位數設定為相同。

出錯

沒有運算出錯。

同一BIN16位元資料塊傳送(32位元)

FMOVL(P)



將BIN16位元資料傳送到指定的元件開始的(n)點(1~4294967295)中。

梯形圖	ST
	ENO:=FMOVL(EN, s, n, d); ENO:=FMOVLP(EN, s, n, d);

FBD/LD

■執行條件

指令	執行條件
FMOVL	
FMOVLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	傳送的資料或儲存了傳送資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	傳送目標的起始元件	—	帶符號BIN16位元	ANY16*1
(n)	傳送數	0~4294967295	無符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

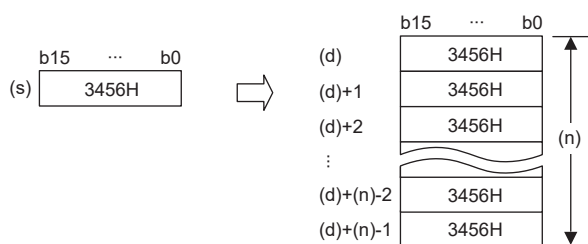
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	○	○	○	○	—	—	—

功能

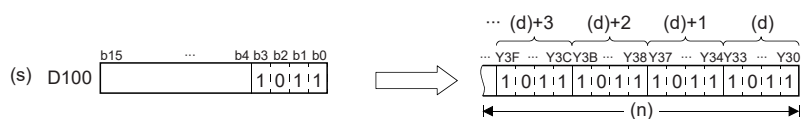
- 將與(s)中指定的元件的BIN16位元資料相同的資料，以(n)點傳送到(d)中指定的元件中。



- 在(s)為字元元件而(d)為位元元件的情況下，對於字元元件，位元元件的位指定中指定的位元數將成為對象。

例

在(d)中指定K1Y30的情況下，(s)中指定的字元元件的低位4位元將成為對象。



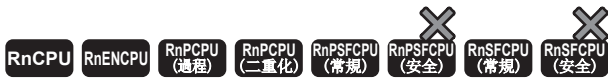
- 在(s)、(d)兩方均為位元元件的情況下，必須將(s)、(d)的位數設定為相同。

出錯

沒有運算出錯。

同一BIN32位元資料塊傳送(16位元)

DFMOV (P)



將BIN32位元資料傳送至指定的元件開始的(n)點(1~65535)。

梯形圖	ST
	ENO:=DFMOV (EN, s, n, d) ; ENO:=DFMOV (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
DFMOV	
DFMOV P	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	傳送資料或儲存了傳送資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	傳送目標的起始元件	—	帶符號BIN32位元	ANY32*1
(n)	傳送數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

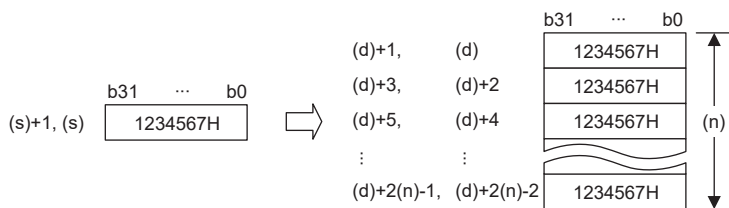
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

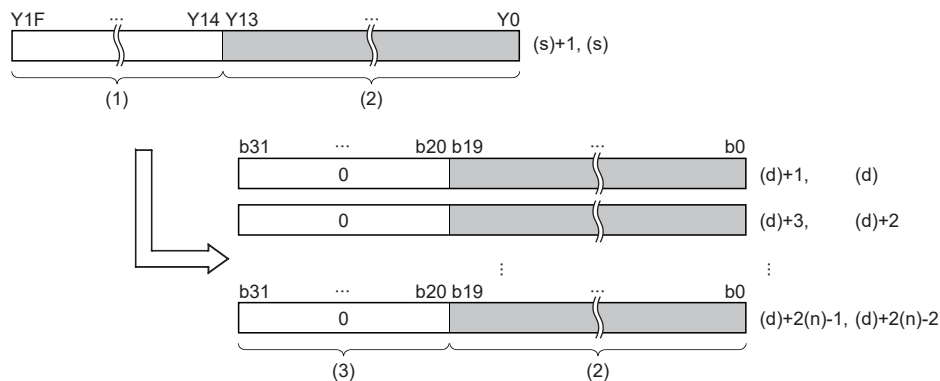
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	—	○	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 將與(s)中指定的元件的BIN32位元資料相同的資料，傳送(n)點至(d)中指定的元件中。

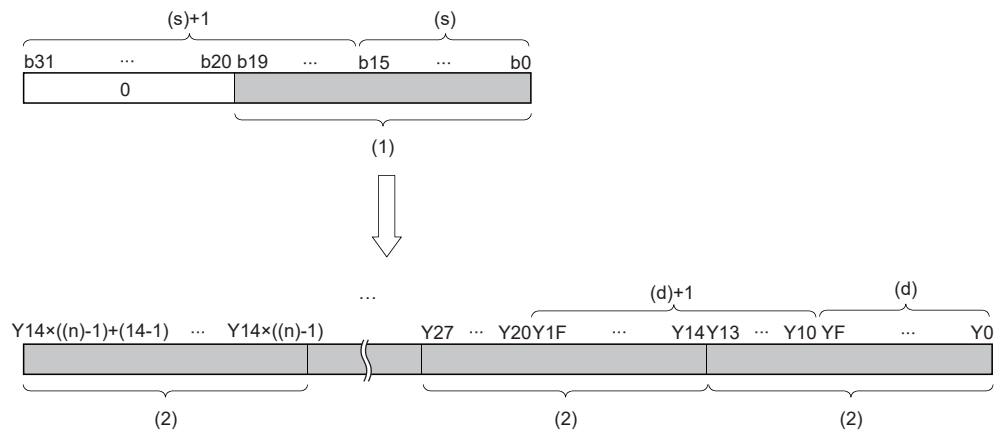


- 在(s)指定位指定的情況下，傳送的資料僅為位指定的量。在(s)中指定K5Y0的情況下，(s)的字元元件的低位20位元(5位分)將成為對象。



- 忽略。
- 20位元分(5位分)的資料
- 0被儲存。

- 在(d)中指定位指定的情況下，對(d)中指定的位指定的資料進行傳送。在(d)中指定K5Y0的情況下，(s)的元件的低位20位元將成為對象。在(s)、(d)的兩方中指定位指定的情況下，也與位數無關對(d)中指定的位指定的資料進行傳送。



- (d)中指定(20位元分(5位分))
- 20位元分(5位分)的資料

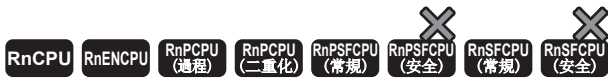
- (n)中指定的值為0的情況下將變為無處理。

出錯

沒有運算出錯。

同一BIN32位元資料塊傳送 (32位元)

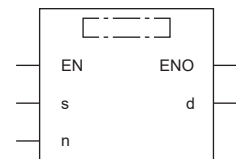
DFMOVLP (P)



將BIN32位元資料傳送至指定的元件開始的 (n) 點 (1~4294967295)。

梯形圖	ST
	ENO:=DFMOVLP (EN, s, n, d) ; ENO:=DFMOVLP (EN, s, n, d) ;

FBD/LD



執行條件

指令	執行條件
DFMOVLP	
DFMOVLP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	傳送資料或儲存了傳送資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	傳送目標的起始元件	—	帶符號BIN32位元	ANY32*1
(n)	傳送數	0~4294967295	無符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

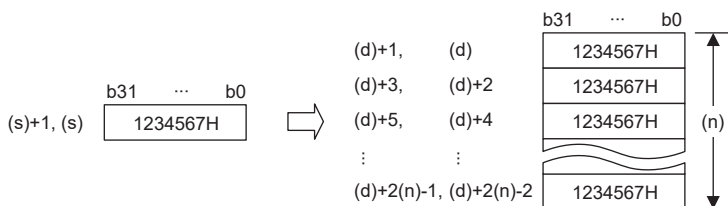
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

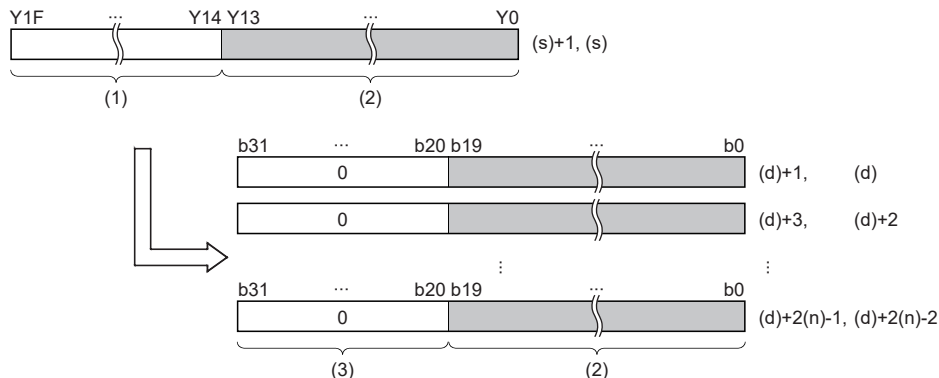
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	—	○	—	○	—	—	—	—	—
(n)	○	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 將與(s)中指定的元件的BIN32位元資料相同的資料，傳送(n)點至(d)中指定的元件中。

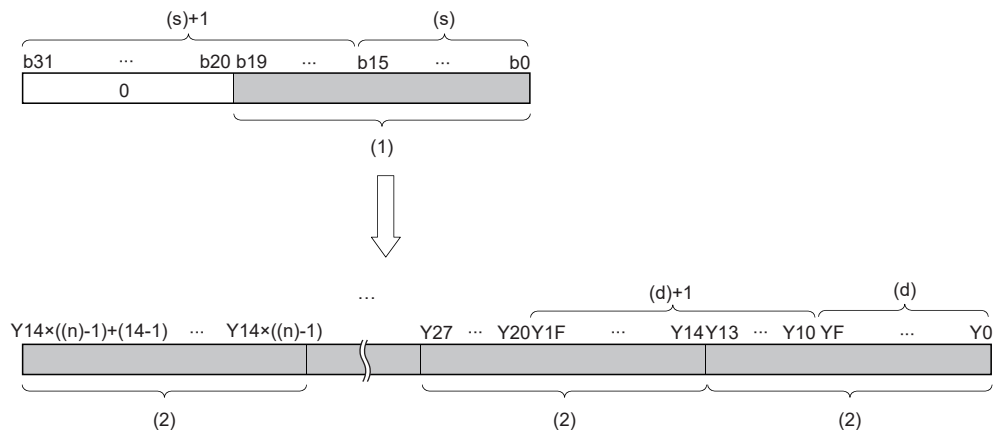


- 在(s)指定位指定的情況下，傳送的資料僅為位指定的量。在(s)中指定K5Y0的情況下，(s)的字元元件的低位20位元(5位分)將成為對象。



- 忽略。
- 20位元分(5位分)的資料
- 0被儲存。

- 在(d)中指定位指定的情況下，對(d)中指定的位指定的資料進行傳送。在(d)中指定K5Y0的情況下，(s)的元件的低位20位元將成為對象。在(s)、(d)的兩方中指定位指定的情況下，也與位數無關對(d)中指定的位指定的資料進行傳送。



- (d)中指定(20位元分(5位分))
- 20位元分(5位分)的資料

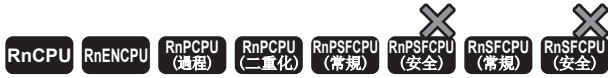
- (n)中指定的值為0的情況下將變為無處理。

出錯

沒有運算出錯。

BIN16位元資料交換

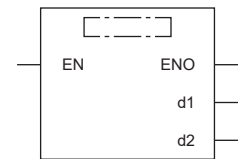
XCH(P)



對指定的BIN16位元資料進行交換。

梯形圖	ST
	ENO:=XCH(EN, d1, d2); ENO:=XCHP(EN, d1, d2);

FBD/LD



執行條件

指令	執行條件
XCH	
XCHP	

設置數據

內容、範圍、資料類型

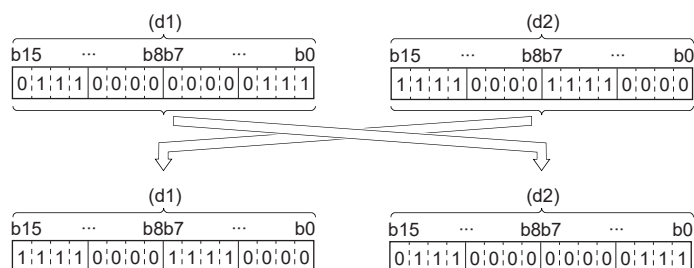
操作數	內容	範圍	資料類型	資料類型(標籤)
(d1)	儲存了交換資料的元件	—	帶符號BIN16位元	ANY16
(d2)	儲存了交換資料的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		Z	雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□		LT、LST、LC	LZ		K	H	E	
(d1)	○	○	○	○	○	—	—	○	—	—	—	—
(d2)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 對(d1)與(d2)的BIN16位元資料進行交換。

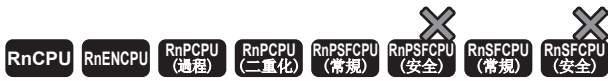


出錯

沒有運算出錯。

BIN32位元資料交換

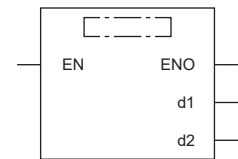
DXCH(P)



對指定的BIN32位元資料進行交換。

梯形圖	ST
	ENO:=DXCH(EN, d1, d2); ENO:=DXCHP(EN, d1, d2);

FBD/LD



執行條件

指令	執行條件
DXCH	
DXCHP	

設置數據

內容、範圍、資料類型

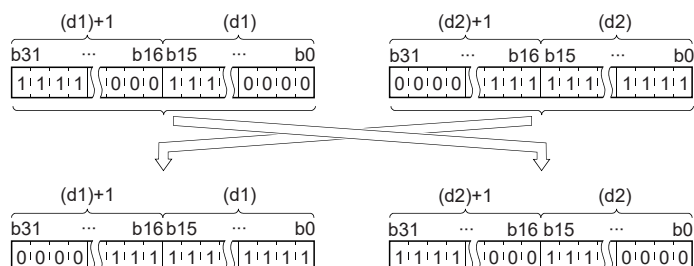
操作數	內容	範圍	資料類型	資料類型(標籤)
(d1)	儲存交換資料的起始元件	—	帶符號BIN32位元	ANY32
(d2)	儲存交換資料的起始元件	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
(d1)	○	○	○	○	○	○	○	○	○	○	○	○
(d2)	○	○	○	○	○	○	○	○	○	○	○	○

功能

- 對(d1)與(d2)的BIN32位元資料進行交換。

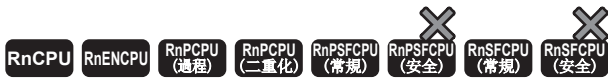


出錯

沒有運算出錯。

BIN16位元資料塊交換

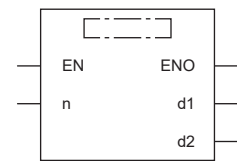
BXCH(P)



對指定的元件開始的(n)點的BIN16位元資料進行交換。

梯形圖	ST
	ENO:=BXCH(EN, n, d1, d2); ENO:=BXCHP(EN, n, d1, d2);

FBD/LD



執行條件

指令	執行條件
BXCH	
BXCHP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d1)	儲存交換資料的起始元件	—	字元	ANY16*1
(d2)	儲存交換資料的起始元件	—	字元	ANY16*1
(n)	交換數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

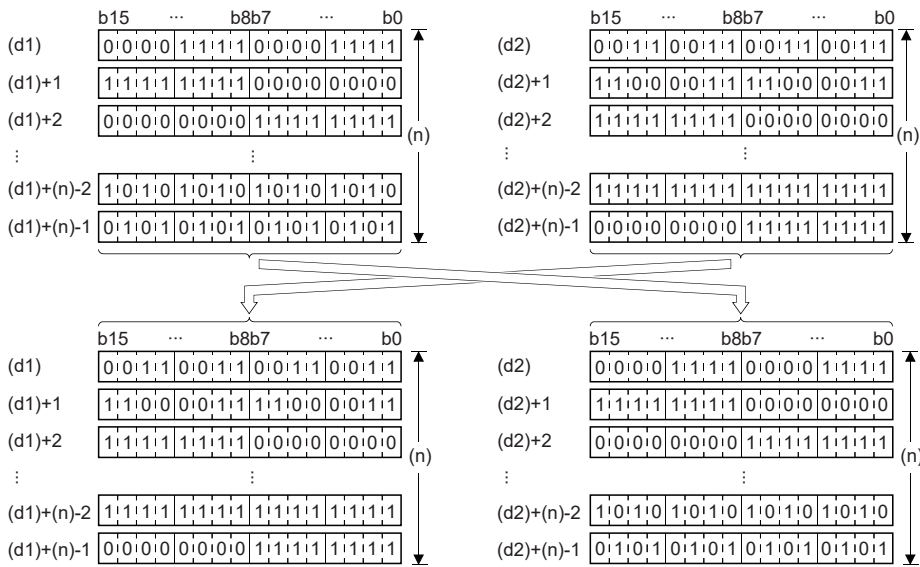
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

操作數	位元		字元				雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ	K、H		E	\$		
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 對(d1)中指定的元件與(d2)中指定的元件開始的(n)點的BIN16位元資料進行交換。

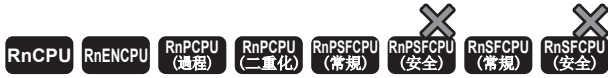


出錯

出錯代碼 (SD0)	內容
2821H	當 (d1)、(d2) 的元件重複時。

BIN16位元資料上下字節交換

SWAP (P)



對指定的元件的上下各8位元的值進行交換。

梯形圖	ST
	ENO:=SWAP (EN, d) ; ENO:=SWAPP (EN, d) ;

FBD/LD

■執行條件

指令	執行條件
SWAP	
SWAPP	

設置數據

■內容、範圍、資料類型

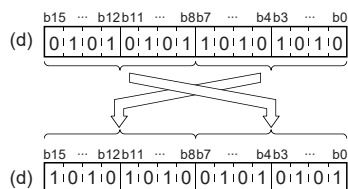
操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存對上下各8位元的值進行交換的資料的起始元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E		\$
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 對 (d) 中指定的元件的上下各8位元的值進行交換。

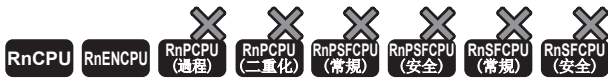


出錯

沒有運算出錯。

BIN32位元資料上下字節交換

DSWAP (P)

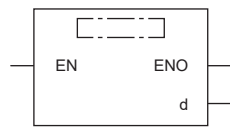


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

對指定的元件的上下各8位元的值進行交換。

梯形圖	ST
	ENO:=DSWAP (EN, d) ; ENO:=DSWAPP (EN, d) ;

FBD/LD



■執行條件

指令	執行條件
DSWAP	
DSWAPP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存對上下各8位元的值進行交換的資料的起始元件	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

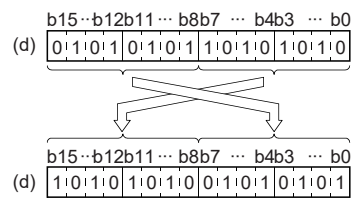
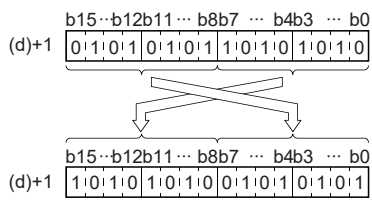
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(d)	○*1	○	○	○	○	○	○	○	—	—	—	—

*1 不能使用FX、FY。

功能

- 對 (d) 中指定的元件的上下各 8 位元的值進行交換。



出錯

沒有運算出錯。

1位元資料傳送

MOVBP (P)

RnCPU RnENCPU RnPCPU (通稱) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對指定的1位元資料進行傳送。

梯形圖	ST
	ENO:=MOVBP (EN, s, d); ENO:=MOVBP (EN, s, d);

FBD/LD

■執行條件

指令	執行條件
MOVBP	
MOVBP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	儲存了傳送資料的元件編號	—	位元	ANY_BOOL
(d)	傳送目標的元件編號	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

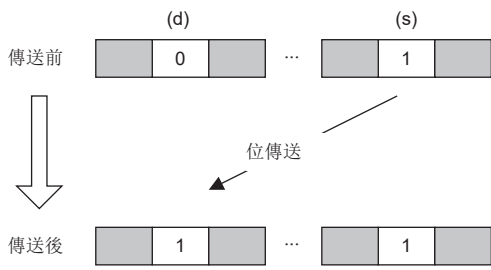
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	○	○	—	—	○	—	○	—	—	—
(d)	○	○	○	—	—	○	—	○	—	—	—

• 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	—
(d)	○	○	—

功能

- 將 (s) 中指定的位元資料傳送至 (d) 中。



出錯

沒有運算出錯。

n位資料傳送

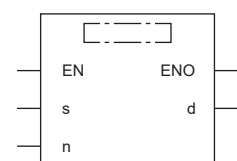
BLKMOVB (P)

RnCPU RnENCPU RnPCPU (通稱) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

對(n)點的位元資料進行批量傳送。

梯形圖	ST
	ENO:=BLKMOVB (EN, s, n, d) ; ENO:=BLKMOVBP (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
BLKMOVB	
BLKMOVBP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	傳送源的塊資料	—	位元	ANY_BOOL*1
(d)	傳送目標的塊資料	—	位元	ANY_BOOL*1
(n)	傳送數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用上述資料類型的安全元件及安全標籤。

■可使用的元件

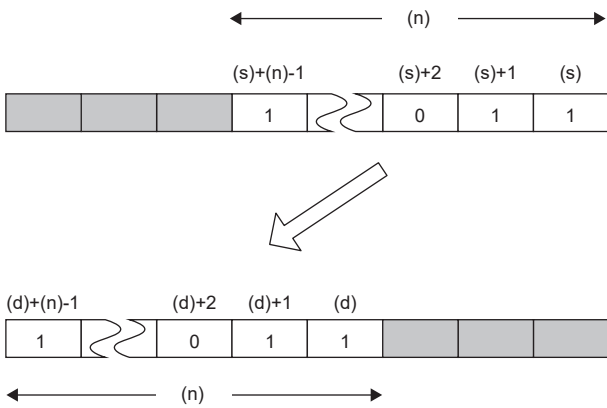
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	—	○	—	—	—	—	○	—	—	—	—	
(d)	○	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

- 在SIL2過程CPU及安全CPU的安全程式中，只能使用下述安全元件及常數。

操作數	位元	字元	常數
	SA\X、SA\Y、SA\M、SA\SM、SA\B	SA\T、SA\ST、SA\C、SA\D、SA\W、SA\SD	K、H
(s)	○	○	—
(d)	○	○	—

功能

- 將從(s)開始的(n)點的位資料批量傳送到(d)開始的(n)點的位元資料。
- 在傳送源與傳送目標重複的情況下，也可進行傳送。



出錯

沒有運算出錯。

第5部分 應用指令

本部分由下述章節構成。

- 7 程式控制
- 8 資料處理
- 9 偵錯、故障診斷
- 10 字元串處理
- 11 實數處理
- 12 隨機數
- 13 元件操作
- 14 定時器、計數器
- 15 就近控制
- 16 斜坡信號
- 17 矩陣輸入
- 18 CPU模組內建資料庫存取功能
- 19 時鐘
- 20 模組存取
- 21 參數設定操作
- 22 CPU模組資料記錄功能
- 23 記錄功能
- 24 內建乙太網路功能用指令

25 PID運算指令

26 PID控制指令

27 多CPU之間專用指令

28 SFC程式用指令

29 二重化系統用指令

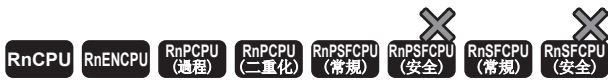
30 安全系統用指令

7 程式控制

7.1 程式分支指令

指標分支

CJ、SCJ、JMP



- CJ：執行同一程式檔案內指定的指標編號的程式。
- SCJ：從下一個掃描開始執行同一程式檔案內指定的指標編號的程式。
- JMP：無條件執行同一程式檔案內指令的指標編號的程式。

梯形圖	ST
	不對應。

FBD/LD
不對應。

■執行條件

指令	執行條件
CJ SCJ	
JMP	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(P)	跳轉目標的指標編號	—	元件名稱	POINTER

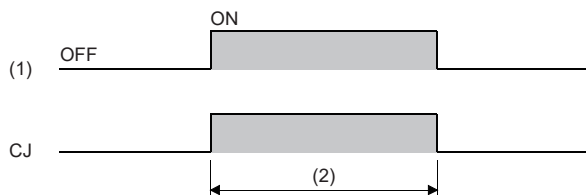
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其它 (P)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(P)	—	—	—	—	—	—	—	—	—	—	—	—	○

功能

■CJ

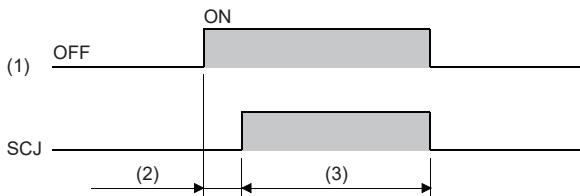
- 執行指令為ON時，執行同一程式檔案內指定的指標編號的程式。
- 執行指令為OFF時，執行下一步的程式。



- (1): 執行指令
(2): 每掃描執行

■SCJ

- 從執行指令由OFF→ON變化的下一個掃描開始，執行同一程式檔案內指定的指標編號的程式。
- 執行指令為OFF及由ON→OFF變化時，執行下一個步的程式。



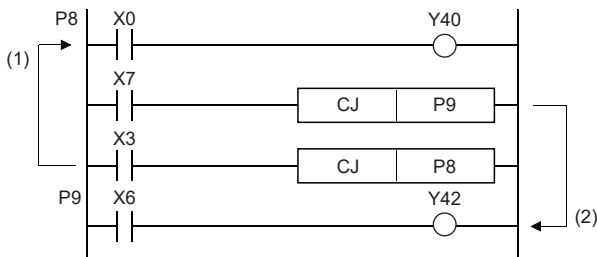
- (1): 執行指令
 (2): 1掃描
 (3): 每掃描執行

■JMP

- 無條件執行同一程式下一個檔案內指定的指標編號的程式。

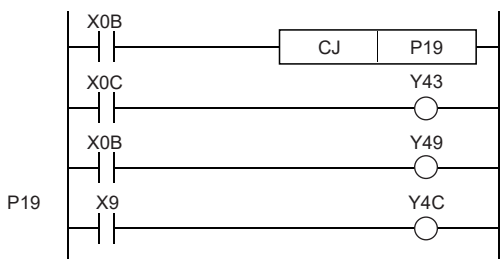
注意事項

- 將定時器的線圈設為ON後，透過CJ指令、SCJ指令、JMP指令對線圈為ON的定時器進行了跳轉的情況下，將無法正常進行計測。
- 透過CJ指令、SCJ指令、JMP指令對OUT指令進行跳轉時掃描時間將變短。
- 透過CJ指令、SCJ指令、JMP指令向後跳轉時掃描時間將變短。
- 對於CJ指令、SCJ指令、JMP指令，可以跳轉至比執行中的步號小的步之處。但是，為了避免看門狗定時器時限到，這期間應考慮從環路跳出的方法。



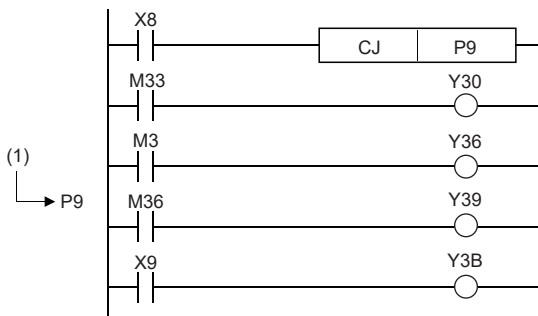
- (1) X3為ON期間，執行環路。
 (2) 將X7設為ON時，從環路中跳出。

- 透過CJ指令、SCJ指令、JMP指令跳過的元件不變化。



- (1) XB為ON時，跳轉至P19的標籤。
 CJ指令執行中即使XB、XC變為ON/OFF，Y43、Y49也不變化。

- 標籤(P□)佔用1步。



- (1) 佔用1步。

- 跳轉指令只能指定同一程式檔案內的指標編號。

出錯

出錯代碼 (SD0)	內容
3380H	(P) 中指定的指標編號在END指令之前不存在時。
	(P) 中指定的指標編號未在同一程式中作為標籤使用時。
	(P) 中指定的指標編號為其它程式中的全局指標時。

跳轉至END

GOEND


RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

跳轉至同一程式檔案內的FEND或END指令。

梯形圖	ST
	不對應。

FBD/LD
不對應。

■執行條件

指令	執行條件
GOEND	

功能

- 跳轉至同一程式檔案內的FEND或END指令。

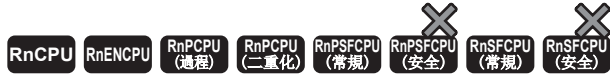
出錯

出錯代碼 (SD0)	內容
3340H	執行FOR指令後，在執行NEXT指令前執行了GOEND指令時。
3381H	執行CALL (P) 指令、ECALL (P) 指令後，執行RET指令前執行了GOEND指令時。
33A1H	在透過中斷指標 (I) 進行的中斷程式中，執行IRET指令前執行了GOEND指令時。

7.2 程式執行控制指令

中斷禁止、中斷允許

DI、EI



- DI: 禁止中斷程式的執行。
- EI: 解除中斷禁止狀態。

梯形圖	ST
	ENO:=DI (EN) ; ENO:=EI (EN) ;

FBD/LD

執行條件

指令	執行條件
DI EI	常時執行

功能

DI

- 即使發生中斷程式的啟動原因，在執行EI指令之前仍禁止中斷程式的執行。
- 在電源投入時或進行了CPU模組重設的情況下，將變為執行了DI指令後的狀態。
- 關於使用DI (指定優先度及其以下的中斷禁止) 指令時的DI (中斷禁止) 指令的動作，請參閱下述內容。

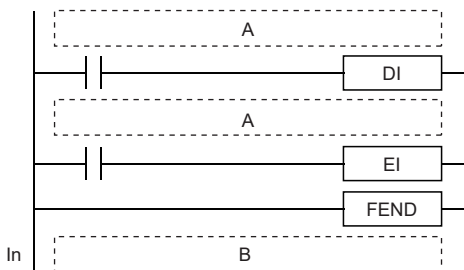
☞ 559頁 DI

- DI (中斷禁止) 指令不能在中斷程式中執行。執行的情況下將變為無處理。

EI

- 解除執行DI (中斷禁止) 指令時的中斷禁止狀態，根據IMASK指令將被允許的中斷指標編號之中斷程式的執行變為可能狀態。當IMASK指令非執行時，I32~I43將變為中斷禁止。
- 關於使用DI (指定優先度及其以下的中斷禁止) 指令時的EI指令的動作，請參閱下述內容。

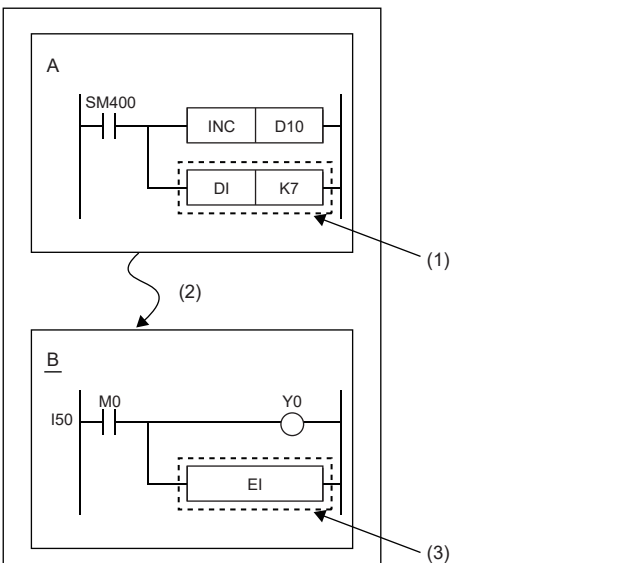
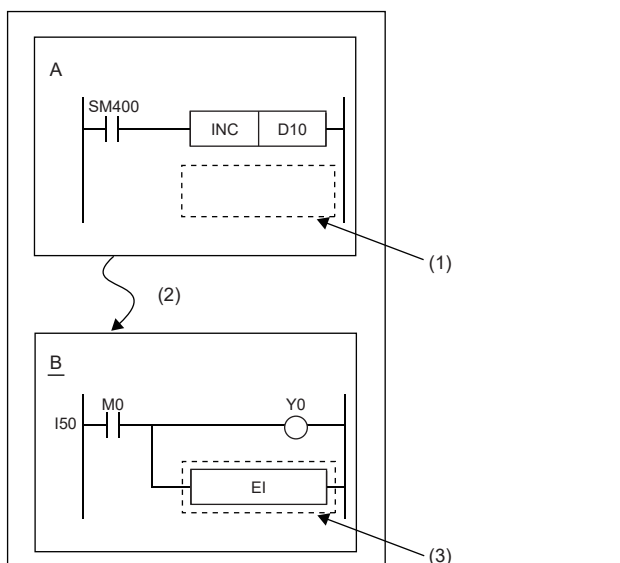
☞ 559頁 DI



A: 順控程式
B: 中斷程式

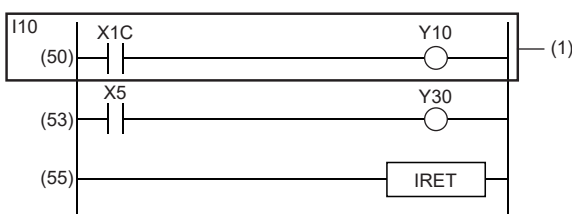
DI~EI指令之間即使發生中斷原因，在DI~EI指令之間的處理結束之前，中斷程式將等待。

- 中斷程式中的EI指令的動作，根據EI指令執行前的DI(指定優先度及其以下的中斷禁止)指令的執行而變化。執行DI(指定優先度及其以下的中斷禁止)指令後的中斷程式中的EI指令可以執行。中斷程式中的EI指令的執行可否如下所示。

執行DI(指定優先度及其以下的中斷禁止)指令後，在透過中斷程式執行EI指令的情況下	在未執行DI(指定優先度及其以下的中斷禁止)指令的狀況中，透過中斷程式執行了EI指令的情況下
 <p>A: 主程式 B: 中斷程式 (1): 執行DI(指定優先度及其以下的中斷禁止)指令 (2): 中斷發生 (3): 可以執行</p>	 <p>A: 主程式 B: 中斷程式 (1): 未執行DI(指定優先度及其以下的中斷禁止)指令 (2): 中斷發生 (3): 不可以執行(無處理)</p>

要點

- 中斷用指標佔用1步。(在下述(1)中，I10: 步50、X1C: 步51、Y10: 步52。)



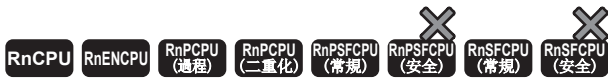
- 主控制中有EI指令、DI指令時與MC指令的執行、非執行無關，執行EI指令、DI指令。

出錯

出錯代碼(SD0)	內容
3362H	在透過DI(中斷禁止)指令、DI(指定優先度及其以下的中斷禁止)指令進行的嵌套超過了16層的情況下。

指定優先度及其以下的中斷禁止

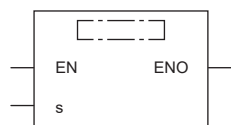
DI



禁止指定的優先度及其以下的中斷程式的執行。

梯形圖	ST
	<pre>ENO:=DI_1(EN, s);</pre>

FBD/LD



(□中放入DI_1。)

■執行條件

指令	執行條件
DI	常時執行

7

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	禁止中斷的優先度	1~8*1	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在R00CPU、R01CPU、R02CPU為3~8。

■可使用的元件

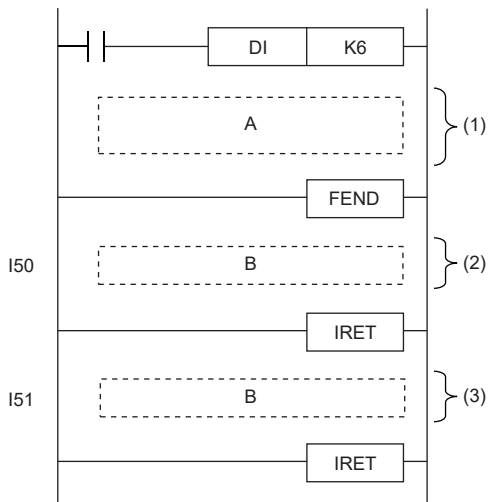
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	—	○	—	○	—	—	○	—	—	—

功能

- 禁止(s)中指定的中斷優先度及其以下的中斷指標編號的中斷程式。

中斷優先度設定

I No.	優先度
I50	5
I51	6

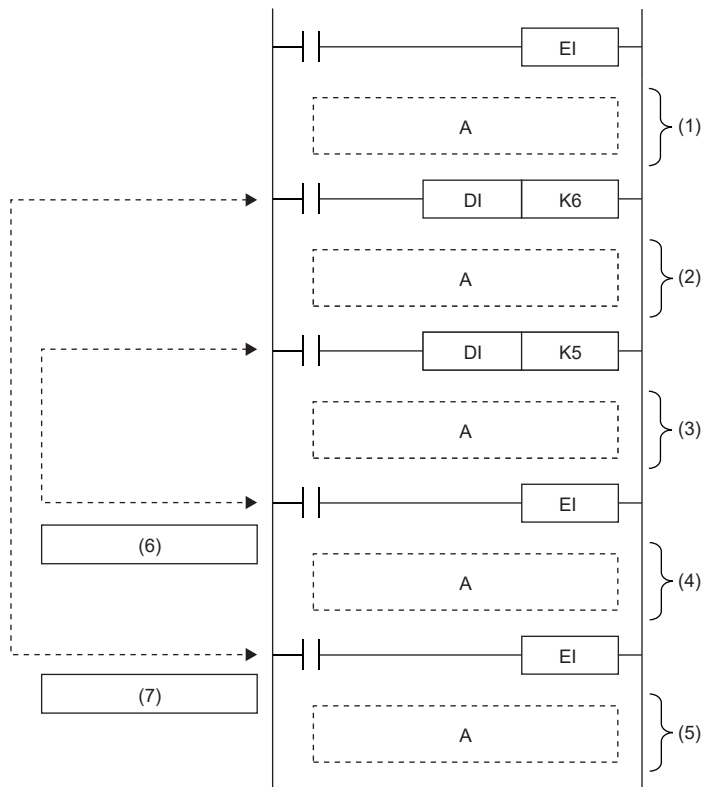


A: 順控程式

B: 中斷程式

- (1): 優先度6~8的中斷禁止區間(優先度1~5的中斷允許區間)
- (2): 因為是優先度5的中斷, 因此可以執行
- (3): 因為是優先度6的中斷, 因此禁止執行

- 透過EI指令的執行，將透過DI指令禁止的優先度的中斷設為允許。但是，在僅透過DI(中斷禁止)指令設為中斷禁止的情況下，僅執行1次EI指令便將所有優先度的中斷設為允許。



A: 順控程式

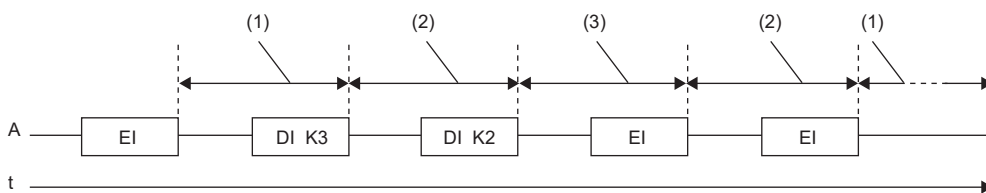
- (1): 所有優先度的中斷允許區間
- (2): 優先度6~8的中斷禁止區間(優先度1~5的中斷允許區間)
- (3): 優先度5~8的中斷禁止區間(優先度1~4的中斷允許區間)
- (4): 優先度6~8的中斷禁止區間(優先度1~5的中斷允許區間)
- (5): 所有優先度的中斷允許區間
- (6): 對應[DI K5]的EI指令
- (7): 對應[DI K6]的EI指令

- 當多次執行DI(指定優先度及其以下的中斷禁止)指令時，指定的中斷禁止優先度不大於當前禁止的中斷禁止優先度的情況下，當前的中斷禁止優先度不被更改。
- DI指令的嵌套最多可達16層。
- 對於中斷指標*1的中斷優先度可透過參數進行設定。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

*1 I0~I15、I50~I1023。

- 對於處於中斷禁止的優先度，可以透過SD758(中斷禁止優先度設定值)進行確認。
- 中斷程式中執行DI(指定優先度及其以下的中斷禁止)指令後，中斷禁止優先度被更改的情況下SD758的值也將被更改。

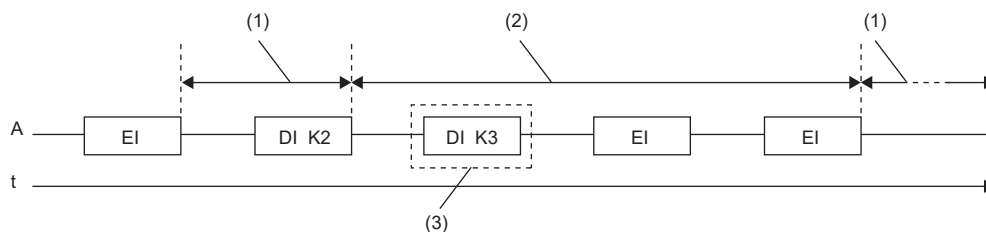
- 執行DI(中斷禁止)指令、DI(指定優先度及其以下的中斷禁止)指令及EI指令時的中斷禁止區間如下所示。
- 多次執行了DI(指定優先度及其以下的中斷禁止)指令的情況下(對高於當前禁止的中斷優先度的優先度的中斷執行了禁止指定的情況下)



A: 掃描執行類型程式

- (1): 所有優先度的中斷允許區間
- (2): 優先度3~8的中斷禁止區間(優先度1、2的中斷允許區間)
- (3): 優先度2~8的中斷禁止區間(優先度1的中斷允許區間)

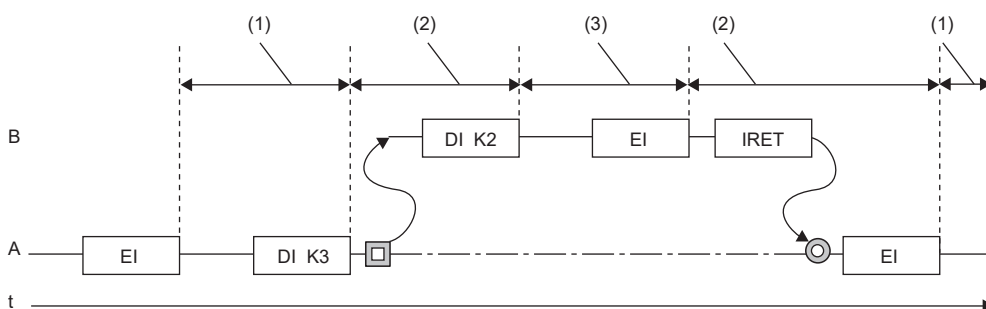
- 多次執行了DI(指定優先度及其以下的中斷禁止)指令的情況下(對低於當前禁止的中斷優先度的優先度的中斷進行了禁止指定的情況下)



A: 掃描執行類型程式

- (1): 所有優先度的中斷允許區間
- (2): 優先度2~8的中斷禁止區間(優先度1的中斷允許區間)
- (3): 因為優先度2及其以下的中斷已處於被禁止狀態,因此中斷禁止優先度不被更改。

- 在透過中斷程式執行DI(指定優先度及其以下的中斷禁止)指令的情況下

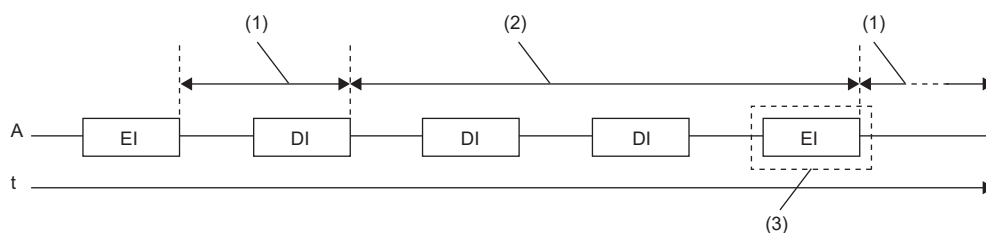


A: 掃描執行類型程式

B: 中斷程式

- (1): 所有優先度的中斷允許區間
- (2): 優先度3~8的中斷禁止區間(優先度1、2的中斷允許區間)
- (3): 優先度2~8的中斷禁止區間(優先度1的中斷允許區間)

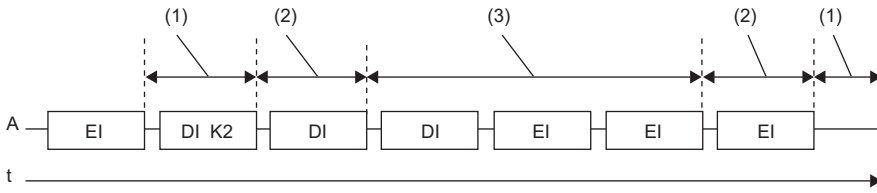
- 在僅執行DI(中斷禁止)指令的情況下



A: 掃描執行類型程式

- (1): 所有優先度的中斷允許區間
- (2): 優先度1~8的中斷禁止區間(所有的中斷禁止區間)
- (3): 由於透過DI(中斷禁止)指令設為中斷禁止,因此透過執行1次EI指令,所有優先度的中斷均被設為允許。

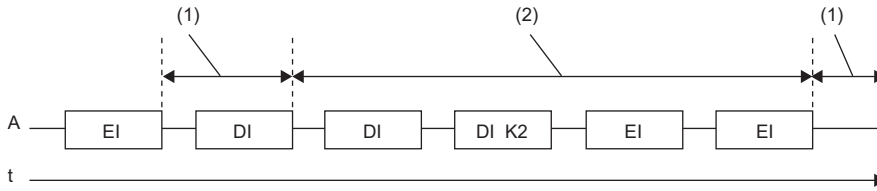
- 在執行DI(指定優先度及其以下的中斷禁止)指令及DI(中斷禁止)指令的情況下(按照DI(指定優先度及其以下的中斷禁止)指令→DI(中斷禁止)指令的順序執行的情況下)



A: 掃描執行類型程式

- (1): 所有優先度的中斷允許區間
- (2): 優先度2~8的中斷禁止區間(優先度1的中斷允許區間)
- (3): 優先度1~8的中斷禁止區間(所有的中斷禁止區間)

- 在執行DI(指定優先度及其以下的中斷禁止)指令及DI(中斷禁止)指令的情況下(按照DI(中斷禁止)指令→DI(指定優先度及其以下的中斷禁止)指令的順序執行的情況下)



A: 掃描執行類型程式

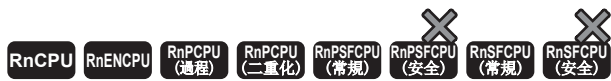
- (1): 所有優先度的中斷允許區間
- (2): 優先度1~8的中斷禁止區間(所有的中斷禁止區間)

出錯

出錯代碼(SD0)	內容
3362H	在透過DI(中斷禁止)指令、DI(指定優先度及其以下的中斷禁止)指令進行的嵌套超過了16層的情況下。
3405H	(s)中指定的優先度超出範圍時。

中斷程式屏蔽

IMASK



將指定的中斷指標編號的中斷程式設為執行允許狀態/執行禁止狀態。

梯形圖	ST
	ENO:=IMASK (EN, s) ;

FBD/LD

■執行條件

指令	執行條件
IMASK	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	中斷屏蔽資料或儲存了中斷屏蔽資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 16)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 根據(s)中指定的元件開始的16點的位元模式，將指定的中斷指標編號的中斷程式設為執行允許狀態或執行禁止狀態。
- 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
- 各位對應的中斷指標編號如下所示。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(s)	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
(s)+1	I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
(s)+2	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
(s)+3	I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
(s)+4	I79	I78	I77	I76	I75	I74	I73	I72	I71	I70	I69	I68	I67	I66	I65	I64
(s)+5	I95	I94	I93	I92	I91	I90	I89	I88	I87	I86	I85	I84	I83	I82	I81	I80
(s)+6	I111	I110	I109	I108	I107	I106	I105	I104	I103	I102	I101	I100	I99	I98	I97	I96
(s)+7	I127	I126	I125	I124	I123	I122	I121	I120	I119	I118	I117	I116	I115	I114	I113	I112
(s)+8	I143	I142	I141	I140	I139	I138	I137	I136	I135	I134	I133	I132	I131	I130	I129	I128
(s)+9	I159	I158	I157	I156	I155	I154	I153	I152	I151	I150	I149	I148	I147	I146	I145	I144
(s)+10	I175	I174	I173	I172	I171	I170	I169	I168	I167	I166	I165	I164	I163	I162	I161	I160
(s)+11	I191	I190	I189	I188	I187	I186	I185	I184	I183	I182	I181	I180	I179	I178	I177	I176
(s)+12	I207	I206	I205	I204	I203	I202	I201	I200	I199	I198	I197	I196	I195	I194	I193	I192
(s)+13	I223	I222	I221	I220	I219	I218	I217	I216	I215	I214	I213	I212	I211	I210	I209	I208
(s)+14	I239	I238	I237	I236	I235	I234	I233	I232	I231	I230	I229	I228	I227	I226	I225	I224
(s)+15	I255	I254	I253	I252	I251	I250	I249	I248	I247	I246	I245	I244	I243	I242	I241	I240

- 在電源接通時或進行了CPU模組重設的情況下，I0～I31、I44～I1023的中斷程式將變為執行允許狀態、I32～I43的中斷程式將變為執行禁止狀態。
- (s)～(s)+15元件的狀態將被儲存到SD1400～SD1463(IMASK指令屏蔽模式)中。

要點

在IMASK指令中，透過批量I0～I255的中斷指標，可使其設為執行允許狀態或執行禁止狀態。

將I256～I1023的中斷指標設為執行允許狀態或執行禁止狀態的情況下，應透過程式將IMASK指令替換為SIMASK指令。

關於SIMASK指令的詳細內容，請參閱下述內容。

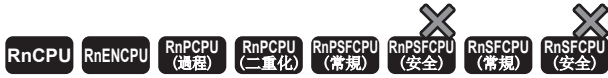
☞ 566頁 SIMASK

出錯

沒有運算出錯。

指定中斷指標的禁止/允許

SIMASK



將指定的中斷指標編號的中斷程式設為執行允許狀態/執行禁止狀態。

梯形圖	ST
	<pre>ENO:=SIMASK (EN, I, s);</pre>

FBD/LD

■執行條件

指令	執行條件
SIMASK	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(I)	對中斷進行允許/禁止設定的中斷指標No.	I0~I1023	元件名稱	POINTER*1
(s)	指定的中斷指標No. 的允許/禁止	0: 禁止 1: 允許	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用元件 (I) 中分配的標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其它 (I)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(I)	—	—	—	—	—	—	—	—	—	—	—	○
(s)	—	—	○	—	○	—	○	○	○	—	—	—

功能

- 將 (I) 中指定的中斷指標編號的中斷程式，按照 (s) 中指定的資料設為執行允許狀態或執行禁止狀態。
- (s) 為 1 的情況下：中斷程式的執行允許狀態
- (s) 為 0 的情況下：中斷程式的執行禁止狀態
- 在電源接通時或進行了 CPU 模組重設的情況下，I0~I31、I44~I1023 的中斷程式將變為執行允許狀態、I32~I43 的中斷程式將變為執行禁止狀態。
- 中斷指標的執行允許狀態/執行禁止狀態將被儲存到 SD1400~SD1463 (IMASK 指令屏蔽模式) 中。

要點

對於 (I) 可以進行變址修飾。透過使用變址修飾後的 SIMASK 指令，可以將 I0~I1023 的中斷指標設為執行允許狀態或執行禁止狀態。

出錯

出錯代碼 (SD0)	內容
3405H	(I) 中指定的中斷指標No. 超過中斷指標No. 的範圍時。 (s) 的值為中斷禁止 (0) / 中斷允許 (1) 以外時。

從中斷程式返回

IRET



表示中斷程式的處理結束。

梯形圖	ST
	不對應。

FBD/LD

不對應。

■執行條件

指令	執行條件
IRET	常時執行

功能

- 表示透過中斷指標 (I) 進行的中斷程式的處理結束。
- 執行IRET指令後，處理將返回至順控程式。

出錯

出錯代碼 (SD0)	內容
33A0H	中斷No. 中對應的指標不存在時。
33A1H	發生中斷後執行IRET指令前，執行了END指令、FEND指令、GOEND指令、STOP指令時。
33A2H	執行透過中斷指標 (I) 進行的中斷程式前執行了IRET指令時。
33A3H	恆定週期執行類型程式中執行了IRET指令時。
33A4H	事件執行類型程式中執行了IRET指令時。

WDT重設

WDT (P)



重設看門狗定時器。

梯形圖	ST
	ENO:=WDT (EN) ; ENO:=WDTP (EN) ;

FBD/LD

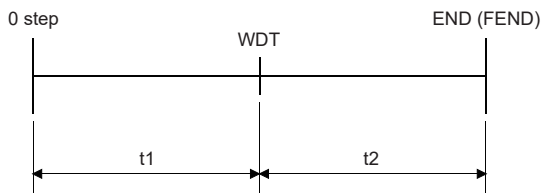
■執行條件

指令	執行條件
WDT	
WDTP	

7

功能

- 透過程式對看門狗定時器進行重設。
- 在掃描時間根據條件超過了看門狗定時器的設定值的情況下使用。在掃描時間超過每個掃描看門狗定時器的設定值的情況下，應在工程工具的參數設定中更改看門狗定時器的設定值。
- 對於從0步開始至WDT (P) 指令為止的t1，及從WDT (P) 指令開始至END (FEND) 指令為止的t2，應勿超過看門狗定時器的設定值。



- 雖然WDT (P) 指令在1個掃描中至少可以使用2次，但應注意至發生異常時的輸出OFF為止需要耗費一定的時間。
- 即使執行WDT (P) 指令，特殊寄存器中儲存的掃描時間的值也不會被清除。因此，特殊寄存器的掃描時間值有可能會大於參數中設定的看門狗定時器的設定值。

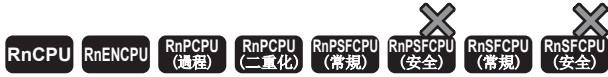
出錯

沒有運算出錯。

7.3 結構化指令

FOR~NEXT

FOR、NEXT



對FOR~NEXT指令之間的處理執行(n)次。

梯形圖	ST
<p>A: 重複程式</p>	不對應。

FBD/LD	
<p>[FOR]</p>	<p>[NEXT]</p>

■執行條件

指令	執行條件
FOR NEXT	常時執行

設置數據

■內容、範圍、資料類型

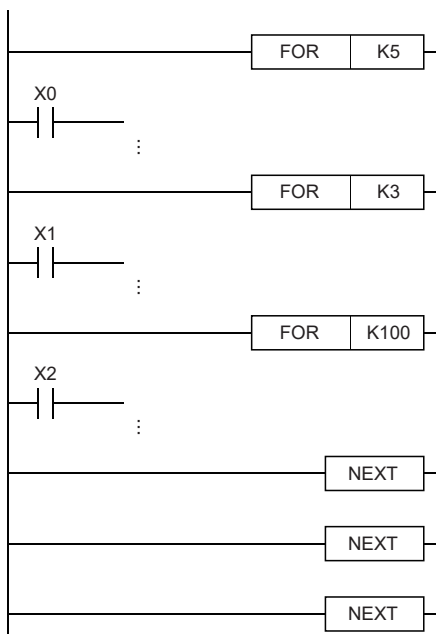
操作數	內容	範圍	資料類型	資料類型(標籤)
(n)	FOR~NEXT指令之間的重複次數	1~32767	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 將FOR~NEXT指令之間的處理無條件執行(n)次時，將進行NEXT指令的下一個步的處理。
- (n)可在1~32767的範圍內指定。指定-32768~0的情況下，將進行與(n)=1相同的處理。
- 不希望執行FOR~NEXT指令之間的處理時，應透過CJ指令、SCJ指令跳轉。
- FOR指令的嵌套最多可達16層。

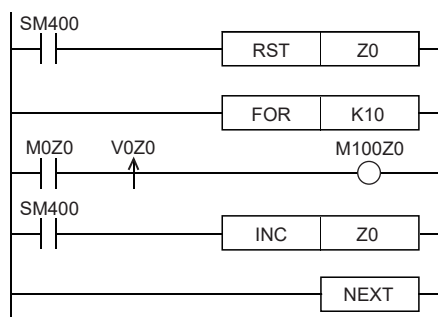


出錯

出錯代碼(SD0)	內容
3340H	執行FOR指令後，執行NEXT指令前執行了END指令、FEND指令、GOEND指令時。 FOR~NEXT指令之間存在有STOP指令時。
3341H	執行FOR指令前執行了NEXT指令時。
3361H	在進行了FOR指令的嵌套的情況下，執行第17層時。

要點

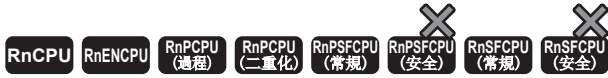
- 在FOR~NEXT指令之間的重複執行中中途結束的情況下，應使用BREAK指令。
☞ 572頁 BREAK(P)
- 在FOR~NEXT指令之間進行變址修飾後的程式的脈衝運算的情況下，應使用EGP指令、EGF指令。但是，動作輸出側不能使用上升沿指令、下降沿指令。



- 不能從FOR~NEXT指令之間外向FOR~NEXT指令內的JMP指令等進行分支。
- 為了建立易於理解的程式，對於成對的FOR指令及NEXT指令應在一個程式塊內使用。

FOR~NEXT強制結束

BREAK (P)



強制結束透過FOR~NEXT指令進行的重複處理，並執行移轉至指定的指標。

梯形圖	ST
	不對應。

FBD/LD



■執行條件

指令	執行條件
BREAK	
BREAKP	

設置數據

■內容、範圍、資料類型

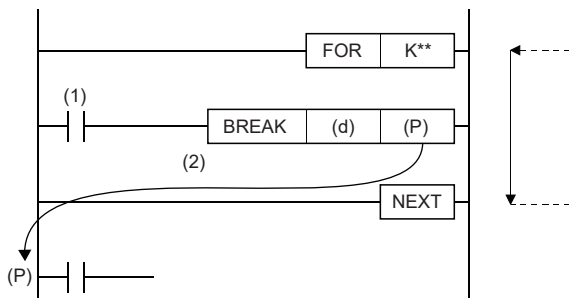
操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存重複剩餘數的元件編號	—	帶符號BIN16位元	ANY16
(P)	強制結束重複處理時的分支目標指標編號	—	元件名稱	POINTER
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其它 (P)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—
(P)	—	—	—	—	—	—	—	—	—	—	—	○

功能

- 強制結束透過FOR~NEXT指令進行的重複處理，將執行移轉至(P)中指定的指標。(P)中只能指定同一程式檔案內的指標。在(P)中指定了其它程式檔案內的指標的情況下，將變為運算出錯。



在未執行BREAK指令的情況下，以FOR指令中指定的次數返回至FOR指令。

- (1): 強制結束條件
 (2): 強制結束條件成立時

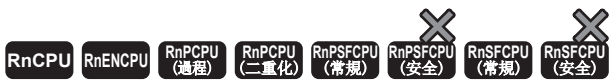
- (d)中將儲存強制結束時的FOR~NEXT指令中重複處理執行次數的剩餘數。但是，重複處理的剩餘數中還包含有BREAK(P)指令執行時的次數。
- BREAK(P)指令只能在FOR~NEXT指令之間使用。
- BREAK(P)指令只能對1個嵌套使用。在強制結束多重嵌套的情況下，應執行嵌套的BREAK(P)指令。

出錯

出錯代碼(SD0)	內容
3342H	在FOR~NEXT指令以外中使用時。
3380H	(P)中指定的指標的跳轉目標不存在時。
	(P)中指定了其它程式檔案的指標時。

子程式調用

CALL (P)



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

執行指定的指標的子程式。

梯形圖	ST
	不對應。

FBD/LD

■執行條件

指令	執行條件
CALL	
CALLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(P)	子程式的起始指標編號	—	元件名稱	POINTER
(s1) ~ (s5) ^{*1}	子程式中作為引數傳送的元件編號	-2147483648~2147483647	位元/帶符號BIN16位元 /帶符號BIN32位元	ANY
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在FBD/LD中不能指定。

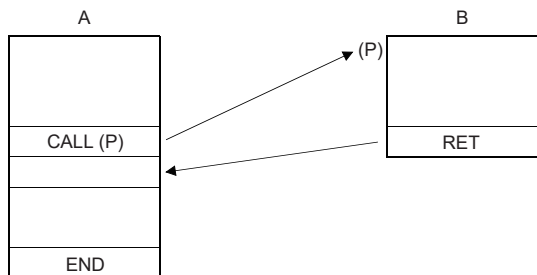
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其它 (P)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(P)	—	—	—	—	—	—	—	—	—	—	—	○
(s1) ~ (s5)	○ ^{*1}	○	○	○	○	○	○	○	○	○	—	—

*1 F以外可以使用。

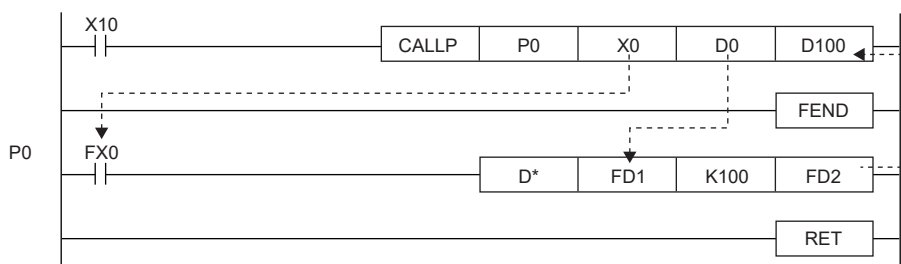
功能

- 當執行CALL (P) 指令時，將執行指標 (P) 的子程式。CALL (P) 指令可以執行同一程式檔案內的指標中指定的子程式及全局指標中指定的子程式。



A: 主程式
B: 子程式

- 子程式中使用函數元件 (FX、FY、FD) 時，在 (s1) ~ (s5) 中指定函數元件對應的元件。(s1) ~ (s5) 中指定的元件的內容如下所示。



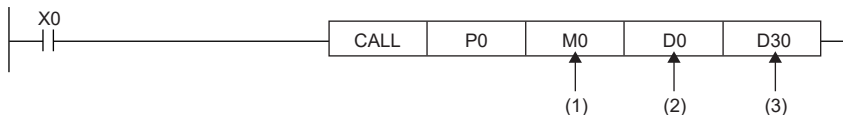
- 執行子程式前位元資料的內容將被傳送到FX中，字元資料的內容將被傳送到FD中。
- 執行子程式後，將被傳送到FY、FD的內容對應的元件中。
- FX、FY的函數元件的處理單位為位元單位。FD的函數元件的處理單位為4字元單位。根據引數中指定的元件的類型，可處理的資料容量有所不同。對於函數元件中指定的元件，應預留出資料容量相應的區域。無法預留出資料容量相應的區域的情況下將變為出錯狀態。
- 各函數元件的資料容量如下所示。

函數元件	使用元件	資料容量
FX、FY	位元元件	1點
	字元元件的位元指定時	1位元
FD	位元元件的位指定時*2	4字元*3
	字元元件	4字元*3

- *2 當位元元件的位指定時，即使 (s1) ~ (s5) 中指定的元件編號不是16的倍數的情況下，也不變為出錯狀態。
- *3 資料容量根據所使用的指令而變化。

例

指定的元件的內容



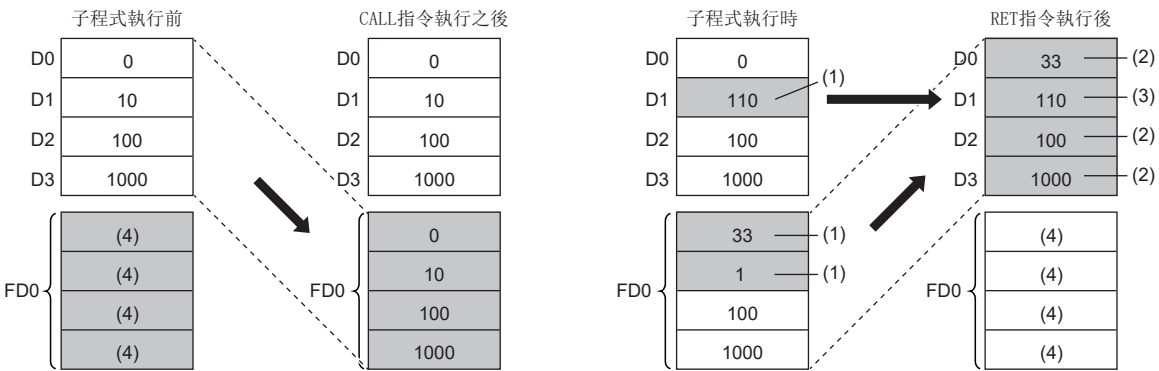
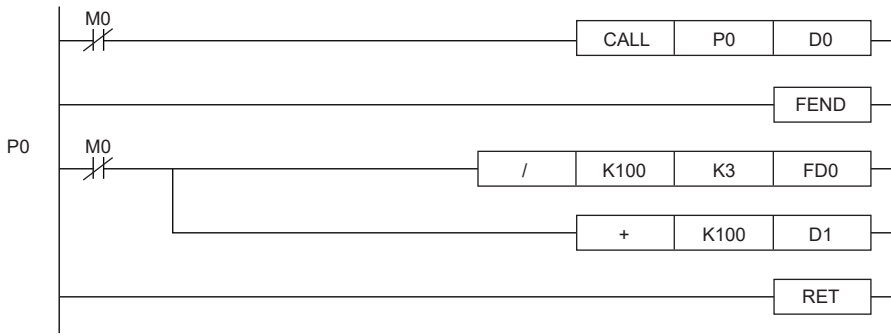
- (1) 佔用M0 (傳送至FX0)
- (2) 佔用D0~D3 (傳送至FD1)
- (3) 佔用D30~D33 (傳送至FD2)

- 在CALL (P) 指令中，可以使用 (s1) ~ (s5)。
- 對於子程式中使用的函數元件數及CALL (P) 指令的引數，應設定為相同的數。此外，應將函數元件與CALL (P) 指令的引數的類型設定為相同。
- CALL (P) 指令中引數中指定的元件編號應勿重複。重複的情況下將無法正常運算。
- 在CALL (P) 指令的引數的元件中使用定時器或計數器的情況下，僅對當前值進行發送接收。

• 應勿將CALL (P) 指令的引數中使用的元件用於子程式。在使用的情况下，將無法正常運算。

例

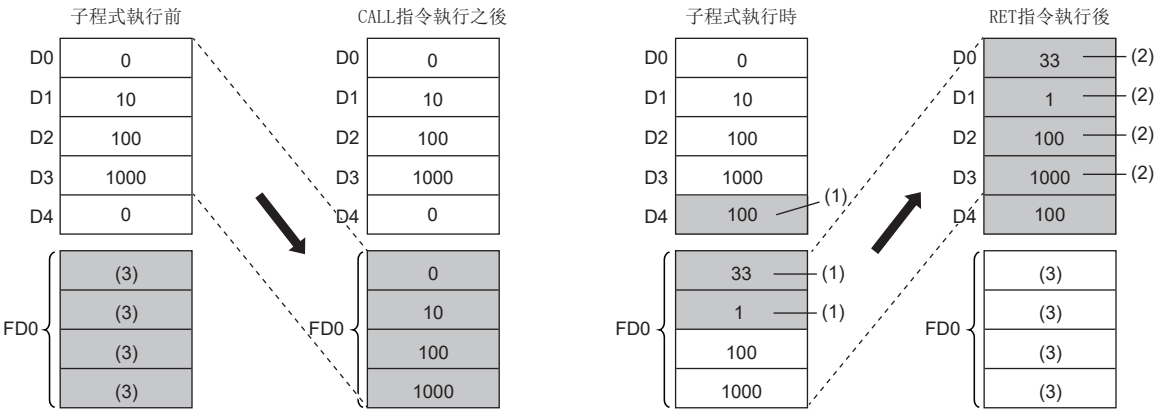
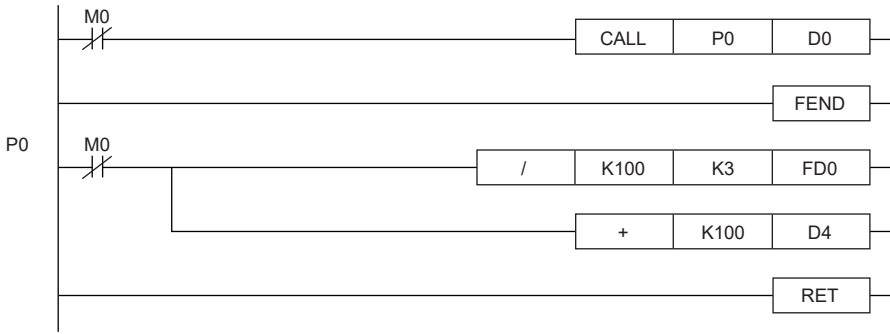
[錯誤的運算] 子程式的FD0中指定D0、子程式中使用D1情况下的動作



- (1): 子程式的執行結果將被儲存。
- (2): 將被替換為函數元件的值。
- (3): D1不反映函數元件的值。
- (4): 將變為不固定。

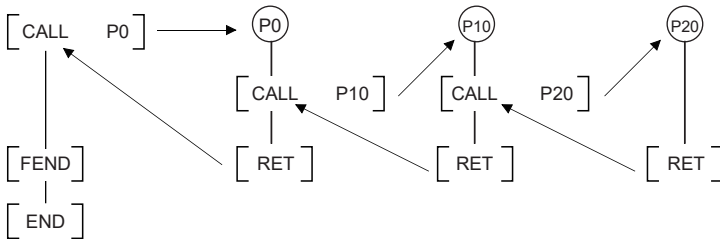
例

[正確的運算]子程式的FD0中指定D0、子程式中使用D4情況下的動作



- (1): 子程式的執行結果將被儲存。
- (2): 將被替換為函數元件的值。
- (3): 將變為不固定。

• CALL (P) 指令的嵌套最多可達16層。但是，嵌套的16層指的是CALL (P) 指令、FCALL (P) 指令、ECALL指令 (P)、EFCALL (P) 指令、XCALL指令中的合計值。



• 對於子程式內建為ON的元件，在子程式非執行時也將被保持。對於執行子程式時設為ON的元件，可以透過FCALL (P) 指令設為OFF。

注意事項

- FBD/LD程式不能建立為子程式。
- 在FBD/LD中，不能執行具有引數的子程式。

出錯

出錯代碼 (SD0)	內容
2820H	(s1)~(s5)中引數中指定的元件無法預留資料容量區域時。
3360H	執行了第17層的嵌套時。
3380H	(P)中指定的指標的子程式不存在時。
3381H	執行CALL (P)指令後，執行RET指令前執行了END指令、FEND指令、GOEND指令、STOP指令時。
3382H	執行CALL (P)指令前執行了RET指令時。

從子程式的返回

RET



表示子程式的結束。

梯形圖	ST
	不對應。

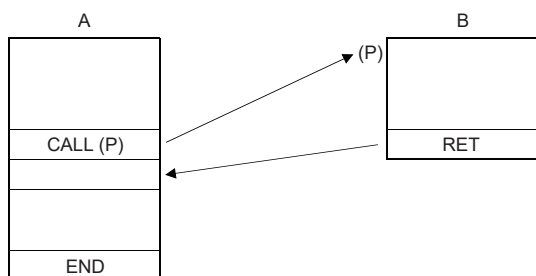
FBD/LD
不對應。

執行條件

指令	執行條件
RET	常時執行

功能

- 表示子程式的結束。
- 執行RET指令時，將返回至調用了子程式的CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令的下一個步處。



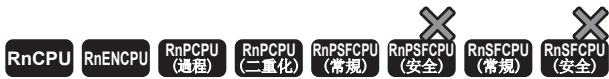
A: 主程式
B: 子程式

出錯

出錯代碼(SD0)	內容
3381H	執行CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令執行後，執行RET指令前執行了END指令、FEND指令、GOEND指令、STOP指令時。
3382H	執行CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令前執行了RET指令時。

子程式的輸出OFF調用

FCALL (P)



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

不進行指定的指標的子程式的執行處理。

梯形圖	ST
	不對應。

FBD/LD

執行條件

指令	執行條件
FCALL	
FCALLP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(P)	子程式的起始指標編號	—	元件名稱	POINTER
(s1) ... (s5)*1	子程式中作為引數傳送的元件編號	-2147483648~2147483647	位元/帶符號BIN16位元 /帶符號BIN32位元	ANY
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在FBD/LD中不能指定。

可使用的元件

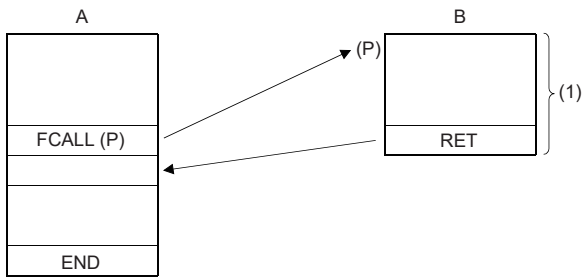
操作數	位元		字元		雙字元			間接指定	常數			其它 (P)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(P)	—	—	—	—	—	—	—	—	—	—	—	○
(s1) ... (s5)	○*1	○	○	○	○	○	○	○	○	—	—	—

*1 F以外可以使用。

功能

- 執行FCALL (P) 指令時，不進行指標 (P) 的子程式的執行處理*2。FCALL (P) 指令可以對同一程式檔案內的指標中指定的子程式及全局指標中指定的子程式進行非執行。

*2 不執行處理時，將進行與將各線圈指令的條件設定為OFF狀態時相同的處理。



A: 主程式

B: 子程式

(1): FCALL (P) 指令的指令由ON→OFF時不執行處理。

- 不執行處理後的各線圈指令的運算結果與條件觸點的ON/OFF無關，其情況如下所示。

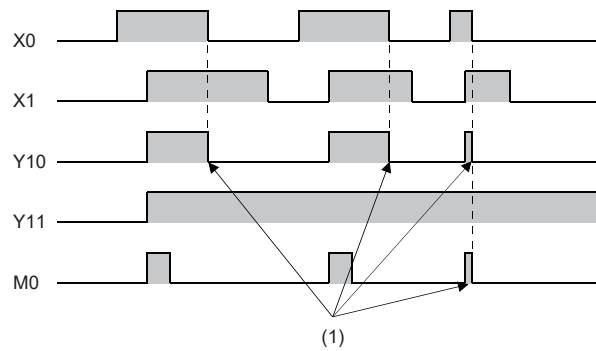
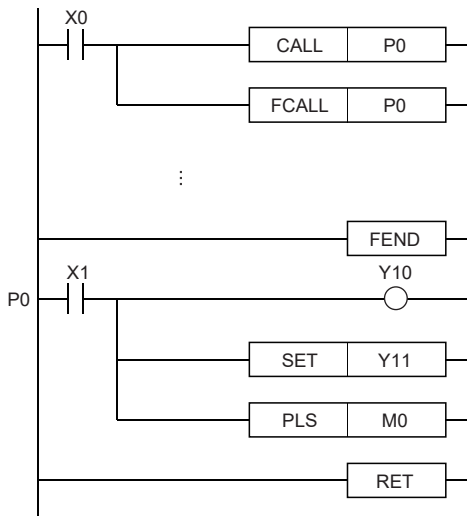
運算中使用的元件	運算結果(元件的狀態)
高速定時器、低速定時器	變為0。
高速累計定時器、低速累計定時器、計數器	保持當前的狀態。
OUT指令中的元件	變為強制OFF。
SET指令、RST指令中的元件、SFT (P) 指令中的元件、基本/應用指令中的元件	保持當前的狀態。
PLS指令、脈衝化指令(□P)	變為與條件觸點OFF相同的處理。

- FCALL (P) 指令與CALL (P) 指令組合使用。在未將FCALL (P) 指令與CALL (P) 指令組合執行的情況下，因為即使將執行指令設為OFF也不進行子程式的不執行處理，因此各線圈指令將保持輸出狀態。

- 因為執行指令變為OFF時不進行子程式的執行處理，因此可以將OUT指令、PLS指令(包括脈衝化指令)強制設為OFF。

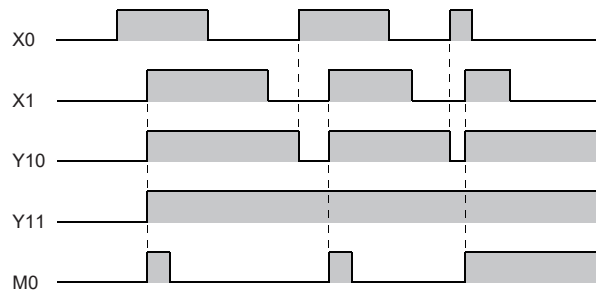
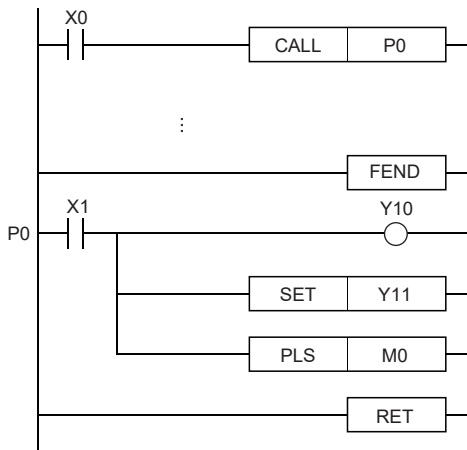
例

使用FCALL(P)指令時

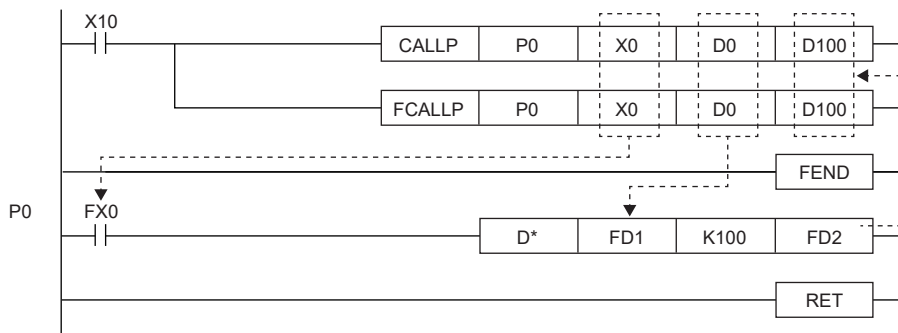


(1): 透過FCALL指令強制OFF

不使用FCALL(P)指令時



- 子程式中使用函數元件(FX、FY、FD)時，在(s1)~(s5)中指定函數元件對應的元件。(s1)~(s5)中指定的元件的內容如下所示。



- 執行子程式前，位元資料的內容將被傳送至FX中，字元資料的內容將被傳送至FD中。
- 執行子程式後，將被傳送到FY、FD的內容對應的元件中。
- FX、FY的函數元件的處理單位為位元單位。FD的函數元件的處理單位為4字元單位。根據引數中指定的元件的類型，可處理的資料容量有所不同。對於函數元件中指定的元件，應預留出資料容量相應的區域。無法預留出資料容量相應的區域的情況下將變為出錯狀態。

• 各函數元件的資料容量如下所示。

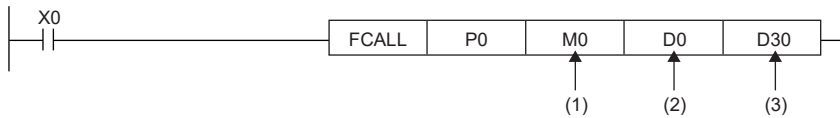
函數元件	使用元件	資料容量
FX、FY	位元元件	1點
	字元元件的位元指定時	1位元
FD	位元元件的位指定時*3	4字元*4
	字元元件	4字元

*3 當位元元件的位指定時，即使(s1)~(s5)中指定的元件編號不是16的倍數的情況下，也不變為出錯狀態。

*4 FD的高位2字元將變為0。

例

指定的元件的內容



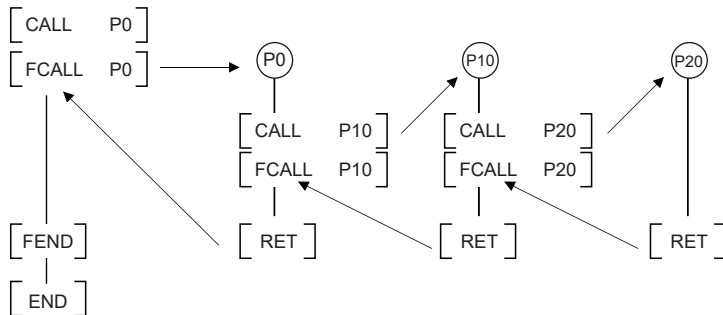
(1) 佔用M0(傳送至FX0)

(2) 佔用D0~D3(傳送至FD1)

(3) 佔用D30~D33(傳送至FD2)

• 在FCALL(P)指令中，可以使用(s1)~(s5)。

• FCALL(P)指令的嵌套最多可達16層。但是，嵌套的16層指的是CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令的合計值。



注意事項

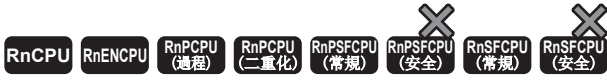
- FBD/LD程式不能建立為子程式。
- 在FBD/LD中，不能執行具有引數的子程式。

出錯

出錯代碼(SD0)	內容
2820H	(s1)~(s5)中引數中指定的元件無法預留資料容量區域時。
3360H	執行了第17層的嵌套時。
3380H	FCALL(P)指令中指定的指標的子程式不存在時。
3381H	執行FCALL(P)指令後，執行RET指令前執行了END指令、FEND指令、GOEND指令、STOP指令時。
3382H	執行FCALL(P)指令前執行了RET指令時。

程式檔案之間子程式調用

ECALL (P)



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(LJMEI SEC iQ-R CPU模組用戶手冊(應用篇))

執行指定程式檔案名稱的指定的指標的子程式。

梯形圖	ST
<p>FILE: 檔案名稱</p>	不對應。

FBD/LD
<p>FILE: 檔案名稱</p>

■執行條件

指令	執行條件
ECALL	
ECALLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(檔案名稱)	調用的程式檔案名稱	—	Unicode字元串	ANYSTRING_DOUBLE
(P)	子程式的起始指標編號	—	元件名稱	POINTER
(s1) ... (s5)*1	子程式中作為引數傳送的元件編號	-2147483648~2147483647	位元/帶符號BIN16位元 /帶符號BIN32位元	ANY
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在FBD/LD中不能指定。

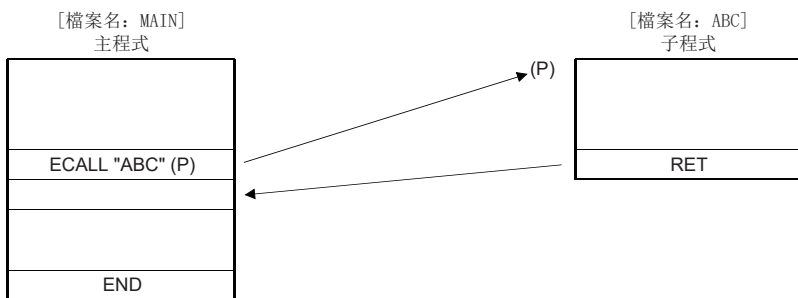
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它 (P)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(檔案名稱)	—	—	○	—	—	—	○	—	—	○	—
(P)	—	—	—	—	—	—	—	—	—	—	○
(s1) ... (s5)	○*1	○	○	○	○	○	○	○	○	—	—

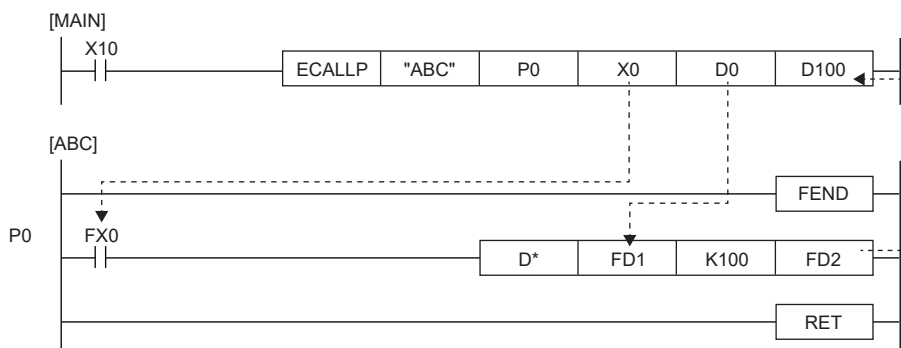
*1 F以外可以使用。

功能

- 執行ECALL (P) 指令時，執行指定程式檔案名的指標 (P) 的子程式。在ECALL (P) 指令中，也可調用使用了其它程式檔案的局部指標的子程式。



- 檔案名稱中只能指定程式記憶體中已儲存的程式檔案。
- 檔案名稱中無需指定副檔名 (. PRG)。(僅 . PRG 檔案作為對象。)
- 子程式中使用函數元件 (FX、FY、FD) 時，在 (s1) ~ (s5) 中指定函數元件對應的元件。(s1) ~ (s5) 中指定的元件的內容如下所示。



- 執行子程式前，位元資料的內容將被傳送至FX中，字元資料的內容將被傳送至FD中。
- 執行子程式後，將被傳送到FY、FD的內容對應的元件中。
- FX、FY的函數元件的處理單位為位元單位。FD的函數元件的處理單位為4字元單位。根據引數中指定的元件的類型，可處理的資料容量有所不同。對於函數元件中指定的元件，應預留出資料容量相應的區域。無法預留出資料容量相應的區域的情況下將變為出錯狀態。
- 各函數元件的資料容量如下所示。

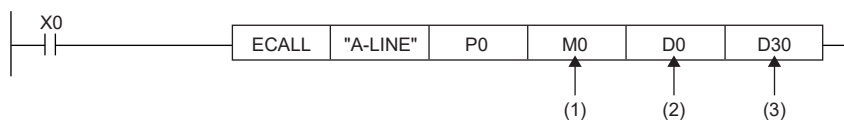
函數元件	使用元件	資料容量
FX、FY	位元元件	1點
	字元元件的位元指定時	1位元
FD	位元元件的位指定時*2	4字元*3
	字元元件	4字元*3

*2 當位元元件的位指定時，即使 (s1) ~ (s5) 中指定的元件編號不是16的倍數的情況下，也不變為出錯狀態。

*3 資料容量根據所使用的指令而變化。

例

指定的元件的內容



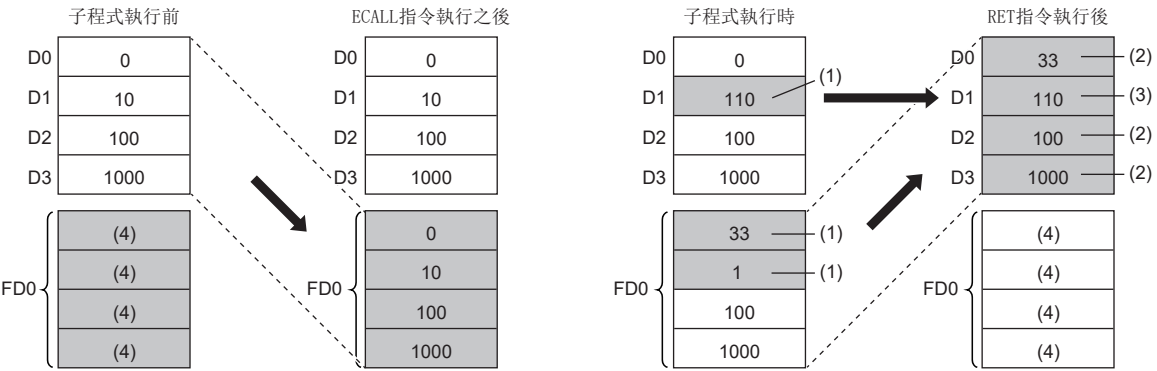
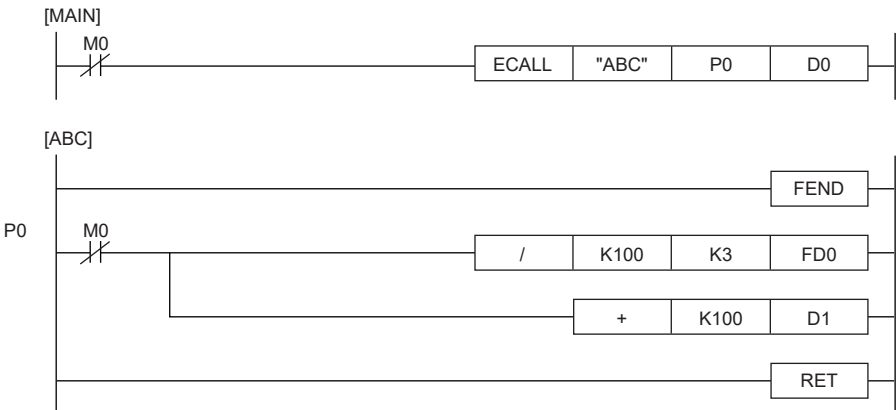
- (1) 佔用M0 (傳送至FX0)
- (2) 佔用D0~D3 (傳送至FD1)
- (3) 佔用D30~D33 (傳送至FD2)

- 在ECALL (P) 指令中，可以使用 (s1) ~ (s5)。

- 應勿將ECALL (P) 指令的引數中使用的元件用於子程式。將ECALL (P) 指令的引數中使用的元件用於子程式的情況下，將無法正常運算。

例

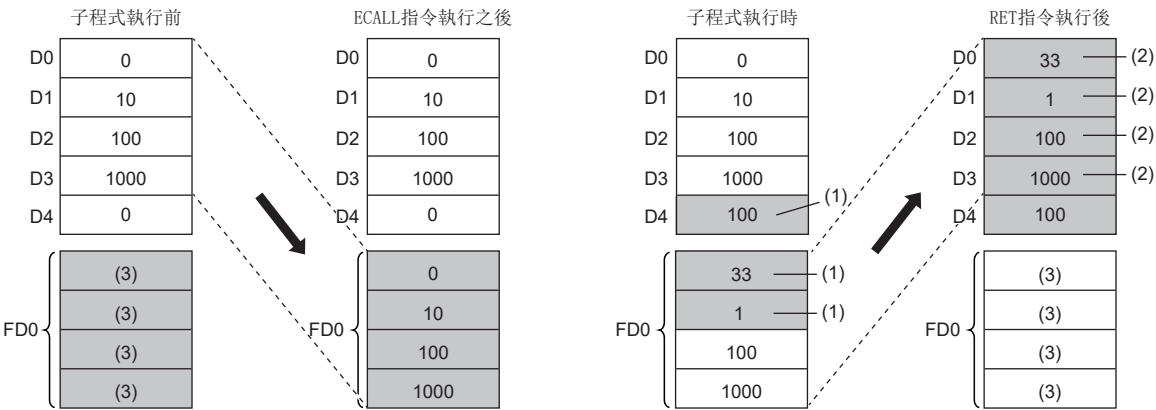
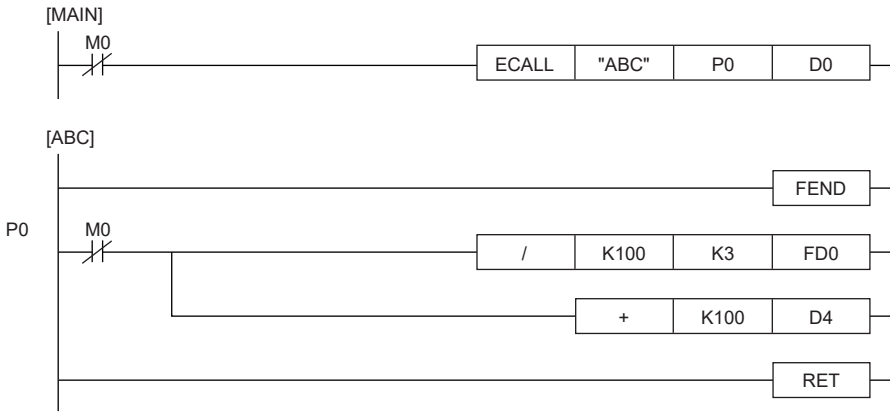
[錯誤的運算] 子程式的FD0中指定D0、子程式中使用D1情況下的動作



- (1): 子程式的執行結果將被儲存。
- (2): 將被替換為函數元件的值。
- (3): D1不反映函數元件的值。
- (4): 將變為不固定。

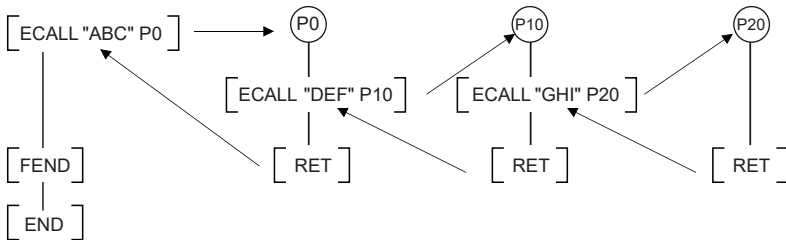
例

[正確的運算]子程式的FD0中指定D0、子程式中使用D4情況下的動作



- (1): 子程式的執行結果將被儲存。
- (2): 將被替換為函數元件的值。
- (3): 將變為不固定。

- ECALL (P) 指令中引數中指定的元件編號應勿重複。在重複的情況下，將無法正常運算。
- ECALL (P) 指令的嵌套最多可達16層。但是，嵌套的16層指的是CALL (P) 指令、FCALL (P) 指令、ECALL (P) 指令、EFCALL (P) 指令、XCALL指令的合計值。



- 對於子程式內建為ON的元件，在子程式非執行時也將被保持。對執行子程式時設為ON的元件，可以透過EFCALL (P) 指令設為OFF。

注意事項

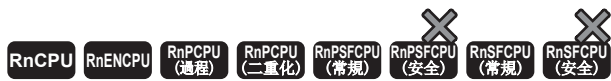
- FBD/LD程式不能建立為子程式。
- 在FBD/LD中，不能執行具有引數的子程式。

出錯

出錯代碼 (SD0)	內容
2820H	(s1) ~ (s5) 中引數中指定的元件無法預留資料容量區域時。
2840H	(檔案名稱) 中指定的檔案不存在時。
2841H	(檔案名稱) 中指定的檔案無法執行時。
3360H	執行了第17層的嵌套時。
3380H	(P) 中指定的指標的子程式不存在時。
3381H	執行ECALL (P) 指令後，執行RET指令前執行了END指令、FEND指令、GOEND指令、STOP指令時。
3382H	執行ECALL (P) 指令前執行了RET指令時。

程式檔案之間子程式輸出OFF調用

EFCALL (P)

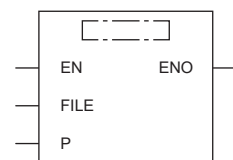


• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(LJMESEC iQ-R CPU模組用戶手冊(應用篇))

不進行指定程式檔案名的指定的指標的子程式的執行處理。

梯形圖	ST
<p>FILE: 檔案名稱</p>	不對應。

FBD/LD



FILE: 檔案名稱

■執行條件

指令	執行條件
EFCALL	
EFCALLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(檔案名稱)	不執行處理的程式檔案名	—	Unicode字元串	ANYSTRING_DOUBLE
(P)	子程式的起始指標編號	—	元件名稱	POINTER
(s1)~(s5)*1	子程式中作為引數傳送的元件編號	-2147483648~2147483647	位元/帶符號BIN16位元 /帶符號BIN32位元	ANY
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在FBD/LD中不能指定。

■可使用的元件

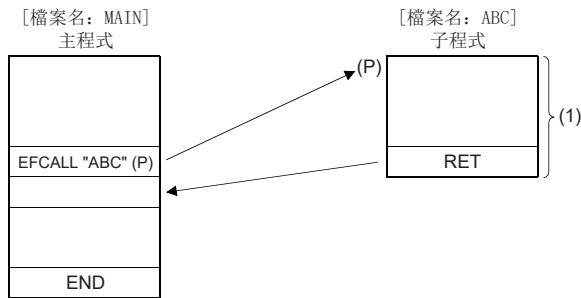
操作數	位元		字元		雙字元			間接指定	常數				其它 (P)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H)G□	Z	LT、 LST、LC	LZ		K	H	E	\$	
(檔案名稱)	—	—	○	—	—	—	—	○	—	—	○	—	—
(P)	—	—	—	—	—	—	—	—	—	—	—	—	○
(s1)~(s5)	○*1	○	○	○	○	○	○	○	○	—	—	—	—

*1 F以外可以使用。

功能

- 當執行EFCALL(P)指令時，不進行(P)中指定的指標的子程式的執行處理*2。在EFCALL(P)指令中，也可調用使用了其它程式檔案的局部指標的子程式。

*2 不執行處理時，進行與將各線圈指令的條件觸點設為OFF狀態時相同的處理。



(1) EFCALL(P)指令的指令由ON→OFF時不執行處理。

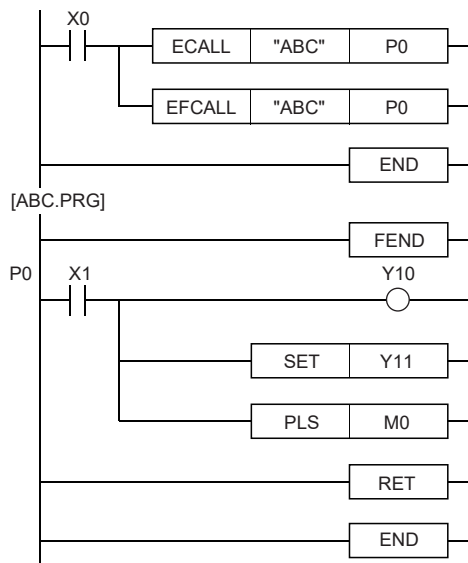
- 不執行處理後的各線圈指令的運算結果與條件觸點的ON/OFF無關，其情況如下所示。

運算中使用的元件	運算結果(元件的狀態)
高速定時器、低速定時器	變為0。
高速累計定時器、低速累計定時器、計數器	保持當前的狀態。
OUT指令中的元件	變為強制OFF。
SET指令、RST指令中的元件、SFT(P)指令中的元件、基本/應用指令中的元件	保持當前的狀態。
PLS指令、脈衝化指令(□P)	變為與條件觸點OFF相同的處理。

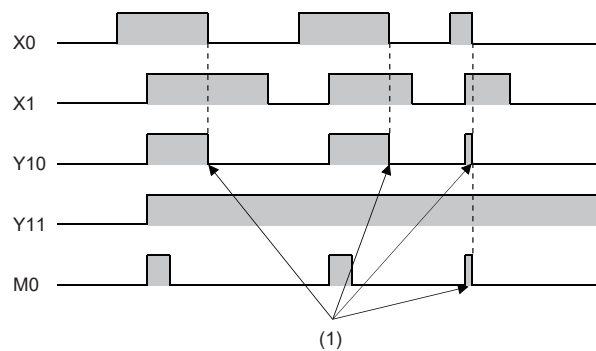
- EFCALL(P)指令與ECALL(P)指令組合使用。在未與ECALL(P)指令組合執行的情況下，因為即使將執行指令設為OFF也不進行子程式的不執行處理，因此各線圈指令將保持輸出狀態。
- 因為執行指令變為OFF時不進行子程式的執行處理，因此可以將OUT指令、PLS指令(包括脈衝化指令)強制設為OFF。

例

使用EFCALL指令時

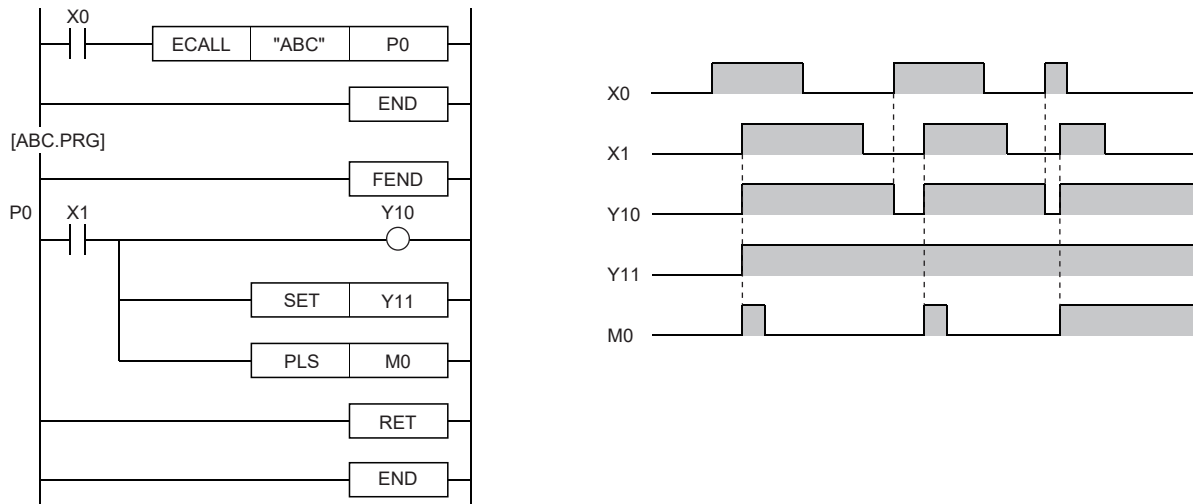


(1): 透過EFCALL指令強制OFF

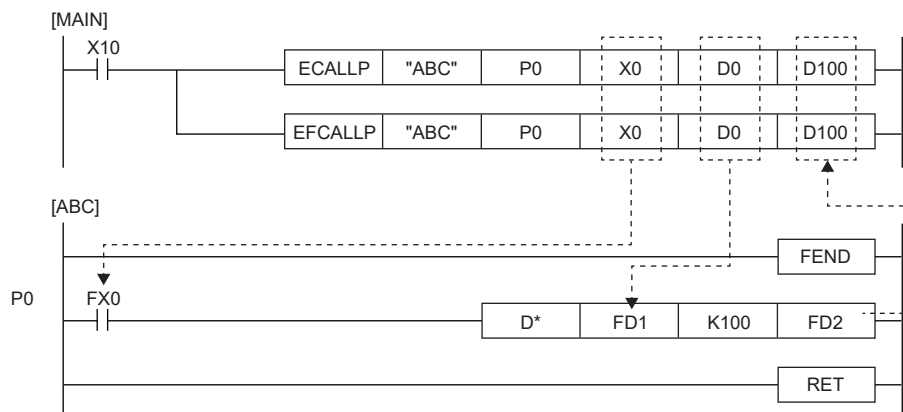


例

不使用EFCALL指令時



- 檔案名稱中只能指定程式記憶體中已儲存的程式檔案。
- 檔案名稱中無需指定副檔名(.PRG)。(僅.PRG檔案作為對象。)
- 子程式中使用函數元件(FX、FY、FD)時，在(s1)～(s5)中指定函數元件對應的元件。(s1)～(s5)中指定的元件的內容如下所示。



- 執行子程式前，位元資料的內容將被傳送至FX中，字元資料的內容將被傳送至FD中。
- 執行子程式後，將被傳送到FY、FD的內容對應的元件中。
- FX、FY的函數元件的處理單位為位元單位。FD的函數元件的處理單位為4字元單位。根據引數中指定的元件的類型，可處理的資料容量有所不同。對於函數元件中指定的元件，應預留出資料容量相應的區域。無法預留出資料容量相應的區域的情況下將變為出錯狀態。

• 各函數元件的資料容量如下所示。

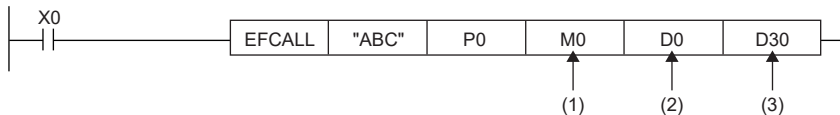
函數元件	使用元件	資料容量
FX、FY	位元元件	1點
	字元元件的位元指定時	1位元
FD	位元元件的位指定時*3	4字元*4
	字元元件	4字元

*3 當位元元件的位指定時，即使(s1)~(s5)中指定的元件編號不是16的倍數的情況下，也不變為出錯狀態。

*4 FD的高位2字元將變為0。

例

指定的元件的內容

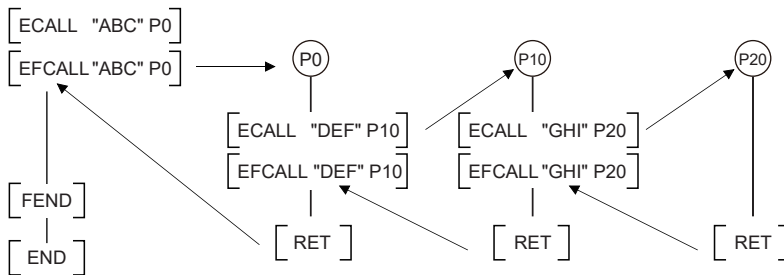


(1) 佔用M0 (傳送至FX0)

(2) 佔用D0~D3 (傳送至FD1)

(3) 佔用D30~D33 (傳送至FD2)

- 在EFCALL(P)指令中，可以使用(s1)~(s5)
- 對於子程式中使用的函數元件數及EFCALL(P)指令的引數，需要設定相同的數。此外，應將函數元件與EFCALL(P)指令的引數的類型設定為相同。
- EFCALL(P)指令的嵌套最多可達16層。但是，嵌套的16層指的是CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令的合計值。



注意事項

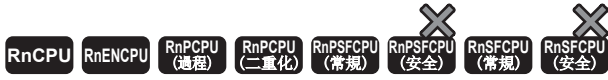
- FBD/LD程式不能建立為子程式。
- 在FBD/LD中，不能執行具有引數的子程式。

出錯

出錯代碼(SD0)	內容
2820H	(s1)~(s5)中引數中指定的元件無法預留資料容量區域時。
2840H	(檔案名稱)中指定的檔案不存在時。
2841H	(檔案名稱)中指定的檔案無法執行時。
3360H	執行了第17層的嵌套時。
3380H	(P)中指定的指標的子程式不存在時。
3381H	執行EFCALL(P)指令後，執行RET指令前執行了END指令、FEND指令、GOEND指令、STOP指令時。
3382H	執行EFCALL(P)指令前執行了RET指令時。

子程式調用

XCALL



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

進行子程式的執行及不執行處理。條件成立時進行子程式CALL，條件由ON→OFF時進行子程式的FCALL。

梯形圖	ST
	不對應。

FBD/LD

■執行條件

指令	執行條件
XCALL	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(P)	子程式的起始指標編號	—	元件名稱	POINTER
(s1) ... (s5)*1	子程式中作為引數傳送的元件編號	-2147483648~2147483647	位元/帶符號BIN16位元 /帶符號BIN32位元	ANY
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在FBD/LD中不能指定。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其它 (P)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(P)	—	—	—	—	—	—	—	—	—	—	—	—	○
(s1) ... (s5)	○*1	○	○	○	○	○	○	○	○	○	—	—	—

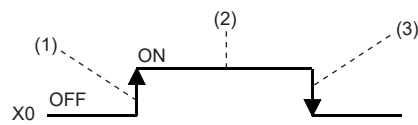
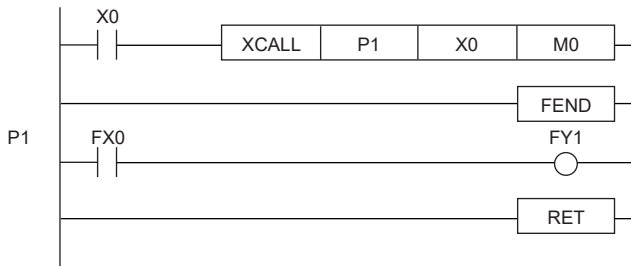
*1 F以外可以使用。

功能

- XCALL指令是進行子程式的執行及不執行處理的指令。
- 子程式的執行時，根據各線圈指令的條件觸點的ON/OFF狀態進行運算。
- 子程式的不執行處理時，進行與將各線圈指令的條件觸點設為OFF狀態時相同的處理。
- 不執行處理後的各線圈指令的運算結果與條件觸點的ON/OFF無關，其情況如下所示。

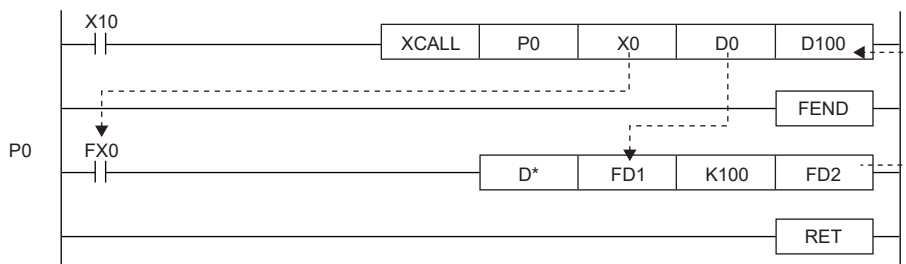
運算中使用的元件	運算結果(元件的狀態)
高速定時器、低速定時器	變為0。
高速累計定時器、低速累計定時器、計數器	保持當前的狀態。
OUT指令中的元件	變為強制OFF。
SET指令、RST指令中的元件、SFT(P)指令中的元件、基本/應用指令中的元件	保持當前的狀態。
PLS指令、脈衝化指令(□P)	變為與條件觸點OFF相同的處理。

- XCALL指令的動作如下所示。



- (1) X0的上升沿(OFF→ON)：執行P1的子程式。
- (2) X0的ON中：執行P1的子程式。(X0的ON中，不包含X0的上升沿。)
- (3) X0的下降沿(ON→OFF)：進行P1的子程式的不執行處理。

- 子程式中使用函數元件 (FX、FY、FD) 時，在 (s1) ~ (s5) 中指定函數元件對應的元件。(s1) ~ (s5) 中指定的元件的內容如下所示。



- 執行子程式前，位元資料的內容將被傳送至FX中，字元資料的內容將被傳送至FD中。
- 執行子程式後，將被傳送到FY、FD的內容對應的元件中。
- FX、FY的函數元件的處理單位為位元單位。FD的函數元件的處理單位為4字元單位。根據引數中指定的元件的類型，可處理的資料容量有所不同。對於函數元件中指定的元件，應預留出資料容量相應的區域。無法預留出資料容量相應的區域的情況下將變為出錯狀態。
- 各函數元件的資料容量如下所示。

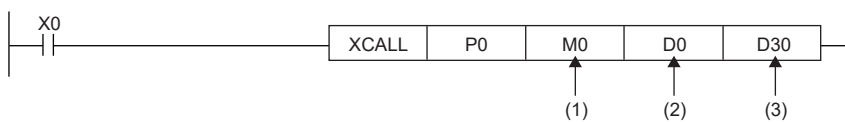
函數元件	使用元件	資料容量
FX、FY	位元元件	1點
	字元元件的位元指定時	1位元
FD	位元元件的位指定時*2	4字元*3
	字元元件	4字元*3

*2 當位元元件的位指定時，即使 (s1) ~ (s5) 中指定的元件編號不是16的倍數的情況下，也不變為出錯狀態。

*3 根據使用的指令而有所不同。

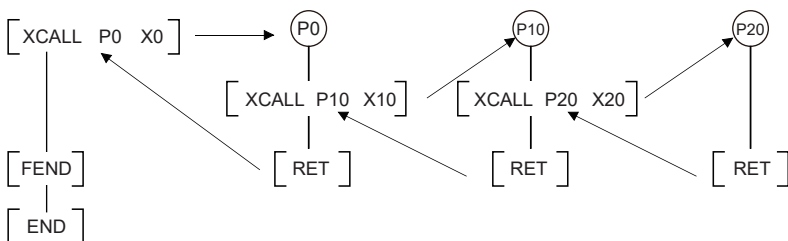
例

指定的元件的內容



- 佔用M0 (傳送至FX0)
- 佔用D0~D3 (傳送至FD1)
- 佔用D30~D33 (傳送至FD2)

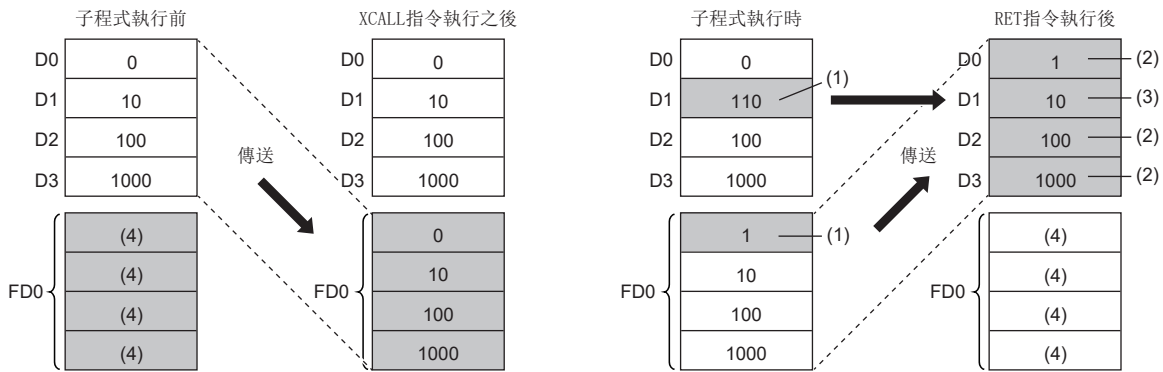
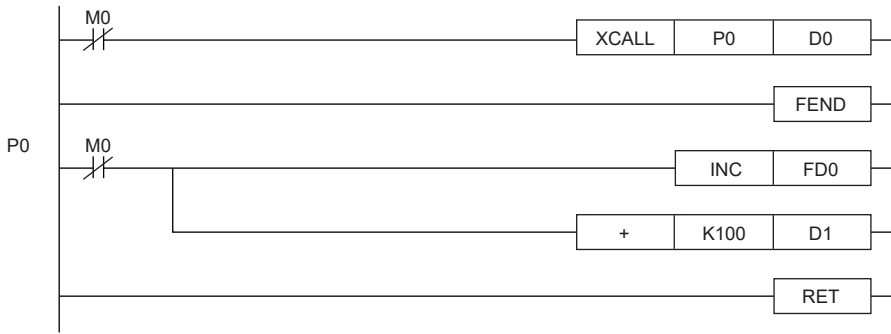
- 在XCALL指令中，可以使用 (s1) ~ (s5)。
- 對於子程式中使用的函數元件數與XCALL指令的引數，需要設定為相同的數。此外，應將函數元件與XCALL指令的引數的類型設定為相同。
- XCALL指令中引數中指定的元件編號應避免重複。在重複的情況下，將無法正常運算。
- XCALL指令的嵌套最多可達16層。但是，嵌套的16層指的是CALL (P) 指令、FCALL (P) 指令、ECALL (P) 指令、EFCALL (P) 指令、XCALL指令的合計值。



- 應勿將XCALL指令的引數中使用的元件用於子程式。將XCALL指令的引數中使用的元件用於子程式的情況下，將無法正常運算。

例

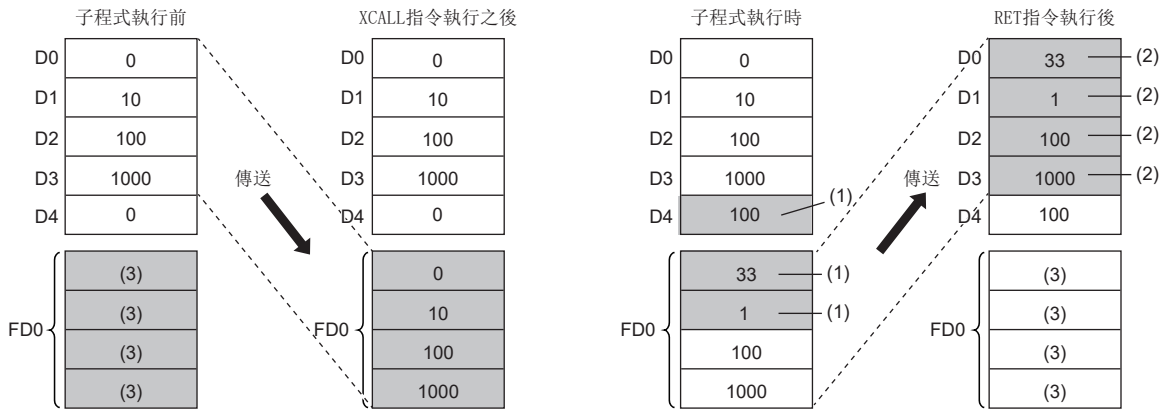
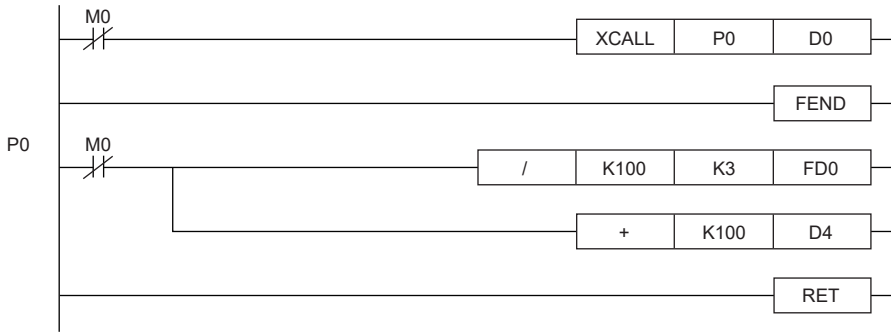
[錯誤的運算]子程式的FD0中指定D0、子程式中使用D1情況下的動作



- (1): 子程式的執行結果將被儲存。
- (2): 將被替換為函數元件的值。
- (3): D1不反映函數元件的值。
- (4): 將變為不固定。

例

[正確的運算]子程式的FD0中指定D0、子程式中使用D4情況下的動作



- (1): 子程式的執行結果將被儲存。
- (2): 將被替換為函數元件的值。
- (3): 將變為不固定。

注意事項

- FBD/LD程式不能建立為子程式。
- 在FBD/LD中，不能執行具有引數的子程式。

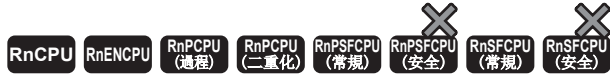
出錯

出錯代碼 (SD0)	內容
2820H	(s1)~(s5)中引數中指定的元件無法預留資料容量區域時。
3360H	執行了第17層的嵌套時。
3380H	(P)中指定的指標的子程式不存在時。
3381H	執行XCALL指令後，執行RET指令前執行了END指令、FEND指令、GOEND指令、STOP指令時。
3382H	執行XCALL指令前執行了RET指令時。

7.4 程式控制用指令

程式待機

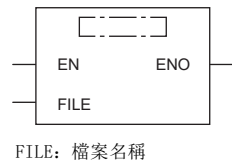
PSTOP (P)



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用時，在系統切換時將不交接執行類型。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
將指定的元件中儲存的檔案名的程式執行類型設為待機類型。

梯形圖	ST
<p>FILE: 檔案名稱</p>	ENO:=PSTOP (EN, 檔案名); ENO:=PSTOPP (EN, 檔案名);

FBD/LD



執行條件

指令	執行條件
PSTOP	
PSTOPP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(檔案名稱)	設為待機狀態的程式的檔案名稱的字元串資料、或儲存了字元串資料的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(檔案名稱)	—	—	○	—	—	—	○	—	—	○	—	—

功能

- 將(檔案名稱)中指定的元件中儲存的檔案名稱的程式設為待機狀態。
- 只能將程式記憶體中儲存的程式設為待機類型。
- 指定的程式在END處理中將變為待機狀態。
- 即使在參數中指定了執行形式的情況下PSTOP (P)指令也將優先。
- 檔案名稱中無需指定副檔名(.PRG)。(僅.PRG檔案作為對象。)

出錯

出錯代碼 (SD0)	內容
2840H	(檔案名稱) 中指定的檔案名稱的程式不存在時。
2841H	(檔案名稱) 中指定的檔案名稱的程式未登錄到參數中時。
2842H	(檔案名稱) 中指定的檔案名稱的程式類型為SFC程式時。

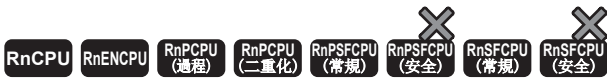
要點

關於程式執行類型的更改，請參閱下述手冊。

 MELSEC iQ-R CPU模組用戶手冊(應用篇)

程式輸出OFF待機

POFF (P)



- 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用時，在系統切換時將不交接執行類型。(□MELSEC iQ-R CPU模組用戶手冊(應用篇))
- 無法在過程CPU(二重化模式)的SFC程式中使用。

更改指定的元件中儲存的檔案名的程式執行類型。

梯形圖	ST
<p>FILE: 檔案名稱</p>	ENO:=POFF (EN, 檔案名稱); ENO:=POFFP (EN, 檔案名稱);

FBD/LD
<p>FILE: 檔案名稱</p>

■執行條件

指令	執行條件
POFF	
POFFP	

7

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(檔案名稱)	將輸出設為OFF後設為待機類型的程式的檔案名稱、或儲存了檔案名稱的元件	—	Unicode字元串	ANYSTRING_DOUBLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(檔案名稱)	—	—	○	—	—	—	—	○	—	—	○	—	

功能

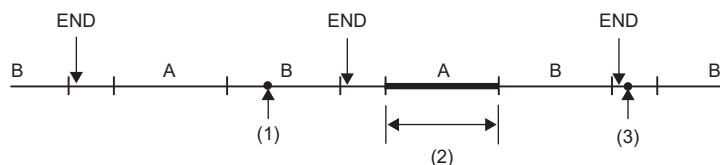
- 更改(檔案名稱)中指定的元件中儲存的檔案名稱的程式的執行類型。掃描執行類型的情況下，在下一個掃描中將輸出設為OFF(不執行處理)。在下一個掃描以後，將變為待機類型。恆定週期執行類型、事件執行類型的情況下，在下一個掃描中將變為掃描執行類型，將輸出設為OFF(不執行處理)。在下一個掃描以後，將變為待機類型。
- 只能將程式記憶體中儲存的程式設為待機類型。
- 即使在參數中指定了執行形式的情況下POFF (P) 指令也將優先。
- 檔案名稱中無需指定副檔名(. PRG)。(僅. PRG檔案作為對象。)

出錯

出錯代碼 (SD0)	內容
1BB0H	(檔案名稱) 中指定的檔案名稱的程式為執行掃描類型的SFC程式時。(僅過程CPU(二重化模式))
2840H	(檔案名稱) 中指定的檔案名稱的程式不存在時。
2841H	(檔案名稱) 中指定的檔案名稱的程式未登錄到參數中時。
2842H	(檔案名稱) 中指定的檔案名稱的程式類型為不支援的程式時。

要點

不執行處理時，將進行與將各線圈指令的條件設定設為OFF狀態時相同的處理。



A、B：執行程式名

(1) 指定“A”後執行POFF指令

(2) “A”不執行處理

(3) 將“A”切換為待機類型程式。

不執行處理後的各線圈指令的運算結果與條件觸點的ON/OFF無關，其情況如下所示。

- OUT指令：強制設為OFF
- SET指令、RST指令、SFT指令、基本指令、應用指令：保持狀態
- PLS指令、PLS化指令(□P)：與條件觸點OFF相同的處理
- OUT T指令：低速/高速定時器的當前值為0
- OUT ST指令、OUT C指令：當前值被保持

關於程式執行類型的更改，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

關於指定SFC程式時的動作，請參閱下述手冊。

📖 MELSEC iQ-R 程式手冊(程式設計篇)

程式掃描執行登錄

PSCAN(P)



- 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用時，在系統切換時將不交接執行類型。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
- 無法在過程CPU(二重化模式)的SFC程式中使用。

將指定的元件中儲存的檔案名的程式執行類型設為掃描執行類型。

梯形圖	ST
<p>FILE: 檔案名稱</p>	ENO:=PSCAN(EN, 檔案名稱); ENO:=PSCANP(EN, 檔案名稱);

FBD/LD
<p>FILE: 檔案名稱</p>

執行條件

指令	執行條件
PSCAN	
PSCANP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(檔案名稱)	設為掃描執行類型的程式的檔案名稱、或儲存了檔案名稱的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

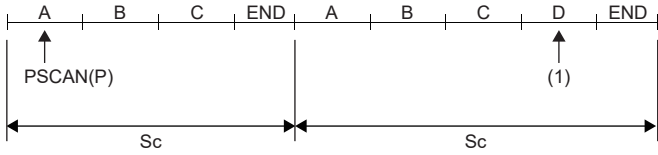
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(檔案名稱)	—	—	○	—	—	—	○	—	—	○	—	—

功能

- 將(檔案名稱)中指定的元件中儲存的檔案名的程式設為掃描執行類型。
- 只能將程式記憶體中儲存的程式設為掃描執行類型。
- 指定的程式在END處理中變為掃描執行類型行。

例

存在程式A、B、C，在程式A內對程式D執行了PSCAN(P)指令的情況下。



Sc: 掃描

(1): 執行D。

- 即使在參數中指定了執行形式的情況下PSCAN(P)指令也將優先。
- 檔案名稱中無需指定副檔名(.PRG)。(僅.PRG檔案作為對象。)

出錯

出錯代碼(SD0)	內容
1BB0H	(檔案名稱)中指定的檔案名稱的程式為待機類型的SFC程式時。(僅過程CPU(二重化模式))
2840H	(檔案名稱)中指定的檔案名稱的程式不存在時。
2841H	(檔案名稱)中指定的檔案名稱的程式未登錄到參數中時。
2842H	(檔案名稱)中指定的檔案名稱的程式類型為不支援的程式時。
3204H	(檔案名稱)中指定的檔案名稱為SFC程式且其他檔案名稱的SFC程式已經啟動的情況。
3424H	(檔案名稱)中指定的檔案名稱為SFC程式且其他檔案名稱的SFC程式已經啟動的情況。

要點

關於程式執行類型的更改，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

關於指定SFC程式時的動作，請參閱下述手冊。

📖 MELSEC iQ-R 程式手冊(程式設計篇)

8 資料處理

8.1 旋轉指令

BIN16位元資料的右旋轉

ROR(P)、RCR(P)



- ROR(P)：將BIN16位元資料在不包含進位標誌的狀況下進行(n)位元右旋轉。
- RCR(P)：將BIN16位元資料在包含進位標誌的狀況下進行(n)位元右旋轉。

梯形圖	ST*1
	ENO:=RORP(EN, n, d); ENO:=RCR(EN, n, d); ENO:=RCRP(EN, n, d);

FBD/LD*1

*1 在ST、FBD/LD中不支援ROR指令。應使用通用函數的ROR。
 ☞ 1422頁 ROR(_E)

■執行條件

指令	執行條件
ROR RCR	
RORP RCRP	

設置數據

■內容、範圍、資料類型

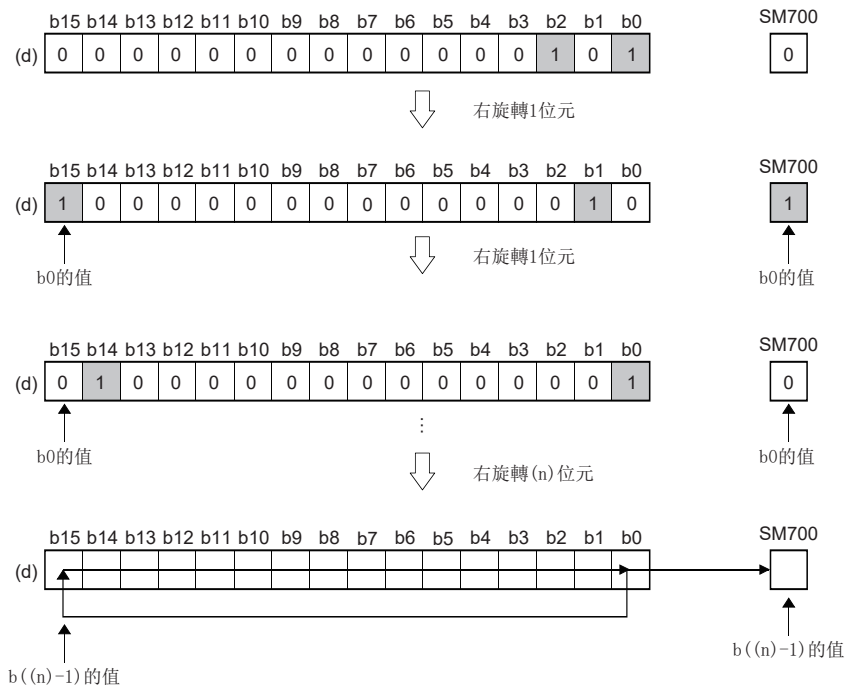
操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	旋轉的元件	—	帶符號BIN16位元	ANY16
(n)	旋轉的位元數	0~15	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

■ROR(P)

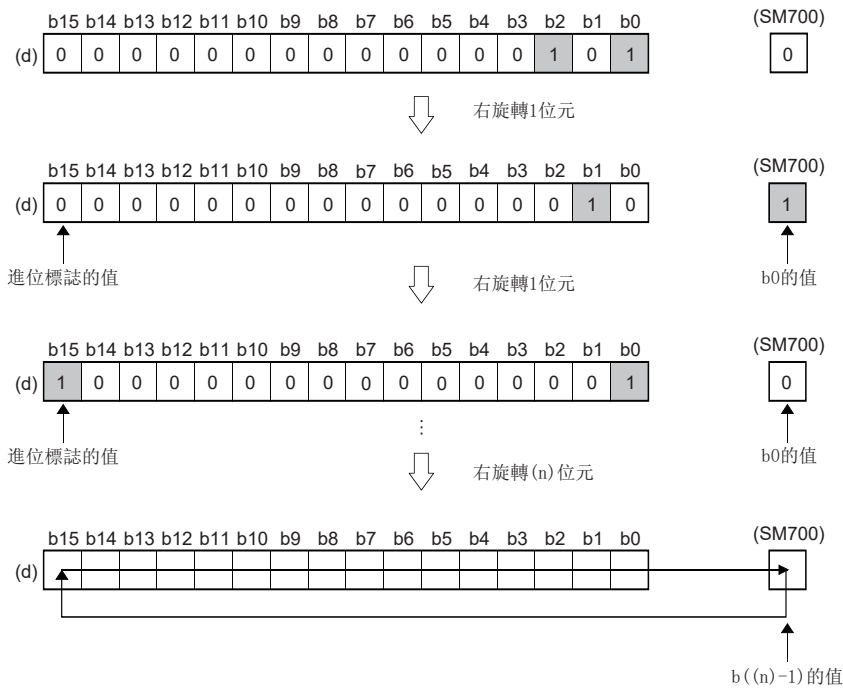
- 將(d)中指定的元件的BIN16位元資料，在不包含SM700(進位標誌)的狀況下進行(n)位元右旋轉。SM700(進位標誌)根據ROR(P)指令執行前的狀態而處於ON或OFF狀態。



- 在(d)中指定了位元元件的情況下，以位指定中指定的元件範圍進行旋轉。此時實際旋轉的位元數將變為(n)÷(位指定中指定的點數)的餘數。例如，(n)=15、(位指定中指定的點數)=12位元時，因為15÷12=1餘3，因此進行3位元右旋轉。
- (n)以0~15進行指定。在(n)中指定16及其以上的值的情況下，以(n)÷16的餘數值進行旋轉。例如(n)=18時，因為18÷16=1餘2，因此進行2位元右旋轉。

■RCR(P)

- 將(d)中指定的元件的BIN16位元資料，在包含SM700(進位標誌)的狀況下進行(n)位元右旋轉。SM700(進位標誌)根據RCR(P)指令執行前的狀態而處於ON或OFF狀態。



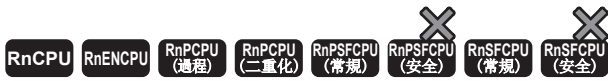
- 在(d)中指定了位元元件的情況下，以位指定中指定的元件範圍進行旋轉。此時實際旋轉的位元數將變為 $(n) \div (\text{位指定中指定的點數})$ 的餘數。例如， $(n)=15$ 、 $(\text{位指定中指定的點數})=12$ 位元時，因為 $15 \div 12=1$ 餘3，因此進行3位元右旋轉。
- (n) 以0~15進行指定。在 (n) 中指定16及其以上的值的情況下，以 $(n) \div 16$ 的餘數值進行旋轉。例如 $(n)=18$ 時，因為 $18 \div 16=1$ 餘2，因此進行2位元右旋轉。

出錯

沒有運算出錯。

BIN16位元資料的左旋轉

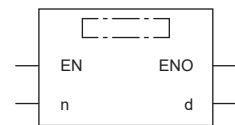
ROL (P)、RCL (P)



- ROL (P)：將BIN16位元資料在不包含進位標誌的狀況下進行 (n) 位元左旋轉。
- RCL (P)：將BIN16位元資料在包含進位標誌的狀況下進行 (n) 位元左旋轉。

梯形圖	ST*1
	ENO:=ROL (EN, n, d) ; ENO:=RCL (EN, n, d) ; ENO:=RCLP (EN, n, d) ;

FBD/LD*1



*1 在ST、FBD/LD中不支援ROL指令。應使用通用函數的ROL。
 ☞ 1420頁 ROL (E)

■執行條件

指令	執行條件
ROL RCL	
ROLP RCLP	

設置數據

■內容、範圍、資料類型

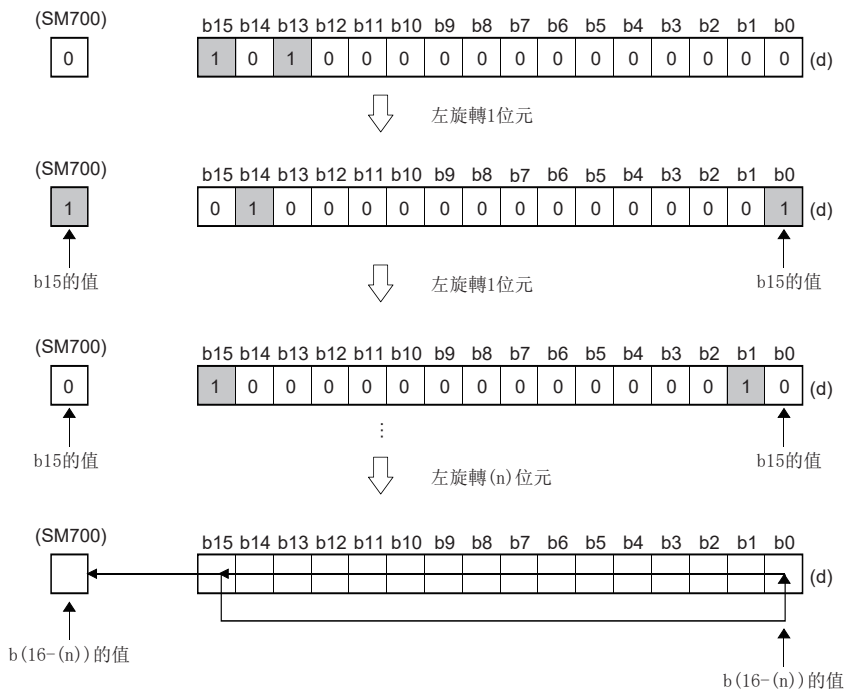
操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	旋轉的元件	—	帶符號BIN16位元	ANY16
(n)	旋轉的位元數	0~15	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(d)	○	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

■ROL (P)

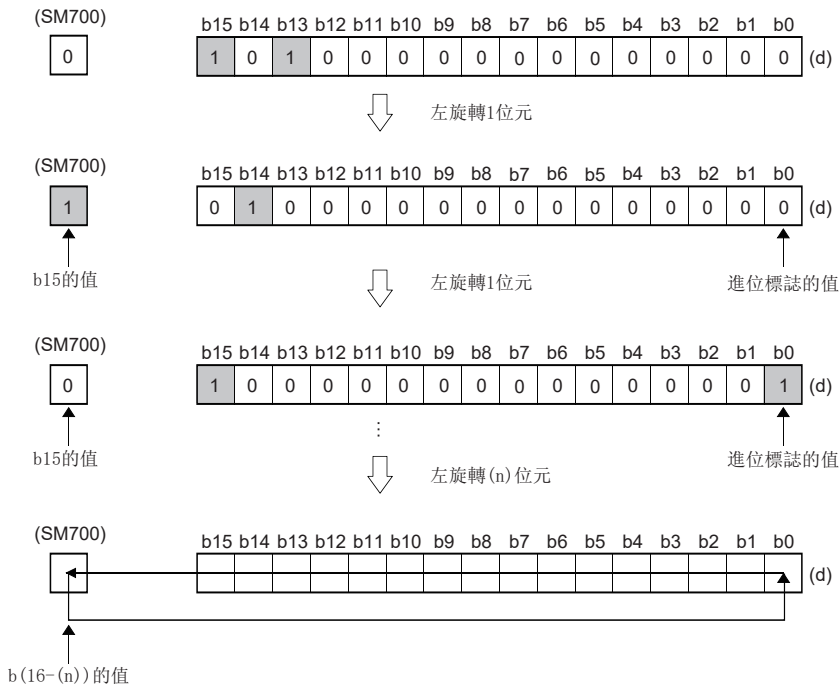
- 將 (d) 中指定的元件的BIN16位元資料，在不包含SM700(進位標誌)的狀況下進行 (n) 位元左旋轉。SM700(進位標誌)根據ROL(P)指令執行前的狀態而處於ON或OFF狀態。



- 在 (d) 中指定了位元元件的情況下，以位指定中指定的元件範圍進行旋轉。此時實際旋轉的位元數將變為 $(n) \div (\text{位指定中指定的點數})$ 的餘數。例如， $(n)=15$ 、(位指定中指定的點數)=12位元時，因為 $15 \div 12=1$ 餘3，因此進行3位元左旋轉。
- (n) 以 $0 \sim 15$ 進行指定。在 (n) 中指定16及其以上的值的情況下，以 $(n) \div 16$ 的餘數值進行旋轉。例如 $(n)=18$ 時，因為 $18 \div 16=1$ 餘2，因此進行2位元左旋轉。

■RCL (P)

- 將(d)中指定的元件的BIN16位元資料，在包含SM700(進位標誌)的狀況下進行(n)位元左旋轉。SM700(進位標誌)根據RCL(P)指令執行前的狀態而處於ON或OFF狀態。



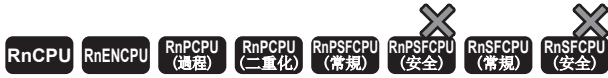
- 在(d)中指定了位元元件的情況下，以位指定中指定的元件範圍進行旋轉。此時實際旋轉的位元數將變為 $(n) \div (\text{位指定中指定的點數})$ 的餘數。例如， $(n)=15$ 、 $(\text{位指定中指定的點數})=12$ 位元時，因為 $15 \div 12=1$ 餘3，因此進行3位元左旋轉。
- (n) 以0~15進行指定。在 (n) 中指定16及其以上的值的情況下，以 $(n) \div 16$ 的餘數值進行旋轉。例如 $(n)=18$ 時，因為 $18 \div 16=1$ 餘2，因此進行2位元左旋轉。

出錯

沒有運算出錯。

BIN32位元資料的右旋轉

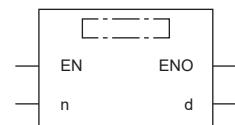
DROR(P)、DRCR(P)



- DROR(P)：將BIN32位元資料在不包含進位標誌的狀況下進行(n)位元右旋轉。
- DRCR(P)：將BIN32位元資料在包含進位標誌的狀況下進行(n)位元右旋轉。

梯形圖	ST*1
	ENO:=DRORP(EN, n, d); ENO:=DRCR(EN, n, d); ENO:=DRCRP(EN, n, d);

FBD/LD*1



*1 在ST、FBD/LD中不支援DROR指令。應使用通用函數的ROR。
 ↖ 1422頁 ROR(E)

執行條件

指令	執行條件
DROR DRCR	
DRORP DRCRP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	旋轉的起始元件	—	帶符號BIN32位元	ANY32
(n)	旋轉的位元數	0~31	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(d)	○	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

■DROR(P)

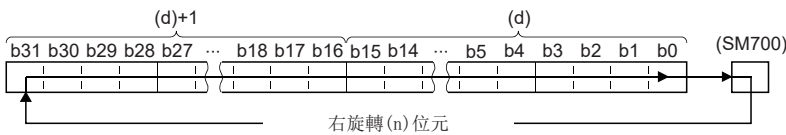
- 將(d)中指定的元件的BIN32位元資料，在不包含SM700(進位標誌)的狀況下進行(n)位元右旋轉。SM700(進位標誌)根據DROR(P)指令執行前的狀態而處於ON或OFF狀態。



- 在(d)中指定了位元元件的情況下，以位指定中指定的元件範圍進行旋轉。此時實際旋轉的位元數將變為 $(n) \div (\text{位指定中指定的點數})$ 的餘數。例如， $(n)=31$ 、(位指定中指定的點數)=24位元時，因為 $31 \div 24=1$ 餘7，因此進行7位元右旋轉。
- (n)以0~31進行指定。在(n)中指定了32及其以上的值的情況下，以 $(n) \div 32$ 的餘數值進行旋轉。例如 $(n)=34$ 時，因為 $34 \div 32=1$ 餘2，因此進行2位元右旋轉。

■DRCR(P)

- 將(d)中指定的元件的BIN32位元資料，在包含SM700(進位標誌)的狀況下進行(n)位元右旋轉。SM700(進位標誌)根據DRCR(P)指令執行前的狀態而處於ON或OFF狀態。



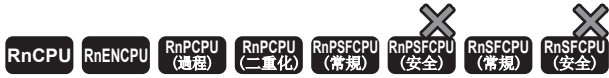
- 在(d)中指定了位元元件的情況下，以位指定中指定的元件範圍進行旋轉。此時實際旋轉的位元數將變為 $(n) \div (\text{位指定中指定的點數})$ 的餘數。例如， $(n)=31$ 、(位指定中指定的點數)=24位元時，因為 $31 \div 24=1$ 餘7，因此進行7位元右旋轉。
- (n)以0~31進行指定。在(n)中指定了32及其以上的值的情況下，以 $(n) \div 32$ 的餘數值進行旋轉。例如 $(n)=34$ 時，因為 $34 \div 32=1$ 餘2，因此進行2位元右旋轉。

出錯

沒有運算出錯。

BIN32位元資料的左旋轉

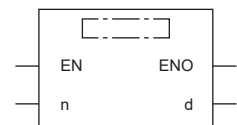
DROL (P)、DRCL (P)



- DROL (P)：將BIN32位元資料在不包含進位標誌的狀況下進行 (n) 位元左旋轉。
- DRCL (P)：將BIN32位元資料在包含進位標誌的狀況下進行 (n) 位元左旋轉。

梯形圖	ST*1
	ENO:=DROL (EN, n, d) ; ENO:=DRCL (EN, n, d) ; ENO:=DRCLP (EN, n, d) ;

FBD/LD*1



*1 在ST、FBD/LD中不支援DROL指令。應使用通用函數的ROL。
 ↖ 1420頁 ROL (E)

■執行條件

指令	執行條件
DROL DRCL	
DROLP DRCLP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	旋轉的起始元件	—	帶符號BIN32位元	ANY32
(n)	旋轉的位元數	0~31	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(d)	○	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

■DROL(P)

- 將(d)中指定的元件的BIN32位元資料，在不包含SM700(進位標誌)的狀況下進行(n)位元左旋轉。SM700(進位標誌)根據DROL(P)指令執行前的狀態而處於ON或OFF狀態。



- 在(d)中指定了位元元件的情況下，以位指定中指定的元件範圍進行旋轉。此時實際旋轉的位元數將變為 $(n) \div (\text{位指定中指定的點數})$ 的餘數。例如， $(n)=31$ 、(位指定中指定的點數)=24位元時，因為 $31 \div 24 = 1$ 餘7，因此進行7位元左旋轉。
- (n)以0~31進行指定。在(n)中指定了32及其以上的值的情況下，以 $(n) \div 32$ 的餘數值進行旋轉。例如 $(n)=34$ 時，因為 $34 \div 32 = 1$ 餘2，因此進行2位元左旋轉。

■DRCL(P)

- 將(d)中指定的元件的BIN32位元資料，在包含SM700(進位標誌)的狀況下進行(n)位元左旋轉。SM700(進位標誌)根據DRCL(P)指令執行前的狀態而處於ON或OFF狀態。



- 在(d)中指定了位元元件的情況下，以位指定中指定的元件範圍進行旋轉。此時實際旋轉的位元數將變為 $(n) \div (\text{位指定中指定的點數})$ 的餘數。例如， $(n)=31$ 、(位指定中指定的點數)=24位元時，因為 $31 \div 24 = 1$ 餘7，因此進行7位元左旋轉。
- (n)以0~31進行指定。在(n)中指定了32及其以上的值的情況下，以 $(n) \div 32$ 的餘數值進行旋轉。例如 $(n)=34$ 時，因為 $34 \div 32 = 1$ 餘2，因此進行2位元左旋轉。

出錯

沒有運算出錯。

8.2 資料表操作指令

從資料表的先輸入資料的讀取

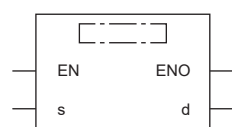
FIFR(P)



將表中最先儲存的資料儲存到指定的元件中。

梯形圖	ST
	ENO:=FIFR(EN, s, d); ENO:=FIFRP(EN, s, d);

FBD/LD



■執行條件

指令	執行條件
FIFR	
FIFRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存從表中讀取的資料的元件	—	帶符號BIN16位元	ANY16
(d)	表的起始元件	—	字元	ANY16* ¹
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

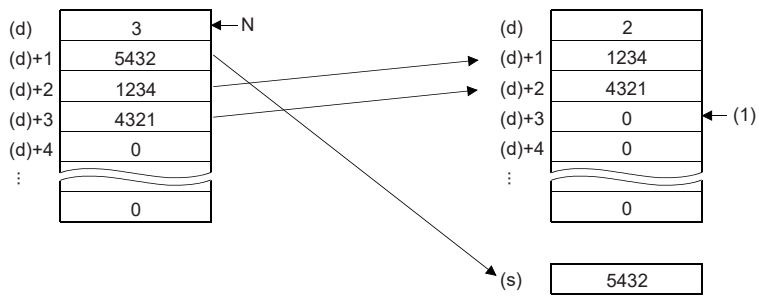
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將 (d) 中指定的表的最先輸入資料 (d) +1 儲存在 (s) 中指定的元件中。執行 FIFR(P) 指令後資料表的資料將逐個向前填充對齊。



N: 資料數

(1): 儲存為0。

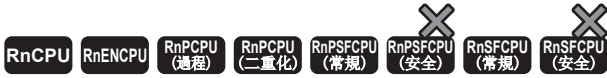
- 當 (d) 中儲存的值為0時，不應執行 FIFR(P) 指令的互鎖。
- 資料數 (d) 可儲存的範圍為 0~65535。

出錯

出錯代碼 (SD0)	內容
3405H	當 (d) 的值為0的狀況下執行 FIFR(P) 指令時。

從資料表的後輸入資料的讀取

FPOP (P)



將表的最後處儲存的資料儲存到指定的元件中。

梯形圖	ST
	ENO:=FPOP (EN, s, d) ; ENO:=FPOPP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
FPOP	
FPOPP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存從表中讀取的資料的元件	—	帶符號BIN16位元	ANY16
(d)	表的起始元件	—	字元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

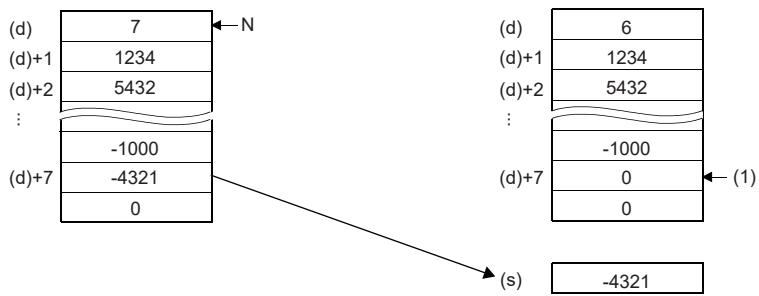
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	○	○	○	○	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 將 (d) 中指定的表的最後處儲存的資料儲存到 (s) 中指定的元件中。執行 FPOP (P) 指令後，儲存了透過 FPOP (P) 指令讀取的資料的元件將變為 0。



N: 資料數

(1): 儲存為 0。

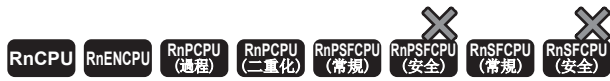
- 當 (d) 中儲存的值為 0 時，不應執行 FPOP (P) 指令的互鎖。
- 資料數 (d) 可儲存的範圍為 0~65535。

出錯

出錯代碼 (SD0)	內容
3405H	當 (d) 的值為 0 的狀況下執行了 FPOP (P) 指令時。

至資料表的資料寫入

FIFW(P)



將BIN16位元資料儲存到指定的資料表中。

梯形圖	ST
	ENO:=FIFW (EN, s, d) ; ENO:=FIFWP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
FIFW	
FIFWP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	寫入至表中的資料或儲存了資料的元件編號	—	帶符號BIN16位元	ANY16
(d)	表的起始元件	—	字元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

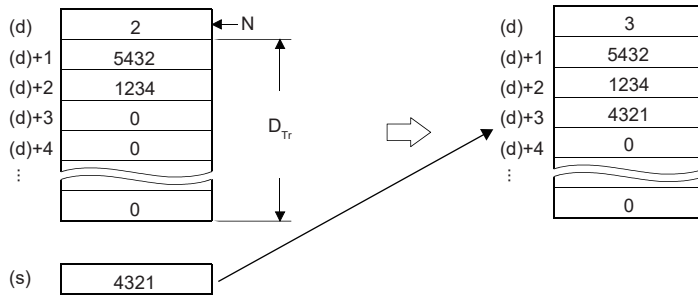
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	○	○	○	○	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 將 (s) 中指定的BIN16位元資料儲存到 (d) 中指定的資料表中。(d) 中儲存了表中儲存的資料數後，(d)+1 及其以後將依次儲存 (s) 中指定的資料。



N: 資料數

D_{tr} : 資料表範圍(由用戶管理)

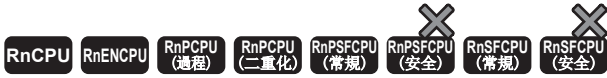
- 首次執行FIFW(P)指令的情況下，應將 (d) 中指定的元件的值清除。
- 因為 (d)+1 及其以後將依次儲存資料，因此應注意資料表的範圍。
- 資料數 (d) 可儲存的範圍為0~65535。

出錯

出錯代碼 (SD0)	內容
3405H	(d) 的值為FFFFH的狀況下執行了FIFW(P)指令時。

資料表的資料插入

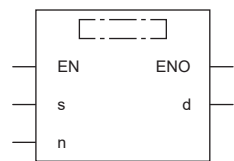
FINS (P)



將BIN16位元資料插入到指定的資料表的第(n)號。

梯形圖	ST
	ENO: =FINS (EN, s, n, d) ; ENO: =FINS (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
FINS	
FINS (P)	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存插入資料的元件	—	帶符號BIN16位元	ANY16
(d)	表的起始元件	—	字元	ANY16*1
(n)	插入的表位置	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

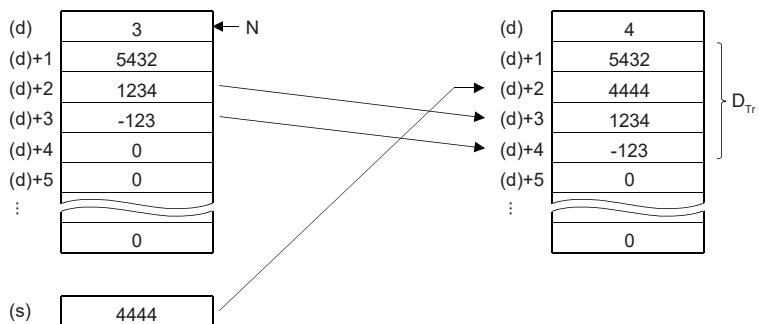
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 將(s)中指定的BIN16位元資料插入到(d)中指定的資料表的第(n)號中。執行FINS(P)指令後，從資料表的第(n)號開始的資料將逐個往下順延。



N: 資料數

D_{Tr} : 資料表範圍(由用戶管理)

在(n)=2的情況下，將被插入到(d)+2中。

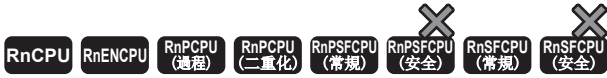
- 資料數(d)可儲存的範圍為0~65535。

出錯

出錯代碼(SD0)	內容
3405H	在(n)的值為0的狀況下執行FINS(P)指令時。
	在(d)的值為FFFFH的狀況下執行FINS(P)指令時。

資料表的資料刪除

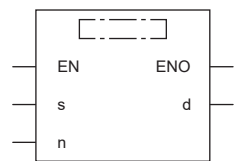
FDEL (P)



刪除資料表第(n)號的資料。

梯形圖	ST
	ENO:=FDEL (EN, s, n, d) ; ENO:=FDELP (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
FDEL	
FDELP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存刪除資料的元件	—	帶符號BIN16位元	ANY16
(d)	表的起始元件	—	字元	ANY16*1
(n)	刪除的表位置	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

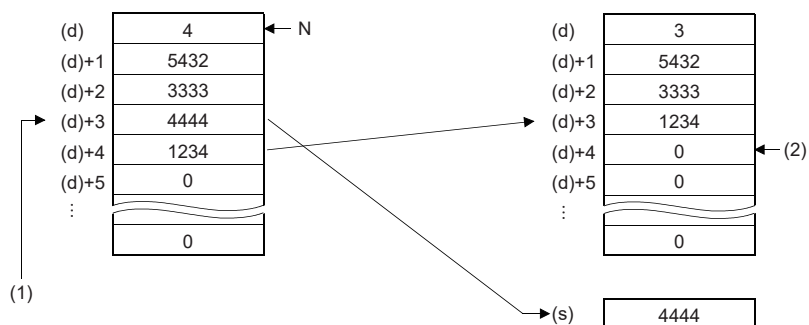
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 將(d)中指定的資料表的第(n)號的資料刪除後，儲存到(s)中指定的元件中。執行FDEL(P)指令後，資料表的第(n)+1號及其以後的資料將逐個依次向前順延。



N: 資料數

(1): (n)=3時，對象為(d)+3。

(2): 儲存為0。

- 資料數(d)可儲存的範圍為0~65535。

出錯

出錯代碼(SD0)	內容
3405H	在(n)的值為0的狀況下執行FDEL(P)指令時。
	在(d)的值為0的狀況下執行FDEL(P)指令時。

8.3 資料讀取/寫入指令

資料記憶體的讀取/寫入

至資料記憶體的資料寫入指令是將任意元件資料寫入到資料記憶體中的指令。

透過將運算中使用的固定值及運算結果寫入到資料記憶體中，可以防止電池過低時資料丟失。

此外，對於寫入至資料記憶體中的資料，可以使用從資料記憶體中的資料讀取指令在任意時機進行讀取。


■執行方法

向元件資料的資料記憶體寫入時，透過SP.DEVST指令進行。

此外，對於資料記憶體中寫入的元件資料，透過S(P).DEVLD指令讀取到指定元件中。

■設定方法

在使用SP.DEVST指令、S(P).DEVLD指令的情況下，需要進行元件資料儲存用檔案設定。

 [CPU參數]⇒[檔案設定]⇒[元件資料儲存用檔案設定]

File Setting for Device Data Storage	
Use Or Not Setting	Use
Capacity	1 K Word
File Name	DEVSTORE

項目	內容
容量	1K~512K(字元)
檔案名	DEVSTORE(固定)

資料記憶體中沒有建立元件資料儲存用檔案的必要空餘容量的情況下，將發生出錯(出錯代碼：21A1H)。

此外，STOP→RUN時將進行實際元件資料儲存用檔案與參數設定是否一致的檢查，不一致的情況下將發生出錯(出錯代碼：21A0H)。

從資料記憶體的資料讀取

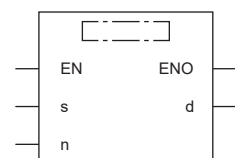
S(P).DEVLD



從資料記憶體上的元件資料儲存用檔案中讀取資料。

梯形圖	ST
	ENO:=S_DEVLD(EN, s, n, d); ENO:=SP_DEVLD(EN, s, n, d);

FBD/LD



■執行條件

指令	執行條件
S. DEVLD	
SP. DEVLD	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	從元件資料儲存用檔案的讀取偏置(以1點16位元的單位指定)	0~524287	無符號BIN32位元	ANY32
(d)	儲存讀取數據的元件	—	字元	ANY16* ¹
(n)	讀取點數	1~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

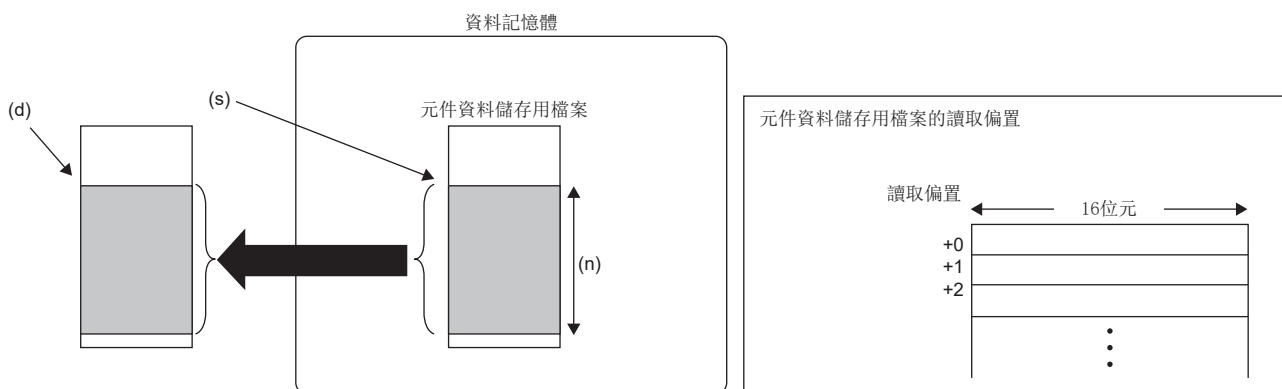
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	—	○	—	—	—	—	○	○	—	—	—	

功能

- 從資料記憶體上的元件資料儲存用檔案的(s)中指定的讀取偏置開始，讀取(n)中指定的點數的元件資料，儲存在(d)中指定的元件中。對於(s)，在元件資料儲存用檔案起始開始的偏置中，以字元偏置(每16位元+1的單位)進行指定。



- 在使用S(P).DEVLD指令的情況下，需要進行元件資料儲存用檔案設定。(☞ 623頁 設定方法)
- 至元件資料儲存用檔案的寫入透過SP.DEVST指令進行。(☞ 626頁 SP.DEVST)

出錯

出錯代碼 (SD0)	內容
2840H	當參數中未進行元件資料儲存用檔案的設定時。
3405H	當(n)的設定資料為0時。

至資料記憶體的資料寫入

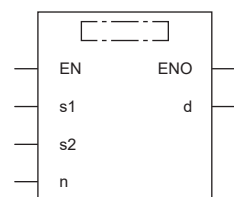
SP.DEVST



將指定點數的元件資料寫入到資料記憶體上的元件資料儲存用檔案中。

梯形圖	ST
	ENO:=SP_DEVST(EN, s1, s2, n, d);

FBD/LD



■執行條件

指令	執行條件
SP.DEVST	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	元件資料儲存用檔案的寫入偏置(以1點16位元的單位指定)	0~524287	無符號BIN32位元	ANY32
(s2)	寫入的起始元件	—	字元	ANY16*1
(n)	寫入點數	1~65535	無符號BIN16位元	ANY16
(d)	(d): 完成元件, (d)+1: 異常完成元件	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

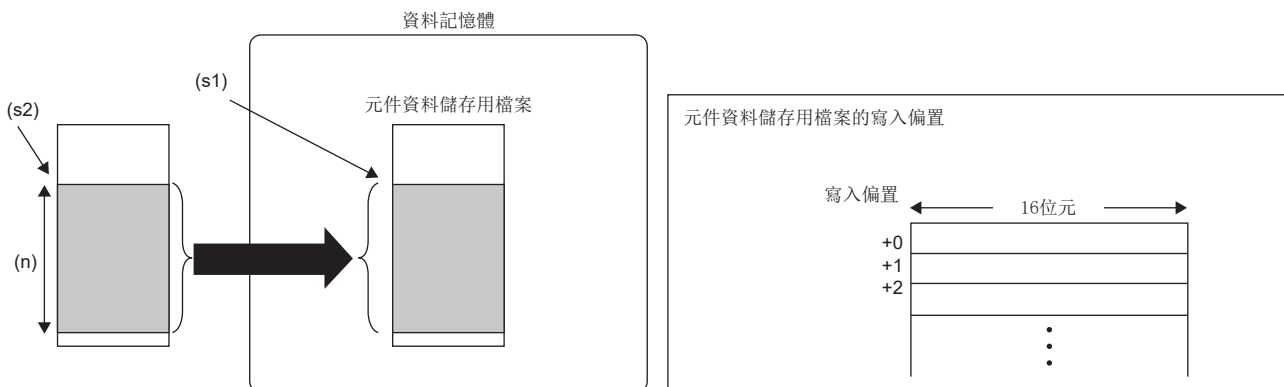
*1 在透過標籤進行設定的情況下, 應在確保動作所需區域的前提下定義數組, 指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	○	—	○	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	
(n)	○	—	○	—	—	—	—	○	—	—	—	
(d)	○	—	○	—	—	—	—	○	—	—	—	

功能

- 將從(s2)中指定的元件開始的(n)中指定的點數的元件資料，寫入到資料記憶體上的元件資料儲存用檔案的(s1)中指定的寫入偏置中。在(s1)是元件資料儲存用檔案起始開始的偏置下，以字元偏置(每個16位元+1的單位)進行指定。



- 對於完成元件(d)，檢測出SP.DEVST指令的處理完成的END指令執行時將自動設為ON，因為透過下一個掃描的END指令設為OFF，因此作為SP.DEVST指令的執行完成標誌使用。
- 在SP.DEVST指令異常完成時，因為異常完成元件(d)+1將以與完成元件(d)相同的時機設為ON/OFF，因此作為SP.DEVST指令的異常完成標誌使用。
- SP.DEVST指令的執行中SM753(檔案存取中)將變為ON。在SM753已處於ON狀態的情況下，不能執行SP.DEVST指令。(執行的情況下將變為無處理。)
- 在執行SP.DEVST指令時檢測到出錯的情況下，完成元件(d)、異常完成元件(d)+1、SM753不變為ON。
- 在使用SP.DEVST指令的情況下，需要進行元件資料儲存用檔案設定。(☞ 623頁 設定方法)
- 對於元件資料儲存用檔案中寫入的元件資料，透過S(P).DEVLD指令讀取到指定元件中。(☞ 624頁 S(P).DEVLD)

注意事項

- 寫入到資料記憶體中的值將變為執行SP.DEVST指令時的值。
- 透過執行SP.DEVST指令，SD634、SD635(資料記憶體寫入次數指標)將增加。寫入CPU模組的資料記憶體的次數存在上限，資料記憶體寫入次數指標超過10萬次後，會發生出錯(出錯代碼：1080H)。
- 為了防止意外指令執行導致資料記憶體寫入次數增加，透過設定SD771(至資料記憶體的寫入指令執行次數指定)，可以對1天中的寫入次數進行限制。寫入次數的限制預設為36次。應根據需要透過SD771變更限制次數。超過設定的寫入次數後，會發生出錯(出錯代碼：3421H)。此外，每天寫入至資料記憶體的指令執行次數將在下述時機初始化為0。
 - 電源OFF→ON時，重設→重設解除時
 - 由於時間的推進，時鐘資料的日期(年、月、日)變更時
 - 由於時鐘資料的更改功能，使得CPU模組內部的時鐘資料的年、月、日變更的時候
- 寫入至元件資料儲存用檔案的時機將變為執行END指令時。執行SP.DEVST指令之後的END指令執行時將寫入到元件資料儲存用檔案中。因此，根據寫入點數，在至元件資料儲存用檔案的寫入有可能會跨越多個掃描的情況下，應透過完成元件判斷寫入是否完成。

出錯

出錯代碼(SD0)	內容
2840H	當參數中未進行元件資料儲存用檔案的設定時。
3405H	當(n)的設定資料為0時。
3421H	執行SP.DEVST指令時，當天的寫入次數超過了SD771中指定的值時。
	執行SP.DEVST指令時，SD771中設定了超過1~32767的範圍的值時。

8.4 檔案操作指令

從指定檔案的資料讀取

SP.FREAD

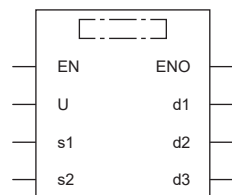


• R00CPU中不能使用。

從SD記憶卡的指定檔案中讀取元件資料。

梯形圖	ST
	<pre>ENO:=SP_FREAD(EN, U, s1, s2, d1, d2, d3);</pre>

FBD/LD



■執行條件

指令	執行條件
SP.FREAD	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	元件名	ANY16
(s1)	驅動器指定	2(固定)*1	字元	ANY16
(d1)	儲存了控制資料的起始元件	☞ 630頁 類型1 ☞ 632頁 類型2	字元	ANY16_ARRAY (要素數: 8)
(s2)	儲存了檔案名的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(d2)	儲存讀取的資料的起始元件	—	字元	ANY16*2
(d3)	處理完成時設為ON的位元元件 (但是, 異常完成時(d3)+1也將ON)	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能設定SD記憶卡的驅動器2。

*2 在透過標籤進行設定的情況下, 應在確保動作所需區域的前提下定義數組, 指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H)G□	Z	LT、 LST、LC	LZ		K、H	E	\$	
(U)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	—	○	—	—	—	—	○	○	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—
(d3)	○	—	○	—	—	—	—	—	—	—	—	—

控制資料(d1)、檔案名(s2)、讀取資料(d2)根據使用的CPU模組而有所不同。

使用的CPU模組		參閱目標
RnCPU	R01CPU、R02CPU	630頁 類型1
	R04CPU、R08CPU、R16CPU、 R32CPU、R120CPU	韌體版本“28”及其以後 630頁 類型1 韌體版本“27”及其以前 632頁 類型2
RnENCPU		韌體版本“28”及其以後 630頁 類型1 韌體版本“27”及其以前 632頁 類型2
RnPCPU		632頁 類型2
RnPSFCPU		632頁 類型2
RnSFCPU		632頁 類型2

■類型1

• 控制資料 (d1)

操作數: (d1)				
元件	項目	內容	設定範圍	設定方
+0	執行/完成類型	指定執行類型。 ■00**H : 二進制讀取 • 0000H: BIN16位元資料 • 0001H: BIN32位元資料 ■01**H : CSV格式轉換讀取 • 0100H: 10進制數 (16位元資料) • 0110H: 10進制數 (32位元資料) • 0120H: 16進制數 (16位元資料) • 0121H: 16進制數 (32位元資料) • 0130H: 字元串 (ASCII資料) • 0140H: 浮點實數 (單精度實數) • 0141H: 浮點實數 (雙精度實數)	0000H 0001H 0100H 0110H 0120H 0121H 0130H 0140H 0141H	用戶
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H: 正常完成 • 0000H以外: 異常完成 (出錯代碼) (參見 680頁 在檔案操作指令中發生的出錯代碼)	—	系統
+2	請求讀取資料數	指定希望讀取的資料數。處理單位及設定範圍根據 (d1)+0 的執行/完成類型有所不同。 ■在 (d1)+0 中指定 “二進制讀取” 時 • BIN16位元資料指定時: 字元單位 (1~65535)*2*3 • BIN32位元資料指定時: 雙字元單位 (1~32767) ■(d1)+0 中指定 “CSV格式轉換讀取” 時 • 10進制數 (16位元資料) 指定時: 要素數 (1~65535)*2*3 • 10進制數 (32位元資料) 指定時: 要素數 (1~32767) • 16進制數 (16位元資料) 指定時: 要素數 (1~65535) • 16進制數 (32位元資料) 指定時: 要素數 (1~32767) • 字元串 (ASCII資料) 指定時: 要素數 (1~1023) • 浮點實數 (單精度實數) 指定時: 要素數 (1~32767) • 浮點實數 (雙精度實數) 指定時: 要素數 (1~16383)	1~65535	用戶
+3	最大讀取資料數	■在 (d1)+0 中指定 “0130H: 字元串 (ASCII資料)” 時 • 要素內的字元數的合計容量 (字元單位) ■在 (d1)+0 中指定 “0130H: 指定為字元串 (ASCII資料)” 以外時 • 固定為0	0、1~65535	用戶
+4 +5	檔案位置	■在 (d1)+0 中指定 “二進制讀取” 時 • 00000000H: 從檔案的起始開始 • 00000001H~FFFFFFFH: 從指定位置開始 (單位取決於 (d1)+7 的資料類型指定。) • FFFFFFFFH: 不能指定 ■(d1)+0 中指定 “CSV格式轉換讀取” 時 • 00000000H: 從檔案的起始開始 • 00000001H~FFFFFFFH: 從指定行開始 • FFFFFFFFH: 從上次的讀取位置繼續	00000000H~ FFFFFFFH	用戶
+6	欄數指定	在 (d1)+0 指定 “二進制讀取” 時, 必須指定0。 在 (d1)+0 中指定 “CSV格式轉換讀取” 時, 指定進行讀取的欄數。 • 0: 無欄位。變為1列。 • 0以外: 變為指定數的欄位。	0000H~FFFFH (0~65535)	用戶
+7	資料類型指定	• 0: 字元 • 1: 偶數字節*2 • 2: (d1)+0 中指定的資料類型的單位 • 3: 奇數字節*1*2*3 “0: 字元” 及 “1: 偶數字節”, “3: 奇數字節” 僅在 (d1)+0 中指定了 “0000H: BIN16位元資料” 或 “0100H: 10進制數 (16位元資料)” 的情況下可以指定。	0、1、2、3	用戶

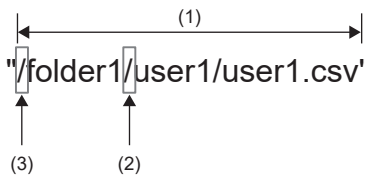
*1 奇數字節的指定, 只有下述機型可以使用。此外, 根據機型不同所對應的韌體版本也不同。

- R01CPU、R02CPU: “19” 及其以後
- R04CPU、R08CPU、R16CPU、R32CPU、R120CPU: “51” 及其以後
- RnENCPU: “51” 及其以後

*2 資料類型指定 (d1)+7 中指定了 “1: 偶數字節” 或者 “3: 奇數字節” 時, 請求讀取資料數 (d1)+2 的設定範圍為 1~32767。

*3 指定奇數字節 (d1)+7 時對欲讀取的奇數字節資料數加入1字節, 使其成為字元單位。並應將已成為字元單位的資料數指定為請求讀取資料數 (d1)+2。

• 檔案名稱 (s2)

操作數: (s2)				
元件	項目	內容	設定範圍	設定方
+0~+□	檔案名稱字元串	<p>指定儲存了檔案的資料夾路徑、檔案名稱的字元串。</p> <ul style="list-style-type: none"> 資料夾路徑+檔案名稱(包含副檔名)不應超過253字元。 資料夾路徑不應超過244字元。(不包含檔案的分隔符號。) 資料夾路徑的階層數不應超過10層。 省略副檔名的情況下應從“.”(點號)開始省略。 檔案名不應超過60字元+點號+3字元。檔案名超過61字元時, 即使有副檔名也將被忽略, 變為“.BIN”或“.CSV”。  <p>(1): 最大253字元 (2): 資料夾路徑、檔案的分隔符號使用“/”或“\”。 (3): 可以省略。省略的情況下, (1)最大為252字元。</p>	Unicode字元串	用戶

• 讀取的資料 (d2)

操作數: (d2)				
元件	項目	內容	設定範圍	設定方
+0	讀取結果資料數	對 (d1)+2中指定的資料數, 設定實際讀取的資料數。 值的單位取決於 (d1)+7的資料類型指定。	—	系統
+1~+□	讀取資料	儲存讀取的資料。	—	系統

■類型2

• 控制資料 (d1)

操作數: (d1)				
元件	項目	內容	設定範圍	設定方
+0	執行/完成類型	指定執行類型。 • 0000H: 二進制讀取 • 0100H: CSV格式轉換讀取	0000H 0100H	用戶
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H: 正常完成 • 0000H以外: 異常完成(出錯代碼) (參見 680頁 在檔案操作指令中發生的出錯代碼)	—	系統
+2	請求讀取資料數	指定希望讀取的資料數。(字元單位) 在(d1)+7的資料類型指定中指定了“偶數字節”的情況下,也要進行字元換算並以字元單位進行設定。	1~65535*1	用戶
+3	未使用	—	—	—
+4 +5	檔案位置	<ul style="list-style-type: none"> ■在(d1)+0中指定“二進制讀取”時 <ul style="list-style-type: none"> • 00000000H: 從檔案的起始開始 • 00000001H~FFFFFFFH: 從指定位置開始(單位取決於(d1)+7的資料類型指定。) • FFFFFFFFH: 不能指定 ■(d1)+0中指定“CSV格式轉換讀取”時 <ul style="list-style-type: none"> • 00000000H: 從檔案的起始開始 • 00000001H~FFFFFFFH: 從指定行開始 • FFFFFFFFH: 從上次的讀取位置繼續 	00000000H~ FFFFFFFH	用戶
+6	欄數指定	在(d1)+0指定“二進制讀取”時,必須指定0。 在(d1)+0中指定“CSV格式轉換讀取”時,指定進行讀取的欄數。 • 0: 無欄位。變為1列。 • 0以外: 變為指定數的欄位。	0000H~FFFFH (0~65535)	用戶
+7	資料類型指定	0: 字元 1: 偶數字節	0、1	用戶

*1 資料類型指定(d1)+7為“1: 偶數字節”的情況下,設定範圍為1~32767。

• 檔案名稱 (s2)

操作數: (s2)				
元件	項目	內容	設定範圍	設定方
+0~+□	檔案名稱字元串	指定檔案名稱的字元串。 省略副檔名的情況下應從“.”(點號)開始省略。 不應超過60字元+點號+3字元。 超過61字元時,即使有副檔名也將被忽略,變為“.BIN”或“.CSV”。	Unicode字元串	用戶

• 讀取的資料 (d2)

操作數: (d2)				
元件	項目	內容	設定範圍	設定方
+0	讀取結果資料數	對(d1)+2中指定的資料數,設定實際讀取的資料數。 值的單位取決於(d1)+7的資料類型指定。	—	系統
+1~+□	讀取資料	儲存讀取的資料。	—	系統

功能

- 從指定檔案中讀取資料。透過控制資料的執行/完成類型，指定檔案的讀取形式。

使用的CPU模組		參閱目標
RnCPU	R01CPU、R02CPU	630頁 類型1
	R04CPU、R08CPU、R16CPU、 R32CPU、R120CPU	韌體版本“28”及其以後 630頁 類型1 韌體版本“27”及其以前 632頁 類型2
RnENCPU		韌體版本“28”及其以後 630頁 類型1 韌體版本“27”及其以前 632頁 類型2
RnPCPU		632頁 類型2
RnPSFCPU		632頁 類型2
RnSFCPU		632頁 類型2

- 讀取的對象僅SD記憶卡。
- 處理完成 (d3) 的位元元件，在檢測出SP. FREAD指令的處理完成掃描的END指令執行時將被自動設為ON，透過下一個掃描的END指令設為OFF。
- 當SP. FREAD指令異常完成時，異常完成 (d3)+1的元件將以與處理完成 (d3) 的元件相同的時機設為ON/OFF。
- SP. FREAD指令執行中，SM753 (檔案存取中) 將變為ON。
- SM753為ON中，不能執行SP. FREAD指令。(執行的情況下將變為無處理。)
- 在執行指令時檢測到出錯的情況下，處理完成 (d3)、異常完成 (d3)+1及SM753不變為ON。
- 指定奇數字節時，資料將以如下形式儲存。(在指定的最後的b8~b15中，儲存有指令執行前的資料。)

	b15	...	b8 b7	...	b0
(d2)					
(d2)+1					
(d2)+2					
(d2)+3					
(d2)+4					
(d2)+5					

- 資料的請求讀取檔案數(d1)+2、檔案位置(d1)+4、(d1)+5及讀取結果資料數(d2)+0的處理單位，透過執行/完成類型(d1)+0及資料類型指定(d1)+7的組合進行指定。

執行/完成類型 (d1)+0	資料類型指定 (d1)+7	處理單位及設定範圍			
		請求讀取資料數 (d1)+2	檔案位置 (d1)+4、(d1)+5	讀取結果資料數 (d2)+0	
二進制讀取	0000H: BIN16位元資料	0: 字元	字元(1~65535)	字元(00000000H~7FFFFFFFH)	字元
		1: 偶數字節	字元(1~32767)	字節(00000000H~FFFFFFFEH)	字節
		2: 透過執行/完成類型指定的資料類型的單位	字元(1~65535)	字元(00000000H~7FFFFFFFH)	字元
		3: 奇數字節	字元(1~32767)	字節(00000000H~FFFFFFFEH)	字節
	0001H: BIN32位元資料	0: 字元 1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	雙字元(1~32767)	雙字元(00000000H~3FFFFFFFH)	雙字元
3: 奇數字節		(不能指定)			
CSV格式轉換讀取	0100H: 10進制數(16位元資料)	0: 字元	要素數(1~65535)	列數	字元
		1: 偶數字節	要素數(1~32767)	列數	字節
		2: 透過執行/完成類型指定的資料類型的單位	要素數(1~65535)	列數	字元
		3: 奇數字節	要素數(1~32767)	列數	字節
	0110H: 10進制數(帶符號32位元資料)	0: 字元 1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	要素數(1~32767)	列數	雙字元
		3: 奇數字節	(不能指定)		
	0120H: 16進制數(16位元資料)	0: 字元 1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	要素數(1~65535)	列數	字元
		3: 奇數字節	(不能指定)		
	0121H: 16進制數(32位元資料)	0: 字元 1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	要素數(1~32767)	列數	雙字元
		3: 奇數字節	(不能指定)		
	0130H: 字元串(ASCII資料)	0: 字元 1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	要素數(1~1023)	列數	要素數
		3: 奇數字節	(不能指定)		
	0140H: 浮點實數(單精度實數)	0: 字元 1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	要素數(1~32767)	列數	雙字元
		3: 奇數字節	(不能指定)		
	0141H: 浮點實數(雙精度實數)	0: 字元 1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	要素數(1~16383)	列數	4字元
		3: 奇數字節	(不能指定)		

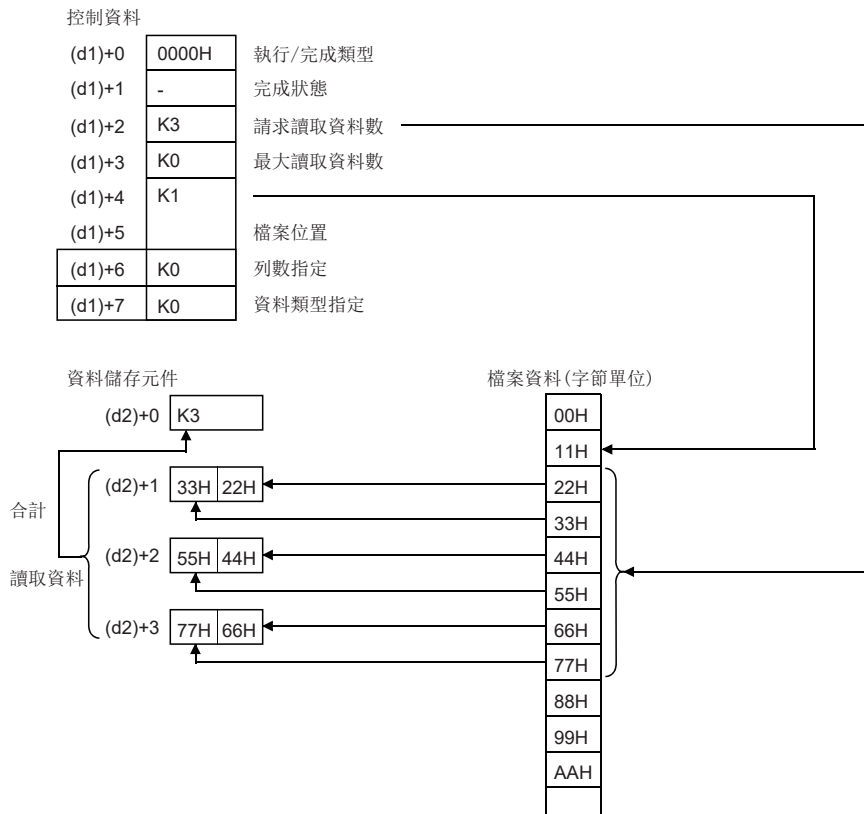
■二進制讀取時

- 省略了對象檔案的副檔名的情況下，副檔名將變為“.BIN”。
- 在被指定不存在的檔案時將變為出錯。
- 與現有檔案的大小相比，在被指定更大檔案位置的情況下，將變為0點的讀取，正常完成。

二進制讀取的示例如下所示。

例

二進制讀取(BIN16位元資料)的示例



■CSV格式轉換讀取時

- 將CSV格式檔案的要素 (Excel®的單元格)按橫向方向依次讀取，儲存到元件中。
- 在省略對象檔案的副檔名的情況下，副檔名將變為“.CSV”。
- 在被指定不存在的檔案時將變為出錯。
- 在指定了比現有檔案的大小更大的檔案位置的情況下，將變為0點的讀取，正常完成。
- 從檔案的起始開始請求讀取資料數(d1)+2中指定的資料將被讀取。讀取指定資料數的資料之前，到達檔案的最終資料的情況下，讀取可讀取的資料。
- 在指定欄數為0的情況下，將忽略CSV格式檔案的換行進行讀取。
- 與CSV檔案內的字元串資料讀取後儲存於元件的值取決於執行/完成類型。

執行/完成類型	CSV內資料(1要素)	元件儲存值	備注
0100H: 10進制數(16位元資料)	-32768~-1	-32768~-1 (32768~65535)	視為帶符號16位元資料時為-32768~-1，視為無符號16位元資料時為32768~65535。作為元件儲存值為相同。
	0~32767	0~32767	—
	32768~65535	-32768~-1 (32768~65535)	視為帶符號16位元資料時為-32768~-1，視為無符號16位元資料時為32768~65535。作為元件儲存值為相同。
	<ul style="list-style-type: none"> • 上述以外的數值 • 包含英文字母與符號的字元串 	0	由於無法轉換因此儲存0。
0110H: 10進制數(32位元資料)	-2147483648~-1	-2147483648~-1 (2147483648~4294967295)	視為帶符號32位元資料時為-2147483648~-1，視為無符號32位元資料時為2147483648~4294967295。作為元件儲存值為相同。
	0~2147483647	0~2147483647	—
	2147483648~4294967295	-2147483648~-1 (2147483648~4294967295)	視為帶符號32位元資料時為-2147483648~-1，視為無符號32位元資料時為2147483648~4294967295。作為元件儲存值為相同。
	<ul style="list-style-type: none"> • 上述以外的數值 • 包含英文字母與符號的字元串 	0	由於無法轉換因此儲存0。
0120H: 16進制數(16位元資料)	0H~FFFFH	0H~FFFFH	—
	<ul style="list-style-type: none"> • 上述以外的數值 • 包含A~F以外的英文字母與符號的字元串 	0000H	由於無法轉換因此儲存0。
0121H: 16進制數(32位元資料)	0H~FFFFFFFFH	0H~FFFFFFFFH	—
	<ul style="list-style-type: none"> • 上述以外的數值 • 包含A~F以外的英文字母與符號的字元串 	00000000H	由於無法轉換因此儲存0。
0130H: 字元串(ASCII資料)	字元串(1999字元為止)	字元串(1999字元為止)	在末尾追加NULL(00H)。CSV內的字元串為偶數字節的情況下，下個1字元中儲存0000H。CSV內的字元串中包含00H的情況下，此00H將被忽略。
	字元串(2000字元及其以上)		1要素內的字元串超過1999字元的情況下，至1999字元為止將作為1要素讀取。第2000字元及其以後的字元將不會被讀取，而進行下一個要素的讀取。
0140H: 浮點實數(單精度實數)	$-2^{128} < \text{資料} \leq -2^{126}$ 、 0、 $2^{-126} \leq \text{資料} < 2^{128}$ 的範圍的值	左邊記載範圍的值	小數點形式/指數形式皆可轉換。
	上述以外的數值	0	由於無法轉換因此儲存0。
0141H: 浮點實數(雙精度實數)	$-2^{1024} < \text{資料} \leq -2^{1022}$ 、 0、 $2^{-1022} \leq \text{資料} < 2^{1024}$ 的範圍的值	左邊記載範圍的值	小數點形式/指數形式皆可轉換。
	上述以外的數值	0	由於無法轉換因此儲存0。

例

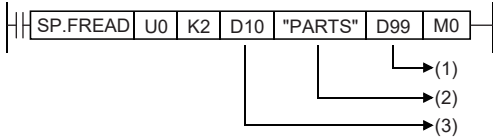
指定CSV格式轉換讀取(字元串(ASCII資料))的情況下

[以CSV格式保存的資料]

PARTS.CSV

No.	Name	Value1	Value2	CR	LF
AA_0001	Prts_A	100	200	CR	LF
BB_0002	Prts_B	300	400	CR	LF

[元件中讀入的資料]



[控制資料]

D10	H0130
D11	H0000
D12	K6
D13	K100
D14	K2
D15	K2
D16	K3
D17	K2

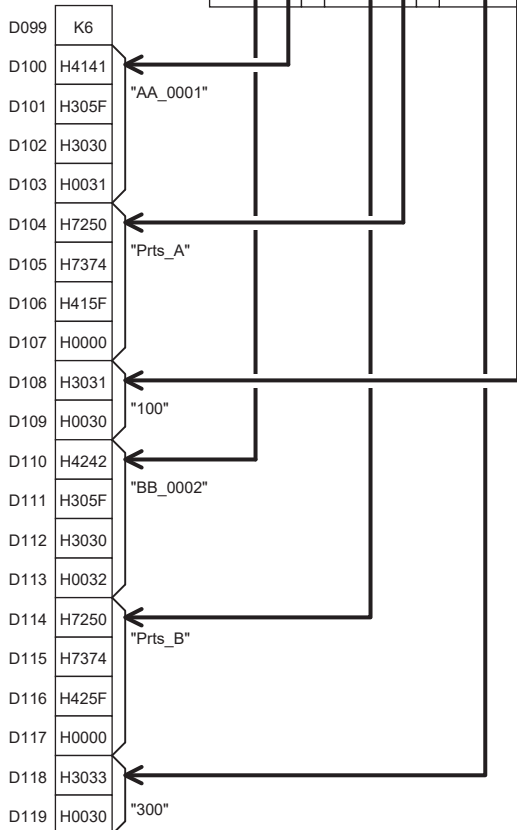
進行左邊框起來地方的讀取。
(從PARTS.CSV的第2列開始每3欄，合計6
要素的讀取)

- (1): 讀取的資料
- (2): 檔案名
- (3): 控制資料

- D10: 執行/完成類型
- D11: 完成狀態
- D12: 請求讀取資料數
- D13: 最大讀取資料數
- D14、D15: 檔案位置
- D16: 欄數指定
- D17: 資料類型指定

PARTS.CSV

No.	Name	Value1	Value2	CR	LF
AA_0001	Prts_A	100	200	CR	LF
BB_0002	Prts_B	300	400	CR	LF



- D99: 讀取結果資料數
- D100~D103: 第2列第1欄的字元串
- D104~D107: 第2列第2欄的字元串
- D108~D109: 第2列第3欄的字元串
- D110~D113: 第3列第1欄的字元串
- D114~D117: 第3列第2欄的字元串
- D118~D119: 第3列第3欄的字元串

例

CSV格式轉換讀取(10進制數(16位元資料))指定時，指定欄數為0的情況下

[透過Excel建立的資料]

	A	B	C
1	Main/sub item		Measured value
2	Length	1	3
3	Temperature	-21	
4			

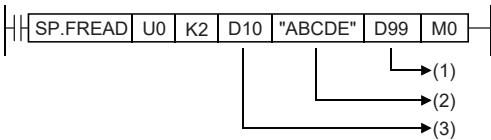


[以CSV形式儲存的資料]

大/小項目	,	,	測定值	CR	LF
長度	,	1	,	3	CR LF
溫度	,	-21	,		CR LF



[元件中讀入的資料]



- (1): 讀取的資料
- (2): 檔案名
- (3): 控制資料

[控制資料]

D10	0100H
D11	-
D12	K9
D13	K0
D14	K0
D15	K0
D16	K0
D17	K0

- D10: 執行/完成類型
- D11: 完成狀態
- D12: 請求讀取資料數
- D13: 最大讀取資料數
- D14、D15: 檔案位置
- D16: 欄數指定
- D17: 資料類型指定

[讀入的資料]

D99	K9
D100	K0
D101	K0
D102	K0
D103	K0
D104	K1
D105	K3
D106	K0
D107	K-21
D108	K0

- D99: 讀取結果資料數
- D100: 因為“大/小項目”為數值以外的資料，因此儲存轉換資料(0)。
- D101: 因為“ ”為數值以外的資料，因此儲存轉換資料(0)。
- D102: 因為“測定值”為數值以外的資料，因此儲存轉換資料(0)。
- D103: 因為“長度”為數值以外的資料，因此儲存轉換資料(0)。
- D104: 因為“1”為數值，因此轉換為二進制值。
- D105: 因為“3”為數值，因此轉換為二進制值。
- D106: 因為“溫度”為數值以外的資料，因此儲存轉換資料(0)。
- D107: 因為“-21”為數值，因此轉換為二進制值。
- D108: 因為“ ”為數值以外的資料，因此儲存轉換資料(0)。

- 即使欄數在各個列中有不相同的情況，也將忽略行進行讀取。

例

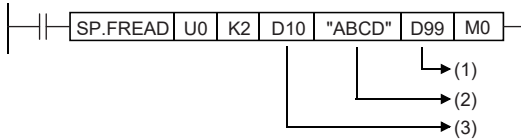
在讀取時的欄數在各個列中不相同的情況下

[以CSV形式儲存的資料]

大/小項目	,	,	測定值	,	餘量	CR	LF
長度	CR	LF					
溫度	,	-21	,	CR	LF		



[元件中讀入的資料]



- (1): 讀取的資料
- (2): 檔案名
- (3): 控制資料

[控制資料]

D10	0100H
D11	-
D12	K7
D13	K0
D14	K0
D15	K0
D16	K0
D17	K0

- D10: 執行/完成類型
- D11: 完成狀態
- D12: 請求讀取資料數
- D13: 最大讀取資料數
- D14、D15: 檔案位置
- D16: 欄數指定
- D17: 資料類型指定

[讀入的資料]

D99	K7
D100	K0
D101	K0
D102	K0
D103	K0
D104	K0
D105	K0
D106	K-21

- D99: 讀取結果資料數
- D100: 因為“大/小項目”為數值以外的資料，因此儲存轉換資料(0)。
- D101: 因為“ ”為數值以外的資料，因此儲存轉換資料(0)。
- D102: 因為“測定值”為數值以外的資料，因此儲存轉換資料(0)。
- D103: 因為“餘量”為數值以外的資料，因此儲存轉換資料(0)。
- D104: 因為“長度”為數值以外的資料，因此儲存轉換資料(0)。
- D105: 因為“溫度”為數值以外的資料，因此儲存轉換資料(0)。
- D106: 因為“-21”為數值，因此轉換為二進制值。

要點

在Excel中，欄數不能在各個列中建立不相同的檔案。在修改CSV檔案的時候發生。

- 在指定欄數為0以外的情況下，作為指定的欄數的表讀取CSV格式的檔案。超過指定欄數的要素將被忽略。

例

CSV格式轉換讀取(10進制數(16位元資料))指定時，指定欄數為0以外的情況下((d1)+6為2的情況下)

[透過Excel建立的資料]

	A	B	C
1	Main/sub item		Measured value
2	Length	1	3
3	Temperature	-21	
4			



[以CSV形式儲存的資料]

大/小項目	,	測定值	CR	LF
長度	,	1	,	3 CR LF
溫度	,	-21	CR	LF

(1)

(1): 超過指定欄數的要素將被忽略。



[元件中讀入的資料]

SP.FREAD	U0	K2	D10	"ABCD"	D99	M0
----------	----	----	-----	--------	-----	----

(2)
(3)
(4)

- (2): 讀取的資料
- (3): 檔案名
- (4): 控制資料

[控制資料]

D10	0100H
D11	-
D12	K6
D13	K0
D14	K0
D15	K0
D16	K2
D17	K0

- D10: 執行/完成類型
- D11: 完成狀態
- D12: 請求讀取資料數
- D13: 最大讀取資料數
- D14、D15: 檔案位置
- D16: 欄數指定
- D17: 資料類型指定

[讀入的資料]

D99	K6
D100	K0
D101	K0
D102	K0
D103	K1
D104	K0
D105	K-21

- D99: 讀取結果資料數
- D100: 因為“大/小項目”為數值以外的資料，因此儲存轉換資料(0)。
- D101: 因為“ ”為數值以外的資料，因此儲存轉換資料(0)。
- D102: 因為“長度”為數值以外的資料，因此儲存轉換資料(0)。
- D103: 因為“1”為數值，因此轉換為二進制值。
- D104: 因為“溫度”為數值以外的資料，因此儲存轉換資料(0)。
- D105: 因為“-21”為數值，因此轉換為二進制值。

- 即使欄數在各個列中有不相同的情況，超過指定欄數的要素將被忽略，不足指定欄數的欄將被補0。

例

在讀取時的欄數在各個列中不相同的情況下

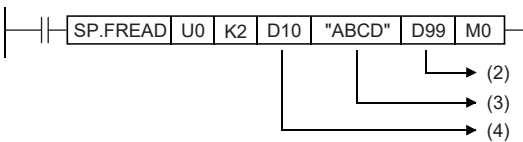
[以CSV形式儲存的資料]

大/小項目	,	,	測定值	,	餘量	CR	LF
長度	CR	LF					
溫度	,	-21	,	CR	LF		

(1)



[元件中讀入的資料]



[控制資料]

D10	0100H
D11	-
D12	K6
D13	K0
D14	K0
D15	K0
D16	K2
D17	K0

[讀入的資料]

D99	K6
D100	K0
D101	K0
D102	K0
D103	K0
D104	K0
D105	K-21

(1): 超過指定欄數的要素將被忽略。

- (2): 讀取的資料
- (3): 檔案名
- (4): 控制資料

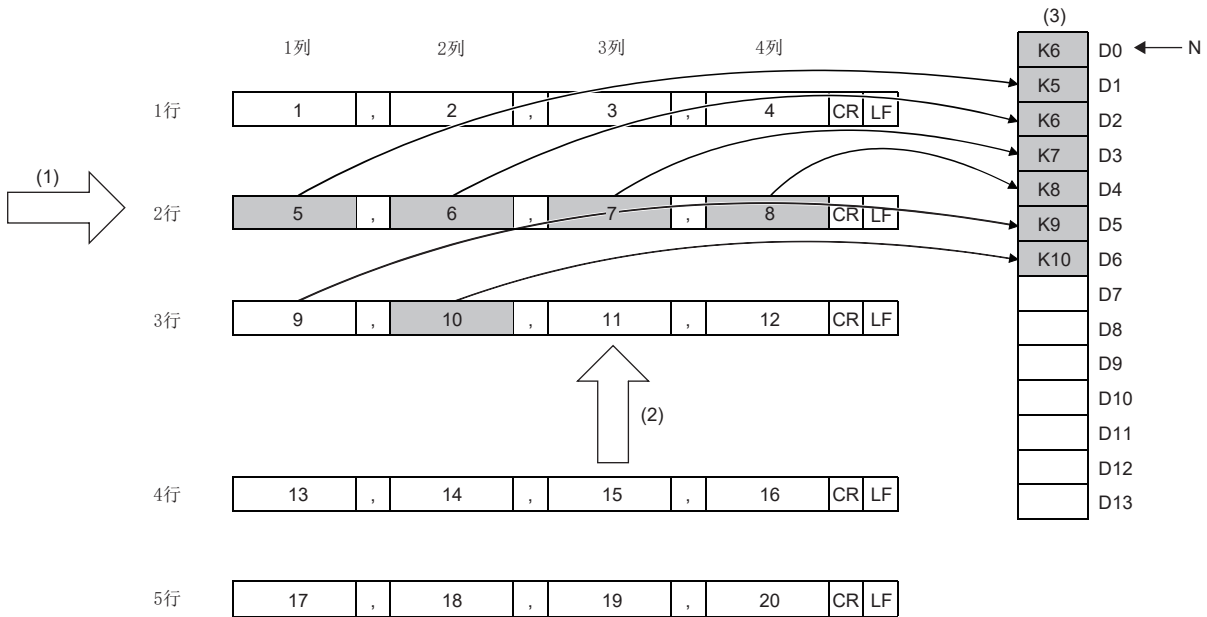
- D10: 執行/完成類型
- D11: 完成狀態
- D12: 請求讀取資料數
- D13: 最大讀取資料數
- D14、D15: 檔案位置
- D16: 欄數指定
- D17: 資料類型指定

- D99: 讀取結果資料數
- D100: 因為“大/小項目”為數值以外的資料，因此儲存轉換資料(0)。
- D101: 因為“ ”為數值以外的資料，因此儲存轉換資料(0)。
- D102: 因為“長度”為數值以外的資料，因此儲存轉換資料(0)。
- D103: 因為沒有相應的要素，因此被填補轉換資料(0)。
- D104: 因為“溫度”為數值以外的資料，因此儲存轉換資料(0)。
- D105: 因為“-21”為數值，因此轉換為二進制值。

- 在CSV格式轉換讀取中，可以將資料分為多次進行讀取。

[指定希望開始讀取的行]

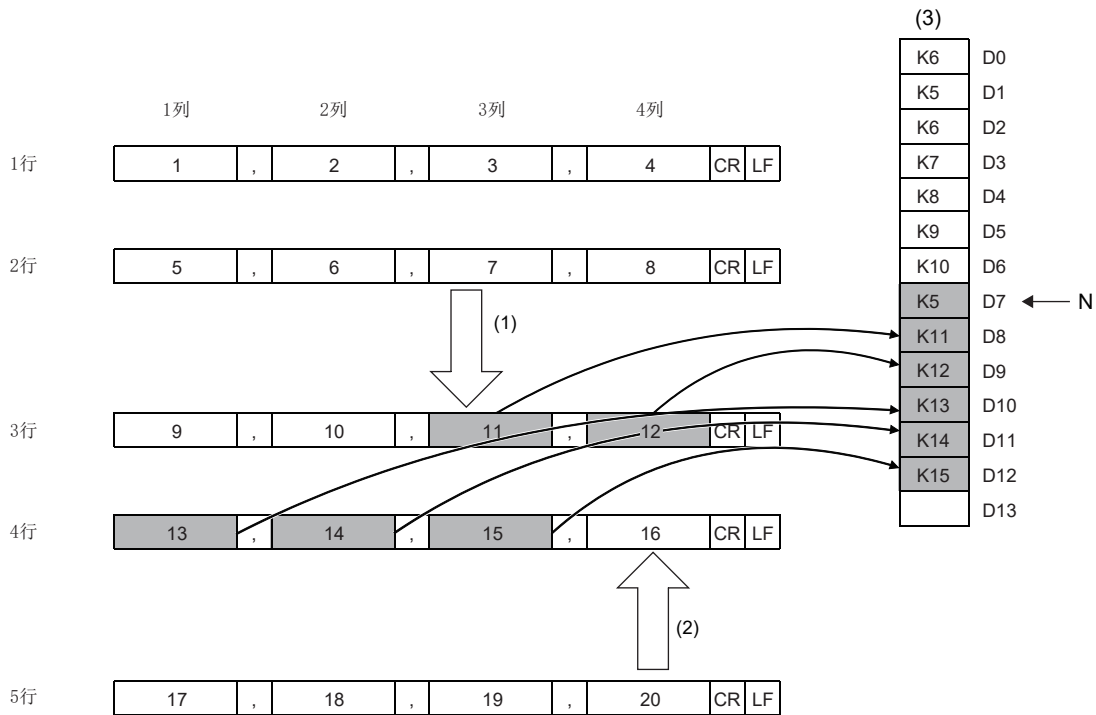
- 執行/完成類型：CSV格式轉換讀取(10進制數(BIN16位元資料))
- 欄數指定：4H
- 資料類型指定：字元
- 檔案位置：2H
- 讀取起始元件：D0
- 讀取結果資料數：6H



- (1): 開始列
 - (2): 下次開始位置
 - (3): 元件儲存資料(讀取資料)
- N: 資料數

[從上次讀取位置開始繼續讀取]

- 執行/完成類型：CSV格式轉換讀取(10進制數(BIN16位元資料))
- 欄數指定：4H
- 資料類型指定：字元
- 檔案位置：FFFFFFFH(從上次讀取位置開始繼續)
- 讀取起始元件：D7
- 讀取結果資料數：5H



- (1): 開始列
 (2): 下次開始位置
 (3): 元件儲存資料(讀取資料)
 N: 資料數

要點

- 在從上次讀取位置開始繼續進行讀取的情況中，如果“執行/完成類型”、“欄數指定”、“資料類型指定”之設定與上次不相同時，將無法從上次的讀取位置開始正常添加。
- 在從上次讀取位置開始繼續讀取資料的過程中，如果執行其它設定的SP.FREAD指令及SP.FWRITE指令，將無法從上次的讀取位置開始正常添加。

注意事項

- 對於SP.FREAD指令，應勿透過中斷程式執行。若透過中斷程式執行，可能導致誤動作。
- CSV格式轉換讀取時，讀取多個要素數的情況下，應在確保各要素的合計容量的元件的前提下在讀取資料區域中執行指令。由於讀取資料從(d2)+1開始儲存，因此作為(d2)確保的必要字元數為((各要素的合計字元數)+1)。
- CSV格式轉換讀取(字元串(ASCII資料))指定時，合計容量(字元單位)應設定為(d1)+3的最大讀取資料數。
[例]從CSV檔案，將1要素的字元數為100字元的字元串讀取100要素的情況下
 $(100(\text{字元})+2(\text{NULL}))\times 100(\text{要素}) = 10200\text{字節} = 5100\text{字元}$
並且，應在最大讀取資料數((d1)+3)設定5100，確保(d2)為5101字元的領域。
- 由於發生檔案讀取，因此執行SP.FREAD指令可能延長掃描時間。

出錯

出錯代碼(SD0)	內容
2820H	在讀取的資料容量超過讀取元件容量的情況下。
3405H	當透過驅動器指定(s1)指定的驅動器為SD記憶卡以外時。 當控制資料(d1)及其以後設定的值超出設定範圍時。 ☞ 630頁 類型1 ☞ 632頁 類型2 無法讀取(s2)中指定的檔案名字元串時。 <ul style="list-style-type: none">• 指定的檔案名字元串的字元數超出範圍時。• 設定了不能使用的值。• 對已指定的檔案名字元串的末尾指定了分隔符號。 ☞ 630頁 類型1 ☞ 632頁 類型2
3427H	(d1)+0的執行/完成類型和(d1)+7的資料類型指定為不能指定的組合時。 ☞ 630頁 類型1

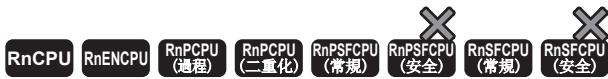
在SP.FREAD指令異常完成的情況下，完成狀態(d1)+1中指定的元件中將儲存出錯代碼。(指令運算出錯的情況下不能儲存。)

關於完成狀態(d1)+1中儲存的出錯代碼，請參閱下述內容。

☞ 680頁 在檔案操作指令中發生的出錯代碼

至指定檔案的資料寫入

SP.FWRITE



• R00CPU中不能使用。

將元件資料寫入到SD記憶卡的指定檔案中。

梯形圖	ST
	<pre>ENO:=SP_FWRITE (EN, U, s1, s2, s3, d1, d2);</pre>

FBD/LD

■執行條件

指令	執行條件
SP.FWRITE	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	元件名	ANY16
(s1)	驅動器指定	2(固定)*1	字元	ANY16
(d1)	儲存了控制資料的起始元件	☞ 647頁 類型1 ☞ 649頁 類型2	字元	ANY16_ARRAY (要素數: 8)
(s2)	儲存了檔案名的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(s3)	儲存了資料的起始元件	—	字元	ANY16*2
(d2)	處理完成時設為ON的位元元件	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能設定SD記憶卡的驅動器2。

*2 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H) G□	Z	LT、 LST、LC	LZ		K、H	E	\$	
(U)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	—	○	—	—	—	—	○	○	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	—	○	—	—	—	—	—	—	—	—	—

控制資料(d1)、檔案名(s2)、寫入資料(s3)根據使用的CPU模組而有所不同。

使用的CPU模組		參閱目標
RnCPU	R01CPU、R02CPU	647頁 類型1
	R04CPU、R08CPU、R16CPU、 R32CPU、R120CPU	韌體版本“28”及其以後 647頁 類型1 韌體版本“27”及其以前 649頁 類型2
RnENCPU		韌體版本“28”及其以後 647頁 類型1 韌體版本“27”及其以前 649頁 類型2
RnPCPU		649頁 類型2
RnPSFCPU		649頁 類型2
RnSFCPU		649頁 類型2

■類型1

• 控制資料 (d1)

操作數: (d1)										
元件	項目	內容	設定範圍	設定方						
+0	執行/完成類型	指定執行類型。 ■00**H : 二進制寫入 • 0000H: BIN16位元資料 • 0001H: BIN32位元資料 ■01**H : CSV格式轉換寫入 • 0100H: 10進制數(帶符號16位元資料) • 0101H: 10進制數(無符號16位元資料) • 0110H: 10進制數(帶符號32位元資料) • 0111H: 10進制數(無符號32位元資料) • 0120H: 16進制數(16位元資料) • 0121H: 16進制數(32位元資料) • 0130H: 字元串(ASCII資料) • 0140H: 浮點實數(單精度實數) • 0141H: 浮點實數(雙精度實數)	0000H 0001H 0100H 0101H 0110H 0111H 0120H 0121H 0130H 0140H 0141H	用戶						
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H: 正常完成 • 0000H以外: 異常完成(出錯代碼)(參見 680頁 在檔案操作指令中發生的出錯代碼)	—	系統						
+2	寫入結果資料數	對(s3)中指定的資料, 輸入實際寫入的資料數。 值的單位取決於(d1)+7的資料類型指定。	—	系統						
+3	使用用途設定區域*4	<table border="1" style="margin-left: 20px;"> <tr> <td style="text-align: right;">b15</td> <td style="text-align: center;">...</td> <td style="text-align: left;">b0</td> </tr> <tr> <td colspan="2" style="text-align: center;">0</td> <td style="text-align: center;">1/0</td> </tr> </table> b0: 寫入開始位置設定*5 當(d1)+0中指定“二進制寫入”時, 必須指定0。 當(d1)+0中指定了“CSV格式轉換寫入”, (d1)+4, (d1)+5中指定了“添加到檔案的最後”時, 應指定開始寫入的位置。 • 0: 添加到檔案的最後 • 1: 將檔案的最後的換行代碼轉換為逗號並添加(從最後的行的後續開始寫入)	b15	...	b0	0		1/0	如左所示	用戶
b15	...	b0								
0		1/0								
+4 +5	檔案位置	■ 當(d1)+0中指定“二進制寫入”時 • 00000000H: 從檔案的起始開始 • 00000001H~FFFFFFFH: 從指定位置開始(單位取決於(d1)+7的資料類型指定。) • FFFFFFFFH: 添加到檔案的最後 ■ (d1)+0中指定了“CSV格式轉換寫入”時 • 00000000H~FFFFFFFH: 從檔案的起始開始 • FFFFFFFFH: 添加到檔案的最後	00000000H~ FFFFFFFH	用戶						
+6	欄數指定	當(d1)+0中指定“二進制寫入”時, 必須指定0。 當(d1)+0中指定了“CSV格式轉換寫入”時, 應指定進行寫入的欄數。 • 0: 無欄位。變為1列。 • 0以外: 變為指定數的欄位。	0000H~FFFFH (0~65535)	用戶						
+7	資料類型指定	• 0: 字元 • 1: 偶數字節*2 • 2: (d1)+0中指定的資料類型的單位 • 3: 奇數字節*1*2*3 *0: 字元”及“1: 偶數字節”, “3: 奇數字節”僅在(d1)+0中指定了“0000H: BIN16位元資料”或“0100H: 10進制數(帶符號16位元資料)”的情況下可以指定。	0、1、2、3	用戶						

*1 奇數字節的指定, 只有下述機型可以使用。此外, 根據機型不同所對應的韌體版本也不同。

- R01CPU、R02CPU: “19”及其以後
- R04CPU、R08CPU、R16CPU、R32CPU、R120CPU: “51”及其以後
- RnENCPU: “51”及其以後

*2 資料類型指定(d1)+7中指定了“1: 偶數字節”或者“3: 奇數字節”時, 請求寫入資料數(s3)+0的設定範圍為1~32767。

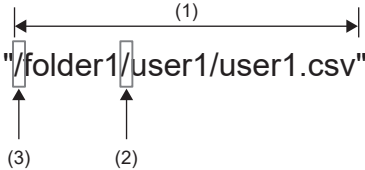
*3 指定奇數字節(d1)+7時對欲寫入的奇數字節資料數加入1字節, 使其成為字元單位。並應將已成為字元單位的資料數指定為請求寫入資料數(s3)+0。

*4 只有下述機型可以使用。此外, 根據機型不同, 所對應的韌體版本也不同。

- R01CPU、R02CPU: “24”及其以後
- Rn(EN)CPU(R01CPU、R02CPU除外): “57”及其以後

*5 僅在執行/完成類型(d1)+0中指定了“CSV格式轉換寫入”, 且檔案位置(d1)+4, (d1)+5中指定了“添加到檔案的最後”時生效。除此以外的情況將出錯。

• 檔案名稱 (s2)

操作數: (s2)				
元件	項目	內容	設定範圍	設定方
+0~+□	檔案名稱字元串	<p>指定儲存了檔案的資料夾路徑、檔案名稱的字元串。</p> <ul style="list-style-type: none"> 資料夾路徑+檔案名稱(包含副檔名)不應超過253字元。 資料夾路徑不應超過244字元。(不包含檔案的分隔符號。) 資料夾路徑的階層數不應超過10層。 省略副檔名的情況下應從“.”(點號)開始省略。 檔案名不應超過60字元+點號+3字元。檔案名超過61字元時, 即使有副檔名也將被忽略, 變為“.BIN”或“.CSV”。  <p>(1): 最大253字元 (2): 資料夾路徑、檔案的分隔符號使用“/”或“\”。 (3): 可以省略。省略的情況下, (1)最大為252字元。</p>	Unicode字元串	用戶

• 寫入資料 (s3)

操作數: (s3)				
元件	項目	內容	設定範圍	設定方
+0	請求寫入資料數	<p>指定請求寫入的資料數。</p> <p>處理單位及設定範圍根據(d1)+0的執行/完成類型有所不同。</p> <p>■當(d1)+0中指定“二進制寫入”時</p> <ul style="list-style-type: none"> BIN16位元資料指定時: 字元單位(1~65535)^{*6*7} BIN32位元資料指定時: 雙字元單位(1~32767) <p>■(d1)+0中指定了“CSV格式轉換寫入”時: 要素數</p> <ul style="list-style-type: none"> 指定10進制數(帶符號16位元資料)時: 字元單位(1~65535)^{*6*7} 指定10進制數(無符號16位元資料)時: 字元單位(1~65535) 指定10進制數(帶符號32位元資料)時: 雙字元單位(1~32767) 指定10進制數(無符號32位元資料)時: 雙字元單位(1~32767) 16進制數(16位元資料)指定時: 字元單位(1~65535) 16進制數(32位元資料)指定時: 雙字元單位(1~32767) 字元串(ASCII資料)指定時: 要素數(1~1023) 浮點實數(單精度實數)指定時: 雙字元單位(1~32767) 浮點實數(雙精度實數)指定時: 4字元單位(1~16383) 	1~65535	用戶
+1~+□	寫入資料	儲存請求寫入的資料。	0000H~FFFFH	用戶

*6 資料類型指定(d1)+7中指定了“1: 偶數字節”或者“3: 奇數字節”時, 請求寫入資料數(s3)+0的設定範圍為1~32767。

*7 指定奇數字節(d1)+7時對欲寫入的奇數字節資料數加入1字節, 使其成為字元單位。並應將已成為字元單位的資料數指定為請求寫入資料數(s3)+0。

■類型2

• 控制資料 (d1)

操作數: (d1)				
元件	項目	內容	設定範圍	設定方
+0	執行/完成類型	指定執行類型。 • 0000H: 二進制寫入 • 0100H: CSV格式轉換寫入	0000H 0100H	用戶
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H: 正常完成 • 0000H以外: 異常完成(出錯代碼) (參見 680頁 在檔案操作指令中發生的出錯代碼)	—	系統
+2	寫入結果資料數	對(s3)中指定的資料, 輸入實際寫入的資料數。 值的單位取決於(d1)+7的資料類型指定。	—	系統
+3	未使用	—	—	—
+4 +5	檔案位置	■當(d1)+0中指定“二進制寫入”時 • 00000000H: 從檔案的起始開始 • 00000001H~FFFFFFFH: 從指定位置開始(單位取決於(d1)+7的資料類型指定。) • FFFFFFFFH: 添加到檔案的最後 ■(d1)+0中指定了“CSV格式轉換寫入”時 • 00000000H~FFFFFFFH: 從檔案的起始開始 • FFFFFFFFH: 添加到檔案的最後	00000000H~ FFFFFFFH	用戶
+6	欄數指定	當(d1)+0中指定“二進制寫入”時, 必須指定0。 當(d1)+0中指定了“CSV格式轉換寫入”時, 應指定進行寫入的欄數。 • 0: 無欄位。變為1列。 • 0以外: 變為指定數的欄位。	0000H~FFFFH (0~65535)	用戶
+7	資料類型指定	0: 字元 1: 偶數字節	0、1	用戶

• 檔案名稱 (s2)

操作數: (s2)				
元件	項目	內容	設定範圍	設定方
+0~+□	檔案名稱字元串	指定檔案名稱的字元串。 省略副檔名的情況下應從“.”(點號)開始省略。 不應超過60字元+點號+3字元。 超過61字元時, 即使有副檔名也將被忽略, 變為“.BIN”或“.CSV”。	Unicode字元串	用戶

• 寫入資料 (s3)

操作數: (s3)				
元件	項目	內容	設定範圍	設定方
+0	請求寫入資料數	指定請求寫入的資料數。(字元單位) 在(d1)+7的資料類型指定中指定了“偶數字節”的情況下, 也要進行字元換算並以字元單位進行設定。	1~65535*1	用戶
+1~+□	寫入資料	儲存請求寫入的資料。	0000H~FFFFH	用戶

*1 資料類型指定(d1)+7為“1: 偶數字節”的情況下, 設定範圍為1~32767。

功能

- 將指定資料數的資料寫入到指定的檔案中。透過控制資料的執行/完成類型，指定檔案的寫入形式。

使用的CPU模組		參閱目標
RnCPU	R01CPU、R02CPU	647頁 類型1
	R04CPU、R08CPU、R16CPU、R32CPU、R120CPU	韌體版本“28”及其以後 647頁 類型1 韌體版本“27”及其以前 649頁 類型2
RnENCPU		韌體版本“28”及其以後 647頁 類型1 韌體版本“27”及其以前 649頁 類型2
RnPCPU		649頁 類型2
RnPSFCPU		649頁 類型2
RnSFCPU		649頁 類型2

- 寫入對象僅為SD記憶卡。
- 處理完成(d2)的位元元件，在檢測出SP.FWRITE指令的處理完成掃描的END指令執行時將被自動設為ON，透過下一個掃描的END指令設為OFF。
- 當SP.FWRITE指令異常完成時，異常完成(d2)+1的元件將與處理完成(d2)的元件在相同的時機設為ON/OFF。
- SP.FWRITE指令執行中，SM753(檔案存取中)將變為ON。
- SM753為ON中，不能執行SP.FWRITE指令。(執行的情況下將變為無處理。)
- 對於執行指令時檢測出的出錯，處理完成(d2)、異常完成(d2)+1及SM753不變為ON。
- 請求寫入檔案數((s3)+0)、檔案位置((d1)+4、(d1)+5)及寫入結果資料數((d1)+2)的處理單位，根據執行/完成類型(d1)+0及資料類型指定(d1)+7的組合進行指定。

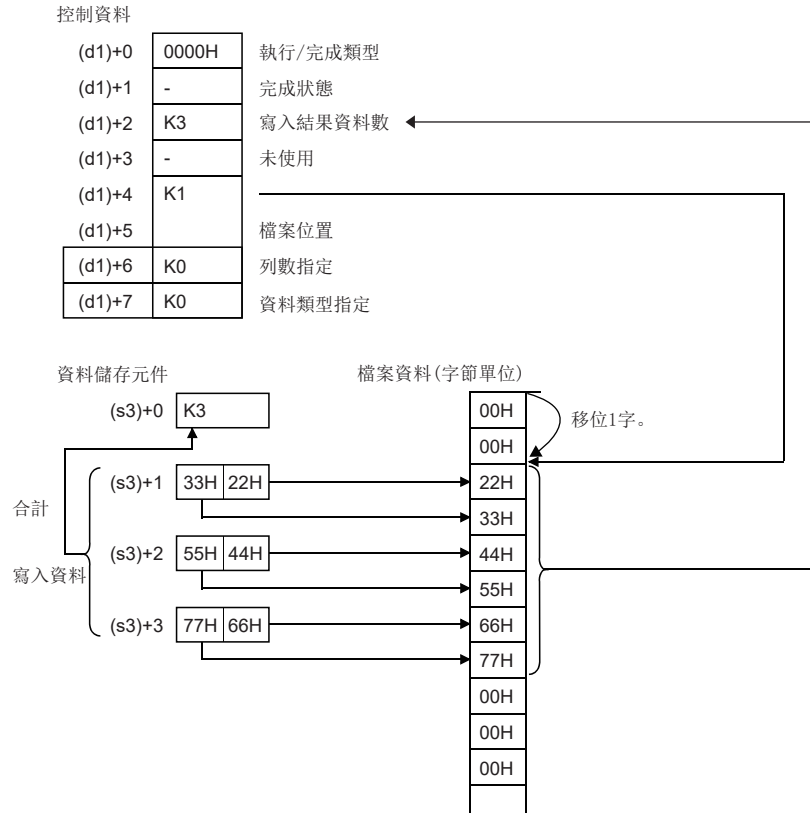
執行/完成類型 (d1)+0	資料類型指定 (d1)+7	處理單位及設定範圍			
		請求寫入資料數 (s3)+0	檔案位置 (d1)+4、(d1)+5	寫入結果資料數 (d1)+2	
二進制寫入	0000H: BIN16位元資料	0: 字元	字元(1~65535)	字元(00000000H~7FFFFFFFH、FFFFFFFFH)	字元
		1: 偶數字節	字元(1~32767)	字節(00000000H~FFFFFFFFH)	字節
		2: 透過執行/完成類型指定的資料類型的單位	字元(1~65535)	字元(00000000H~7FFFFFFFH、FFFFFFFFH)	字元
		3: 奇數字節	字元(1~32767)	字節(00000000H~FFFFFFFFH)	字節
	0001H: BIN32位元資料	0: 字元 1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	雙字元(1~32767)	雙字元(00000000H~3FFFFFFFH、FFFFFFFFH)	雙字元
		3: 奇數字節	(不能指定)		

執行/完成類型 (d1)+0		資料類型指定 (d1)+7	處理單位及設定範圍		
			請求寫入資料數 (s3)+0	檔案位置 (d1)+4、(d1)+5	寫入結果資料數 (d1)+2
CSV格式轉換寫入	0100H: 10進制數(帶符號16位元資料)	0: 字元	字元(1~65535)	起始/末尾*2	字元
		1: 偶數字節	字元(1~32767)	起始/末尾*2	字節
		2: 透過執行/完成類型指定的資料類型的單位	字元(1~65535)	起始/末尾*2	字元
		3: 奇數字節	字元(1~32767)	起始/末尾*2	字節
	0101H: 10進制數(無符號16位元資料)	0: 字元	(不能指定)		
		1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	字元(1~65535)	起始/末尾*2	字元
	0110H: 10進制數(帶符號32位元資料)	0: 字元	(不能指定)		
		1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	雙字元(1~32767)	起始/末尾*2	雙字元
	0111H: 10進制數(無符號32位元資料)	0: 字元	(不能指定)		
		1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	雙字元(1~32767)	起始/末尾*2	雙字元
	0120H: 16進制數(16位元資料)	0: 字元	(不能指定)		
		1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	字元(1~65535)	起始/末尾*2	字元
	0121H: 16進制數(32位元資料)	0: 字元	(不能指定)		
		1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	雙字元(1~32767)	起始/末尾*2	雙字元
	0130H: 字元串(ASCII資料)	0: 字元	(不能指定)		
		1: 偶數字節	(不能指定)		
		2: 透過執行/完成類型指定的資料類型的單位	要素數(1~1023)	起始/末尾*2	要素數
	0140H: 浮點實數(單精度實數)	0: 字元	(不能指定)		
		1: 偶數字節	(不能指定)		
2: 透過執行/完成類型指定的資料類型的單位		雙字元(1~32767)	起始/末尾*2	雙字元	
0141H: 浮點實數(雙精度實數)	0: 字元	(不能指定)			
	1: 偶數字節	(不能指定)			
	2: 透過執行/完成類型指定的資料類型的單位	4字元(1~16383)	起始/末尾*2	4字元	
		3: 奇數字節	(不能指定)		

*2 設定值為00000000H~FFFFFFFFEH(檔案的起始開始)或FFFFFFFFFH(添加在檔案的最後)的其中之一。

■二進制寫入時

- 在省略對象檔案的副檔名的情況下，副檔名將變為“.BIN”。
- 在指定不存在的檔案的情況下，在新建相應檔案後，從起始開始添加儲存資料。此時新建檔案的屬性被設定為存檔屬性。
- 在指定存在的檔案的情況下，將從存在的檔案的起始開始進行儲存。在寫入資料的過程中，若其超過現有大小的情況下，超過部分的資料將被添加儲存。
- 與現有檔案的大小相比，在指定更大檔案位置的情況下，將變為以0點寫入而正常完成。
- 在添加儲存資料的過程中，若有來自於媒介的空餘區域不足之情況，將變為出錯狀態。此時，寫入及添加儲存成功的部分將變為原樣寫入狀態，且在添加儲存可添加儲存的部分後異常結束。
- 在二進制寫入(BIN16位元資料)的情況下指定了請求寫入資料數、檔案位置時的資料寫入方法如下所示。

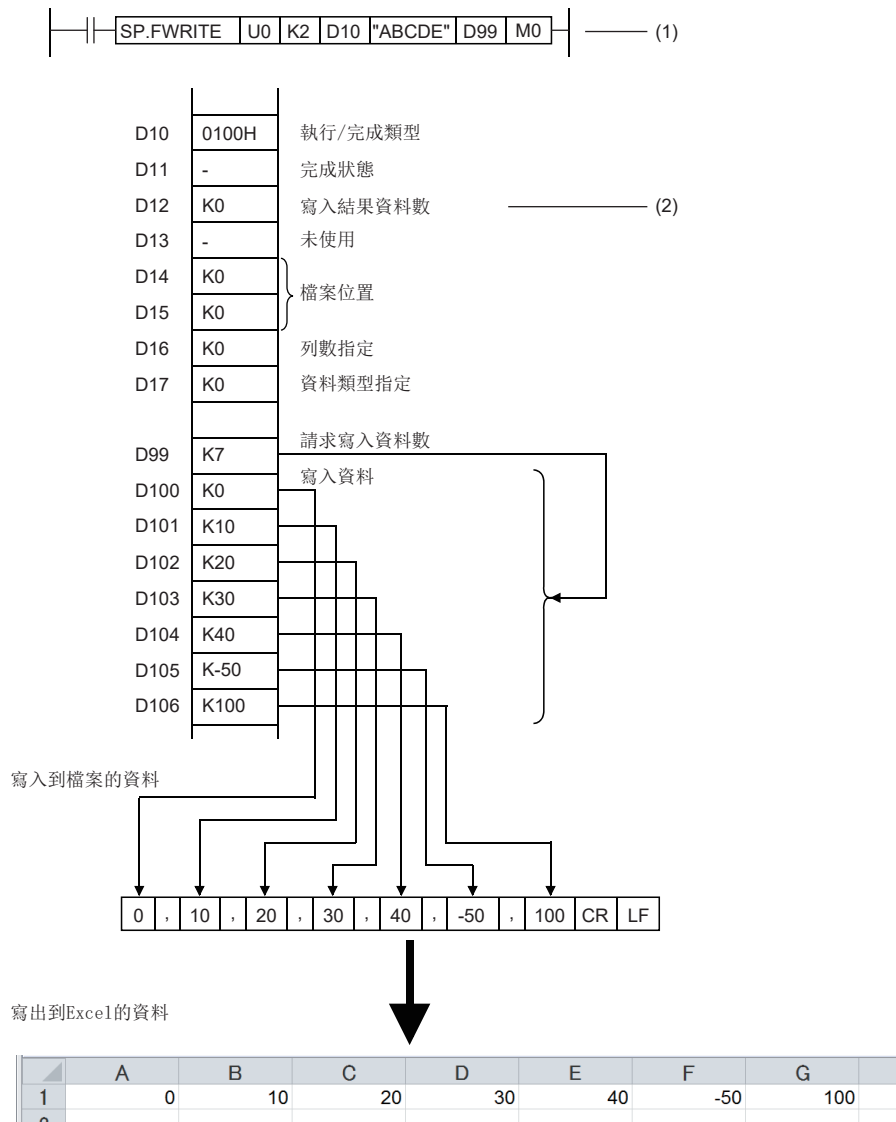


■CSV格式轉換寫入時

- 在省略副檔名的情況下，副檔名將變為“.CSV”。
- 在指定存在的檔案時，其情況如下所示。
 - 當(d1)+4、(d1)+5中設定FFFFFFFF以外時，將刪除全部檔案內容後從起始開始儲存資料。
 - 當(d1)+4、(d1)+5中設定了FFFFFFFF時，將從檔案的最後開始儲存資料。此時，(d1)+3中指定了1時，將檔案最後的換行代碼轉換為逗號，並保存資料。但是，檔案的最後為換行代碼以外時則不轉換為逗號，從檔案的最後保存資料。
- 在指定不存在的檔案的情況下，在新建相應檔案後，從起始開始添加儲存資料。此時的新建檔案的屬性將被設定為存檔屬性。
- 在資料添加儲存的途中，若有來自於媒介的空餘區域不足的情況，將變為出錯狀態。此時，添加儲存成功的部分將變為原樣寫入狀態，在添加儲存可添加儲存的部分後異常結束。
- 在指定欄數為0的情況下，作為1列的CSV格式檔案儲存。

例

在CSV格式轉換寫入時指定欄數為0的情況下



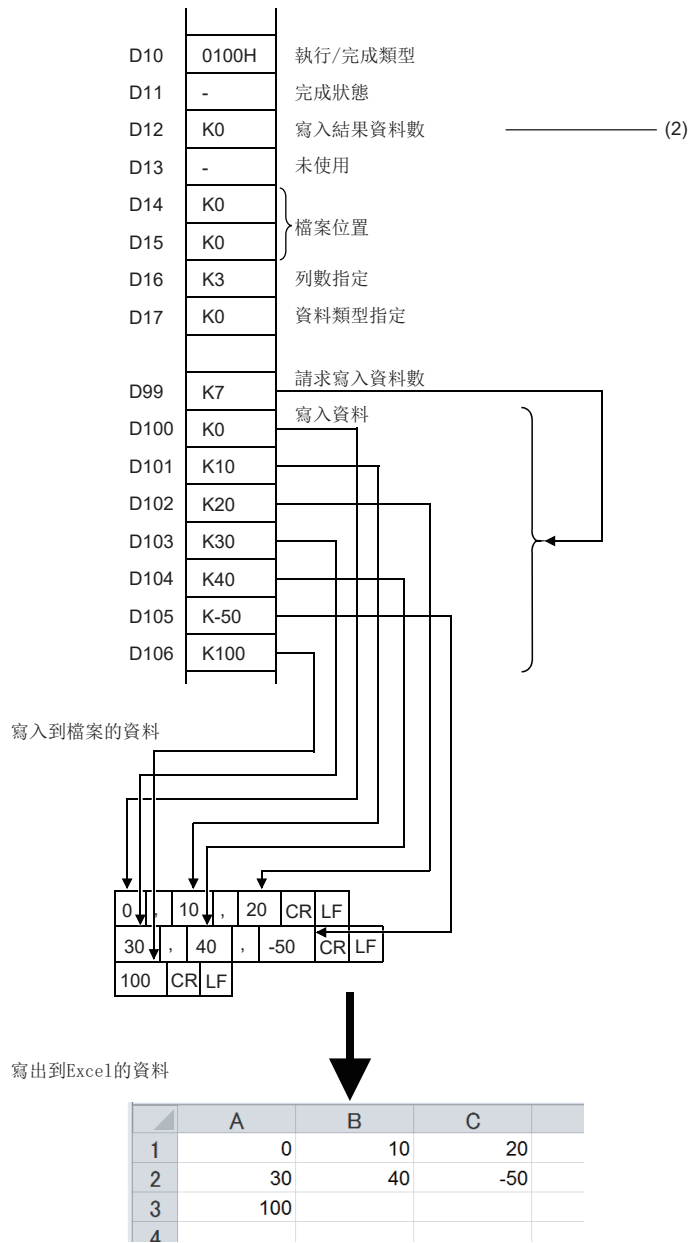
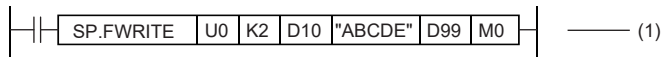
(1) 以字元單位進行指定。

(2) 正常完成時，變為與寫入資料數相同。

- 在指定欄數為0以外的情況下，作為指定欄數的表儲存CSV格式檔案。

例

在CSV格式轉換寫入時指定欄數為0以外的情況下

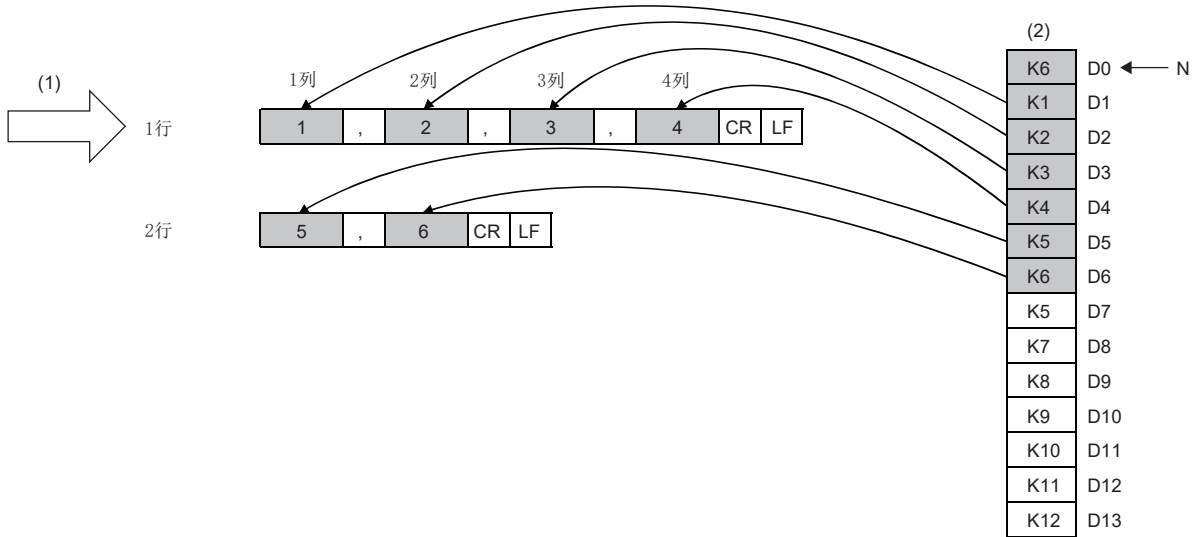


- (1) 以字元單位進行指定。
- (2) 正常完成時，變為與寫入資料數相同。

- 添加資料時的情況如下所示。

[指定進行寫入的檔案] (即使檔案已存在，也將其刪除後再次新建。)

- 執行/完成類型：CSV格式轉換寫入(10進制數(帶符號16位元資料))
- 檔案位置：00000000H(從檔案的起始開始)
- 欄數指定：4H
- 資料類型指定：字元
- 寫入起始元件：D0
- 請求寫入資料數：6H



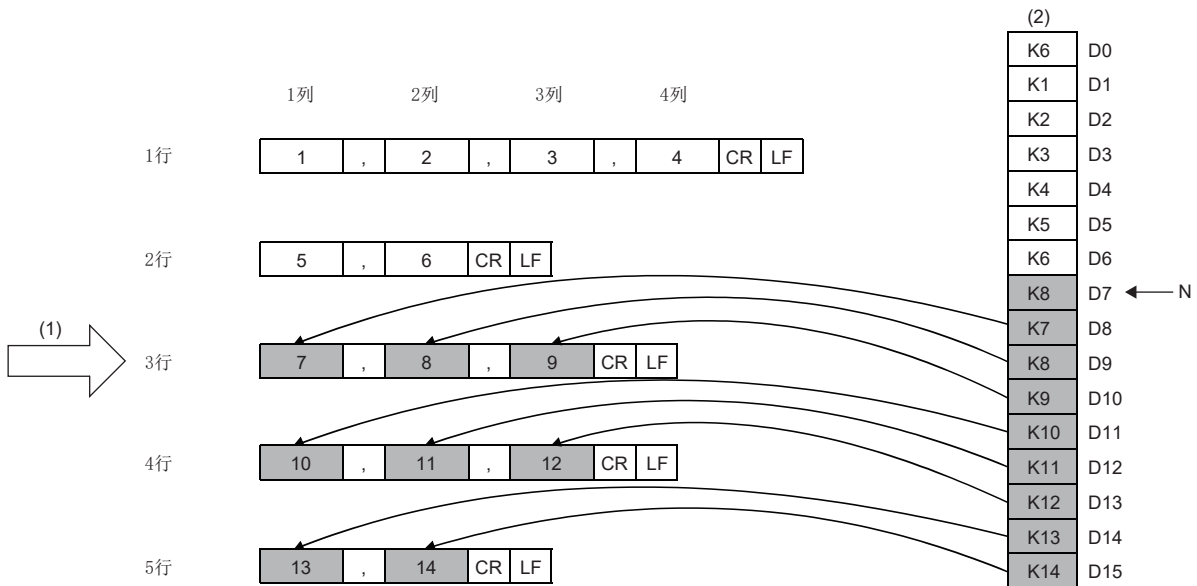
(1): 開始列

(2): 元件儲存資料(寫入資料)

N: 資料數

[添加到檔案的最後]

- 執行/完成類型：CSV格式轉換寫入(10進制數(帶符號16位元資料))
- 使用用途設定區域：0H
- 檔案位置：FFFFFFFH(添加到檔案的最後)
- 欄數指定：3H
- 資料類型指定：字元
- 寫入起始元件：D7
- 請求寫入資料數：8H



(1): 本次數開始位置

(2): 元件儲存資料(寫入資料)

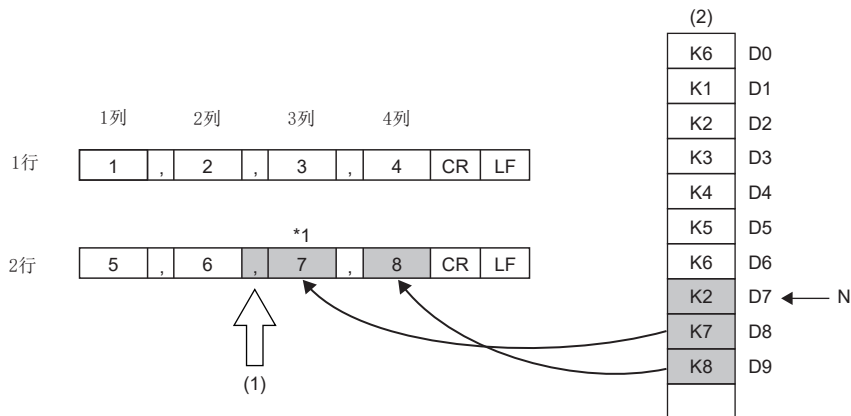
N: 資料數

要點

- 對於“請求寫入資料數”，應設定為“欄數指定”的整數倍。若不是整數倍，欄數將錯亂。
- 因為最後資料的後面必須輸入換行代碼，因此使用用途設定區域為0時，將從新的一行的起始開始添加。
- 在添加到檔案的最後的情況下，若將“欄數指定”從上次寫入時進行更改時，欄數將錯亂。

[從檔案最後的行的後續開始添加]

- 執行/完成類型：CSV格式轉換寫入(10進制數(帶符號16位元資料))
- 使用用途設定區域：1H
- 檔案位置：FFFFFFFH(添加到檔案的最後)
- 資料類型指定：字元
- 寫入起始元件：D7
- 請求寫入資料數：2H



(1): 本次開始位置

(2): 元件儲存資料(寫入資料)

N: 資料數

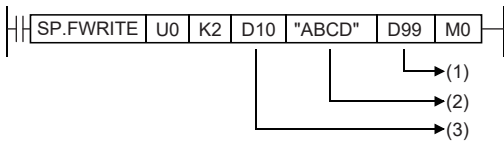
*1 使用用途設定區域設定為1時，檔案末尾的換行代碼將轉換為逗號，可以從最後行的後續開始添加資料。

• 執行/完成類型中指定“字元串(ASCII資料)”的示例如下所示。

例

CSV格式寫入(字元串(ASCII資料))的情況下

[寫入檔案的資料]

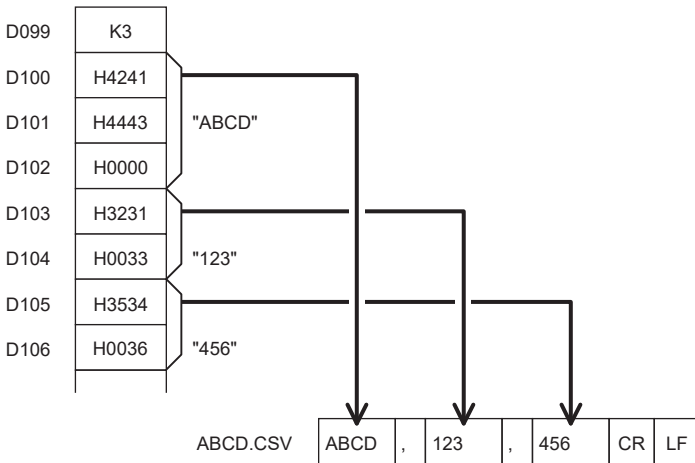


- (1): 寫入資料
- (2): 檔案名
- (3): 控制資料

[控制資料]

D10	H0130
D11	H0000
D12	-
D13	-
D14	K0
D15	K0
D16	K0
D17	K2

- D10: 執行/完成類型: 字元串(ASCII資料)
- D11: 完成狀態
- D12: 寫入結果資料數
- D13: (未使用)
- D14、D15: 檔案位置
- D16: 欄數指定
- D17: 資料類型指定



- D99: 請求寫入資料數
- D100~D102: 第1列第1欄中寫入的字元串
- D103~D104: 第1列第2欄中寫入的字元串
- D105~D106: 第1列第3欄中寫入的字元串

	A	B	C	D
1	ABCD	123	456	
2				

要點

- 必須在1要素的字元串的終端設定00H(NULL)。字元串為偶數字節的情況下，應在下一個1字元中設定0000H(2字節份的NULL)。
- 1要素的最大字元數為1999字元。超過此字元數且未儲存00H(NULL)的情況下，第2000字元及其以後將不會被寫入，而進行下個要素的寫入。
- 1次指令執行中可寫入的最大要素數為1023。

- 執行/完成類型((d1)+0)中設定“0140H: 浮點實數(單精度實數)”或“0141H: 浮點實數(雙精度實數)”的情況下, (s3)+1及其以後設定的值與寫入至CSV檔案的內容如下所示。

執行/完成類型((d1)+0)	寫入檔案((s3)+1及其以後)中設定的值	寫入CSV檔案的內容
0140H: 浮點實數(單精度實數)	$-2^{128} < \text{資料} \leq -2^{126}$ 、 0、 $2^{-126} \leq \text{資料} < 2^{128}$ 的範圍的值	左邊記載的值(小數部分位數0~7)以指數形式寫入。
	上述以外的數值	由於無法轉換因此寫入0。
0141H: 浮點實數(雙精度實數)	$-2^{1024} < \text{資料} \leq -2^{1022}$ 、 0、 $2^{-1022} \leq \text{資料} < 2^{1024}$ 的範圍的值	左邊記載的值(小數部分位數0~15)以指數形式寫入。
	上述以外的數值	由於無法轉換因此寫入0。

- SD記憶卡中寫入CSV格式檔案時的檔案大小(合計字節數)的計算方法如下所示。

[合計字節數]=[最終列以外的合計字節數]+[最終列的字節數]

([各列的字節數]=[欄數*^{*2}]+1+[列內的各資料值的字節數的合計*^{*3}])

- *2 最終列以外將變為指定的欄數。對於最終列的欄數, 因為寫入資料數指定的欄數有可能不同, 因此按下述方式計算。

- 計算除去最終列的列數。(除去最終列的列數=請求寫入資料數÷欄數(餘數捨去))
- 計算最終列的欄數。(最終列的欄數=請求寫入資料數-(除去最終列的列數×欄數))

- *3 各資料值的字節數按下表計算。

資料值的符號	各資料值的字節數	字節數的範圍	示例
正	位數	1~5(字元指定時) 1~3(字節指定時)	• 12345→5字節 • 67→2字節
負	位數+1	2~6(字指定時) 2~4(字節指定時)	• -12345→6字節 • -67→3字節

注意事項

- 對於SP.FWRITE指令, 應勿透過中斷程式執行。若透過中斷程式執行, 可能導致誤動作。
- 由於發生檔案寫入, 因此執行SP.FWRITE指令可能延長掃描時間。

出錯

出錯代碼 (SD0)	內容
2820H	當請求寫入資料數 (s3)+0 中指定的值超出設定範圍時，或超出 (s3)+1 及其以後的元件/標籤記憶體的各設定區域時。
3405H	當透過驅動器指定 (s1) 指定的驅動器為SD記憶卡以外時。 當控制資料 (d1) 及其以後設定的值超出設定範圍時。 ☞ 647頁 類型1 ☞ 649頁 類型2 無法讀取 (s2) 中指定的檔案名字元串時。 <ul style="list-style-type: none">• 指定的檔案名字元串的字元數超出範圍時。• 設定了不能使用的值。• 對已指定的檔案名字元串的末尾指定了分隔符號。 ☞ 647頁 類型1 ☞ 649頁 類型2
3427H	(d1)+0 的執行/完成類型和 (d1)+7 的資料類型指定為不能指定的組合時。 (d1) 的執行/完成類型和 (d1)+3 的寫入開始位置與 (d1)+4 的文檔位置指定為無法指定的組合時。 ☞ 647頁 類型1

在SP.FWRITE指令異常完成的情況下，完成狀態 (d1)+1 中指定的元件中將儲存出錯代碼。(指令運算出錯的情況下不能儲存。)
關於完成狀態 (d1)+1 中儲存的出錯代碼，請參閱下述內容。

☞ 680頁 在檔案操作指令中發生的出錯代碼

刪除指定檔案

SP. FDELETE



- R00CPU中不能使用。
- R01CPU、R02CPU可以在韌體版本為“08”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“40”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

刪除SD記憶卡內指定的檔案或資料夾。

梯形圖	ST
	ENO:=SP_FDELETE(EN, U, s1, s2, d1, d2);

FBD/LD

■執行條件

指令	執行條件
SP. FDELETE	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	元件名	ANY16
(s1)	驅動器指定	2(固定)*1	字元	ANY16
(d1)	儲存了控制資料的起始元件	☞ 662頁 控制資料(d1)	字元	ANY16_ARRAY (要素數: 2)
(s2)	儲存了檔案名稱/資料夾名稱的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(d2)	處理完成時設為ON的位元元件	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能設定SD記憶卡的驅動器2。

功能

- 刪除 (s1) 指定驅動器的 (s2) 中指定的檔案或資料夾。
- SP. FDELETE 指令執行中，SM753 (檔案存取中) 將變為 ON。SM753 為 ON 中，不能執行 SP. FDELETE 指令。(執行的情況下將變為無處理。)
- 處理完成 (d2) 的位元元件，在檢測出 SP. FDELETE 指令的處理完成掃描的 END 指令執行時將被自動設為 ON，透過下一個掃描的 END 指令變為 OFF。
- 當 SP. FDELETE 指令異常完成時，異常完成 (d2)+1 的元件將與處理完成 (d2) 的元件在相同的時機設為 ON/OFF。
- 在執行指令時檢測到運算出錯的情況下，處理完成 (d2)、異常完成 (d2)+1 將不變為 ON。

注意事項

- 請勿透過中斷程式執行 SP. FDELETE 指令。若透過中斷程式執行，可能導致誤動作。
- 指令執行中即使將 CPU 模組的狀態更改為 RUN→STOP，本指令的處理仍會繼續進行。
- 指令在處理中途異常完成的情況下，將無法還原已刪除的檔案或資料夾。
- 刪除對象檔案的檔案大小及檔案數量較大的情況下，指令完成的時間將會變長。
- 請勿從其他功能中存取 SP. FDELETE 指令操作中的檔案。(可能會造成檔案毀損或是導致出錯。)
- 請勿對其他功能存取中的檔案/資料夾進行操作。

出錯

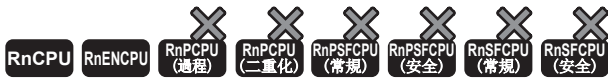
出錯代碼 (SD0)	內容
3405H	<p>當透過驅動器指定 (s1) 指定的驅動器為 SD 記憶卡以外時。</p> <p>無法讀取 (s2) 中指定的檔案名稱/資料夾名稱字元串時。</p> <ul style="list-style-type: none"> • 檔案名稱字元串未指定任何字元。 • 已指定的檔案名稱字元串指定了 254 字元及其以上。 • 已指定的資料夾路徑指定了 245 字元及其以上。 • 已指定的資料夾路徑指定了 11 階層及其以上。 • 在指定的檔案名字元串的末尾或各個分隔符號的前方指定了半形點號。 • 指定檔案時，對已指定的檔案名字元串的末尾指定了分隔符號。 <p>對 (s2) 中指定的資料夾路徑內直接存放於根資料夾下的系統檔案 (\$MELPRJ\$) 進行指定時。</p>

在 SP. FDELETE 指令異常完成的情況下，完成狀態 (d1)+1 中指定的元件中將儲存出錯代碼。(指令運算出錯的情況下不能儲存。)
關於完成狀態 (d1)+1 中儲存的出錯代碼，請參閱下述內容。

☞ 680 頁 在檔案操作指令中發生的出錯代碼

複製指定檔案

SP.FCOPY

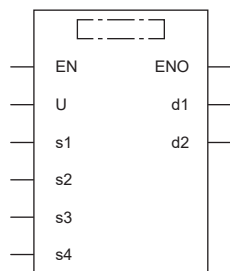


- R00CPU中不能使用。
- R01CPU、R02CPU可以在韌體版本為“08”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“40”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

複製SD記憶卡內指定的檔案或資料夾。指定資料夾時，複製整體資料夾或者資料夾內的所有的檔案、子資料夾。

梯形圖	ST
	<pre>ENO:=SP_FCOPY(EN, U, s1, s2, s3, s4, d1, d2);</pre>

FBD/LD



■執行條件

指令	執行條件
SP.FCOPY	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	元件名	ANY16
(d1)	儲存了控制資料的起始元件	665頁 控制資料 (d1)	字元	ANY16_ARRAY (要素數: 2)
(s1)	指定複製源驅動器	2(固定)*1	字元	ANY16
(s2)	儲存了複製源的檔案名稱/資料夾名稱的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(s3)	指定複製目標驅動器	2(固定)*1	字元	ANY16
(s4)	儲存複製目標資料夾路徑的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(d2)	處理完成時設為ON的位元元件	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能設定SD記憶卡的驅動器2。

操作數: (d1)				
元件	項目	內容	設定範圍	設定方
+0	使用用途設定區域	<p>• b2為1 (複製指定資料夾內的所有的檔案、子資料夾) 時的動作 (指定資料夾內存在子資料夾的情況下) 複製指定資料夾內的檔案、子資料夾。</p> <p>• b2為1 (複製指定資料夾內的所有的檔案、子資料夾) 時的動作 (指定資料夾內不存在子資料夾 (只有檔案) 的情況下) 複製指定資料夾內的檔案、子資料夾。</p> <p>■b1: 覆寫設定 指定複製目標中存在和複製源相同名稱的檔案及資料夾時的動作。</p> <ul style="list-style-type: none"> • 0: 不覆寫 • 1: 覆寫 <p>將b0指定了1 (指定資料夾) 以及將b1指定了0 (不覆寫) 的情況下, 與複製源同名的檔案或資料夾會跳過複製。(不會異常完成。)</p>	如左所示	用戶
+1	完成狀態	<p>指令完成時儲存完成狀態。</p> <ul style="list-style-type: none"> • 0000H: 正常完成 • 0000H以外: 異常完成 (出錯代碼) (參見 680 頁 在檔案操作指令中發生的出錯代碼) 	—	系統

*1 只有下述機型可以使用。此外, 根據機型不同, 所對應的韌體版本也不同。

- R01CPU、R02CPU: “18” 及其以後
- Rn (EN) CPU (除去R01CPU、R02CPU): “50” 及其以後

■複製源檔案名稱/資料夾名稱(s2)

操作數: (s2)				
元件	項目	內容	設定範圍	設定方
+0~+□	檔案名稱/資料夾名稱	<p>指定檔案時，將對儲存了複製源檔案的資料夾路徑+檔案名稱進行指定。 指定資料夾時，將對複製源的資料夾路徑進行指定。</p> <ul style="list-style-type: none"> 檔案名稱應指定至副檔名。 資料夾路徑+檔案名稱(包含副檔名)不應超過253字元。 資料夾路徑不應超過244字元。(不包含檔案的分隔符號。) 檔案名稱或是資料夾路徑，應指定不包含分隔符號至少1個字元。 指定檔案時，請勿在字元串的末尾指定分隔符號。 請勿在字元串的末尾或各個分隔符號的前方指定半形點號。 資料夾路徑的階層數不應超過10層。 <p>■指定檔案的情況下</p> <p>(1): 最大253字元 (2): 資料夾路徑、檔案的分隔符號使用“/”或“\”。 (3): 可以省略。省略的情況下，(1)最大為252字元。</p> <p>■指定資料夾的情況下</p> <p>(4): 最大244字元 (5): 資料夾路徑的分隔符號使用“/”或“\”。 (6): 可以省略。省略的情況下，(4)最大為243字元。 (7): 可以省略</p>	Unicode字元串	用戶

■複製目標資料夾路徑(s4)

操作數: (s4)				
元件	項目	內容	設定範圍	設定方
+0~+□	資料夾路徑	<p>指定複製目標的資料夾路徑。</p> <ul style="list-style-type: none"> 資料夾路徑不應超過244字元。(不包含檔案的分隔符號。) 資料夾路徑，應指定不包含分隔符號至少1個字元。 請勿在字元串的末尾或各個分隔符號的前方指定半形空格。 請勿在字元串的末尾或各個分隔符號的前方指定半形點號。 資料夾路徑的階層數不應超過10層。 <p>(1): 最大244字元 (2): 資料夾路徑的分隔符號使用“/”或“\”。 (3): 可以省略。省略的情況下，(1)最大為243字元。 (4): 可以省略</p>	Unicode字元串	用戶

功能

- 將(s1)指定驅動器的(s2)中指定的檔案或資料夾，複製至(s3)指定驅動器的(s4)中的指定資料夾。指定(s2)中的資料夾時，存在和複製目標相同名稱的檔案或子資料夾的情況下，(d1)的位元1的值(覆寫設定)為0也不會變為異常完成。(跳過複製。)
- (s4)中指定複製目標的資料夾不存在的情況下，資料夾將自動作成。
- SP.FCOPY指令執行中，SM753(檔案存取中)將變為ON。SM753為ON中，不能執行SP.FCOPY指令。(執行的情況下將變為無處理。)
- 處理完成(d2)的位元元件，在檢測出SP.FCOPY指令的處理完成掃描的END指令執行時將被自動設為ON，透過下一個掃描的END指令設為OFF。
- 當SP.FCOPY指令異常完成時，異常完成(d2)+1的元件將與處理完成(d2)的元件在相同的時機設為ON/OFF。
- 在執行指令時檢測到運算出錯的情況下，處理完成(d2)、異常完成(d2)+1將不變為ON。

注意事項

- 請勿透過中斷程式執行SP.FCOPY指令。若透過中斷程式執行，可能導致誤動作。
- 指令執行中即使將CPU模組的狀態更改為RUN→STOP，本指令的處理仍會繼續進行。
- 複製後的資料夾路徑+檔案名稱以及資料夾路徑請勿超出字元數限制。(可能會造成檔案無法存取或是導致出錯。)
- 指令在處理的中途異常完成的情況下，檔案/資料夾可能以處理中的狀態留存。
- 複製對象檔案的檔案大小及檔案數量較大的情況下，指令完成的時間將會變長。
- 請勿從其他功能中存取SP.FCOPY指令操作中的檔案。(可能會造成檔案毀損或是導致出錯。)
- 請勿對其他功能存取中的檔案/資料夾進行操作。

出錯

出錯代碼(SD0)	內容
3405H	當透過驅動器指定(s1)，(s3)指定的驅動器為SD記憶卡以外時。 無法讀取在(s2)中指定的檔案名稱/資料夾名稱字元串時。 <ul style="list-style-type: none">檔案名稱字元串未指定任何字元。已指定的檔案名稱字元串指定了254字元及其以上。已指定的資料夾路徑指定了245字元及其以上。已指定的資料夾路徑指定了11階層及其以上。指定檔案時，對已指定的檔案名字元串的末尾指定了分隔符號。在指定的檔案名字元串的末尾或各個分隔符號的前方指定了半形點號。 無法讀取在(s4)中指定的資料夾路徑字元串時。 <ul style="list-style-type: none">資料夾路徑字元串未指定任何字元。已指定的資料夾路徑字元串指定了245字元及其以上。已指定的資料夾路徑字元串指定了11階層及其以上。在指定的資料夾路徑的字元串末尾或各個分隔符號的前方指定了半形空格。在指定的資料夾路徑的字元串末尾或各個分隔符號的前方指定了半形點號。 對(s2)，(s4)中指定的資料夾路徑內直接存放於根資料夾下的系統檔案(\$MELPRJ\$)進行指定時。

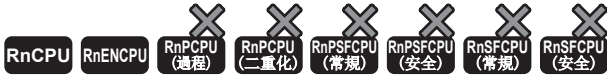
在SP.FCOPY指令異常完成的情況下，完成狀態(d1)+1中指定的元件中將儲存出錯代碼。(指令運算出錯的情況下不能儲存。)

關於完成狀態(d1)+1中儲存的出錯代碼，請參閱下述內容。

☞ 680頁 在檔案操作指令中發生的出錯代碼

移動指定檔案

SP.FMOVE

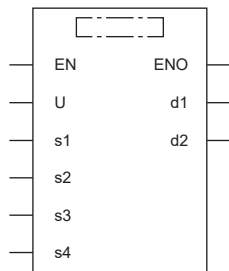


- R00CPU中不能使用。
- R01CPU、R02CPU可以在韌體版本為“08”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“40”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

移動SD記憶卡內指定的檔案或資料夾。指定資料夾時，移動整體資料夾或者資料夾內的所有的檔案、子資料夾。

梯形圖	ST
	ENO: =SP_FMOVE (EN, U, s1, s2, s3, s4, d1, d2);

FBD/LD



■執行條件

指令	執行條件
SP.FMOVE	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(U)	虛擬	—	元件名	ANY16
(d1)	儲存了控制資料的起始元件	670頁 控制資料 (d1)	字元	ANY16_ARRAY (要素數: 2)
(s1)	移動源驅動器指定	2(固定)*1	字元	ANY16
(s2)	儲存了移動源檔案名稱/資料夾名稱的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(s3)	移動目標驅動器指定	2(固定)*1	字元	ANY16
(s4)	儲存移動目標資料夾路徑的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(d2)	處理完成時設為ON的位元元件	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能設定SD記憶卡的驅動器2。

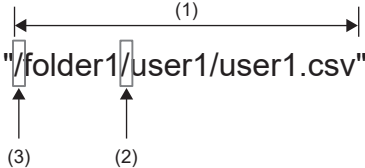
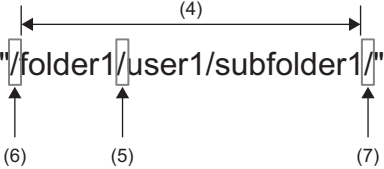
操作數：(d1)

元件	項目	內容	設定範圍	設定方
+0	使用用途設定區域	<p>• b2為1 (移動指定資料夾內的所有檔案、子資料夾) 時的動作 (指定資料夾內存在子資料夾的情況下) 移動指定資料夾內的檔案、子資料夾。</p> <p>• b2為1 (移動指定資料夾內的所有檔案、子資料夾) 時的動作 (指定資料夾內不存在子資料夾 (只有檔案) 的情況下) 移動指定資料夾內的檔案、子資料夾。</p> <p>■b1: 覆寫設定 指定移動目標中存在和移動源相同名稱的檔案及資料夾時的動作。 • 0: 不覆寫 • 1: 覆寫</p>	如左所示	用戶
+1	完成狀態	<p>指令完成時儲存完成狀態。</p> <ul style="list-style-type: none"> • 0000H: 正常完成 • 0000H以外: 異常完成 (出錯代碼) (參見 680 頁 在檔案操作指令中發生的出錯代碼) 	—	系統

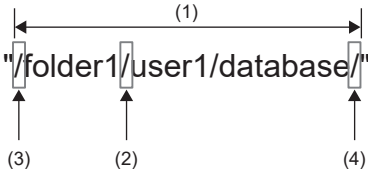
*1 只有下述機型可以使用。此外，根據機型不同所對應的韌體版本也不同。

- R01CPU、R02CPU: “18” 及其以後
- Rn (EN) CPU (除去 R01CPU、R02CPU): “50” 及其以後

■檔案名稱/資料夾名稱 (s2)

操作數: (s2)				
元件	項目	內容	設定範圍	設定方
+0~+□	檔案名稱/資料夾名稱	<p>指定檔案時，將對儲存了移動源檔案的資料夾路徑+檔案名稱進行指定。 指定資料夾時，將對移動源的資料夾的資料夾路徑進行指定。</p> <ul style="list-style-type: none"> 檔案名稱應指定至副檔名。 資料夾路徑+檔案名稱(包含副檔名)不應超過253字元。 資料夾路徑不應超過244字元。(不包含檔案的分隔符號。) 檔案名稱或是資料夾路徑，應指定不包含分隔符號至少1個字元。 指定檔案時，請勿在字元串的末尾指定分隔符號。 請勿在字元串的末尾或各個分隔符號的前方指定半形點號。 資料夾路徑的階層數不應超過10層。 <p>■指定檔案的情況下</p>  <p>(1): 最大253字元 (2): 資料夾路徑、檔案的分隔符號使用“/”或“\”。 (3): 可以省略。省略的情況下，(1)最大為252字元。</p> <p>■指定資料夾的情況下</p>  <p>(4): 最大244字元 (5): 資料夾路徑的分隔符號使用“/”或“\”。 (6): 可以省略。省略的情況下，(4)最大為243字元。 (7): 可以省略</p>	Unicode字元串	用戶

■移動目標資料夾路徑 (s4)

操作數: (s4)				
元件	項目	內容	設定範圍	設定方
+0~+□	資料夾路徑	<p>指定移動目標的資料夾路徑。</p> <ul style="list-style-type: none"> 資料夾路徑不應超過244字元。(不包含檔案的分隔符號。) 資料夾路徑，應指定不包含分隔符號至少1個字元。 請勿在字元串的末尾或各個分隔符號的前方指定半形空格。 請勿在字元串的末尾或各個分隔符號的前方指定半形點號。 資料夾路徑的階層數不應超過10層。  <p>(1): 最大244字元 (2): 資料夾路徑、檔案的分隔符號使用“/”或“\”。 (3): 可以省略。省略的情況下，(1)最大為243字元。 (4): 可以省略</p>	Unicode字元串	用戶

功能

- 將(s1)指定的驅動器中(s2)指定的檔案或資料夾，移動至(s3)指定的驅動器中(s4)指定的檔案或資料夾。(s4)指定移動目標的資料夾不存在的情況下，資料夾將自動作成。
- SP.FMOVE指令執行中，SM753(檔案存取中)將變為ON。SM753為ON中，不能執行SP.FMOVE指令。(執行的情況下將變為無處理。)
- 處理完成(d2)的位元元件，在檢測出SP.FMOVE指令的處理完成掃描的END指令執行時將被自動設為ON，透過下一個掃描的END指令設為OFF。
- 當SP.FMOVE指令異常完成時，異常完成(d2)+1的元件將與處理完成(d2)的元件在相同的時機設為ON/OFF。
- 在執行指令時檢測到運算出錯的情況下，處理完成(d2)、異常完成(d2)+1將不變為ON。

注意事項

- 請勿透過中斷程式執行SP.FMOVE指令。若透過中斷程式執行，可能導致誤動作。
- 指令執行中即使將CPU模組的狀態更改為RUN→STOP，本指令的處理仍會繼續進行。
- 移動後的資料夾路徑+檔案名稱以及資料夾路徑請勿超出字元數限制。(可能會造成檔案毀損或是導致出錯。)
- 發生覆寫的情況下，將可能會延長完成指令的時間。
- 發生覆寫的情況下，應確保有與操作對象相同容量大小的可用空間。
- 指令在處理的中途異常完成的情況下，檔案/資料夾可能以處理中途的狀態留存。
- 發生覆寫時，移動對象檔案的檔案大小及檔案數量較大的情況下，指令完成的時間將會變長。
- 請勿從其他功能中存取SP.FMOVE指令操作中的檔案。(可能會造成檔案無法存取或是導致出錯。)
- 請勿對其他功能存取中的檔案/資料夾進行操作。

出錯

出錯代碼(SD0)	內容
3405H	<p>當透過驅動器指定(s1)指定的驅動器為SD記憶卡以外時。</p> <p>無法讀取(s2)中指定的檔案名稱/資料夾名稱字元串時。</p> <ul style="list-style-type: none"> 檔案名稱字元串未指定任何字元。 已指定的檔案名稱字元串指定了254字元及其以上。 已指定的資料夾路徑指定了245字元及其以上。 已指定的資料夾路徑指定了11階層及其以上。 指定檔案時，對已指定的檔案名字元串的末尾指定了分隔符號。 在指定的檔案名字元串的末尾或各個分隔符號的前方指定了半形點號。 <p>無法讀取在(s4)中指定的資料夾路徑字元串時。</p> <ul style="list-style-type: none"> 資料夾路徑字元串未指定任何字元。 已指定的資料夾路徑字元串指定了245字元及其以上。 已指定的資料夾路徑字元串指定了11階層及其以上。 在指定的資料夾路徑的字元串末尾或各個分隔符號的前方指定了半形空格。 在指定的資料夾路徑的字元串末尾或各個分隔符號的前方指定了半形點號。 <p>對(s2)，(s4)中指定的資料夾路徑內直接存放於根資料夾下的系統檔案(\$MELPRJ\$)進行指定時。</p>

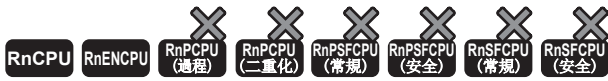
在SP.FMOVE指令異常完成的情況下，完成狀態(d1)+1中指定的元件中將儲存出錯代碼。(指令運算出錯的情況下不能儲存。)

關於完成狀態(d1)+1中儲存的出錯代碼，請參閱下述內容。

☞ 680頁 在檔案操作指令中發生的出錯代碼

更改指定檔案名稱

SP. FRENAME



- R00CPU中不能使用。
- R01CPU、R02CPU可以在韌體版本為“08”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“40”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

更改SD記憶卡內指定的檔案名稱或資料夾名稱。

梯形圖	ST
	<pre>ENO:=SP_FRENAME (EN, U, s1, s2, s3, d1, d2);</pre>

FBD/LD

■執行條件

指令	執行條件
SP. FRENAME	

設置數據

■內容、範圍、資料類型

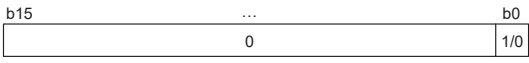
操作數	內容	範圍	資料類型	資料類型 (標籤)
(U)	虛擬	—	元件名	ANY16
(s1)	驅動器指定	2 (固定)*1	字元	ANY16
(d1)	儲存了控制資料的起始元件	☞ 675頁 控制資料 (d1)	字元	ANY16_ARRAY (要素數: 2)
(s2)	儲存了更改後的檔案名稱/資料夾名稱的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(s3)	儲存了所要更改的檔案名稱/資料夾名稱的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(d2)	處理完成時設為ON的位元元件	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能設定SD記憶卡的驅動器2。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其它 (U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(U)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	—	○	—	—	—	—	○	○	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(s3)	—	—	○	—	—	—	—	○	—	—	○	—
(d2)	○	—	○	—	—	—	—	○	—	—	—	—

■控制資料 (d1)

操作數：(d1)				
元件	項目	內容	設定範圍	設定方
+0	使用用途設定區域	 <p>b0: 對象類型設定 將指定對象指定為檔案或資料夾。</p> <ul style="list-style-type: none"> • 0: 指定檔案 • 1: 指定資料夾 	如左所示	用戶
+1	完成狀態	指令完成時儲存完成狀態。	—	系統

■所要更改的檔案名稱/資料夾名稱 (s2)

操作數：(s2)				
元件	項目	內容	設定範圍	設定方
+0~+□	檔案名稱/資料夾名稱	<p>指定檔案時，將對儲存了更改名稱的檔案的資料夾路徑+檔案名稱進行指定。 指定資料夾時，將對儲存了更改名稱的資料夾的資料夾路徑進行指定。</p> <ul style="list-style-type: none"> • 檔案名稱應指定至副檔名。 • 資料夾路徑+檔案名稱 (包含副檔名) 不應超過253字元。 • 資料夾路徑不應超過244字元。(不包含檔案的分隔符號。) • 檔案名稱或是資料夾路徑，應指定不包含分隔符號至少1個字元。 • 指定檔案時，請勿在字元串的末尾指定分隔符號。 • 請勿在字元串的末尾或各個分隔符號的前方指定半形點號。 • 資料夾路徑的階層數不應超過10層。 <p>■指定檔案的情況下</p> <p>(1): 最大253字元 (2): 資料夾路徑、檔案的分隔符號使用“/”或“\”。 (3): 可以省略。省略的情況下，(1)最大為252字元。</p> <p>■指定資料夾的情況下</p> <p>(4): 最大244字元 (5): 資料夾路徑的分隔符號使用“/”或“\”。 (6): 可以省略。省略的情況下，(4)最大為243字元。 (7): 可以省略</p>	Unicode字元串	用戶

■更改後的檔案名稱/資料夾名稱(s3)

操作數：(s3)				
元件	項目	內容	設定範圍	設定方
+0~+□	檔案名稱/資料夾名稱	指定更改後的檔案名稱，資料夾名稱的字元串。 • 檔案名稱應指定至副檔名。 • 請勿指定資料夾路徑。 • 檔案名稱(包含副檔名)不應超過252字元。 • 資料夾名稱不應超過243字元。 • 檔案名稱或是資料夾名稱，應指定不包含分隔符號至少1個字元。 • 請勿指定分隔符號。 • 請勿在字元串的末尾指定半形點號。	Unicode字元串	用戶

功能

- 將(s1)指定的驅動器的(s2)中指定的檔案名稱，或資料夾名稱更改為(s3)中指定的檔案名稱或資料夾名稱。在(s3)指定的檔案名稱或資料夾名稱已存在的情況下，將異常完成。
- SP.FRENAME指令執行中，SM753(檔案存取中)將變為ON。SM753為ON中，不能執行SP.FRENAME指令。(執行的情況下將變為無處理。)
- 處理完成(d2)的位元元件，在檢測出SP.FRENAME指令的處理完成掃描的END指令執行時將被自動設為ON，透過下一個掃描的END指令設為OFF。
- 當SP.FRENAME指令異常完成時，異常完成(d2)+1的元件將與處理完成(d2)的元件在相同的時機設為ON/OFF。
- 在執行指令時檢測到運算出錯的情況下，處理完成(d2)、異常完成(d2)+1將不變為ON。

注意事項

- 請勿透過中斷程式執行SP.FRENAME指令。若透過中斷程式執行，可能導致誤動作。
- 指令執行中即使將CPU模組的狀態更改為RUN→STOP，本指令的處理仍會繼續進行。
- 更改後的資料夾路徑+檔案名稱以及資料夾路徑請勿超出字元數限制。(可能會造成檔案無法存取或是導致出錯。)
- 請勿從其他功能中存取SP.FRENAME指令操作中的檔案。(可能會造成檔案毀損或是導致出錯。)
- 請勿對其他功能存取中的檔案/資料夾進行操作。

出錯

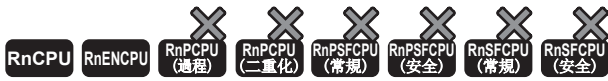
出錯代碼(SD0)	內容
3405H	當透過驅動器指定(s1)指定的驅動器為SD記憶卡以外時。 無法讀取(s2)中指定的檔案名稱/資料夾名稱字元串時。 • 檔案名稱字元串未指定任何字元。 • 已指定的檔案名稱字元串指定了254字元及其以上。 • 已指定的資料夾路徑指定了245字元及其以上。 • 已指定的資料夾路徑指定了11階層及其以上。 • 指定檔案時，對已指定的檔案名字元串的末尾指定了分隔符號。 • 在指定的檔案名字元串的末尾或各個分隔符號的前方指定了半形點號。 無法讀取(s3)中指定的檔案名稱/資料夾名稱字元串時。 • 檔案名稱/資料夾名稱字元串未指定任何字元。 • 已指定的檔案名稱字元串指定了253字元及其以上。 • 已指定的資料夾名稱字元串指定了244字元及其以上。 • 已指定的檔案名稱字元串指定了分隔符號。 • 已指定的檔案名稱字元串末尾指定了半形點號。 對(s2)中指定的資料夾路徑內直接存放於根資料夾下的系統檔案(\$MELPRJ\$)進行指定時。 在(s3)中指定的檔案名稱/資料夾名稱指定為\$MELPRJ\$時。

在SP.FRENAME指令異常完成的情況下，完成狀態(d1)+1中指定的元件中將儲存出錯代碼。(指令運算出錯的情況下不能儲存。)
關於完成狀態(d1)+1中儲存的出錯代碼，請參閱下述內容。

☞ 680頁 在檔案操作指令中發生的出錯代碼

取得指定檔案的狀態

SP.FSTATUS



- R00CPU中不能使用。
- R01CPU、R02CPU可以在韌體版本為“08”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“40”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

取得SD記憶卡內指定的檔案或資料夾的狀態。

梯形圖	ST
	ENO:=SP_FSTATUS (EN, U, s1, s2, d1, d2, d3);

FBD/LD

■執行條件

指令	執行條件
SP.FSTATUS	

設置數據

■內容、範圍、資料類型

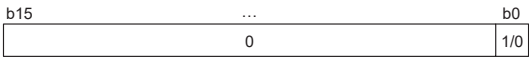
操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	元件名	ANY16
(s1)	驅動器指定	2(固定)*1	字元	ANY16
(d1)	儲存了控制資料的起始元件	☞ 678頁 控制資料(d1)	字元	ANY16_ARRAY (要素數: 2)
(s2)	儲存了檔案名稱/資料夾名稱的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(d2)	儲存檔案狀態的起始元件	—	字元	ANY16_ARRAY (要素數: 10)
(d3)	處理完成時設為ON的位元元件	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能設定SD記憶卡的驅動器2。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其它 (U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(U)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	—	○	—	—	—	—	○	○	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—
(d3)	○	—	○	—	—	—	—	○	—	—	—	—

■控制資料 (d1)

操作數：(d1)				
元件	項目	內容	設定範圍	設定方
+0	使用用途設定區域	 <p>b0: 對象類型設定 將指定對象指定為檔案或資料夾。</p> <ul style="list-style-type: none"> • 0: 指定檔案 • 1: 指定資料夾 	如左所示	用戶
+1	完成狀態	指令完成時儲存完成狀態。	—	系統

■檔案名稱/資料夾名稱 (s2)

操作數：(s2)				
元件	項目	內容	設定範圍	設定方
+0~+□	檔案名稱/資料夾名稱	<p>指定檔案時，將對儲存了取得狀態的檔案的資料夾路徑+檔案名稱進行指定。 指定資料夾時，將對儲存了取得狀態的資料夾的資料夾路徑進行指定。</p> <ul style="list-style-type: none"> • 檔案名稱應指定至副檔名。 • 資料夾路徑+檔案名稱(包含副檔名)不應超過253字元。 • 資料夾路徑不應超過244字元。(不包含檔案的分隔符號。) • 檔案名稱或是資料夾路徑，應指定不包含分隔符號至少1個字元。 • 指定檔案時，請勿在字元串的末尾指定分隔符號。 • 請勿在字元串的末尾或各個分隔符號的前方指定半形點號。 • 資料夾路徑的階層數不應超過10層。 <p>■指定檔案的情況下</p> <p>(1): 最大253字元 (2): 資料夾路徑、檔案的分隔符號使用“/”或“\”。 (3): 可以省略。省略的情況下，(1)最大為252字元。</p> <p>■指定資料夾的情況下</p> <p>(4): 最大244字元 (5): 資料夾路徑的分隔符號使用“/”或“\”。 (6): 可以省略。省略的情況下，(4)最大為243字元。 (7): 可以省略</p>	Unicode字元串	用戶

■檔案狀態 (d2)

操作數：(d2)			
元件	項目	範圍	設定方
+0	檔案屬性 bit0: 讀取專用檔案的情況下為ON bit1: 隱藏檔案的情況下為ON bit2: 系統檔案的情況下為ON bit3: 預留(固定為0) bit4: 目錄的情況下為ON bit5: 保存記錄的情況下為ON bit6~15: 預留(固定為0)	如左所示	系統
+1	預留	0	
+2~+3	檔案大小(字節單位)	0~4294967294*1	
+4	最後更新時間: 年	0, 1980~2079*1	
+5	最後更新時間: 月	0~12	
+6	最後更新時間: 日	0~31	
+7	最後更新時間: 時	0~23	
+8	最後更新時間: 分	0~59	
+9	最後更新時間: 秒		

*1 在CPU模組以外的環境(OS)中存取檔案/資料夾的情況下，所取得的值將依存於此環境(OS)。

功能

- 取得在(s1)指定的驅動器中(s2)指定的檔案或資料夾的狀態，將儲存於(d2)及其之後的元件。(s2)中指定資料夾的情況下，(d2)+2~(d2)+3中將儲存0。
- SP.FSTATUS指令執行中，SM753(檔案存取中)將變為ON。SM753為ON中，不能執行SP.FSTATUS指令。(執行的情況下將變為無處理。)
- 處理完成(d3)的位元元件，在檢測出SP.FSTATUS指令的處理完成掃描的END指令執行時將被自動設為ON，透過下一個掃描的END指令設為OFF。
- 當SP.FSTATUS指令異常完成時，因為異常完成(d3)+1的元件將以與處理完成(d3)的元件相同的時機設為ON/OFF。
- 在執行指令時檢測到運算出錯的情況下，處理完成(d2)、異常完成(d2)+1將不變為ON。

注意事項

- 請勿透過中斷程式執行SP.FSTATUS指令。若透過中斷程式執行，可能導致誤動作。
- 指令執行中即使將CPU模組的狀態更改為RUN→STOP，本指令的處理仍會繼續進行。

出錯

出錯代碼(SD0)	內容
3405H	當透過驅動器指定(s1)指定的驅動器為SD記憶卡以外時。 無法讀取(s2)中指定的檔案名稱/資料夾名稱字元串時。 <ul style="list-style-type: none"> 檔案名稱字元串未指定任何字元。 已指定的檔案名稱字元串指定了254字元及其以上。 已指定的資料夾路徑指定了245字元及其以上。 已指定的資料夾路徑指定了11階層及其以上。 指定檔案時，對已指定的檔案名字元串的末尾指定了分隔符號。 在指定的檔案名字元串的末尾或各個分隔符號的前方指定了半形點號。

在SP.FSTATUS指令異常完成的情況下，完成狀態(d1)+1中指定的元件中將儲存出錯代碼。(指令運算出錯的情況下不能儲存。)關於完成狀態(d1)+1中儲存的出錯代碼，請參閱下述內容。

☞ 680頁 在檔案操作指令中發生的出錯代碼

在檔案操作指令中發生的出錯代碼

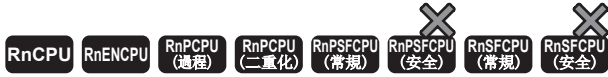
檔案操作指令的完成狀態中儲存的出錯代碼的一覽如下所示。

出錯代碼	出錯內容	處理方法
8000H	SM606 (SD記憶卡強制使用停止指示) 為ON。	在SM606為ON的情況下，將SM606設為OFF後，應進行SD記憶卡強制使用停止解除。
	未安裝SD記憶卡。	應安裝SD記憶卡。
	SD記憶卡未被加載。	應加載SD記憶卡。
8001H	對SD記憶卡的存取失敗。	<ul style="list-style-type: none"> 應確認檔案名字元串的指定是否沒有錯誤。 應確認其他功能中是否存取了檔案。 應採取抗噪聲措施。 將CPU模組重設後，應設為RUN。再次顯示相同出錯的情況下，可能是SD記憶卡的硬體異常。請與附近的三菱電機的分公司、代理商商談。
	SD記憶卡處於寫保護狀態。	應將SD記憶卡的防寫開關設為無效(允許寫入)。
	檔案屬性處於唯讀狀態。	應解除唯讀的設定。
	SD記憶卡中所儲存的檔案大小超過了SD記憶卡的可用空間或最大容量。	<ul style="list-style-type: none"> 應刪除SD記憶卡內不需要的檔案，預留出可用空間。 檔案大小應修改為不超過最大容量。
8002H	指定的檔案或資料夾不存在。	<ul style="list-style-type: none"> 應確認指定的檔案或資料夾是否存在。 應確認指定的資料夾路徑是否存在。
	SD記憶卡的可用空間不足。	應刪除SD記憶卡內不需要的檔案，預留出可用空間。
8003H	從檔案讀取的資料數的合計超過(d1)+3(最大讀取資料數)。	應調整(d1)+2(請求讀取資料數)或(d1)+3(最大讀取資料數)。
8004H	刪除對象資料夾不是空白資料夾。	應確認控制資料的設定。
8005H	存在相同名稱的檔案或資料夾。	應更改檔案名稱或資料名稱。
8006H	<ul style="list-style-type: none"> 執行指令後的資料夾路徑中存在超過244字元的資料夾。 執行指令後的檔案路徑中存在超過253字元的檔案。 	應設定執行指令後的檔案路徑或資料夾路徑中，不超過最大字元數。
	複製目標或移動目標中，指定下述資料夾/子資料夾。 <ul style="list-style-type: none"> 與複製源相同資料夾/子資料夾 與移動源相同資料夾/子資料夾 	應更改複製目標或移動目標的路徑。

8.5 資料控制指令

BIN16位元資料上下限限位控制

LIMIT(P) (_U)



根據指定的輸入值 (BIN16位元值) 是否在上下限限位值的範圍內，對輸出值進行控制。

梯形圖	ST*1	
	ENO:=LIMITP (EN, s1, s2, s3, d);	ENO:=LIMIT_U (EN, s1, s2, s3, d);

FBD/LD*1

*1 在ST、FBD/LD中不支援LIMIT指令、LIMIT_U指令。應使用通用函數的LIMIT。
 1432頁 LIMIT(_E)

■執行條件

指令	執行條件
LIMIT LIMIT_U	
LIMITP LIMITP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	LIMIT(P)	下限限位值(最小輸出極限值)	-32768~32767	帶符號BIN16位元	ANY16_S
	LIMIT(P)_U		0~65535	無符號BIN16位元	ANY16_U
(s2)	LIMIT(P)	上限限位值(最大輸出極限值)	-32768~32767	帶符號BIN16位元	ANY16_S
	LIMIT(P)_U		0~65535	無符號BIN16位元	ANY16_U
(s3)	LIMIT(P)	透過上下限限位控制進行控制的輸入值	-32768~32767	帶符號BIN16位元	ANY16_S
	LIMIT(P)_U		0~65535	無符號BIN16位元	ANY16_U
(d)	LIMIT(P)	儲存透過上下限限位控制進行控制的輸出值的元件	—	帶符號BIN16位元	ANY16_S
	LIMIT(P)_U			無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

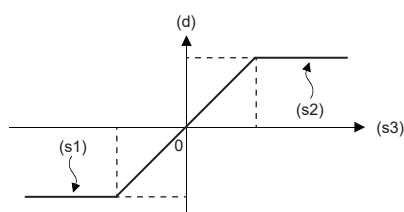
■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- (s3)中指定的輸入值(BIN16位元值)根據(s1)、(s2)中指定的上下限限位值的範圍，對(d)中指定的元件中儲存的輸出值進行控制。輸出值按下述方式被控制。

條件	輸出值中儲存的值
下限限位值(s1) > 輸入值(s3) 時	下限限位值(s1)
上限限位值(s2) < 輸入值(s3) 時	上限限位值(s2)
下限限位值(s1) ≤ 輸入值(s3) ≤ 上限限位值(s2) 時	輸入值(s3)



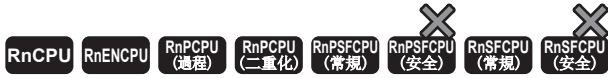
- 僅透過上限限位值進行控制的情況下，在(s1)中指定的下限限位值中設定設定資料範圍的最小值。
- 僅進行下限限位值的控制的情況下，在(s2)中指定的上限限位值中設定設定資料範圍的最大值。

出錯

出錯代碼(SD0)	內容
3405H	(s1)中指定的下限限位值大於(s2)中指定的上限限位值時。

BIN32位元資料上下限限位控制

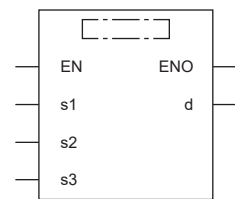
DLIMIT(P) (_U)



根據指定的輸入值 (BIN32位元值) 是否在上下限限位值的範圍內，對輸出值進行控制。

梯形圖	ST*1	
	ENO:=DLIMITP (EN, s1, s2, s3, d);	ENO:=DLIMITP_U (EN, s1, s2, s3, d);

FBD/LD*1



*1 在ST、FBD/LD中不支援DLIMIT指令、DLIMIT_U指令。應使用通用函數的LIMIT。
 1432頁 LIMIT(_E)

■執行條件

指令	執行條件
DLIMIT DLIMIT_U	
DLIMITP DLIMITP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)	
(s1)	DLIMIT(P)	儲存下限限位值 (最小輸出極限值) 的元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DLIMIT(P)_U		0~4294967295	無符號BIN32位元	ANY32_U
(s2)	DLIMIT(P)	儲存上限限位值 (最大輸出極限值) 的元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DLIMIT(P)_U		0~4294967295	無符號BIN32位元	ANY32_U
(s3)	DLIMIT(P)	儲存透過上下限限位控制進行控制的輸入值的元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DLIMIT(P)_U		0~4294967295	無符號BIN32位元	ANY32_U
(d)	DLIMIT(P)	儲存透過上下限限位控制進行控制的輸出值的起始元件	—	帶符號BIN32位元	ANY32_S
	DLIMIT(P)_U			無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

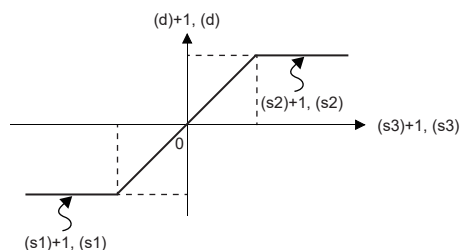
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—
(s3)	○	○	○	○	○	○	○	○	○	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—

功能

- (s3)中指定的輸入值(BIN32位元值)根據(s1)、(s2)中指定的上下限限位值的範圍，對(d)中指定的元件中儲存的輸出值進行控制。輸出值按下述方式被控制。

條件	輸出值中儲存的值
下限限位值((s1)、(s1)+1)>輸入值((s3)、(s3)+1)時	下限限位值((s1)、(s1)+1)
上限限位值((s2)、(s2)+1)<輸入值((s3)、(s3)+1)時	上限限位值((s2)、(s2)+1)
下限限位值((s1)、(s1)+1)≤輸入值((s3)、(s3)+1)≤上限限位值((s2)、(s2)+1)時	輸入值((s3)、(s3)+1)



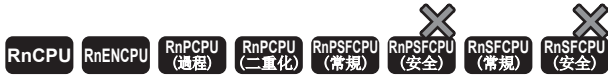
- 僅進行上限限位值的控制的情況下，在(s1)中指定的下限限位值中設定設定資料範圍的最小值。
- 僅進行下限限位值的控制的情況下，在(s2)中指定的上限限位值中設定設定資料範圍的最大值。

出錯

出錯代碼(SD0)	內容
3405H	(s1)中指定的下限限位值大於(s2)中指定的上限限位值時。

BIN16位元資料死區控制

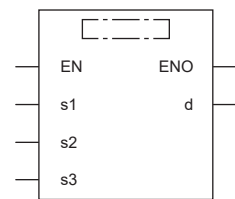
BAND(P) (_U)



根據指定的輸入值 (BIN16位元值) 是否在死區的上下限範圍內，對輸出值進行控制。

梯形圖	ST	
	ENO:=BAND(EN, s1, s2, s3, d); ENO:=BANDP(EN, s1, s2, s3, d);	ENO:=BAND_U(EN, s1, s2, s3, d); ENO:=BANDP_U(EN, s1, s2, s3, d);

FBD/LD



■執行條件

指令	執行條件
BAND BAND_U	
BANDP BANDP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	BAND(P)	-32768~32767	帶符號BIN16位元	ANY16_S
	BAND(P)_U	0~65535	無符號BIN16位元	ANY16_U
(s2)	BAND(P)	-32768~32767	帶符號BIN16位元	ANY16_S
	BAND(P)_U	0~65535	無符號BIN16位元	ANY16_U
(s3)	BAND(P)	-32768~32767	帶符號BIN16位元	ANY16_S
	BAND(P)_U	0~65535	無符號BIN16位元	ANY16_U
(d)	BAND(P)	—	帶符號BIN16位元	ANY16_S
	BAND(P)_U	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

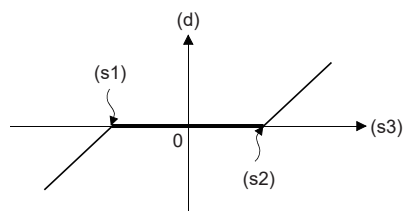
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	
(s3)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	

功能

- (s3)中指定的輸入值(BIN16位元值)根據(s1)、(s2)中指定的死區的上下限範圍，對(d)中指定的元件中儲存的輸出值進行控制。輸出值按下述方式被控制。

條件	輸出值中儲存的值
死區下限值(s1)>輸入值(s3)時	輸入值(s3)-死區下限值(s1)
死區上限值(s2)<輸入值(s3)時	輸入值(s3)-死區上限值(s2)
死區下限值(s1)≤輸入值(s3)≤死區上限值(s2)時	0



- BAND(P)指令的情況下，運算結果超出-32768~32767的範圍時，其情況如下例所示。

例

(s1)為10、(s3)為-32768時，則輸出值=-32768-10=8000H-000AH=7FF6H=32758。

- BAND(P)_U指令的情況下，運算結果超出0~65535的範圍時，其情況如下例所示。

例

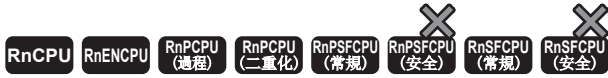
(s1)為100、(s3)為50時，則輸出值=50-100=0032H-0064H=FFCEH=65486。

出錯

出錯代碼(SD0)	內容
3405H	(s1)中指定的下限值大於(s2)中指定的上限值時。

BIN32位元資料死區控制

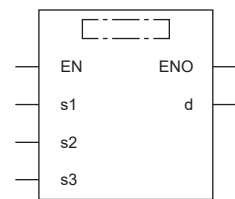
DBAND(P) (_U)



根據指定的輸入值 (BIN32位元值) 是否在死區的上下限範圍內，對輸出值進行控制。

梯形圖	ST	
	ENO:=DBAND (EN, s1, s2, s3, d) ; ENO:=DBANDP (EN, s1, s2, s3, d) ;	ENO:=DBAND_U (EN, s1, s2, s3, d) ; ENO:=DBANDP_U (EN, s1, s2, s3, d) ;

FBD/LD



■執行條件

指令	執行條件
DBAND DBAND_U	
DBANDP DBANDP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)	
(s1)	DBAND (P)	儲存死區 (無輸出區域) 的下限值的元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DBAND (P)_U		0~4294967295	無符號BIN32位元	ANY32_U
(s2)	DBAND (P)	儲存死區 (無輸出區域) 的上限值的元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DBAND (P)_U		0~4294967295	無符號BIN32位元	ANY32_U
(s3)	DBAND (P)	儲存透過死區控制進行控制的輸入值的元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DBAND (P)_U		0~4294967295	無符號BIN32位元	ANY32_U
(d)	DBAND (P)	儲存透過死區控制進行控制的輸出值的起始元件	—	帶符號BIN32位元	ANY32_S
	DBAND (P)_U			無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

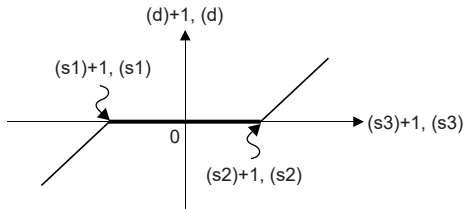
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s1)	○	○	○	○	○	○	○	○	○	—	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—	—
(s3)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

功能

- (s3)中指定的輸入值(BIN32位元值)根據(s1)、(s2)中指定的死區的上下限範圍，對(d)中指定的元件中儲存的輸出值進行控制。輸出值按下述方式被控制。

條件	輸出值中儲存的值
死區下限值((s1)、(s1)+1)>輸入值((s3)、(s3)+1)時	輸入值((s3)、(s3)+1)-死區下限值((s1)、(s1)+1)
死區上限值((s2)、(s2)+1)<輸入值((s3)、(s3)+1)時	輸入值((s3)、(s3)+1)-死區上限值((s2)、(s2)+1)
死區下限值((s1)、(s1)+1)≤輸入值((s3)、(s3)+1)≤死區上限值((s2)、(s2)+1)時	0



- DBAND(P)指令的情況下，運算結果超出-2147483648~2147483647的範圍時，其情況如下例所示。

例

(s1)、(s1)+1為1000，(s3)、(s3)+1為-2147483648時，則輸出值=-2147483648-1000=80000000H-000003E8H=7FFFFC18H=2147482648。

- DBAND(P)_U指令的情況下，運算結果超出0~4294967295的範圍時，其情況如下例所示。

例

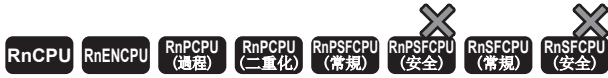
(s1)、(s1)+1為100，(s3)、(s3)+1為50時，則輸出值=50-100=00000032H-00000064H=FFFFFFCEH=4294967246。

出錯

出錯代碼(SD0)	內容
3405H	(s1)中指定的下限值大於(s2)中指定的上限值時。

BIN16位元資料區域控制

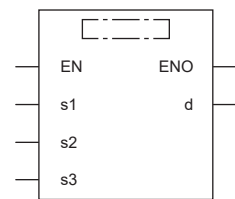
ZONE (P) (_U)



對指定的輸入值 (BIN16位元值) 附加偏置值。

梯形圖	ST	
	ENO:=ZONE (EN, s1, s2, s3, d) ; ENO:=ZONEP (EN, s1, s2, s3, d) ;	ENO:=ZONE_U (EN, s1, s2, s3, d) ; ENO:=ZONEP_U (EN, s1, s2, s3, d) ;

FBD/LD



■執行條件

指令	執行條件
ZONE ZONE_U	
ZONEP ZONEP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	ZONE (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	ZONE (P)_U	0~65535	無符號BIN16位元	ANY16_U
(s2)	ZONE (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	ZONE (P)_U	0~65535	無符號BIN16位元	ANY16_U
(s3)	ZONE (P)	-32768~32767	帶符號BIN16位元	ANY16_S
	ZONE (P)_U	0~65535	無符號BIN16位元	ANY16_U
(d)	ZONE (P)	—	帶符號BIN16位元	ANY16_S
	ZONE (P)_U	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

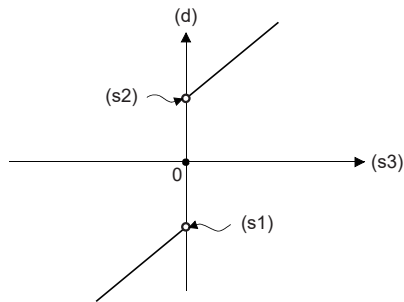
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s1)	○	○	○	○	○	—	—	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—
(s3)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	—	—	—	—

功能

- 在 (s3) 中指定的輸入值 (BIN16位元值) 中附加 (s1) 或 (s2) 中指定的偏置值後，儲存到 (d) 中指定的元件編號中。偏置值按下述方式被控制。

條件	輸出值中儲存的值
輸入值 (s3) < 0 時	輸入值 (s3) + 負的偏置值 (s1)
輸入值 (s3) = 0 時	0
輸入值 (s3) > 0 時	輸入值 (s3) + 正的偏置值 (s2)



- ZONE (P) 指令的情況下，運算結果超出 -32768 ~ 32767 的範圍時，其情況如下例所示。

例

(s1) 為 -100、(s3) 為 -32768 時，則輸出值 = -32768 + (-100) = 8000H - FF9CH = 7F9CH = 32668。

- ZONE (P) _U 指令的情況下，運算結果超出 0 ~ 65535 的範圍時，其情況如下例所示。

例

(s2) 為 100、(s3) 為 65535 時，則輸出值 = 65535 + 100 = FFFFH - 0064H = 0063H = 99。

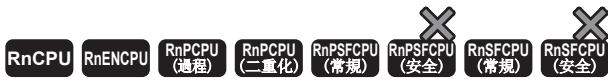
- ZONE (P) _U 指令的情況下，(s1) 不作為虛擬資料處理使用。

出錯

沒有運算出錯。

BIN32位元資料區域控制

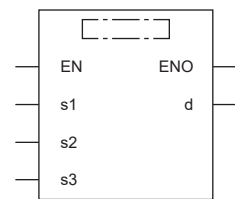
DZONE (P) (_U)



在指定的輸入值 (BIN32位元值) 中附加偏置值。

梯形圖	ST	
	ENO:=DZONE (EN, s1, s2, s3, d) ; ENO:=DZONEP (EN, s1, s2, s3, d) ;	ENO:=DZONE_U (EN, s1, s2, s3, d) ; ENO:=DZONEP_U (EN, s1, s2, s3, d) ;

FBD/LD



■執行條件

指令	執行條件
DZONE DZONE_U	
DZONEP DZONEP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	DZONE (P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DZONE (P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(s2)	DZONE (P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DZONE (P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(s3)	DZONE (P)	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DZONE (P)_U	0~4294967295	無符號BIN32位元	ANY32_U
(d)	DZONE (P)	—	帶符號BIN32位元	ANY32_S
	DZONE (P)_U	—	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

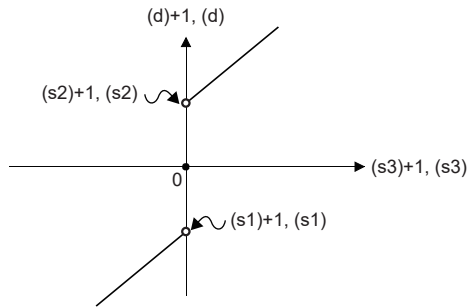
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—	—
(s3)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 在 (s3) 中指定的輸入值 (BIN32 位元值) 中附加 (s1) 或 (s2) 中指定的偏置值後，儲存到 (d) 中指定的元件編號中。偏置值按下述方式被控制。

條件	輸出值中儲存的值
輸入值 ((s3)、(s3)+1) < 0 時	輸入值 ((s3)、(s3)+1) + 負的偏置值 (s1)、(s1)+1
輸入值 ((s3)、(s3)+1) = 0 時	0
輸入值 ((s3)、(s3)+1) > 0 時	輸入值 ((s3)、(s3)+1) + 正的偏置值 (s2)、(s2)+1



- DZONE(P) 指令的情況下，運算結果超出 -2147483648 ~ 2147483647 的範圍時，其情況如下例所示。

例

(s1)、(s1)+1 為 -1000，(s3)、(s3)+1 為 -2147483648 時，則輸出值 = -2147483648 + (-1000) = 80000000H - FFFFFC18H = 7FFFFC18H = 2147482648。

- DZONE(P)_U 指令的情況下，運算結果超出 0 ~ 4294967295 的範圍時，其情況如下例所示。

例

(s2)、(s2)+1 為 1000，(s3)、(s3)+1 為 4294967295 時，則輸出值 = 4294967295 + 1000 = FFFFFFFFH - 00003E8H = 000003E7H = 999。

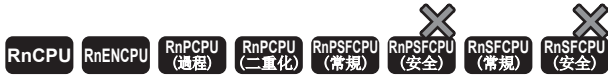
- DZONE(P)_U 指令的情況下，(s1)、(s1)+1 不作為虛擬資料處理使用。

出錯

沒有運算出錯。

BIN16位元單位定標(各點座標資料)

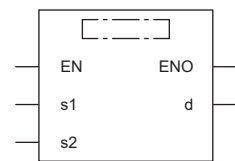
SCL(P) (_U)



對於定標用轉換資料(16位元資料單位)，以指定的輸入值進行定標(各點座標資料)。

梯形圖	ST	
	ENO:=SCL(EN, s1, s2, d); ENO:=SCLP(EN, s1, s2, d);	ENO:=SCL_U(EN, s1, s2, d); ENO:=SCLP_U(EN, s1, s2, d);

FBD/LD



執行條件

指令	執行條件
SCL SCL_U	
SCLP SCLP_U	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	SCL(P)	進行定標的輸入值或儲存了輸入值的元件	-32768~32767	帶符號BIN16位元	ANY16_S
	SCL(P)_U		0~65535	無符號BIN16位元	ANY16_U
(s2)	SCL(P)	儲存了定標用轉換資料的起始元件	—	帶符號BIN16位元*1	ANY16_S*2
	SCL(P)_U		—	無符號BIN16位元*1	ANY16_U*2
(d)	SCL(P)	儲存透過定標進行控制的輸出值的元件	—	帶符號BIN16位元	ANY16_S
	SCL(P)_U		—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

*1 (s2)的座標點數將變為無符號BIN16位元。

*2 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

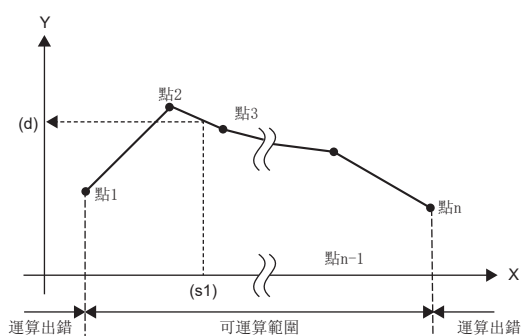
可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s1)	○	○	○	○	○	—	—	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

功能

- 對於(s2)中指定的定標用轉換資料(16位元資料單位)，透過在(s1)中指定的輸入值進行定標，將運算結果儲存到(d)中指定的元件編號中。定標轉換是基於(s2)中指定的元件及其以後儲存的定標用轉換資料進行。

設定項目 (n表示(s2)中指定的座標點數。)		元件分配
座標點數		(s2)
點1	X座標	(s2)+1
	Y座標	(s2)+2
點2	X座標	(s2)+3
	Y座標	(s2)+4
⋮		
點n	X座標	(s2)+2n-1
	Y座標	(s2)+2n



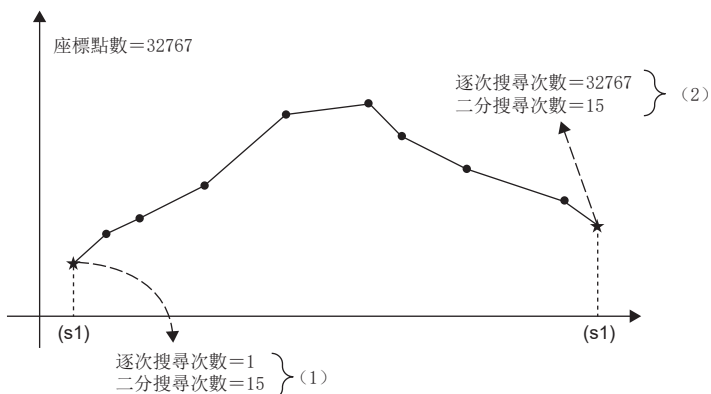
- 運算結果不是整數值的情況下，對小數點以下第1位進行四捨五入。
- 定標用轉換資料的X座標資料應設定為遞增。
- (s1)應在定標用轉換資料範圍內((s2)的元件值)進行設定。
- 多個點指定了同一X座標的情況下，將輸出點No. 為最大的點的Y座標的值。
- (s2)中指定的定標用轉換資料的座標點數應在1~65535內。

注意事項

- 搜尋方法及搜尋次數範圍根據SM755(定標資料檢查設定)的ON/OFF而有所不同。

SM755	搜尋方法	搜尋次數範圍
OFF	逐次搜尋	1≤次數≤65535
ON	二分搜尋	1≤次數≤16

- 定標用轉換資料以遞增被排序的情況下，根據SM755的狀態搜尋方法有所不同，處理速度也不相同。處理速度取決於搜尋次數，搜尋次數越少則處理速度越快。
 - 逐次搜尋的處理速度變快的情況下
在座標點數為最大時，因為(s1)位於座標點1~15之間的情況下，逐次搜尋次數將變為≤15，因此逐次搜尋的處理速度將變快。
 - 二分搜尋的處理速度變快的情況下
因為最多搜尋次數為16次，因此(s1)位於座標點17及其以後的情況下，將變為二分搜尋次數≤逐次搜尋次數，二分搜尋的處理速度將變快。



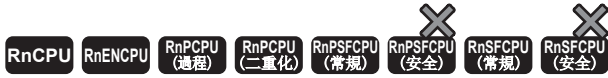
- (1) 因為逐次搜尋次數<二分搜尋次數，因此逐次搜尋的處理速度變快。
 (2) 因為逐次搜尋次數>二分搜尋次數，因此二分搜尋的處理速度變快。

出錯

出錯代碼(SD0)	內容
3405H	比定標用轉換資料的(s1)靠前的點的X座標資料未設定為遞增時。 (但是，SM755為ON時，不檢測本出錯。)
	(s1)中指定的輸入值超出設定的定標用轉換資料的範圍時。
	(s2)的元件開始的座標點數超出1~65535的範圍時。

BIN32位元單位定標(各點座標資料)

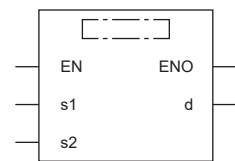
DSCL(P) (_U)



對於定標用轉換資料(32位元資料單位)，以指定的輸入值進行定標(各點座標資料)。

梯形圖	ST	
	ENO:=DSCL(EN, s1, s2, d); ENO:=DSCLP(EN, s1, s2, d);	ENO:=DSCL_U(EN, s1, s2, d); ENO:=DSCLP_U(EN, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
DSCL DSCL_U	
DSCLP DSCLP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	DSCL(P)	進行定標的輸入值或儲存了輸入值的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DSCL(P)_U		0~4294967295	無符號BIN32位元	ANY32_U
(s2)	DSCL(P)	儲存了定標用轉換資料的起始元件	—	帶符號BIN32位元*1	ANY32_S*2
	DSCL(P)_U			無符號BIN32位元*1	ANY32_U*2
(d)	DSCL(P)	儲存透過定標進行控制的輸出值的起始元件	—	帶符號BIN32位元	ANY32_S
	DSCL(P)_U			無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

*1 (s2)+0~(s2)+1的座標點數將變為無符號BIN32位元。

*2 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

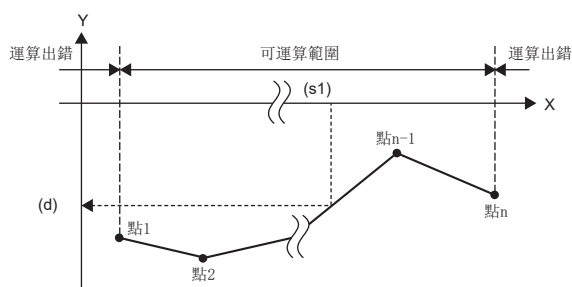
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—

功能

- 對於(s2)中指定的定標用轉換資料(32位元資料單位)，透過在(s1)中指定的輸入值進行定標，將運算結果儲存到(d)中指定的元件編號中。定標轉換是基於(s2)中指定的元件及其以後儲存的定標用轉換資料進行。

設定項目 (n表示(s2)中指定的座標點數。)		元件分配
座標點數		(s2)+1、(s2)
點1	X座標	(s2)+3、(s2)+2
	Y座標	(s2)+5、(s2)+4
點2	X座標	(s2)+7、(s2)+6
	Y座標	(s2)+9、(s2)+8
⋮		
點n	X座標	(s2)+4n-1、(s2)+4n-2
	Y座標	(s2)+4n+1、(s2)+4n



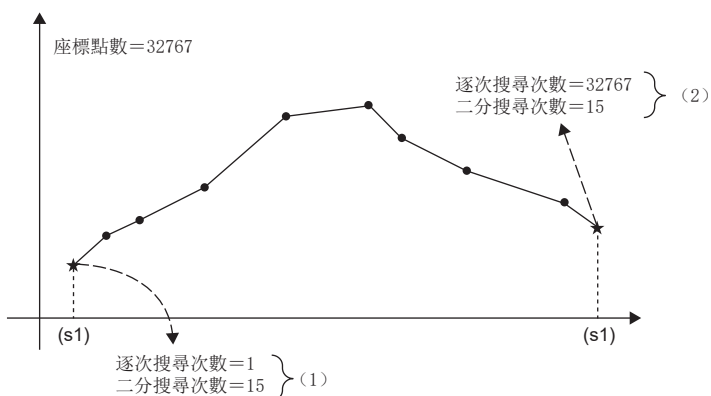
- 運算結果不是整數值的情況下，對小數點以下第1位進行四捨五入。
- 定標用轉換資料的X座標資料應設定為遞增。
- 對於(s1)應在定標用轉換資料範圍內((s2)、(s2)+1的元件值)進行設定。
- 多個點指定了同一X座標的情況下，將輸出點No. 為最大的點的Y座標的值。
- (s2)中指定的定標用轉換資料的座標點數應在1~4294967295內。

注意事項

- 搜尋方法及搜尋次數範圍根據SM755(定標資料檢查設定)的ON/OFF而有所不同。

SM755	搜尋方法	搜尋次數範圍
OFF	逐次搜尋	1≤次數≤4294967295
ON	二分搜尋	1≤次數≤32

- 定標用轉換資料以遞增被排序的情況下，根據SM755的狀態搜尋方法有所不同，處理速度也不相同。處理速度取決於搜尋次數，搜尋次數越少則處理速度越快。
- 逐次搜尋的處理速度變快的情況下
在座標點數為最大時，因為(s1)位於座標點1~15之間的情況下，逐次搜尋次數將變為≤15，因此逐次搜尋的處理速度將變快。
- 二分搜尋的處理速度變快的情況下
最多搜尋次數為32次，因此(s1)位於座標點33及其以後的情況下，將變為二分搜尋次數≤逐次搜尋次數，二分搜尋的處理速度將變快。



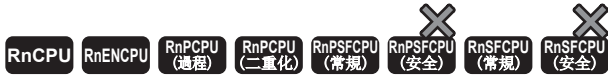
- (1) 因為逐次搜尋次數<二分搜尋次數，因此逐次搜尋的處理速度變快。
 (2) 因為逐次搜尋次數>二分搜尋次數，因此二分搜尋的處理速度變快。

出錯

出錯代碼 (SD0)	內容
3405H	比定標用轉換資料的 (s1) 靠前的點的X座標資料未設定為遞增時。 (但是, SM755為ON時, 不檢測本出錯。)
	(s1) 中指定的輸入值超出設定的定標用轉換資料的範圍時。
	(s2) 的元件開始的座標點數超出1~4294967295的範圍時。

BIN16位元單位定標(各X/Y座標資料)

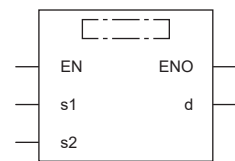
SCL2(P) (_U)



對於定標用轉換資料(16位元資料單位)，以指定的輸入值進行定標(各X/Y座標資料)。

梯形圖	ST	
	ENO:=SCL2(EN, s1, s2, d); ENO:=SCL2P(EN, s1, s2, d);	ENO:=SCL2_U(EN, s1, s2, d); ENO:=SCL2P_U(EN, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
SCL2 SCL2_U	
SCL2P SCL2P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	SCL2(P)	進行定標的輸入值或儲存了輸入值的元件	-32768~32767	帶符號BIN16位元	ANY16_S
	SCL2(P)_U		0~65535	無符號BIN16位元	ANY16_U
(s2)	SCL2(P)	儲存了定標用轉換資料的起始元件	—	帶符號BIN16位元*1	ANY16_S*2
	SCL2(P)_U			無符號BIN16位元*1	ANY16_U*2
(d)	SCL2(P)	儲存透過定標進行控制的輸出值的元件	—	帶符號BIN16位元	ANY16_S
	SCL2(P)_U			無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

*1 (s2)的座標點數將變為無符號BIN16位元。

*2 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

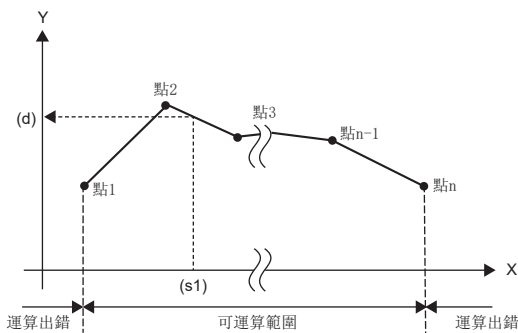
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 對於(s2)中指定的定標用轉換資料(16位元資料單位)，透過在(s1)中指定的輸入值進行定標，將運算結果儲存到(d)中指定的元件編號中。定標轉換是基於(s2)中指定的元件及其以後儲存的定標用轉換資料進行。

設定項目 (n表示(s2)中指定的座標點數。)		元件分配
座標點數		(s2)
X座標	點1	(s2)+1
	點2	(s2)+2
	⋮	⋮
	點n	(s2)+n
Y座標	點1	(s2)+n+1
	點2	(s2)+n+2
	⋮	⋮
	點n	(s2)+2n



- 運算結果不是整數值的情況下，對小數點以下第1位進行四捨五入。
- 定標用轉換資料的X座標資料應設定為遞增。
- (s1)應在定標用轉換資料範圍內((s2)的元件值)進行設定。
- 多個點指定了同一X座標的情況下，將輸出點No. 為最大的點的Y座標的值。
- 應將定標用轉換資料的座標點數設定在1~65535的範圍內。

注意事項

定標用轉換資料以遞增排序的情況下，根據SM755(定標資料檢查設定)的狀態搜尋方法有所不同，因此處理速度也不相同。關於其詳細內容，請參閱SCL(P) (U)指令。

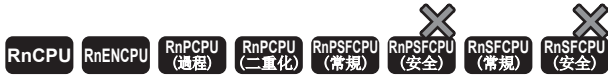
☞ 693頁 SCL(P) (U)

出錯

出錯代碼(SD0)	內容
3405H	X座標資料未設定為遞增時。
	(s1)中指定的輸入值超出設定的定標用轉換資料的範圍時。
	(s2)的元件開始的座標點數超出1~65535的範圍時。

BIN32位元單位定標(各X/Y座標資料)

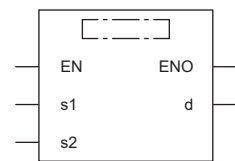
DSCL2(P) (_U)



對於定標用轉換資料(32位元資料單位)，以指定的輸入值進行定標(各X/Y座標資料)。

梯形圖	ST	
	ENO:=DSCL2(EN, s1, s2, d); ENO:=DSCL2P(EN, s1, s2, d);	ENO:=DSCL2_U(EN, s1, s2, d); ENO:=DSCL2P_U(EN, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
DSCL2 DSCL2_U	
DSCL2P DSCL2P_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s1)	DSCL2(P)	進行定標的輸入值或儲存了輸入值的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DSCL2(P)_U		0~4294967295	無符號BIN32位元	ANY32_U
(s2)	DSCL2(P)	儲存了定標用轉換資料的起始元件	—	帶符號BIN32位元*1	ANY32_S*2
	DSCL2(P)_U			無符號BIN32位元*1	ANY32_U*2
(d)	DSCL2(P)	儲存透過定標進行控制的輸出值的起始元件	—	帶符號BIN32位元	ANY32_S
	DSCL2(P)_U			無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

*1 (s2) ~ (s2)+1的座標點數將變為無符號BIN32位元。

*2 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

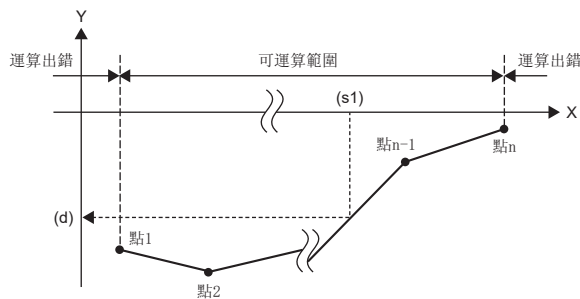
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—

功能

- 對於(s2)中指定的定標用轉換資料(32位元資料單位)，透過在(s1)中指定的輸入值進行定標，將運算結果儲存到(d)中指定的元件編號中。定標轉換是基於(s2)中指定的元件及其以後儲存的定標用轉換資料進行。

設定項目 (n表示(s2)中指定的座標點數。)		元件分配
座標點數		(s2)+1、(s2)
X座標	點1	(s2)+3、(s2)+2
	點2	(s2)+5、(s2)+4
	⋮	⋮
	點n	(s2)+2n+1、(s2)+2n
Y座標	點1	(s2)+2n+3、(s2)+2n+2
	點2	(s2)+2n+5、(s2)+2n+4
	⋮	⋮
	點n	(s2)+4n+1、(s2)+4n



- 運算結果不是整數值的情況下，對小數點以下第1位進行四捨五入。
- 定標用轉換資料的X座標資料應設定為遞增。
- 對於(s1)，應在定標用轉換資料範圍內((s2)~(s2)+1的元件值)進行設定。
- 多個點指定了同一X座標的情況下，將輸出點No. 為最大的點的Y座標的值。
- 應將定標用轉換資料的座標點數設定在1~4294967295的範圍內。

注意事項

定標用轉換資料以遞增排序的情況下，根據SM755(定標資料檢查設定)的狀態搜尋方法有所不同，因此處理速度也不相同。關於詳細內容，請參閱DSCL(P) (_U) 指令。

☞ 696頁 DSCL(P) (_U)

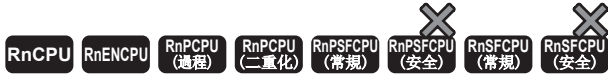
出錯

出錯代碼(SD0)	內容
3405H	X座標資料未設定為遞增時。
	(s1)中指定的輸入值超出設定的定標用轉換資料的範圍時。
	(s2)的元件開始的座標點數超出1~4294967295的範圍時。

8.6 資料處理指令

BIN16位元資料搜尋

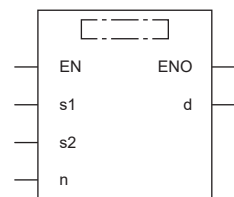
SERDATA (P)



搜尋 (s1) 中指定的BIN16位元資料及從 (s2) 中指定的BIN16位元資料開始的 (n) 點。

梯形圖	ST
	ENO:=SERDATA (EN, s1, s2, n, d) ; ENO:=SERDATAP (EN, s1, s2, n, d) ;

FBD/LD



■執行條件

指令	執行條件
SERDATA	
SERDATAP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	搜尋資料或儲存了搜尋資料的元件	-32768~32767	帶符號BIN16位元	ANY16
(s2)	被搜尋的資料或儲存了被搜尋的資料的起始元件	—	帶符號BIN16位元	ANY16*1
(d)	儲存搜尋結果的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 2)
(n)	搜尋數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

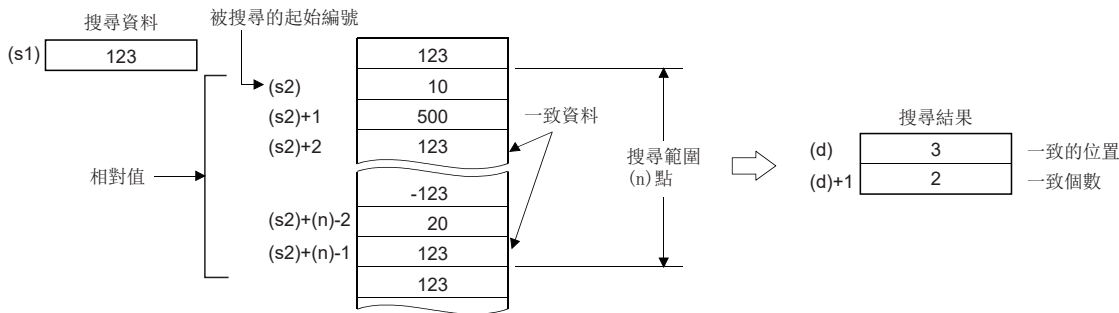
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

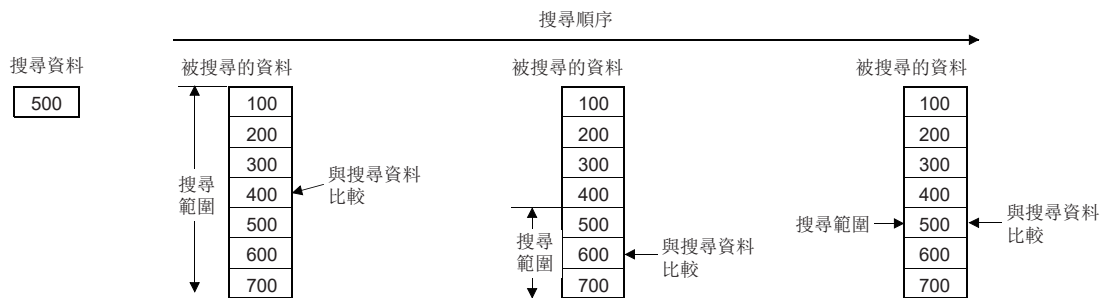
- 將(s1)中指定的元件的BIN16位元資料作為關鍵字，從(s2)中指定的元件開始以BIN16位元單位搜尋(n)點。將與關鍵字一致的個數儲存在(d)+1中指定的元件中，而最先一致的元件編號將從(s2)開始的相對值儲存在(d)中指定的元件中。



- (n) 為0的情況下，將變為無處理。
- 搜尋的結果為未發現一致資料的情況下，(d)、(d)+1中指定的元件將變為0。

要點

- 透過SERDATA (P) 指令搜尋的資料以遞增排序的情況下，將SM702(搜尋方法)^{*1}設為ON時，透過使用二分搜尋法的搜尋處理可變快。搜尋的資料未以遞增排序的情況下如果將SM702設為ON，將無法獲得正常的搜尋結果。透過二分搜尋法進行搜尋的示例如下所示。



*1 SM702是用於設定搜尋方法的特殊繼電器。

[SM702為OFF的情況下]: 變為逐次搜尋法(線形搜尋法)。
是比較從搜尋的資料的起始開始與搜尋資料的方法。

[SM702為ON的情況下]: 變為二分搜尋法。

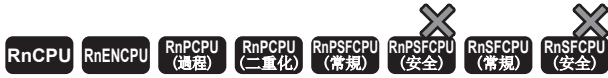
對於以遞增排序的資料，找出搜尋範圍中間的值，根據該值與希望搜尋的值的size將搜尋範圍縮小至某一方。重複此操作，搜尋要搜尋的資料。

出錯

沒有運算出錯。

BIN32位元資料搜尋

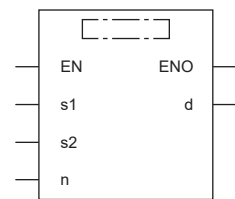
DSERDATA (P)



搜尋 (s1) 中指定的BIN32位元資料及從 (s2) 中指定的BIN32位元資料開始的 (n) 點。

梯形圖	ST
	ENO:=DSERDATA (EN, s1, s2, n, d) ; ENO:=DSERDATAP (EN, s1, s2, n, d) ;

FBD/LD



■執行條件

指令	執行條件
DSERDATA	
DSERDATAP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	搜尋資料或儲存了搜尋資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(s2)	被搜尋的資料或儲存了被搜尋的資料的起始元件	—	帶符號BIN32位元	ANY32*1
(d)	儲存搜尋結果的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 2)
(n)	搜尋數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

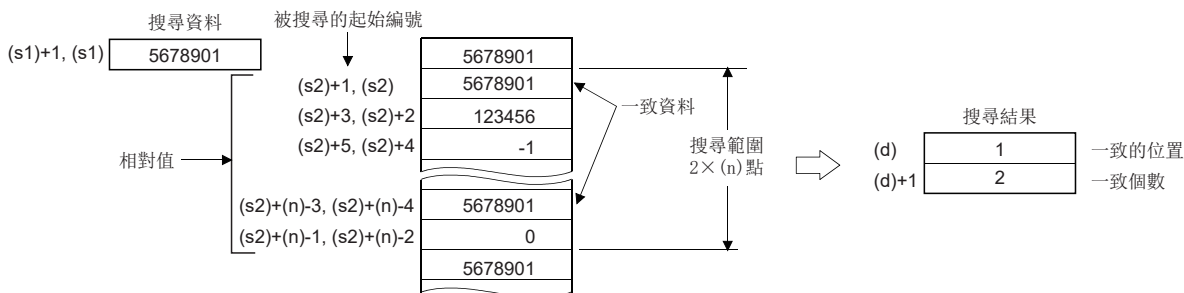
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	○	○	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

功能

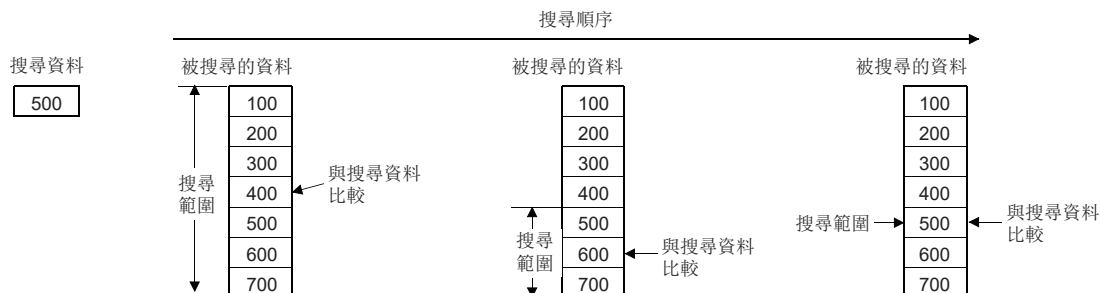
- 將(s1)中指定的元件的BIN32位元資料作為關鍵字，從(s2)中指定的元件開始以BIN32位元單位搜尋(n)點(以16位元單位 $2 \times (n)$ 點)。將與關鍵字一致的個數儲存到(d)+1中指定的元件中，將最先一致的元件編號從(s2)開始的相對值儲存到(d)中指定的元件中。



- (n) 為0的情況下，將變為無處理。
- 搜尋結果為未發現一致資料的情況下，(d)、(d)+1中指定的元件將變為0。

要點

- 透過DSERDATA(P)指令搜尋的資料以遞增排序的情況下，將SM702(搜尋方法)*1設為ON時，透過使用二分搜尋法的搜尋處理可變快。搜尋的資料未以遞增排序的情況下如果將SM702設為ON，將無法獲得正常的搜尋結果。透過二分搜尋法進行搜尋的示例如下所示。



*1 SM702是用於設定搜尋方法的特殊繼電器。

[SM702為OFF的情況下]: 變為逐次搜尋法(線形搜尋法)。

是比較從搜尋的資料的起始開始與搜尋資料的方法。

[SM702為ON的情況下]: 變為二分搜尋法。

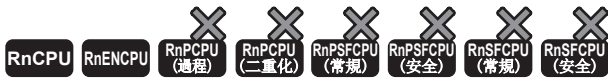
對於以遞增排序的資料，找出搜尋範圍中間的值，根據該值與希望搜尋的值的size將搜尋範圍縮小至某一方。重複此操作，搜尋要搜尋的資料。

出錯

沒有運算出錯。

BIN16位元資料搜尋(最小・相同・最大)

SERMM(P)

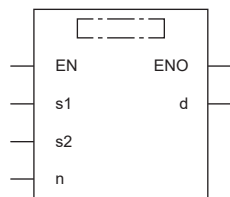


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

搜尋(s1)中指定的BIN16位元資料開始的(n)點，(s2)中指定的BIN16位元資料和相同資料及最大值、最小值。

梯形圖	ST
	ENO:=SERMM(EN, s1, s2, n, d); ENO:=SERMMP(EN, s1, s2, n, d);

FBD/LD



■執行條件

指令	執行條件
SERMM	
SERMMP	

8

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	搜尋相同資料、最大值、最小值的起始元件	—	帶符號BIN16位元	ANY16
(s2)	搜尋相同資料的值或儲存目標元件	—	帶符號BIN16位元	ANY16
(d)	搜尋相同資料、最大值、最小值之後儲存個數的起始元件	—	無符號BIN16位元	ANY16_ARRAY (要素數: 5)
(n)	搜尋相同資料、最大值、最小值的個數	0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	○*1	○	○	○	○	—	○	○	—	—	—
(d)	—	—	○	○	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	○	○	—	—	—

*1 不能使用FX、FY。

功能

- 以(s1)為起始的(n)個資料，搜尋與(s2)的BIN16位元資料相同的資料，並將結果儲存在(d)~(d)+4中。
- 存在相同資料時，在以(d)為起始的5點元件中儲存相同資料的個數、初次/最終位置和最大值、最小值的位置。
- 不存在相同資料時，在以(d)為起始的5點元件中儲存相同資料的個數、初次/最終位置和最大值、最小值的位置。但是，在以(d)為起始的3點元件(相同資料的個數、初次/最終位置)中儲存0。
- 資料中最小值、最大值為多個時，儲存每個後側的位置。
- 在(n)中指定了0的情況下，將變為無處理。
- 搜尋結果表的配置和資料示例如下所示。(n=10)

被搜尋元件 (s1)	被搜尋資料 (s1)的值(示例)	比較資料 (s2)的值(示例)	資料的位置	搜尋結果		
				最大值(d)+4	一致(d)	最小值(d)+3
(s1)	100	100	0	—	○(初次)	—
(s1)+1	111		1	—	—	—
(s1)+2	100		2	—	○	—
(s1)+3	98		3	—	—	—
(s1)+4	123		4	—	—	—
(s1)+5	66		5	—	—	○
(s1)+6	100		6	—	○(最終)	—
(s1)+7	95		7	—	—	—
(s1)+8	210		8	○	—	—
(s1)+9	88		9	—	—	—

- 根據上述示例搜尋結果表如下所示。

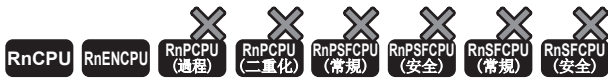
元件編號	內容	搜尋結果項目
(d)	3	相同資料的個數
(d)+1	0	相同資料的位置(初次)
(d)+2	6	相同資料的位置(最終)
(d)+3	5	最小值的最終位置
(d)+4	8	最大值的最終位置

出錯

沒有運算出錯。

BIN32位元資料搜尋(最小・相同・最大)

DSERMM(P)

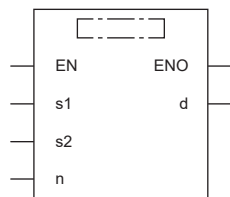


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

搜尋(s1)中指定的BIN32位元資料開始的(n)點及與(s2)中指定的BIN32位元資料相同的資料及最大值、最小值。

梯形圖	ST
	ENO:=DSERMM(EN, s1, s2, n, d); ENO:=DSERMM(EN, s1, s2, n, d);

FBD/LD



■執行條件

指令	執行條件
DSERMM	
DSERMM	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	搜尋相同資料、最大值、最小值的起始元件	—	帶符號BIN32位元	ANY32
(s2)	搜尋相同資料的值或儲存目標元件	—	帶符號BIN32位元	ANY32
(d)	搜尋相同資料、最大值、最小值之後儲存個數的起始元件	—	無符號BIN32位元	ANY32_ARRAY (要素數: 5)
(n)	搜尋相同資料、最大值、最小值的個數	0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	○*1	○	○	○	○	○	○	○	○	—	—
(d)	—	—	○	○	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	○	○	—	—	—

*1 不能使用FX、FY。

功能

- 對於(s1)+1、(s1)為起始的(n)個資料，搜尋與(s2)+1、(s2)的BIN32位元資料相同的資料，並將結果儲存在[(d)+1、(d)]～[(d)+9、(d)+8]中。
- 存在相同資料時，以(d)+1、(d)為起始的5點元件中儲存相同資料的個數、初次/最終位置和最大值、最小值的位置。
- 不存在相同資料時，以(d)+1、(d)為起始的5點元件中儲存相同資料的個數、初次/最終位置和最大值、最小值的位置。但是，以(d)+1、(d)為起始的3點元件(相同資料的個數、初次/最終位置)中儲存0。
- 資料中最小值、最大值為多個時，儲存每個後側的位置。
- 在(n)中指定了0的情況下，將變為無處理。
- 搜尋結果表的配置和資料示例如下所示。(n=10)

被搜尋元件 (s1)	被搜尋資料 (s1)的值(示例)	比較資料 (s2)的值(示例)	資料的位置	搜尋結果		
				最大值 (d)+9、(d)+8	一致(d)	最小值 (d)+7、(d)+6
(s1)+1、(s1)	100000	100000	0	—	○(初次)	—
(s1)+3、(s1)+2	110100		1	—	—	—
(s1)+5、(s1)+4	100000		2	—	○	—
(s1)+7、(s1)+6	98000		3	—	—	—
(s1)+9、(s1)+8	123000		4	—	—	—
(s1)+11、(s1)+10	66000		5	—	—	○
(s1)+13、(s1)+12	100000		6	—	○(最終)	—
(s1)+15、(s1)+14	95000		7	—	—	—
(s1)+17、(s1)+16	910000		8	○	—	—
(s1)+19、(s1)+18	910000		9	○	—	—

- 根據上述示例搜尋結果表如下所示。

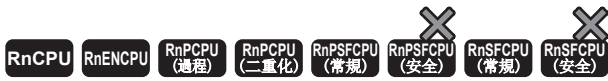
元件編號	內容	搜尋結果項目
(d)+1、(d)	3	相同資料的個數
(d)+3、(d)+2	0	相同資料的位置(初次)
(d)+5、(d)+4	6	相同資料的位置(最終)
(d)+7、(d)+6	5	最小值的最終位置
(d)+9、(d)+8	9	最大值的最終位置

出錯

沒有運算出錯。

BIN16位元資料位元檢查

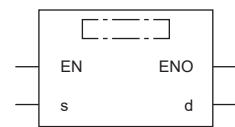
SUM(P)



儲存指定的元件的BIN16位元資料中處於1的位元的總數。

梯形圖	ST
	ENO:=SUM(EN, s, d); ENO:=SUMP(EN, s, d);

FBD/LD



■執行條件

指令	執行條件
SUM	
SUMP	

設置數據

■內容、範圍、資料類型

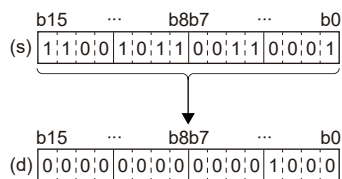
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	對處於1的位元的總數進行計數的元件	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存位元的總數的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\(\H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 將(s)中指定的元件的BIN16位元資料中處於1的位元的總數儲存到(d)中指定的元件中。
1的總數以BIN儲存。(左述的示例中為8個。)

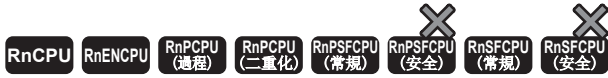


出錯

沒有運算出錯。

BIN32位元資料位元檢查

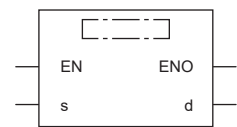
DSUM(P)



儲存指定的元件的BIN32位元資料中處於1的位元的總數。

梯形圖	ST
	ENO:=DSUM(EN, s, d); ENO:=DSUMP(EN, s, d);

FBD/LD



■執行條件

指令	執行條件
DSUM	
DSUMP	

設置數據

■內容、範圍、資料類型

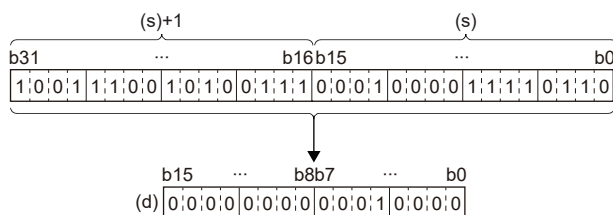
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	對處於1的位元的總數進行計數的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存位元的總數的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- (s)中指定的元件的BIN32位元資料中，將處於1的位元的總數儲存到(d)中指定的元件中。
1的總數以BIN儲存。(左述的示例中為16個。)

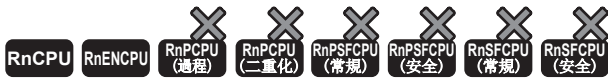


出錯

沒有運算出錯。

BIN16位元資料的位元判定

BON(P)



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

檢查指定的元件的(n)位元的狀態為ON還是OFF，並將結果儲存在(d)中指定的元件。

梯形圖	ST
	ENO:=BON(EN, s, n, d); ENO:=BONP(EN, s, n, d);

FBD/LD

■執行條件

指令	執行條件
BON	
BONP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	資料儲存目標元件	—	帶符號BIN16位元	ANY16
(d)	儲存了結果的位元元件	—	位元	ANY_BOOL
(n)	判定的位元位置	0~15	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

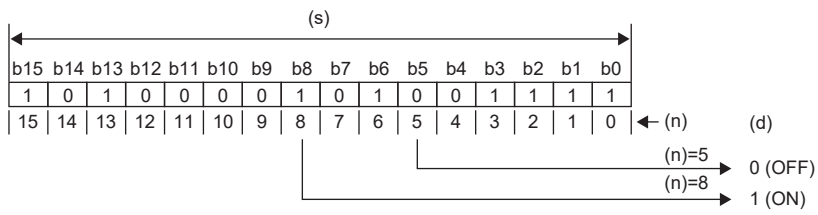
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\(\H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○*1	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○*2	○	—	—	—	○	—	—	—	—	
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	

*1 不能使用FX、FY。
 *2 不能使用T、ST、C。

功能

- 檢查(s)中指定的元件的(n)位元的狀態為ON還是OFF，並將結果儲存在(d)中指定的元件。
- 上述結果若為ON則(d)=ON、若為OFF則(d)=OFF。

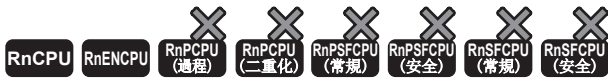


出錯

出錯代碼 (SD0)	內容
3405H	(n)中指定的值超出下述範圍時。 0~15

BIN32位元資料的位元判定

DBON (P)



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

檢查指定的元件的 (n) 位元的狀態為ON還是OFF，並將結果儲存在 (d) 中指定的元件。

梯形圖	ST
	ENO:=DBON (EN, s, n, d) ; ENO:=DBONP (EN, s, n, d) ;

FBD/LD

■執行條件

指令	執行條件
DBON	
DBONP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	資料儲存目標元件	—	帶符號BIN32位元	ANY32
(d)	儲存了結果的位元元件	—	位元	ANY_BOOL
(n)	判定的位元位置	0~31	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

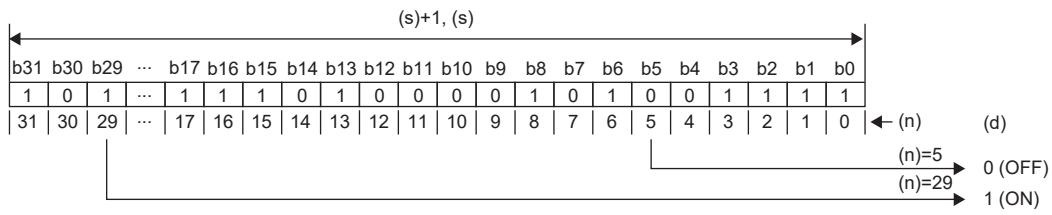
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○*1	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○*2	○	—	—	—	○	—	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	—

*1 不能使用FX、FY。
 *2 不能使用T、ST、C。

功能

- 檢查(s)中指定的元件的(n)位元的狀態為ON還是OFF，並將結果儲存在(d)中指定的元件。
- 上述結果若為ON則(d)=ON、若為OFF則(d)=OFF。

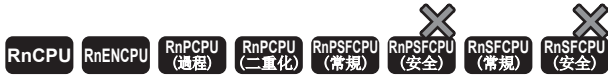


出錯

出錯代碼 (SD0)	內容
3405H	(n)中指定的值超出下述範圍時。 0~31

BIN16位元資料最大值搜尋

MAX(P) (_U)



從指定的元件搜尋從(n)點的BIN16位元資料開始的最大值。

梯形圖	ST*1	
	ENO:=MAXP(EN, s, n, d);	ENO:=MAXP_U(EN, s, n, d);

FBD/LD*1

*1 在ST、FBD/LD中不支援MAX指令、MAX_U指令。應使用通用函數的MAX。
 ↖ 1430頁 MAX(_E)、MIN(_E)

執行條件

指令	執行條件
MAX MAX_U	
MAXP MAXP_U	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	MAX(P)	—	帶符號BIN16位元	ANY16_S*1
	MAX(P)_U		無符號BIN16位元	ANY16_U*1
(d)	MAX(P)	—	帶符號BIN16位元	ANY16_S_ARRAY (要素數: 3)
	MAX(P)_U		無符號BIN16位元	ANY16_U_ARRAY (要素數: 3)
(n)	搜尋資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

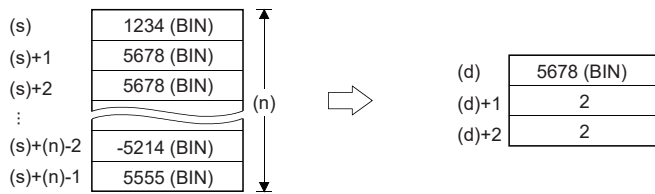
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 從(s)中指定的元件開始的(n)點的BIN16位元資料中搜尋最大值後，將最大值儲存到(d)中指定的元件中。從(s)中指定的元件中搜尋，將儲存了最先檢測到的最大值的元件編號從(s)開始的第幾點儲存到(d)+1中、將最大值的個數儲存到(d)+2中。



(d): 最大值

(d)+1: 位置

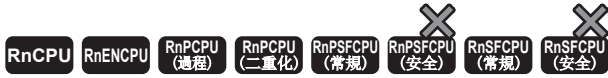
(d)+2: 個數

出錯

沒有運算出錯。

BIN32位元資料最大值搜尋

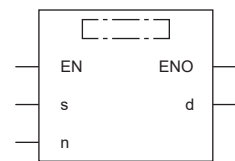
DMAX(P) (_U)



從指定的元件搜尋從(n)點的BIN32位元資料開始的最大值。

梯形圖	ST*1	
	ENO:=DMAXP (EN, s, n, d) ;	ENO:=DMAXP_U (EN, s, n, d) ;

FBD/LD*1



*1 在ST、FBD/LD中不支援DMAX指令、DMAX_U指令。應使用通用函數的MAX。

☞ 1430頁 MAX(_E)、MIN(_E)

■執行條件

指令	執行條件
DMAX DMAX_U	
DMAXP DMAXP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	DMAX(P)	—	帶符號BIN32位元	ANY32_S*1
	DMAX(P)_U		無符號BIN32位元	ANY32_U*1
(d)	DMAX(P)	—	帶符號BIN32位元	—*2 (ANY32_S_ARRAY)
	DMAX(P)_U		無符號BIN32位元	—*2 (ANY32_U_ARRAY)
(n)	搜尋資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

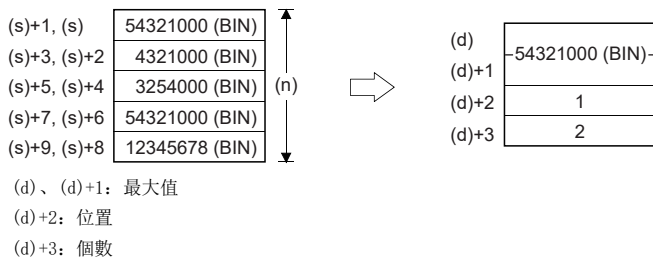
*2 無論使用的程式語言為何，透過元件指定。請勿指定標籤。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 從(s)中指定的元件開始的(n)點的BIN32位元資料中搜尋最大值後，將最大值儲存到(d)、(d)+1中指定的元件中。從(s)中指定的元件中搜尋，將儲存了最先檢測到的最大值的元件編號從(s)開始的第幾點儲存到(d)+2中、將最大值的個數儲存到(d)+3中。

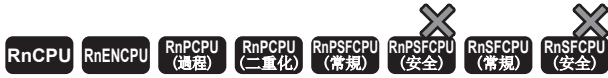


出錯

沒有運算出錯。

BIN16位元資料最小值搜尋

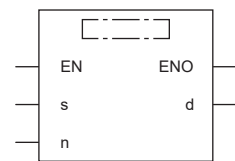
MIN(P) (_U)



從指定的元件搜尋從(n)點的BIN16位元資料開始的最小值。

梯形圖	ST*1	
	ENO:=MINP (EN, s, n, d);	ENO:=MINP_U (EN, s, n, d);

FBD/LD*1



*1 在ST、FBD/LD中不支援MIN指令、MIN_U指令。應使用通用函數的MIN。

☞ 1430頁 MAX(_E)、MIN(_E)

■執行條件

指令	執行條件
MIN MIN_U	
MINP MINP_U	

8

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	MIN(P)	—	帶符號BIN16位元	ANY16_S*1
	MIN(P)_U		無符號BIN16位元	ANY16_U*1
(d)	MIN(P)	—	帶符號BIN16位元	ANY16_S_ARRAY (要素數: 3)
	MIN(P)_U		無符號BIN16位元	ANY16_U_ARRAY (要素數: 3)
(n)	搜尋資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

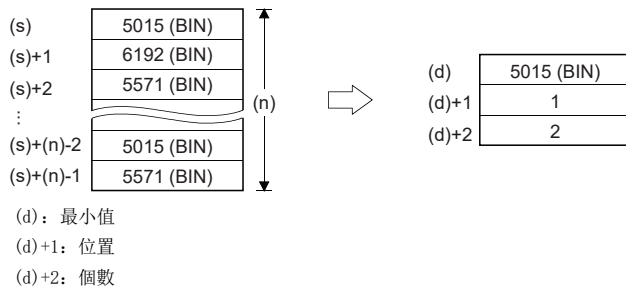
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 從(s)中指定的元件開始至(n)點為止的BIN16位元資料中搜尋最小值，將最小值儲存到(d)中指定的元件中。從(s)中指定的元件中搜尋，將儲存了最先檢測到的最小值的元件編號從(s)開始的第幾點儲存到(d)+1中、將最小值的個數儲存到(d)+2中。

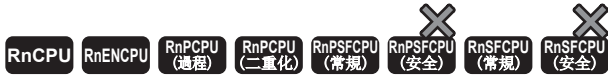


出錯

沒有運算出錯。

BIN32位元資料最小值搜尋

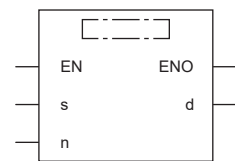
DMIN(P) (_U)



從指定的元件搜尋從(n)點的BIN32位元資料開始的最小值。

梯形圖	ST*1	
	ENO:=DMINP (EN, s, n, d) ;	ENO:=DMINP_U (EN, s, n, d) ;

FBD/LD*1



*1 在ST、FBD/LD中不支援DMIN指令、DMIN_U指令。應使用通用函數的MIN。

☞ 1430頁 MAX(_E)、MIN(_E)

■執行條件

指令	執行條件
DMIN DMIN_U	
DMINP DMINP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	DMIN(P)	—	帶符號BIN32位元	ANY32_S*1
	DMIN(P)_U		無符號BIN32位元	ANY32_U*1
(d)	DMIN(P)	—	帶符號BIN32位元	—*2 (ANY32_S_ARRAY)
	DMIN(P)_U		無符號BIN32位元	—*2 (ANY32_U_ARRAY)
(n)	搜尋資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

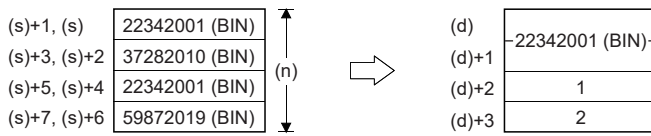
*2 無論使用的程式語言為何，透過元件指定。請勿指定標籤。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 從(s)中指定的元件開始至(n)點為止的BIN32位元資料中搜尋最小值，將最小值儲存到(d)、(d)+1中指定的元件中。從(s)中指定的元件中搜尋，將儲存了最先檢測到最小值的元件編號從(s)開始的第幾點儲存到(d)+2中、將最小值的個數儲存到(d)+3中。



(d)、(d)+1: 最小值

(d)+2: 位置

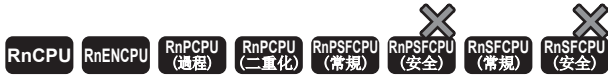
(d)+3: 個數

出錯

沒有運算出錯。

BIN16位元資料排序

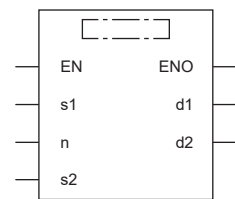
SORTD(_U)



將指定的 (n) 點的BIN16位元資料以遞增/遞減進行排序 (按順序排列)。

梯形圖	ST
	ENO:=SORTD (EN, s1, n, s2, d1, d2) ; ENO:=SORTD_U (EN, s1, n, s2, d1, d2) ;

FBD/LD



■執行條件

指令	執行條件
SORTD SORTD_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	SORTD	—	帶符號BIN16位元	ANY16_S*1
	SORTD_U		無符號BIN16位元	ANY16_U*1
(n)	排序資料數	0~65535	無符號BIN16位元	ANY16
(s2)	透過1次執行進行比較的資料數	0~65535	無符號BIN16位元	ANY16
(d1)	透過排序完成設為ON的位元元件編號	—	位元	ANY_BOOL
(d2)	系統使用元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

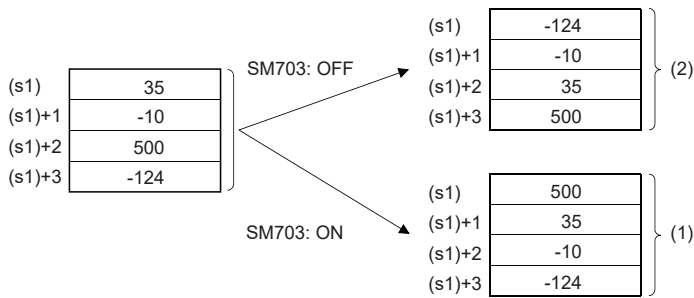
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d1)	○	—	○*1	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	

*1 T、C、ST不能使用。

功能

- 將從(s1)開始的(n)點的BIN16位元資料以遞增或遞減進行排序(按順序排列)。對於排序順序，SM703(排序)為OFF時以遞增進行排序、ON時以遞減進行排序。



(1): 按遞減排序。

(2): 按遞增排序。

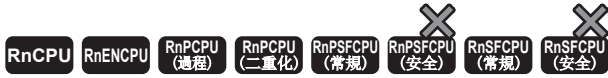
- 透過SORTD(_U)指令進行的排序需要數個掃描。執行完成為止的掃描次數為，將排序執行完成為止的最大執行次數與在(s2)中指定的1次執行下之比較的資料數相除後的值(小數點以下捨去)。增加(s2)的值時，雖然排序完成為止的掃描次數將變少，但掃描時間將延長。
- 排序執行完成為止的最大執行次數透過以下公式計算： $(n) \times (n-1) \div 2$ (次)。例如，(n)=10的情況下，變為 $10 \times (10-1) \div 2 = 45$ 次。此時設定為(s2)=2時，排序完成為止變為 $45 \div 2 = 22.5$ ，需要23次掃描。
- 對於(d1)中指定的元件(完成元件)，透過SORTD(_U)指令執行開始變為OFF，排序完成時變為ON。排序完成後，因為(d1)中指定的元件將保持為ON狀態不變，因此應根據需要將其設為OFF。
- 執行SORTD(_U)指令時(d2)中指定的元件開始的2點由系統使用。應勿對(d2)中指定的元件開始的2點進行更改。進行了更改的情況下，可能出錯。(出錯代碼: 3405H)
- 排序執行中更改了(n)的情況下，將以更改後的排序資料數進行排序。
- 排序執行中將執行指令設為了OFF的情況下，排序將中斷。再次將執行指令設為ON的情況下，將從頭開始重新排序。
- 排序執行完成後，在連續進行下一個排序的情況下，需要將執行指令設為一次OFF後，再次將執行指令設為ON。

出錯

出錯代碼(SD0)	內容
2821H	從(s1)開始的(n)點的元件範圍與從(d2)開始的2點的元件範圍重複時。
3405H	(s2)為0時。
	第2個掃描及其以後，系統使用的(d2)的值大於或等於(n)的值時。
	第2個掃描及其以後，系統使用的(d2)的值處於 $(d2) < (d2)+1$ 的狀態時。

BIN32位元資料排序

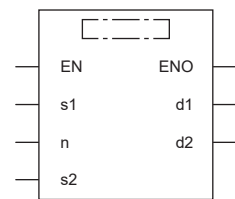
DSORTD(_U)



將指定的 (n) 點的BIN32位元資料以遞增/遞減進行排序 (按順序排列)。

梯形圖	ST
	ENO:=DSORTD (EN, s1, n, s2, d1, d2); ENO:=DSORTD_U (EN, s1, n, s2, d1, d2);

FBD/LD



■執行條件

指令	執行條件
DSORTD DSORTD_U	

■設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	DSORTD	—	帶符號BIN32位元	ANY32_S*1
	DSORTD_U		無符號BIN32位元	ANY32_U*1
(n)	排序資料數	0~65535	無符號BIN16位元	ANY16
(s2)	透過1次執行進行比較的資料數	0~65535	無符號BIN16位元	ANY16
(d1)	透過排序完成設為ON的位元元件編號	—	位元	ANY_BOOL
(d2)	系統使用元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

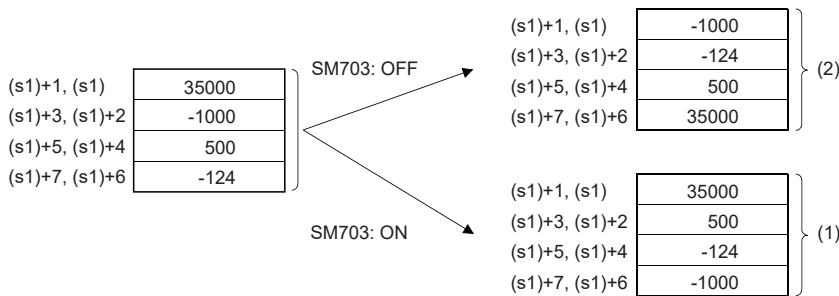
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d1)	○	—	○*1	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	

*1 T、C、ST不能使用。

功能

- 將從(s1)開始的(n)點的BIN32位元資料以遞增或遞減進行排序(按順序排列)。對於排列順序，SM703(排序)為OFF時以遞增進行排序、ON時以遞減進行排序。



- (1): 按遞減排序。
(2): 按遞增排序。

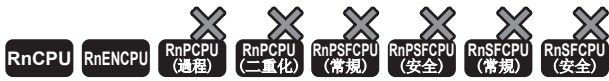
- 透過DSORTD(_U)指令進行排序時，需要數個掃描。執行完成為止的掃描次數為，將排序執行完成為止的最大執行次數與在(s2)中指定的1次執行下之比較的資料數相除後的值(小數點以下捨去)。增加(s2)的值時，雖然排序完成為止的掃描次數將變少，但掃描時間將延長。
- 排序執行完成為止的最大執行次數透過以下公式計算： $(n) \times (n-1) \div 2$ (次)。例如，(n)=10的情況下，變為 $10 \times (10-1) \div 2 = 45$ 次。此時設定為(s2)=2時，排序完成為止變為 $45 \div 2 = 22.5$ ，需要23次掃描。
- 對於(d1)中指定的元件(完成元件)，透過DSORTD(_U)指令開始執行變為OFF，排序完成時將變為ON。排序完成後，因為(d1)中指定的元件將保持為ON狀態不變，因此應根據需要將其設為OFF。
- 執行DSORTD(_U)指令時，從(d2)中指定的元件開始的2點由系統使用。應勿對(d2)中指定的元件開始的2點進行更改。進行了更改的情況下，可能出錯。(出錯代碼: 3405H)
- 排序執行中更改了(n)的情況下，將以更改後的排序資料數進行排序。
- 排序執行中將執行指令設為OFF的情況下，排序將中斷。再次將執行指令設為ON的情況下，將從頭開始重新排序。
- 排序執行完成後，在連續進行下一個排序的情況下，需要將執行指令設為一次OFF後，再次將執行指令設為ON。

出錯

出錯代碼(SD0)	內容
2821H	從(s1)開始的 $2 \times (n)$ 點的元件範圍與從(d2)開始的2點的元件範圍重複時。
3405H	(s2)為0時。
	第2個掃描及其以後，系統使用的(d2)的值大於或等於(n)的值時。
	第2個掃描及其以後，系統使用的(d2)的值處於 $(d2) < (d2)+1$ 的狀態時。

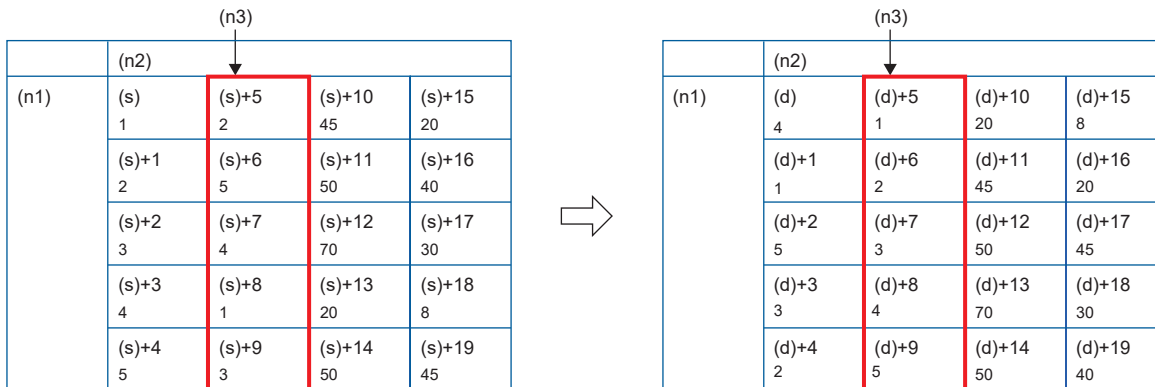
BIN16位元資料表排序

SORTTBL(_U)



- R00CPU、R01CPU、R02CPU可以在韌體版本為“08”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“40”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

將(s)中指定的縱向方向連續的((n1)×(n2))點的BIN16位元資料表(排序源)的資料列，以(n3)欄的資料為基準按遞增或遞減排列，並儲存至(d)指定的((n1)×(n2))點的BIN16位元資料表(排列後)。



要點

以橫向方向連續的元件配置表格時，應使用下述指令。

☞ 733頁 SORTTBL2(_U)

梯形圖	ST	
	<code>ENO:=SORTTBL(EN, s, n1, n2, n3, d);</code>	<code>ENO:=SORTTBL_U(EN, s, n1, n2, n3, d);</code>

FBD/LD

執行條件

指令	執行條件
SORTTBL SORTTBL_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	SORTTBL	儲存了資料表的起始元件	帶符號BIN16位元	ANY16_S*1
	SORTTBL_U		無符號BIN16位元	ANY16_U*1
(n1)	資料(列)數	1~ 32	無符號BIN16位元	ANY16_U
(n2)	資料(欄)數	1~ 6	無符號BIN16位元	ANY16_U
(d)	SORTTBL	儲存運算結果的起始元件	帶符號BIN16位元	ANY16_S*1
	SORTTBL_U		無符號BIN16位元	ANY16_U*1
(n3)	作為排序基準的資料(欄)的欄編號	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(n1)	○	○	○	○	○	—	—	○	○	—	—	—	
(n2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n3)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 將(s)中指定的縱向方向連續的((n1)×(n2))點的BIN16位元資料表(排序源)的資料列，以(n3)欄的資料為基準按遞增或遞減排序，並儲存至(d)指定的((n1)×(n2))點的BIN16位元資料表(排序後)。
- 排序源為(n1)=3，(n2)=4時，配置資料表的配置示例如下所示。此外，資料表排序後的情況下，應將(s)替換為(d)。

例

資料表配置(作為資料(列)數(n1)=3，資料(欄)數(n2)=4的情況下)

—	欄編號1	欄編號2	欄編號3	欄編號4
列編號1	(s)	(s)+3	(s)+6	(s)+9
列編號2	(s)+1	(s)+4	(s)+7	(s)+10
列編號3	(s)+2	(s)+5	(s)+8	(s)+11

- 根據指令執行開始時的SM774(資料表排列執行指令順序)的ON/OFF狀態進行排序的設定。

SM774 設定值	排列順序
OFF	遞增
ON	遞減

- 執行指令設為ON後，SD774(指令執行中)的位元2將變為ON，開始進行排序。排序在(n1)掃描後的END指令執行時完成，SD774的位元2在設為OFF的同時，完成標籤SD774的位元0也將變為ON。完成標籤在下次的掃描的END指令執行時將變為OFF。
- SD774的位元2為ON時，將無法執行SORTTBL(_U)指令。(執行的情況下將變為無處理。)

■運行示例

下述的排序源資料(資料列數(n1)=5, 資料欄數(n2)=4)以各條件進行排序時的運行示例如下所示。

		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(s) 1	(s)+5 150	(s)+10 45	(s)+15 20
	列編號2	(s)+1 2	(s)+6 180	(s)+11 50	(s)+16 40
	列編號3	(s)+2 3	(s)+7 160	(s)+12 70	(s)+17 30
	列編號4	(s)+3 4	(s)+8 100	(s)+13 20	(s)+18 8
	列編號5	(s)+4 5	(s)+9 150	(s)+14 50	(s)+19 45

要點

在第1欄中輸入管理編號等序列編號後, 將可從其內容中判斷原本的列編號。

例

欄編號(n3)=2, SM774=OFF(遞增)的排序結果

		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(s) 1	(s)+5 150	(s)+10 45	(s)+15 20
	列編號2	(s)+1 2	(s)+6 180	(s)+11 50	(s)+16 40
	列編號3	(s)+2 3	(s)+7 160	(s)+12 70	(s)+17 30
	列編號4	(s)+3 4	(s)+8 100	(s)+13 20	(s)+18 8
	列編號5	(s)+4 5	(s)+9 150	(s)+14 50	(s)+19 45



		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(d) 4	(d)+5 100	(d)+10 20	(d)+15 8
	列編號2	(d)+1 1	(d)+6 150	(d)+11 45	(d)+16 20
	列編號3	(d)+2 5	(d)+7 150	(d)+12 50	(d)+17 45
	列編號4	(d)+3 3	(d)+8 160	(d)+13 70	(d)+18 30
	列編號5	(d)+4 2	(d)+9 180	(d)+14 50	(d)+19 40

例

欄編號(n3)=3, SM774=ON(遞減)的排序結果

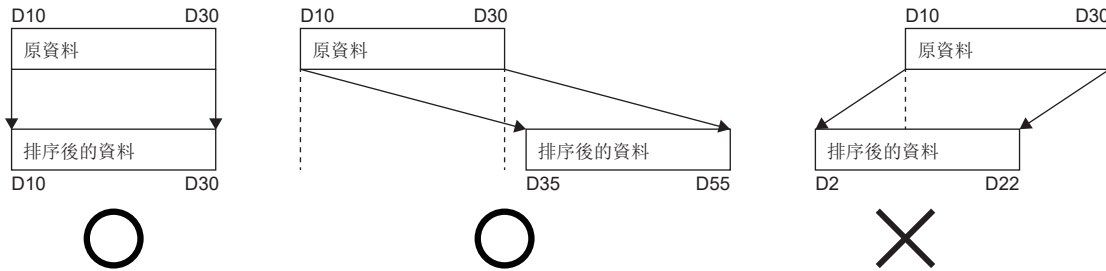
		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(s) 1	(s)+5 150	(s)+10 45	(s)+15 20
	列編號2	(s)+1 2	(s)+6 180	(s)+11 50	(s)+16 40
	列編號3	(s)+2 3	(s)+7 160	(s)+12 70	(s)+17 30
	列編號4	(s)+3 4	(s)+8 100	(s)+13 20	(s)+18 8
	列編號5	(s)+4 5	(s)+9 150	(s)+14 50	(s)+19 45



		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(d) 3	(d)+5 160	(d)+10 70	(d)+15 30
	列編號2	(d)+1 2	(d)+6 180	(d)+11 50	(d)+16 40
	列編號3	(d)+2 5	(d)+7 150	(d)+12 50	(d)+17 45
	列編號4	(d)+3 1	(d)+8 150	(d)+13 45	(d)+18 20
	列編號5	(d)+4 4	(d)+9 100	(d)+14 20	(d)+19 8

注意事項

- 運行中，請勿變更操作數和資料的內容。排序完成時仍可能會出現排列順序不正確的情況。
- 指定(s)和(d)為相同元件的情況下，排序前的資料將會被排序後的資料順序所取代。因此，執行完成(SD774的位元0為0N)之前，請勿變更(s)的內容。
- 指定(s)和(d)時，應指定為相同元件範圍或是不重複範圍的元件。

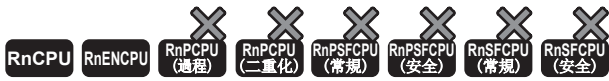


出錯

出錯代碼 (SD0)	內容
2820H	(s) 指定的元件範圍超出相應元件的範圍時。
	(d) 指定的元件範圍超出相應元件的範圍時。
2821H	(s) 指定的元件範圍與 (d) 指定的元件範圍有部分重複時。
3405H	(n1) 中指定的值超出下述範圍時。 1 ~ 32
	(n2) 中指定的值超出下述範圍時。 1 ~ 6
	(n3) 中指定的值超出下述範圍時。 1 ~ (n2)

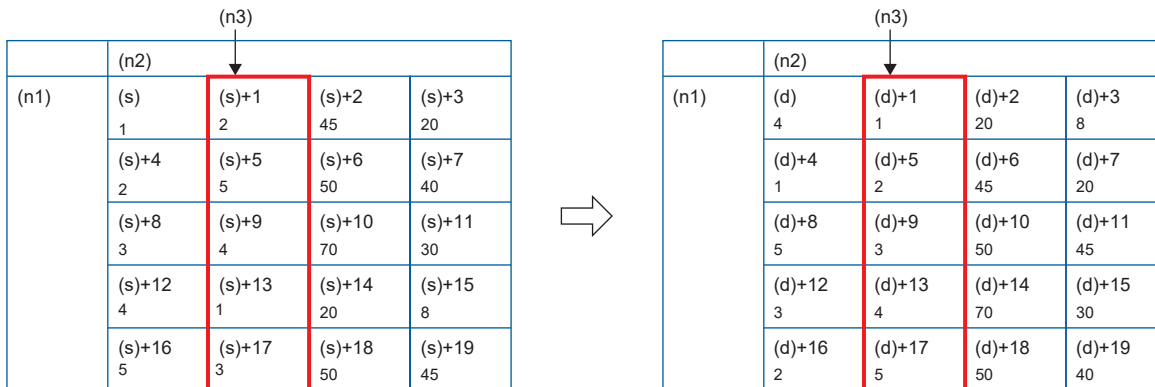
BIN16位元資料表排序2

SORTTBL2(_U)



- R00CPU、R01CPU、R02CPU可以在韌體版本為“08”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“40”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

將(s)中的指定橫向方向連續的((n1)×(n2))點的BIN16位元資料表(排序源)，以(n3)欄的資料為基準按遞增或遞減進行排列，並儲存至(d)指定((n1)×(n2))點的BIN16位元資料表(排列後)。



要點

以縱向方向連續的元件配置表格時，應使用下述指令。

☞ 729頁 SORTTBL(_U)

梯形圖	ST	
	ENO:=SORTTBL2(EN, s, n1, n2, n3, d);	ENO:=SORTTBL2_U(EN, s, n1, n2, n3, d);

FBD/LD

執行條件

指令	執行條件
SORTTBL2 SORTTBL2_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	SORTTBL2	—	帶符號BIN16位元	ANY16_S*1
	SORTTBL2_U		無符號BIN16位元	ANY16_U*1
(n1)	資料(列)數	1~ 32	無符號BIN16位元	ANY16_U
(n2)	資料(欄)數	1~ 6	無符號BIN16位元	ANY16_U
(d)	SORTTBL2	—	帶符號BIN16位元	ANY16_S*1
	SORTTBL2_U		無符號BIN16位元	ANY16_U*1
(n3)	作為排序基準的資料(欄)的欄編號	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(n1)	○	○	○	○	○	—	—	○	○	—	—	—	
(n2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n3)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 將(s)中指定的橫向方向連續的((n1)×(n2))點的資料表(排序源)的資料列，以(n3)欄的資料為基準按遞增或遞減進行排序，並儲存至(d)指定的((n1)×(n2))點的資料表(排序後)。
- 排序源為(n1)=3，(n2)=4時，資料表配置如下所示。此外，資料表排序後的情況下，應將(s)替換為(d)。

例

資料表配置(作為資料(列)數(n1)=3，資料(欄)數(n2)=4的情況下)

—	欄編號1	欄編號2	欄編號3	欄編號4
列編號1	(s)	(s)+1	(s)+2	(s)+3
列編號2	(s)+4	(s)+5	(s)+6	(s)+7
列編號3	(s)+8	(s)+9	(s)+10	(s)+11

- 根據指令執行開始時的SM774(資料表排列執行指令順序)的ON/OFF狀態進行排序的設定。

SM774 設定值	排列順序
OFF	遞增
ON	遞減

- 執行指令設為ON後，SD774(指令執行中)的位元8將變為ON，開始進行排序。排序在(n1)掃描後的END指令執行時完成，SD774的位元8在設為OFF的同時，完成標籤SD774的位元6也將變為ON。完成標籤在下次的掃描的END指令執行時將變為OFF。
- SD774的位元8為ON時，將無法執行SORTTBL2(_U)指令。(執行的情況下將變為無處理。)

■運行示例

下述的排序源資料(資料列數(n1)=5, 資料欄數(n2)=4)以各條件進行排序時的運行示例如下所示。

		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(s) 1	(s)+1 150	(s)+2 45	(s)+3 20
	列編號2	(s)+4 2	(s)+5 180	(s)+6 50	(s)+7 40
	列編號3	(s)+8 3	(s)+9 160	(s)+10 70	(s)+11 30
	列編號4	(s)+12 4	(s)+13 100	(s)+14 20	(s)+15 8
	列編號5	(s)+16 5	(s)+17 150	(s)+18 50	(s)+19 45

要點

在第1欄中輸入管理編號等序列編號後, 將可從其內容中判斷原本的列編號。

例

欄編號(n3)=2, SM774=OFF(遞增)的排序結果

		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(s) 1	(s)+1 150	(s)+2 45	(s)+3 20
	列編號2	(s)+4 2	(s)+5 180	(s)+6 50	(s)+7 40
	列編號3	(s)+8 3	(s)+9 160	(s)+10 70	(s)+11 30
	列編號4	(s)+12 4	(s)+13 100	(s)+14 20	(s)+15 8
	列編號5	(s)+16 5	(s)+17 150	(s)+18 50	(s)+19 45



		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(d) 4	(d)+1 100	(d)+2 20	(d)+3 8
	列編號2	(d)+4 1	(d)+5 150	(d)+6 45	(d)+7 20
	列編號3	(d)+8 5	(d)+9 150	(d)+10 50	(d)+11 45
	列編號4	(d)+12 3	(d)+13 160	(d)+14 70	(d)+15 30
	列編號5	(d)+16 2	(d)+17 180	(d)+18 50	(d)+19 40

例

欄編號(n3)=3, SM774=ON(遞減)的排序結果

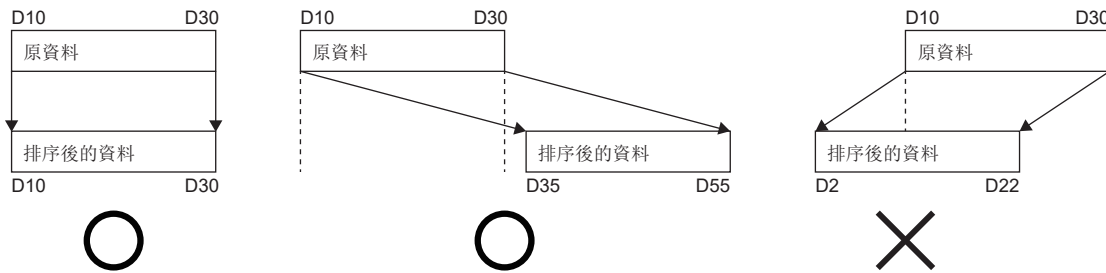
		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(s) 1	(s)+1 150	(s)+2 45	(s)+3 20
	列編號2	(s)+4 2	(s)+5 180	(s)+6 50	(s)+7 40
	列編號3	(s)+8 3	(s)+9 160	(s)+10 70	(s)+11 30
	列編號4	(s)+12 4	(s)+13 100	(s)+14 20	(s)+15 8
	列編號5	(s)+16 5	(s)+17 150	(s)+18 50	(s)+19 45



		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(d) 3	(d)+1 160	(d)+2 70	(d)+3 30
	列編號2	(d)+4 2	(d)+5 180	(d)+6 50	(d)+7 40
	列編號3	(d)+8 5	(d)+9 150	(d)+10 50	(d)+11 45
	列編號4	(d)+12 1	(d)+13 150	(d)+14 45	(d)+15 20
	列編號5	(d)+16 4	(d)+17 100	(d)+18 20	(d)+19 8

注意事項

- 運行中，請勿變更操作數和資料的內容。排序完成時仍可能會出現排列順序不正確的情況。
- 指定(s)和(d)為相同元件的情況下，排序前的資料將會被排序後的資料順序所取代。因此，執行完成(SD774的位元6為0N)之前，請勿變更(s)的內容。
- 指定(s)和(d)時，應指定為相同元件範圍或是不重複範圍的元件。

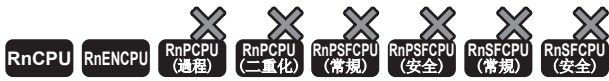


出錯

出錯代碼 (SD0)	內容
2820H	(s) 指定的元件範圍超出相應元件的範圍時。
	(d) 指定的元件範圍超出相應元件的範圍時。
2821H	(s) 指定的元件範圍與 (d) 指定的元件範圍有部分重複時。
3405H	(n1) 中指定的值超出下述範圍時。 1 ~ 32
	(n2) 中指定的值超出下述範圍時。 1 ~ 6
	(n3) 中指定的值超出下述範圍時。 1 ~ (n2)

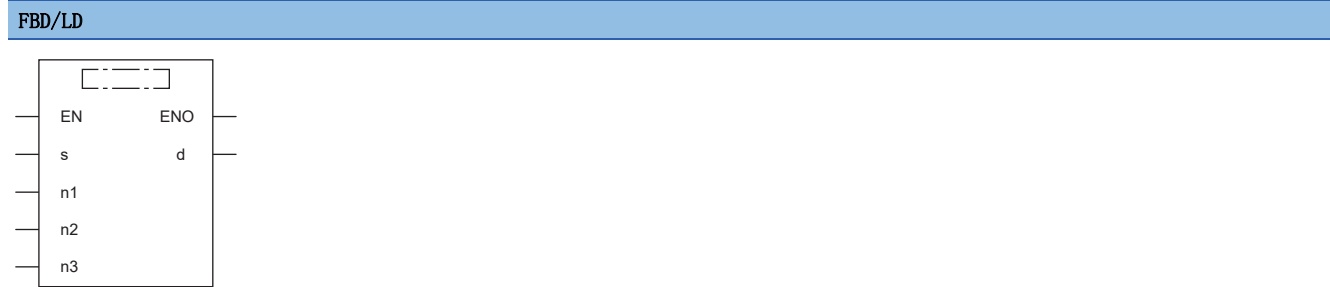
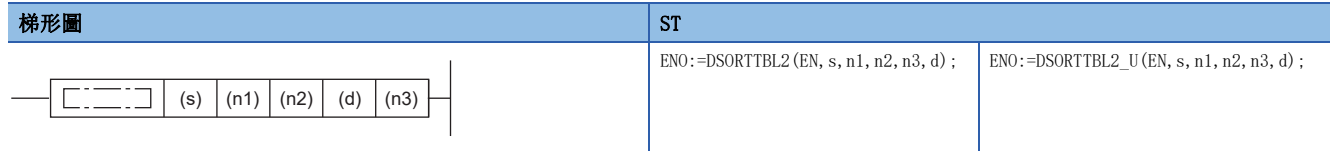
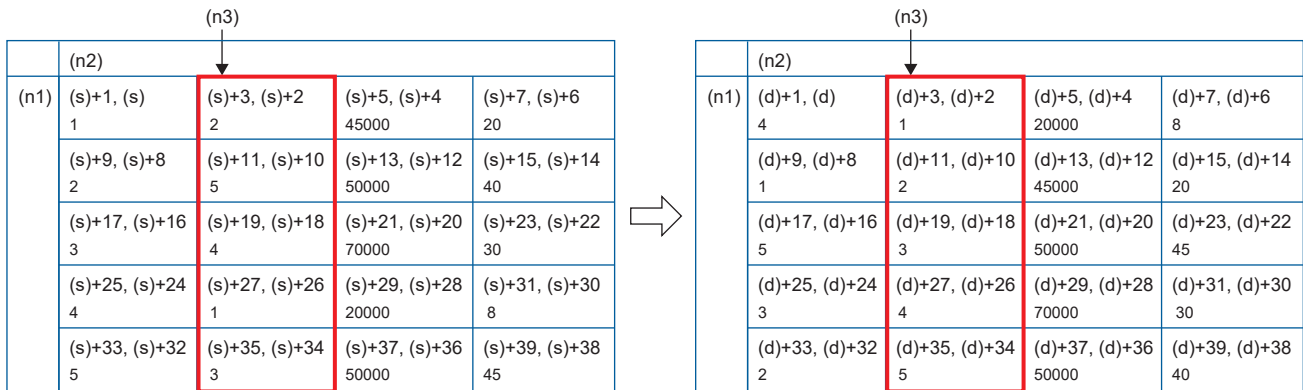
BIN32位元資料表排序2

DSORTTBL2(_U)



- R00CPU、R01CPU、R02CPU可以在韌體版本為“08”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“40”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

將(s)中指定的橫向方向連續的((n1)×(n2))點的BIN32位元資料表(排序源)的資料列，以(n3)欄的資料為基準按遞增或遞減進行排序，並儲存至(d)指定的((n1)×(n2))點的BIN32位元資料表(排序後)。



■執行條件

指令	執行條件
DSORTTBL2 DSORTTBL2_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	DSORTTBL2	—	帶符號BIN32位元	ANY32_S*1
	DSORTTBL2_U		無符號BIN32位元	ANY32_U*1
(n1)	資料(列)數	1~ 32	無符號BIN16位元	ANY16_U
(n2)	資料(欄)數	1~ 6	無符號BIN16位元	ANY16_U
(d)	DSORTTBL2	—	帶符號BIN32位元	ANY32_S*1
	DSORTTBL2_U		無符號BIN32位元	ANY32_U*1
(n3)	作為排序基準的資料(欄)的欄編號	—	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	—	—	○	—	—	○*1	—	○	—	—	—	—	
(n1)	○	○	○	○	○	—	—	○	○	—	—	—	
(n2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	—	—	○	—	—	○*1	—	○	—	—	—	—	
(n3)	○	○	○	○	○	—	—	○	○	—	—	—	

*1 不能使用LT, LST。

功能

- 將(s)指定的橫向方向連續的((n1)×(n2))點的BIN32位元資料表(排序源)，以(n3)欄的資料為基準按遞增或遞減進行排序，並儲存至(d)指定的((n1)×(n2))點的BIN32位元資料表(排序後)。
- 排序源為(n1)=3，(n2)=4時，資料表配置如下所示。此外，資料表排序後的情況下，應將(s)替換為(d)。

例

資料表配置(作為資料(列)數(n1)=3，資料(欄)數(n2)=4的情況下)

—	欄編號1	欄編號2	欄編號3	欄編號4
列編號1	(s)+1、(s)	(s)+3、(s)+2	(s)+5、(s)+4	(s)+7、(s)+6
列編號2	(s)+9、(s)+8	(s)+11、(s)+10	(s)+13、(s)+12	(s)+15、(s)+14
列編號3	(s)+17、(s)+16	(s)+19、(s)+18	(s)+21、(s)+20	(s)+23、(s)+22

- 根據指令執行開始時的SM774(資料表排列執行指令順序)的ON/OFF狀態進行排序的設定。

SM774 設定值	排列順序
OFF	遞增
ON	遞減

- 執行指令設為ON後，SD774(指令執行中)的位元11將變為ON，開始進行排序。排序在(n1)掃描後的END指令執行時完成，SD774的位元11在設為OFF的同時，完成標籤SD774的位元9也將變為ON。完成標籤在下次的掃描的END指令執行時將變為OFF。
- SD774的位元11為ON時，將無法執行DSORTTBL2(U)指令。(執行的情況下將變為無處理。)

■運行示例

下述的排序源資料(資料列數(n1)=5, 資料欄數(n2)=4)以各條件進行排序時的運行示例如下所示。

		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(s)+1, (s) 1	(s)+3, (s)+2 150	(s)+5, (s)+4 45000	(s)+7, (s)+6 20
	列編號2	(s)+9, (s)+8 2	(s)+11, (s)+10 180	(s)+13, (s)+12 50000	(s)+15, (s)+14 40
	列編號3	(s)+17, (s)+16 3	(s)+19, (s)+18 160	(s)+21, (s)+20 70000	(s)+23, (s)+22 30
	列編號4	(s)+25, (s)+24 4	(s)+27, (s)+26 100	(s)+29, (s)+28 20000	(s)+31, (s)+30 8
	列編號5	(s)+33, (s)+32 5	(s)+35, (s)+34 150	(s)+37, (s)+36 50000	(s)+39, (s)+38 45

要點

在第1欄中輸入管理編號等序列編號後, 將可從其內容中判斷原本的列編號。

例

欄編號(n3)=2, SM774=OFF(遞增)的排序結果

		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(s)+1, (s) 1	(s)+3, (s)+2 150	(s)+5, (s)+4 45000	(s)+7, (s)+6 20
	列編號2	(s)+9, (s)+8 2	(s)+11, (s)+10 180	(s)+13, (s)+12 50000	(s)+15, (s)+14 40
	列編號3	(s)+17, (s)+16 3	(s)+19, (s)+18 160	(s)+21, (s)+20 70000	(s)+23, (s)+22 30
	列編號4	(s)+25, (s)+24 4	(s)+27, (s)+26 100	(s)+29, (s)+28 20000	(s)+31, (s)+30 8
	列編號5	(s)+33, (s)+32 5	(s)+35, (s)+34 150	(s)+37, (s)+36 50000	(s)+39, (s)+38 45



		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(d)+1, (d) 4	(d)+3, (d)+2 100	(d)+5, (d)+4 20000	(d)+7, (d)+6 8
	列編號2	(d)+9, (d)+8 1	(d)+11, (d)+10 150	(d)+13, (d)+12 45000	(d)+15, (d)+14 20
	列編號3	(d)+17, (d)+16 5	(d)+19, (d)+18 150	(d)+21, (d)+20 50000	(d)+23, (d)+22 45
	列編號4	(d)+25, (d)+24 3	(d)+27, (d)+26 160	(d)+29, (d)+28 70000	(d)+31, (d)+30 30
	列編號5	(d)+33, (d)+32 2	(d)+35, (d)+34 180	(d)+37, (d)+36 50000	(d)+39, (d)+38 40

例

欄編號(n3)=3, SM774=ON(遞減)的排序結果

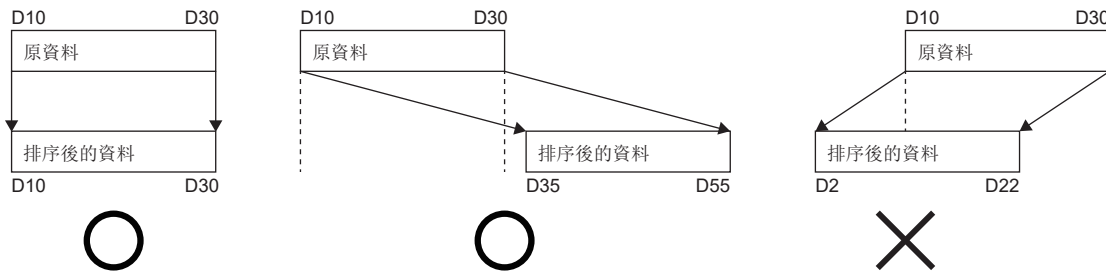
		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(s)+1, (s) 1	(s)+3, (s)+2 150	(s)+5, (s)+4 45000	(s)+7, (s)+6 20
	列編號2	(s)+9, (s)+8 2	(s)+11, (s)+10 180	(s)+13, (s)+12 50000	(s)+15, (s)+14 40
	列編號3	(s)+17, (s)+16 3	(s)+19, (s)+18 160	(s)+21, (s)+20 70000	(s)+23, (s)+22 30
	列編號4	(s)+25, (s)+24 4	(s)+27, (s)+26 100	(s)+29, (s)+28 20000	(s)+31, (s)+30 8
	列編號5	(s)+33, (s)+32 5	(s)+35, (s)+34 150	(s)+37, (s)+36 50000	(s)+39, (s)+38 45



		(n2)			
		欄編號1	欄編號2	欄編號3	欄編號4
		管理號碼	尺寸X	尺寸Y	尺寸Z
(n1)	列編號1	(d)+1, (d) 3	(d)+3, (d)+2 160	(d)+5, (d)+4 70000	(d)+7, (d)+6 30
	列編號2	(d)+9, (d)+8 2	(d)+11, (d)+10 180	(d)+13, (d)+12 50000	(d)+15, (d)+14 40
	列編號3	(d)+17, (d)+16 5	(d)+19, (d)+18 150	(d)+21, (d)+20 50000	(d)+23, (d)+22 45
	列編號4	(d)+25, (d)+24 1	(d)+27, (d)+26 150	(d)+29, (d)+28 45000	(d)+31, (d)+30 20
	列編號5	(d)+33, (d)+32 4	(d)+35, (d)+34 100	(d)+37, (d)+36 20000	(d)+39, (d)+38 8

注意事項

- 運行中，請勿變更操作數和資料的內容。排序完成時仍可能會出現排列順序不正確的情況。
- 指定(s)和(d)為相同元件的情況下，排序前的資料將會被排序後的資料順序所取代。因此，執行完成(SD774的位元9為0N)之前，請勿變更(s)的內容。
- 指定(s)和(d)時，應指定為相同元件範圍或是不重複範圍的元件。

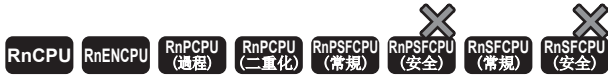


出錯

出錯代碼 (SD0)	內容
2820H	(s) 指定的元件範圍超出相應元件的範圍時。
	(d) 指定的元件範圍超出相應元件的範圍時。
2821H	(s) 指定的元件範圍與 (d) 指定的元件範圍有部分重複時。
3405H	(n1) 中指定的值超出下述範圍時。 1 ~ 32
	(n2) 中指定的值超出下述範圍時。 1 ~ 6
	(n3) 中指定的值超出下述範圍時。 1 ~ (n2)

BIN16位元資料合計值計算

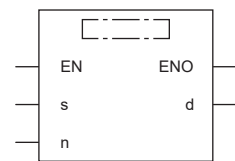
WSUM(P) (_U)



將指定的元件開始的(n)點的BIN16位元資料全部進行加法運算。

梯形圖	ST	
	ENO:=WSUM(EN, s, n, d); ENO:=WSUMP(EN, s, n, d);	ENO:=WSUM_U(EN, s, n, d); ENO:=WSUMP_U(EN, s, n, d);

FBD/LD



■執行條件

指令	執行條件
WSUM WSUM_U	
WSUMP WSUMP_U	

8

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	WSUM(P)	—	帶符號BIN16位元	ANY16_S*1
	WSUM(P)_U		無符號BIN16位元	ANY16_U*1
(d)	WSUM(P)	—	帶符號BIN32位元	ANY32_S
	WSUM(P)_U		無符號BIN32位元	ANY32_U
(n)	資料個數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

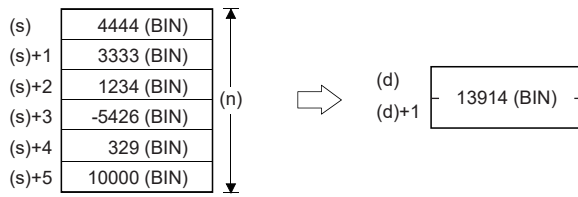
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	○	○	○	○	○	○	○	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 將 (s) 中指定的元件開始的 (n) 點的BIN16位元資料全部進行加法運算後，儲存到 (d) 中指定的元件中。

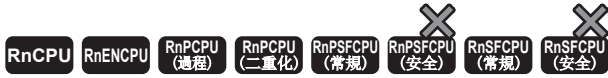


出錯

沒有運算出錯。

BIN32位元資料合計值計算

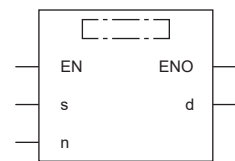
DWSUM(P) (_U)



將從指定的元件開始的 (n) 點的BIN32位元資料全部進行加法運算。

梯形圖	ST	
	ENO:=DWSUM (EN, s, n, d) ; ENO:=DWSUMP (EN, s, n, d) ;	ENO:=DWSUM_U (EN, s, n, d) ; ENO:=DWSUMP_U (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
DWSUM DWSUM_U	
DWSUMP DWSUMP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	DWSUM (P) DWSUM (P)_U	—	帶符號BIN32位元 無符號BIN32位元	ANY32_S*1 ANY32_U*1
(d)	DWSUM (P) DWSUM (P)_U	—	帶符號BIN64位元 無符號BIN64位元	ANY32_ARRAY (要素數: 2)
(n)	資料個數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

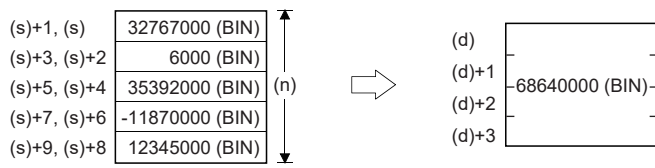
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	○	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

功能

- 將從(s)中指定的元件開始的(n)點的BIN32位元資料全部進行加法運算後，儲存到(d)中指定的元件中。

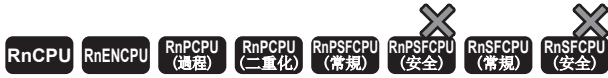


出錯

沒有運算出錯。

BIN16位元資料平均值計算

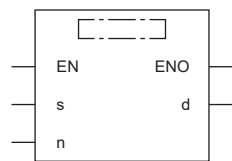
MEAN(P) (_U)



對指定的元件開始的(n)點的(16位元資料單位)的平均值進行計算。

梯形圖	ST	
	ENO:=MEAN(EN, s, n, d); ENO:=MEANP(EN, s, n, d);	ENO:=MEAN_U(EN, s, n, d); ENO:=MEANP_U(EN, s, n, d);

FBD/LD



■執行條件

指令	執行條件
MEAN MEAN_U	
MEANP MEANP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	MEAN(P) MEAN(P)_U	儲存了進行平均值計算的資料的起始元件	—	帶符號BIN16位元 ANY16_S*1 無符號BIN16位元 ANY16_U*1
(d)	MEAN(P) MEAN(P)_U	儲存平均值的元件	—	帶符號BIN16位元 ANY16_S 無符號BIN16位元 ANY16_U
(n)		資料數或儲存了資料數的元件編號	0~65535	無符號BIN16位元 ANY16
EN		執行條件	—	位元 BOOL
ENO		執行結果	—	位元 BOOL

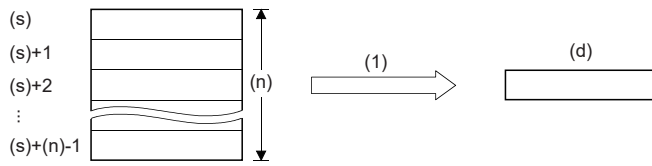
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 對(s)中指定的元件開始的(n)點的BIN16位元資料的平均值進行計算後，儲存到(d)中指定的元件中。



(1): 平均值

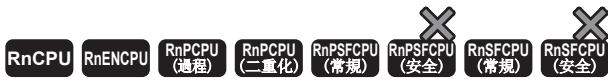
- 計算結果不是整數值的情況下，小數點以下將被捨去。
- (n)中指定的值為0的情況下，將變為無處理。

出錯

沒有運算出錯。

BIN32位元資料平均值計算

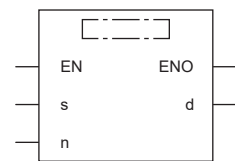
DMEAN(P) (_U)



對指定的元件開始的(n)點的(32位元資料單位)的平均值進行計算。

梯形圖	ST	
	ENO:=DMEAN(EN, s, n, d); ENO:=DMEANP(EN, s, n, d);	ENO:=DMEAN_U(EN, s, n, d); ENO:=DMEANP_U(EN, s, n, d);

FBD/LD



■執行條件

指令	執行條件
DMEAN DMEAN_U	
DMEANP DMEANP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	DMEAN(P) DMEAN(P)_U	—	帶符號BIN32位元	ANY32_S*1
			無符號BIN32位元	ANY32_U*1
(d)	DMEAN(P) DMEAN(P)_U	—	帶符號BIN32位元	ANY32_S
			無符號BIN32位元	ANY32_U
(n)	資料數或儲存了資料數的元件編號	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

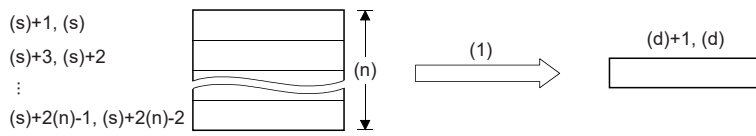
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	—	—	○	—	—	○	—	○	—	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 對(s)中指定的元件開始的(n)點的BIN32位元資料的平均值進行計算，儲存到(d)中指定的元件中。



(1): 平均值

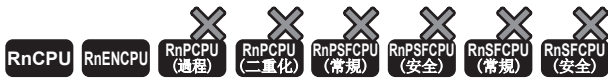
- 計算結果不是整數值的情況下，小數點以下將被捨去。
- (n)中指定的值為0的情況下，將變為無處理。

出錯

沒有運算出錯。

計算BIN16位元平方根

SQRT (P)



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

對指定的BIN16位元資料的平方根進行運算。

梯形圖	ST*1
	ENO:=SQRTP (EN, s, d) ;

FBD/LD*1

*1 SQRT指令不支援ST語言、FBD/LD語言。應使用通用函數的SQRT。(參見1390頁 SQRT(_E))

■執行條件

指令	執行條件
SQRT	
SQRTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存進行平方根運算的資料的元件	0~65535	無符號BIN16位元	ANY16
(d)	儲存運算平方根的元件	—	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\(\H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○*1	○	○	○	○	—	—	○	—	—	—
(d)	○*1	○	○	○	○	—	—	○	—	—	—

*1 不能使用FX、FY。

功能

- (s)中指定的BIN16位元資料的平方根進行運算，運算結果儲存在(d)中。運算結果捨去小數點以後的數字變為整數。 $\sqrt{(s)} \rightarrow (d)$

出錯

沒有運算出錯。

計算BIN32位元平方根

DSQRT (P)



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

對指定的BIN32位元資料的平方根進行運算。

梯形圖	ST
	ENO:=DSQRT(EN, s, d); ENO:=DSQRT(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
DSQRT	
DSQRT(EN, s, d)	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存進行平方根運算的資料的元件	0~4294967295	無符號BIN32位元	ANY32
(d)	儲存運算平方根的元件	—	無符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○*1	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○*1	○	○	○	○	○	○	○	○	—	—	—	—

*1 不能使用FX、FY。

功能

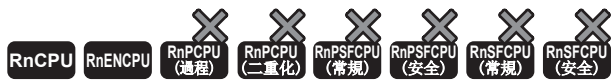
- (s)中指定的BIN32位元資料的平方根進行運算，運算結果儲存在(d)中。運算結果捨去小數點以後的數字變為整數。
 $\sqrt{(s)+1}$, (s)→(d)

出錯

沒有運算出錯。

CRC運算

CRC (P)



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

以(s)中指定的元件為起始，(n)點的8位元資料(字節單位)生成CRC值並儲存在(d)中。

梯形圖	ST
	ENO:=CRC (EN, s, n, d) ; ENO:=CRCP (EN, s, n, d) ;

FBD/LD

■執行條件

指令	執行條件
CRC	
CRCP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存CRC值的生成對象的資料的起始元件	—	帶符號BIN16位元	ANY16
(d)	生成的CRC值的儲存目標元件	—	帶符號BIN16位元	ANY16
(n)	儲存求得CRC值的8位元資料(字節單位)數或8位元資料(字節單位)數的元件	0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○*1	—	○	—	—	—	—	○	—	—	—	—	—
(d)	○*1	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	—

*1 不能使用FX、FY。

功能

- 以(s)中指定的元件為起始，(n)點的8位元資料(字節單位)生成CRC值並儲存在(d)中。CRC值(CRC-16)的生成多項式中使用“ $X^{16}+X^{15}+X^2+1$ ”。該指令中計算時使用的模式有16位元轉換模式和8位元轉換模式。轉換模式可以透過SM772(CCD/CRC指令轉換模式指定)的ON/OFF選擇。
- 在(n)中指定了0的情況下，將變為無處理。
- 在本指令中，計算通訊等所使用的出錯檢查方法中的1個CRC(Cyclic Redundancy Check)值。出錯檢查方法除CRC以外還有奇偶檢查與和校驗(校驗和)，可以利用CCD(P)指令求得水平奇偶值與和校驗值。(☞ 753頁 CCD(P))
- 各轉換模式的動作如下所示。

■16位元轉換模式(SM772=OFF時)

對(s)的元件的高位8位元(字節單位)和低位8位元(字節單位)進行CRC運算。運算結果儲存在指定到(d)的元件1點的16位元。

例

在(n)=6的情況下

16位元轉換模式時，下述影線部分的6字節為運算對象。因為CRC值為“A57BH”，所以“A57BH”儲存在(d)的元件中。

	10進制	16進制	
		高位	低位
D0	24932	61H	64H
D1	4219	10H	7BH
D2	-1333	FAH	CBH
D3	-1	FFH	FFH
D4	32761	7FH	F9H
D5	10000	27H	10H

■8位元轉換模式(SM772=ON時)

僅對(s)的元件的低位8位元(低位字節)進行CRC運算。對於運算結果，使用指定到(d)的元件中的2點，並將低位8位元(字節單位)儲存於(d)中、高位8位元(字節單位)儲存於(d)+1中。

例

在(n)=6的情況下

8位元轉換模式時，下述影線部分的6字節為運算對象。因為CRC值為“BDA1H”，所以將“A1H”儲存在(d)中、將“BDH”儲存在(d)+1中。

	10進制	16進制	
		高位	低位
D0	24932	61H	64H
D1	4219	10H	7BH
D2	-1333	FAH	CBH
D3	-1	FFH	FFH
D4	32761	7FH	F9H
D5	10000	27H	10H

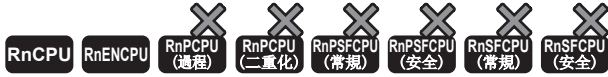
出錯

出錯代碼(SD0)	內容
3405H	透過指定(s)、(d)位元元件的位指定來指定4位指定以外的情況。

8.7 檢查代碼指令

檢查代碼

CCD(P)

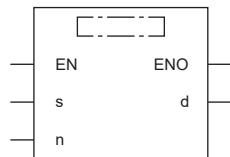


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

計算儲存在(s)~(s)+(n)-1中的資料的加法運算資料和水平奇偶，將加法運算資料儲存在(d)、水平奇偶儲存在(d)+1中。

梯形圖	ST
	ENO:=CCD(EN, s, n, d); ENO:=CCDP(EN, s, n, d);

FBD/LD



■執行條件

指令	執行條件
CCD	
CCDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	對象的起始元件	—	帶符號BIN16位元	ANY16
(d)	計算的資料的儲存目標的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 2)
(n)	資料數	0~65535	無符號BIN16位元	ANY16_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○*1	—	○	—	—	—	—	○	—	—	—	—	
(d)	○*1	—	○	—	—	—	—	○	—	—	—	—	
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	

*1 不能使用FX、FY。

功能

- 計算儲存在(s)~(s)+(n)-1中的資料的加法運算資料和水平奇偶，將加法運算資料儲存在(d)、水平奇偶儲存在(d)+1中。該指令中計算時使用的模式有16位元轉換模式和8位元轉換模式。轉換模式可以透過SM772 (CCD/CRC指令轉換模式指定)的ON/OFF選擇。
- 在(n)中指定了0的情況下，將變為無處理。
- 儲存在(d)中的加法運算結果發生溢出時，進行以下動作。SM700(進位標誌)不變為ON。
 $32767+2=-32767$ (7FFFH+0002H=8001H) $-32767+-2=32767$ (8001H+FFFEH=7FFFH)
- 在本指令中計算通訊等使用的出錯檢查方法的水平奇偶值與和校驗值。出錯檢查方法除CCD(P)指令以外還有CRC(Cyclic Redundancy Check)。(☞ 751頁 CRC(P))
- 各轉換模式的動作如下所示。

■16位元轉換模式(SM772=OFF時)

以(s)為起始的(n)點資料，上下各8位元資料的加法運算資料和水平奇偶資料儲存在(d)和(d)+1的元件中。

例

在(n)=6的情況下

<加法運算值的計算>

16位元轉換模式時，加法運算資料為下述影線部分的6字節的和。因為此處加法運算資料為“0315H”，所以將“0315H”儲存在(d)的元件中。

	10進制	16進制	
		高位	低位
D0	24932	61H	64H
D1	4219	10H	7BH
D2	-1333	FAH	CBH
D3	-1	FFH	FFH
D4	32761	7FH	F9H
D5	10000	27H	10H

<水平奇偶值的計算>

16位元轉換模式時，上述影線部分為水平奇偶的計算對象。

奇偶值為計算各個位元ON(1)的個數，若最終ON(1)的個數為奇數則為ON(1)的值，若最終為偶數則為OFF(0)的值。將該水平奇偶值儲存在(d)+1的元件中。

在下表中，將5FH儲存在(d)+1的元件中。

位元/水平奇偶值	b7	b6	b5	b4	b3	b2	b1	b0	值
D0的高位8位元	0	1	1	0	0	0	0	1	61H
D0的低位8位元	0	1	1	0	0	1	0	0	64H
D1的高位8位元	0	0	0	1	0	0	0	0	10H
D1的低位8位元	0	1	1	1	1	0	1	1	7BH
D2的高位8位元	1	1	1	1	1	0	1	0	FAH
D2的低位8位元	1	1	0	0	1	0	1	1	CBH
水平奇偶值	0	1	0	1	1	1	1	1	5FH

■8位元轉換模式(SM772=ON時)

以(s)為起始的(n)點資料(僅低位8位元)，加法運算資料和水平奇偶資料儲存在(d)和(d)+1的元件中。

例

在(n)=6的情況下

<加法運算值的計算>

8位元轉換模式時，加法運算資料為下述影線部分的6字節的和。因為此處加法運算資料為“03B2H”，所以將“03B2H”儲存在(d)的元件中。

	10進制	16進制	
		高位	低位
D0	24932	61H	64H
D1	4219	10H	7BH
D2	-1333	FAH	CBH
D3	-1	FFH	FFH
D4	32761	7FH	F9H
D5	10000	27H	10H

<水平奇偶值的計算>

8位元轉換模式時，上述影線部分為水平奇偶的計算對象。奇偶值為計算各個位元ON(1)的個數，若最終ON(1)的個數為奇數則為ON(1)的值，若最終為偶數則為OFF(0)的值。將該水平奇偶值儲存在(d)+1的元件中。

在下表中，將C2H儲存在(d)+1的元件中。

位元/水平奇偶值	b7	b6	b5	b4	b3	b2	b1	b0	值
D0的低位8位元	0	1	1	0	0	1	0	0	64H
D1的低位8位元	0	1	1	1	1	0	1	1	7BH
D2的低位8位元	1	1	0	0	1	0	1	1	CBH
D3的低位8位元	1	1	1	1	1	1	1	1	FFH
D4的低位8位元	1	1	1	1	1	0	0	1	F9H
D5的低位8位元	0	0	0	1	0	0	0	0	10H
水平奇偶值	1	1	0	0	0	0	1	0	C2H

出錯

沒有運算出錯。

9 偵錯、故障診斷

9.1 偵錯、故障診斷指令

出錯顯示或報警器重設

LEDR



對CPU模組的報警器顯示及可繼續運轉的自我診斷出錯的顯示進行重設。

梯形圖	ST
	ENO:=LEDR (EN);

FBD/LD

■執行條件

指令	執行條件
LEDR	

功能

- 對CPU模組的報警器顯示及可繼續運轉的自我診斷出錯的顯示進行重設。透過執行1次指令，出錯顯示及報警器二者均將進行重設。
- 發生自我診斷出錯時的動作如下所示。
 - 發生了可繼續運轉的自我診斷出錯時。
 當CPU模組顯示為可繼續運轉的自我診斷出錯時，CPU模組前面的ERROR LED將熄燈。此時因為SM0、SM1 (最新自我診斷出錯)、SD0 (最新自我診斷出錯代碼) 的內容不被重設，因此應透過程式進行重設。
 - 發生了電池出錯時。
 若更換電池後執行LEDR指令，CPU模組前面的BATTERY LED將熄燈。此時SM51 (電池過低鎖存) 也將變為OFF。
- 報警器 (F) 為ON時的動作如下所示。
 - USER LED將熄燈。
 - SD62 (報警器編號)、SD63 (報警器個數)、SD64~79 (報警器檢測編號表) 將全部被清除。

執行前	執行後
SD62 200	SD62 0
SD63 15	SD63 0
SD64 200	SD64 0
SD65 99	SD65 0
SD66 5	SD66 0
SD67 255	SD67 0
⋮	⋮
SD77	SD77 0
SD78 83	SD78 0
SD79 0	SD79 0

(1)

(1): 全部清除。

出錯

沒有運算出錯。

運轉繼續異常指令

PALERT (P)



將CPU模組設為繼續運行出錯。

梯形圖	ST
	ENO:=PALERT (EN, s) ; ENO:=PALERTP (EN, s) ;

FBD/LD



執行條件

指令	執行條件
PALERT	
PALERTP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存在出錯代碼1810H的詳細資訊2中的資料或儲存資料的元件編號	-32768~32767	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

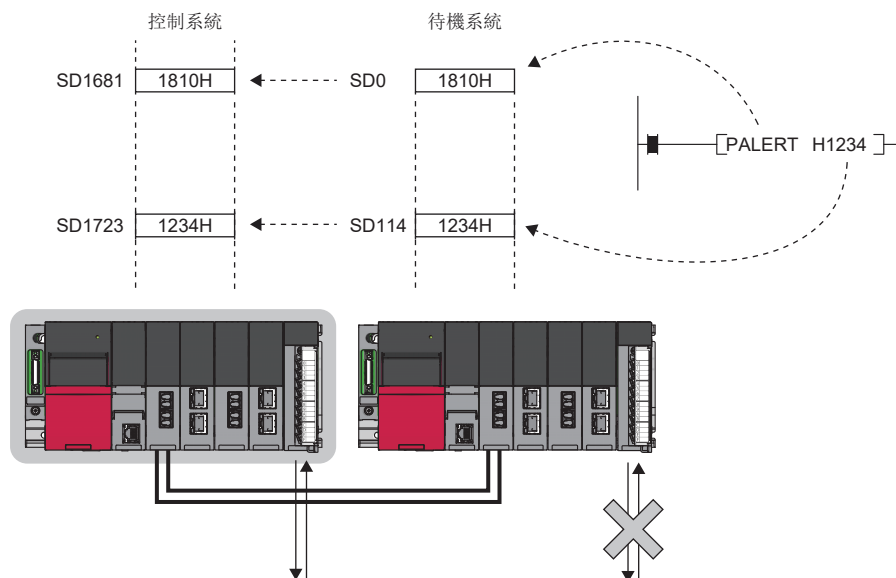
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	—	○	—	○	—	○	○	—	—	—	—

功能

- 將CPU模組設為繼續運行出錯(出錯代碼: 1810H)。執行PALERT (P)指令後, SMO (最新自我診斷出錯(包含報警器ON))為ON、CPU模組的ERROR LED將亮燈。SM1 (最新自我診斷出錯(不包含報警器ON))不為ON。
- 將(s)中指定的BIN16位元資料儲存在出錯代碼1810H的詳細資訊2中。

要點

- 透過使用PALERT(P)指令模擬系統啟動時發生繼續運轉出錯，可用於偵錯。此外，透過確認出錯代碼1810H的詳細資訊2，可以確認指令的發生位置。
- 透過將使用了PALERT(P)指令的程式作為雙系統執行程式，即使是待機系統也可以檢測出外部機器或網路中發生的異常。由於待機系統中執行的PALERT(P)指令的出錯代碼及詳細資訊2中儲存的資料將會儲存到控制系統和待機系統雙方的特殊寄存器中，因此可以透過控制系統的SD1681(最新自我診斷出錯代碼(其它系統))和SD1723(詳細資訊2(其它系統))確認待機系統中檢測出的異常內容。(但是，在控制系統中可以確認的待機系統的出錯代碼僅為最新的出錯。)



出錯

透過執行PALERT(P)指令，沒有發生的出錯代碼1810H以外的運算出錯。

運轉停止異常指令

PABORT



停止執行程式，將CPU模組設為停止出錯。

梯形圖	ST
	ENO:=PABORT (EN, s);

FBD/LD

■執行條件

指令	執行條件
PABORT	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存在出錯代碼3070H的詳細資訊2中的資料或儲存資料的元件編號	-32768~32767	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s)	○	—	○	—	○	—	○	○	—	—	—

功能

- 停止執行程式，將CPU模組設為停止出錯(出錯代碼：3070H)。執行PABORT指令後，SM0(最新自我診斷出錯(包含報警器ON))為ON、CPU模組的ERROR LED滅燈。SM1(最新自我診斷出錯(不包含報警器ON))不為ON。
- 將(s)中指定的BIN16位元資料儲存在出錯代碼3070H的詳細資訊2中。

要點 

- 透過使用PABORT指令模擬啟動系統時發生停止出錯，可用於偵錯。此外，透過確認出錯代碼3070H的詳細資訊2，可以特別指定指令的發生位置。
- 在二重化系統中，連接在待機系統的裝置等中發生問題時，透過PABORT指令設待機系統為停止出錯，可以抑制發生問題側的系統切換。
- 透過將使用了PABORT指令的程式作為雙系統執行程式，即使是待機系統也可以檢測出外部機器或網路中發生的異常。待機系統中執行的PABORT指令的出錯代碼和詳細資訊2中儲存的資料也可以從控制系統進行確認。
(☞ 758頁 PALERT(P))

出錯

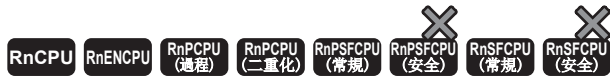
透過執行PABORT指令，沒有發生的出錯代碼3070H以外的運算出錯。

10 字元串處理

10.1 字元串處理指令

字元串比較

LD\$□、AND\$□、OR\$□

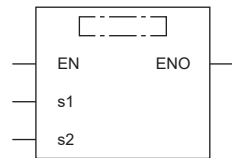


將字元串資料透過常開觸點處理進行比較運算。

梯形圖	ST*1
	ENO:=LDSTRING_□(EN, s1, s2); ENO:=ANDSTRING_□(EN, s1, s2); ENO:=ORSTRING_□(EN, s1, s2); (□中放入EQ、NE、GT、LE、LT、GE。)*2

(□中放入\$=、\$<>、\$>、\$<=、\$<、\$>=。)

FBD/LD



(□中放入LDSTRING_、ANDSTRING_、ORSTRING_與EQ、NE、GT、LE、LT、GE的組合。)*2

*1 在版本為“1.035M”及其以後的工程工具中對應。
 *2 EQ為=、NE為<>、GT為>、LE為<=、LT為<、GE為>=。

■執行條件

指令	執行條件
LD\$□、AND\$□、OR\$□	常時執行

設置數據

■內容、範圍、資料類型

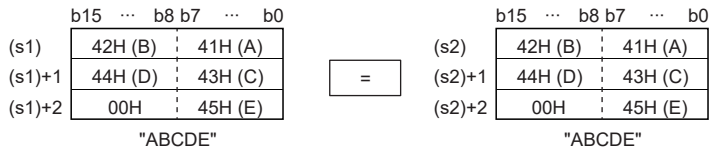
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	比較資料或儲存了比較資料的起始元件	—	字元串	ANYSTRING_SINGLE
(s2)	比較資料或儲存了比較資料的起始元件	—	字元串	ANYSTRING_SINGLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	○	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—

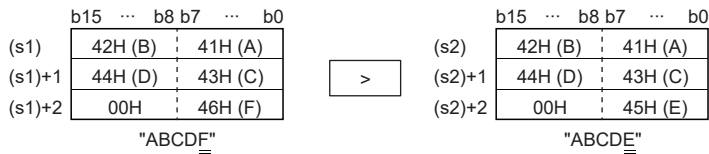
功能

- 將(s1)中指定的字元串資料與(s2)中指定的字元串資料透過常開觸點處理進行比較運算。
- 比較運算時，將字元串的ASCII碼從字元串的起始處開始進行逐個字元比較。
- (s1)、(s2)的字元串的是從指定的元件編號開始至儲存了00H的元件編號為止。
- 在所有字元串一致的情況下，比較結果即為一致。



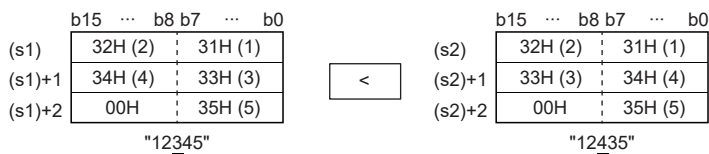
□內指令符號(梯形圖、FBD/LD)	比較運算結果
\$=、EQ	導通狀態 (ENO為ON)
\$<、NE	非導通狀態 (ENO為OFF)
\$>、GT	非導通狀態 (ENO為OFF)
\$<=、LE	導通狀態 (ENO為ON)
\$<、LT	非導通狀態 (ENO為OFF)
\$>=、GE	導通狀態 (ENO為ON)

- 在不同字元串的情況下，字元代碼較大的字元串將變大。



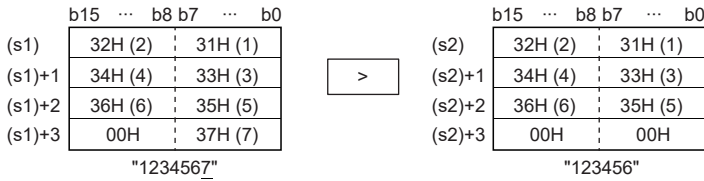
□內指令符號(梯形圖、FBD/LD)	比較運算結果
\$=、EQ	非導通狀態 (ENO為OFF)
\$<、NE	導通狀態 (ENO為ON)
\$>、GT	導通狀態 (ENO為ON)
\$<=、LE	非導通狀態 (ENO為OFF)
\$<、LT	非導通狀態 (ENO為OFF)
\$>=、GE	導通狀態 (ENO為ON)

- 在不同字元串的情況下，以首個不同字元代碼的大小決定字元串的大小。



□內指令符號(梯形圖、FBD/LD)	比較運算結果
\$=、EQ	非導通狀態 (ENO為OFF)
\$<、NE	導通狀態 (ENO為ON)
\$>、GT	非導通狀態 (ENO為OFF)
\$<=、LE	導通狀態 (ENO為ON)
\$<、LT	導通狀態 (ENO為ON)
\$>=、GE	非導通狀態 (ENO為OFF)

- 在(s1)與(s2)中的字元串資料的長度不同的情況下，較長的字元串資料將變大。



□內指令符號(梯形圖、FBD/LD)	比較運算結果
\$=、EQ	非導通狀態 (ENO為OFF)
\$<、NE	導通狀態 (ENO為ON)
\$>、GT	導通狀態 (ENO為ON)
\$<=、LE	非導通狀態 (ENO為OFF)
\$<、LT	非導通狀態 (ENO為OFF)
\$>=、GE	導通狀態 (ENO為ON)

- 在(s1)、(s2)的字元串超過16383字元的情況下，將變為非導通(ENO為OFF)。
- 在FBD/LD中使用LDSTRING_□指令時，應將左母線或常時ON的變數部件/常數部件指定到EN。
- 在將ORSTRING_□指令於FBD/LD使用的的情況下，將EN指定為TRUE時ENO將ON。EN不成為執行條件。

出錯

沒有運算出錯。

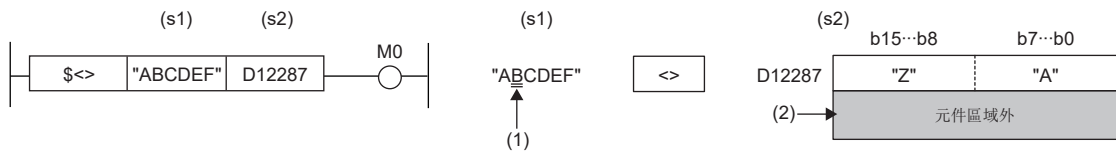
要點

在字元串比較指令中，在進行字元串比較的時，還進行下述檢查。

- 是否超出元件區域範圍的檢查
- 是否為字元串不超過16383字元的檢查

因此，在元件區域內不存在00H、或字元串超過了16383字元的情況下，檢測出字元不一致時，將在不變為非導通(ENO為OFF)的狀況下輸出比較運算結果。

下述示例表示元件區域的最終元件No. 為D12287情況下的運算結果。



因為(1) (s1)的第2字元與(s2)不相同，(s1)≠(s2)，因此運算結果為導通(ENO為ON)。

因為(2) D12287及其以後超出了元件區域，因此對D12287為止的字元串資料實施比較處理。

因為檢測出字元串的不一致，因此條件成立、結束處理。

字元串的合併

\$+(P) [操作數為2個的情況下]



連接字元串資料。

梯形圖	ST
	不對應。 (☞ 768頁 \$+(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 768頁 \$+(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
\$+	
\$+P	

設置數據

■內容、範圍、資料類型

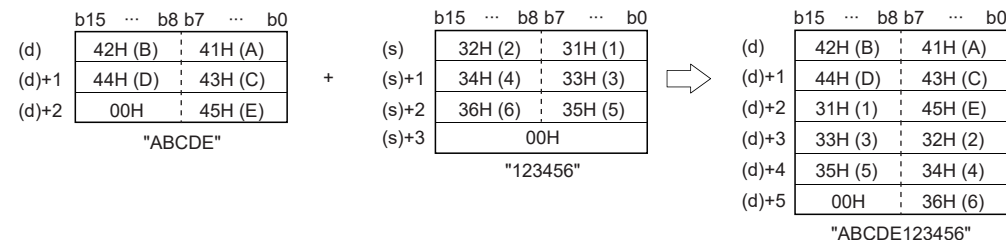
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	連接資料或儲存了資料的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存了被連接資料的起始元件	—	字元串	ANYSTRING_SINGLE

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 將(s)中指定的元件編號及其以後儲存的字元串連接到(d)中指定的元件編號及其以後儲存的字元串資料的後面，儲存到(d)中指定的元件編號及其以後。



- 對於字元串合併，將忽略表示(d)中指定的字元串的結束的00H，並在(d)的最終字元處連接(s)中指定的字元串。

出錯

出錯代碼 (SD0)	內容
2820H	(s) 中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在00H時。
	(d) 中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在00H時。
2821H	當 (s) 與 (d) 中指定的字元串儲存元件編號重複時。
3405H	當 (s) 的字元串超過了16383字元時。
	當 (d) 的字元串超過了16383字元時。
3406H	當 (s)+(d) 的字元串超過了16383字元時。
	(d) 中指定的元件編號及其以後，相應元件/標籤記憶體各設定區域的最終編號為止的點數無法儲存全部合併的字元串時。

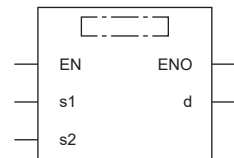
\$+(P) [操作數為3個的情況下]



連接字元串資料。

梯形圖	ST
	ENO:=STRINGPLUS (EN, s1, s2, d) ; ENO:=STRINGPLUSP (EN, s1, s2, d) ;

FBD/LD



(□中放入STRINGPLUS、STRINGPLUSP。)

■執行條件

指令	執行條件
\$+	
\$+P	

設置數據

■內容、範圍、資料類型

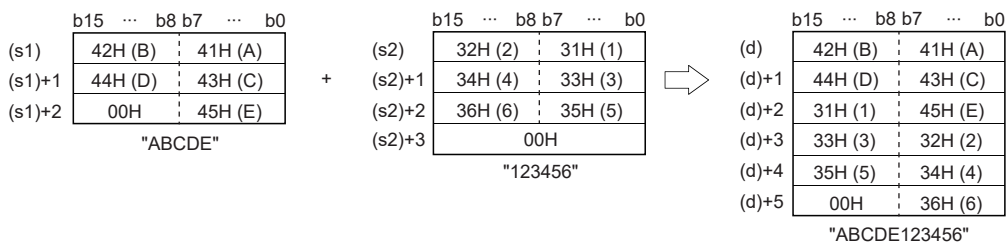
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	連接資料或儲存了資料的起始元件	—	字元串	ANYSTRING_SINGLE
(s2)	被連接的資料或儲存了被連接的資料的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存連接結果的起始元件	—	字元串	ANYSTRING_SINGLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s1)	—	—	○	—	—	—	—	○	—	—	○	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 將 (s2) 中指定的元件編號及其以後儲存的字元串連接到 (s1) 中指定的元件編號及其以後儲存的字元串資料的後面，儲存到 (d) 中指定的元件編號及其以後。



- 當字元串合併時，將忽略表示 (s1) 中指定的字元串的結束的 00H，並在 (s1) 的最終字元處連接 (s2) 中指定的字元串。

出錯

出錯代碼 (SD0)	內容
2820H	(s1) 中指定的元件編號及其以後，相應元件編號為止之間不存在 00H 時。
	(s2) 中指定的元件編號及其以後，相應元件編號為止之間不存在 00H 時。
2821H	當 (s2) 與 (d) 中指定的字元串的儲存元件編號重複時。
3405H	當 (s1) 的字元串超過 16383 字元時。
	當 (s2) 的字元串超過 16383 字元時。
3406H	當 (d) 中儲存的字元串為超出可輸出範圍的資料時。 <ul style="list-style-type: none"> 合併後的字元串超過 16383 字元 指定的元件編號及其以後，相應元件/標籤記憶體各設定區域的最終編號為止的點數無法儲存合併後的全部字元串

字元串傳送

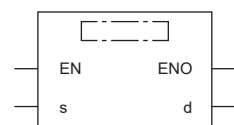
\$MOV (P)



將字元串資料傳送到指定的元件編號及其以後。

梯形圖	ST*1
	ENO:=STRINGMOV (EN, s, d) ; ENO:=STRINGMOVP (EN, s, d) ;

FBD/LD



(□中放入STRINGMOV、STRINGMOVP。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
\$MOV	
\$MOVP	

設置數據

■內容、範圍、資料類型

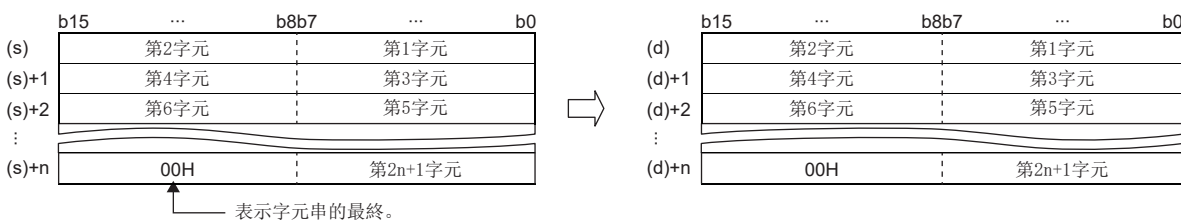
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	傳送字元串(最大255字元)或儲存了字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存傳送字元串的起始元件	—	字元串	ANYSTRING_SINGLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

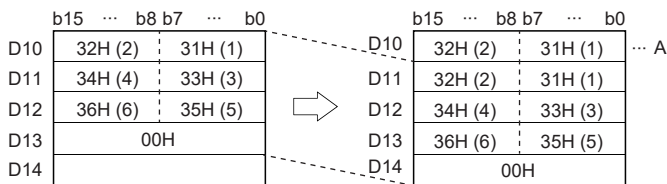
操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	○	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 將(s)中指定的字元串資料傳送到(d)中指定的元件編號及其以後。在字元串的傳送中，對(s)中指定的字元串、或從元件編號開始至儲存了00H的元件編號為止的字元串進行一次傳送。

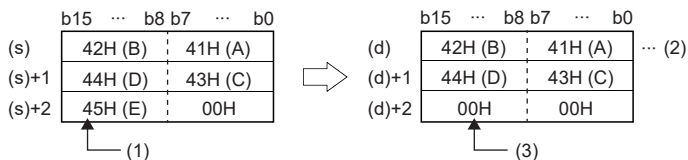


- 即使儲存了傳送的字元串資料的元件範圍(s)~(s)+n與儲存傳送後的字元串資料的元件範圍(d)~(d)+n為重複的情況下，也將正常進行處理。例如，當D10~D13中儲存的字元串傳送到D11~D14中時，其情況如下所示。



A: 變為傳送前的字元串。

- 在(s)+n的低位字節中儲存了00H的情況下，(d)+n的高位字節、低位字節均將儲存00H。



(1): 不傳送高位字節。

(2): 變為傳送前的字元串。

(3): 00H自動保存在高位字節。

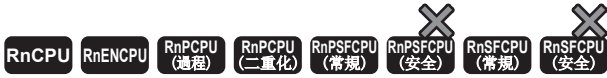
- (s)中指定了漢字等2字節資料時，轉換為移位JIS代碼。

出錯

出錯代碼(SD0)	內容
2820H	(s)中指定的元件編號及其以後，元件/標籤記憶體的各設定區域之間不存在00H時。
3405H	當(s)的字元串超過了16383字元時。
3406H	(d)中指定的元件編號及其以後，相應元件/標籤記憶體的各設定區域的最終編號為止的點數中，無法儲存指定的全部字元串時。

Unicode對應字元串傳送

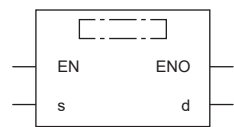
\$MOV(P)_WS



將Unicode字元串資料傳送到指定的元件編號及其以後。

梯形圖	ST*1
	ENO:=STRINGMOV_WS (EN, s, d); ENO:=STRINGMOVP_WS (EN, s, d);

FBD/LD



(□中放入STRINGMOV_WS、STRINGMOVP_WS。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
\$MOV_WS	
\$MOVP_WS	

設置數據

■內容、範圍、資料類型

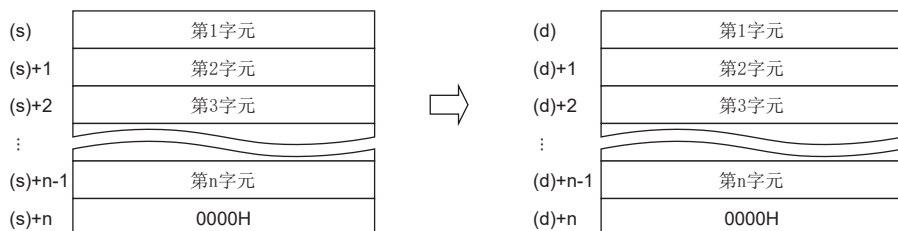
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	傳送的Unicode字元串(最大255字元)或儲存了Unicode字元串的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(d)	儲存傳送的Unicode字元串的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	○	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 將(s)中指定的Unicode字元串資料，傳送到(d)中指定的元件編號及其以後。在Unicode字元串的傳送中，對(s)中指定的Unicode字元串、或從元件編號開始至儲存了0000H的元件編號為止的Unicode字元串進行一次傳送。



- 即使儲存了傳送的Unicode字元串資料的元件範圍(s)~(s)+n與儲存傳送後的字元串資料的元件範圍(d)~(d)+n為重複的情況下，也將正常進行處理。例如，當D10~D13中儲存的字元串傳送到D11~D14中時，其情況如下所示。



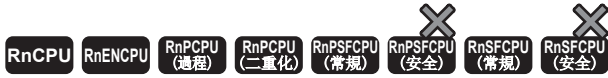
A: 傳送前的值的狀態

出錯

出錯代碼 (SD0)	內容
2820H	(s) 中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在0000H時。
3405H	當(s)的Unicode字元串超過了16383字元時。
3406H	(d) 中指定的元件編號及其以後，相應元件/標籤記憶體各設定區域的最終編號為止的點數中，無法儲存指定的全部Unicode字元串時。

BIN16位元資料→10進制ASCII轉換

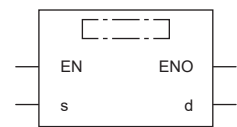
BINDA(P) (_U)



將BIN16位元資料轉換為10進制數ASCII碼。

梯形圖	ST	
	ENO:=BINDA(EN, s, d); ENO:=BINDAP(EN, s, d)	ENO:=BINDA_U(EN, s, d); ENO:=BINDAP_U(EN, s, d)

FBD/LD



■執行條件

指令	執行條件
BINDA BINDA_U	
BINDAP BINDAP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s)	BINDA(P)	進行ASCII轉換的BIN資料	-32768~32767	帶符號BIN16位元	ANY16_S
	BINDA(P)_U		0~65535	無符號BIN16位元	ANY16_U
(d)	儲存轉換結果的起始元件	—	字元串	ANYSTRING_SINGLE	
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 將(s)中指定的BIN16位元資料，以10進制數表示時的數值轉換為ASCII碼後，儲存到(d)中指定的元件編號及其以後。
- 根據SM705(轉換位數切換)的狀態，(d)中設定的10進制ASCII資料的形式有所不同。

SM705*1的狀態	(d)的儲存形式	參閱目標
OFF	以固定位數(符號+5位)儲存	776頁 SM705(轉換位數切換)為OFF的情況下的動作
ON	根據(s)的值各位向前填充對齊並儲存	776頁 SM705(轉換位數切換)為ON的情況下的動作

*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

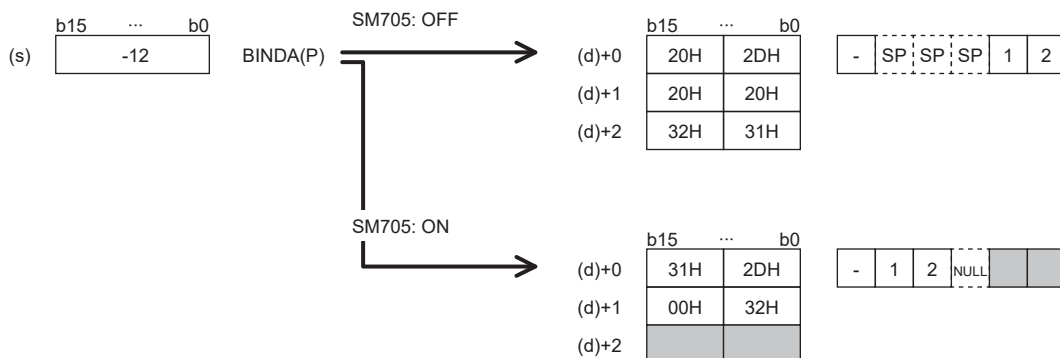
SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■動作的概要

SM705(轉換位數切換)為OFF的情況下與ON的情況下，其各自的動作如下所示。

例

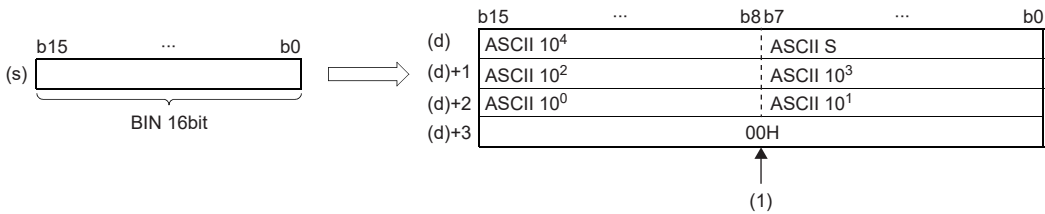
(s)中數值儲存“-12”的狀態，並執行BINDA(P)指令的情況下



- SM705(轉換位數切換)為OFF的情況下為位數固定。第1個字元為符號，在上述的示例中為2DH(-)。(s)為0或正數的情況下，第1個字元為20H(空格)。)數值部分向右填充對齊並儲存。數值部分未滿5位時，高位位的ASCII碼為20H(空格)。
- SM705(轉換位數切換)為ON的情況下為向前填充對齊。數值部分未滿5位時，在終端儲存00H。

■SM705 (轉換位數切換) 為OFF的情況下的動作

(d) ~ (d)+2中, 10進制ASCII資料以固定位數儲存。



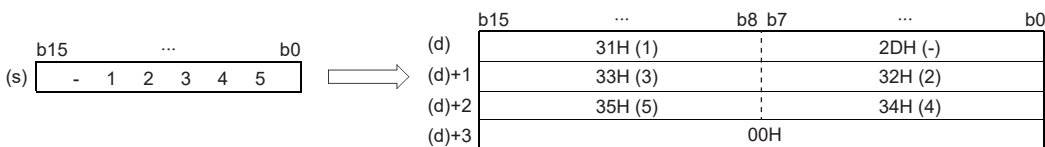
- ASCII S: 符號的ASCII碼*1
- ASCII 10⁴: 萬位的ASCII碼*2
- ASCII 10³: 千位的ASCII碼*3
- ASCII 10²: 百位的ASCII碼*4
- ASCII 10¹: 十位的ASCII碼*5
- ASCII 10⁰: 個位的ASCII碼

(1): (d)+3中, SM701 (輸出字元數切換) 為OFF時儲存00H, ON時不變化。

- *1 0或正數的情況下儲存20H(空格), 負數的情況下儲存2DH(-)。
- *2 數值部分為4位及其以下的情況下, 在ASCII 10⁴中儲存20H(空格)。
- *3 數值部分為3位及其以下的情況下, 在ASCII 10³中儲存20H(空格)。
- *4 數值部分為2位及其以下的情況下, 在ASCII 10²中儲存20H(空格)。
- *5 數值部分為1位的情況下, 在ASCII 10¹中儲存20H(空格)。

例

BINDA(P)指令中, (s)中指定“-12345”的情況下



■SM705 (轉換位數切換) 為ON的情況下的動作

(d)中, 10進制ASCII資料以向前填充對齊儲存。

(s)的值與(d)中儲存的值的示例如下所示。

(s) 的值	(d) ~ (d)+2 的內容	(s) 的值	(d) ~ (d)+2 的內容																																													
<ul style="list-style-type: none"> • 0 • 正數(數值部分 1位) 	<ul style="list-style-type: none"> • (d)的高位字節中儲存00H。 • (d)+1 ~ (d)+2不變化。 <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(d)</td><td>00H</td><td></td><td>ASCII 10⁰</td><td></td> </tr> <tr> <td>(d)+1</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>(d)+2</td><td></td><td></td><td></td><td></td> </tr> </table> </div>	b15	...	b8 b7	...	b0	(d)	00H		ASCII 10 ⁰		(d)+1					(d)+2					<ul style="list-style-type: none"> • 正數(數值部分 2位) • 負數(數值部分 1位) 	<ul style="list-style-type: none"> • (d)+1中儲存00H。 • (d)+2不變化。 <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(d)</td><td>ASCII 10⁰</td><td></td><td>ASCII 10¹ / 2DH (-)</td><td></td> </tr> <tr> <td>(d)+1</td><td></td><td></td><td>00H</td><td></td> </tr> <tr> <td>(d)+2</td><td></td><td></td><td></td><td></td> </tr> </table> </div>	b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁰		ASCII 10 ¹ / 2DH (-)		(d)+1			00H		(d)+2									
b15	...	b8 b7	...	b0																																												
(d)	00H		ASCII 10 ⁰																																													
(d)+1																																																
(d)+2																																																
b15	...	b8 b7	...	b0																																												
(d)	ASCII 10 ⁰		ASCII 10 ¹ / 2DH (-)																																													
(d)+1			00H																																													
(d)+2																																																
<ul style="list-style-type: none"> • 正數(數值部分 5位) • 負數(數值部分 4位) 	<ul style="list-style-type: none"> • (d)+2的高位字節中儲存00H。 <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(d)</td><td>ASCII 10³</td><td></td><td>ASCII 10⁴ / 2DH (-)</td><td></td> </tr> <tr> <td>(d)+1</td><td>ASCII 10¹</td><td></td><td>ASCII 10²</td><td></td> </tr> <tr> <td>(d)+2</td><td>00H</td><td></td><td>ASCII 10⁰</td><td></td> </tr> </table> </div>	b15	...	b8 b7	...	b0	(d)	ASCII 10 ³		ASCII 10 ⁴ / 2DH (-)		(d)+1	ASCII 10 ¹		ASCII 10 ²		(d)+2	00H		ASCII 10 ⁰		<ul style="list-style-type: none"> • 負數(數值部分 5位) 	<ul style="list-style-type: none"> • (1): (d)+3中, 僅SM701 (輸出字元數切換) 為OFF時儲存00H。 <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td> </tr> <tr> <td>(d)</td><td>ASCII 10⁴</td><td></td><td>2DH (-)</td><td></td> </tr> <tr> <td>(d)+1</td><td>ASCII 10²</td><td></td><td>ASCII 10³</td><td></td> </tr> <tr> <td>(d)+2</td><td>ASCII 10⁰</td><td></td><td>ASCII 10¹</td><td></td> </tr> <tr> <td>(d)+3</td><td></td><td></td><td>00H</td><td></td> </tr> </table> <p style="text-align: center;">↑ (1)</p> </div>	b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁴		2DH (-)		(d)+1	ASCII 10 ²		ASCII 10 ³		(d)+2	ASCII 10 ⁰		ASCII 10 ¹		(d)+3			00H	
b15	...	b8 b7	...	b0																																												
(d)	ASCII 10 ³		ASCII 10 ⁴ / 2DH (-)																																													
(d)+1	ASCII 10 ¹		ASCII 10 ²																																													
(d)+2	00H		ASCII 10 ⁰																																													
b15	...	b8 b7	...	b0																																												
(d)	ASCII 10 ⁴		2DH (-)																																													
(d)+1	ASCII 10 ²		ASCII 10 ³																																													
(d)+2	ASCII 10 ⁰		ASCII 10 ¹																																													
(d)+3			00H																																													

- ASCII 10⁴: 萬位的ASCII碼
- ASCII 10³: 千位的ASCII碼
- ASCII 10²: 百位的ASCII碼
- ASCII 10¹: 十位的ASCII碼
- ASCII 10⁰: 個位的ASCII碼

- 運算位數未滿最大位數(符號+5位)的情況下, 無論是SM701 (輸出字元數切換) 的ON/OFF, 字元串的終端將儲存00H。字元串的終端為低位字節的情況下, 高位字節中也將儲存00H。
- 運算位數為最大位數(符號+5位)的情況下, SM701 (輸出字元數切換) 為OFF時, (d)+3中將儲存00H。SM701 (輸出字元數切換) 為ON時(d)+3不變化。

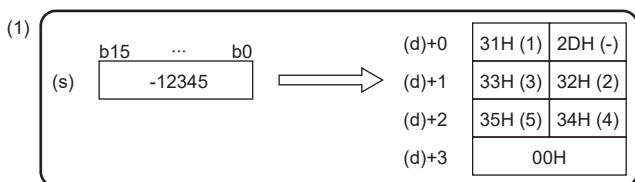
出錯

沒有運算出錯。

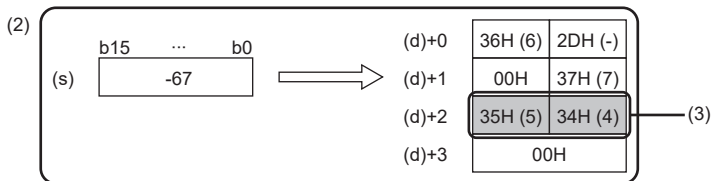
注意事項

SM705(轉換位數切換)為ON時，(d)中運算結果將只儲存有效位數部分。因此，連續執行BINDA(P)(_U)指令，運算結果儲存在同一元件的情況下，(d)中的一部分中有可能上次的運算結果不被覆蓋儲存而殘留。

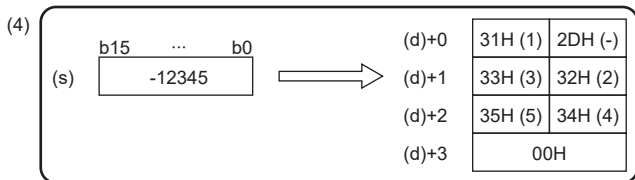
[例] (s)為“-12345”時執行BINDA(P)指令，且(s)為“-67”時執行BINDA(P)指令的示例



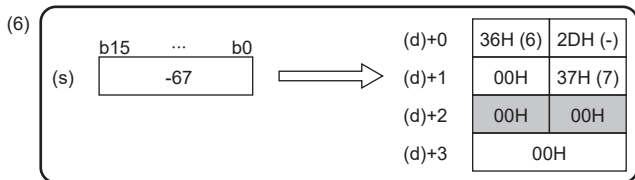
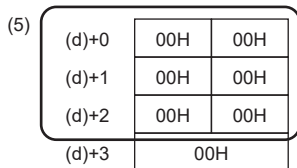
- (1) 將“-12345”轉換為字元串
- (2) 將“-67”轉換為字元串
- (3) (d)+2中留有上次轉換結果的一部分



想避免此狀況，應建立程式使儲存區域(d)+0~(d)+2全部清除後執行BINDA(P)(_U)指令。

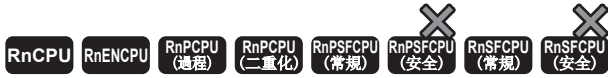


- (4) 將“-12345”轉換為字元串
- (5) 將(d)+0~(d)+2清除
- (6) 將“-67”轉換為字元串



BIN32位元資料→10進制ASCII轉換

DBINDA(P) (_U)



將BIN32位元資料轉換為10進制數ASCII碼。

梯形圖	ST	
	ENO:=DBINDA (EN, s, d) ; ENO:=DBINDAP (EN, s, d) ;	ENO:=DBINDA_U (EN, s, d) ; ENO:=DBINDAP_U (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
DBINDA DBINDA_U	
DBINDAP DBINDAP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(s)	DBINDA(P)	進行ASCII轉換的BIN資料	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
	DBINDA(P)_U		0~4294967295	無符號BIN32位元	ANY32_U
(d)		儲存轉換結果的起始元件	—	字元串	ANYSTRING_SINGLE
EN		執行條件	—	位元	BOOL
ENO		執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—

功能

- 將(s)中指定的BIN32位元資料，以10進制數表示時的各位的數值轉換為ASCII碼後，儲存到(d)中指定的元件編號及其以後。
- 根據SM705(轉換位數切換)的狀態，(d)中設定的10進制ASCII資料的形式有所不同。

SM705*1的狀態	(d)的儲存形式	參閱目標
OFF	以固定位數(符號+10位)儲存	780頁 SM705(轉換位數切換)為OFF的情況下的動作
ON	根據(s)的值各位向前填充對齊並儲存	781頁 SM705(轉換位數切換)為ON的情況下的動作

*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

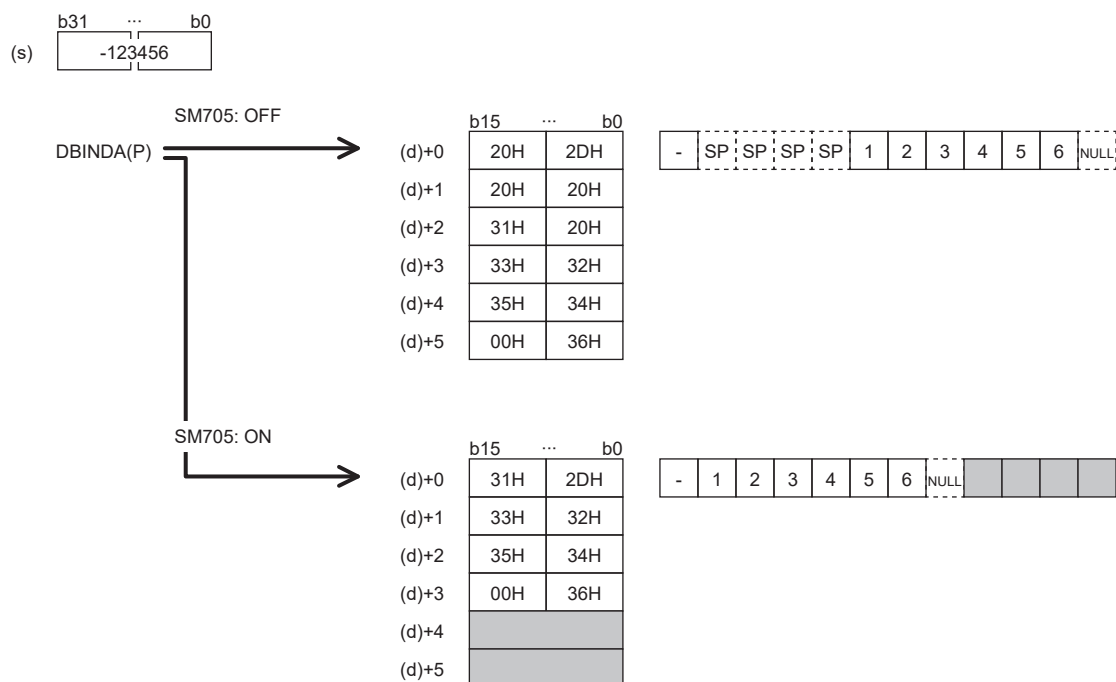
SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■動作的概要

SM705(轉換位數切換)為OFF的情況下與ON的情況下，其各自的動作如下所示。

例

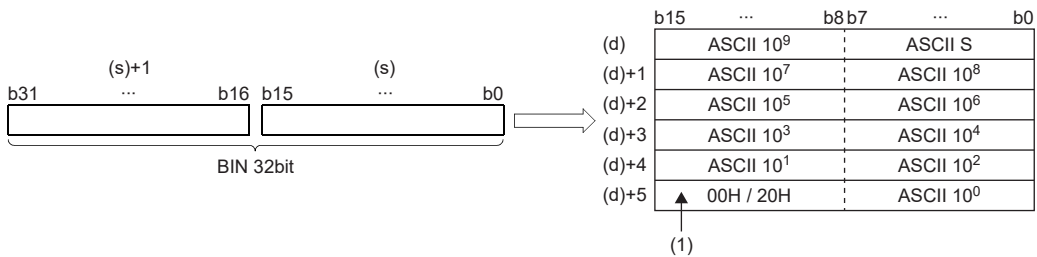
(s)中儲存數值“-123456”的狀態，並執行DBINDA(P)指令的情況下



- SM705(轉換位數切換)為OFF的情況下為位數固定。第1個字元為符號，在上述的示例中為2DH(-)。(s)為0或正數的情況下，第1個字元為20H(空格)。)數值部分向右填充對齊並儲存。數值部分未滿10位時，高位位的ASCII碼為20H(空格)。
- SM705(轉換位數切換)為ON的情況下為向前填充對齊。數值部分未滿10位時，在終端儲存00H。

■SM705 (轉換位數切換) 為OFF的情況下的動作

(d) ~ (d)+5中, 10進制ASCII資料以固定位數儲存。



ASCII S: 符號的ASCII碼*1

ASCII 10⁹: 十億位的ASCII碼*2

ASCII 10⁸: 億位的ASCII碼*3

⋮

ASCII 10¹: 十位的ASCII碼*4

ASCII 10⁰: 個位的ASCII碼

(1): (d)+5的高位字節中, SM701(輸出字元數切換)為OFF時儲存00H, ON時儲存20H(空格)。

*1 0或正數的情況下儲存20H(空格), 負數的情況下儲存2DH(-)。

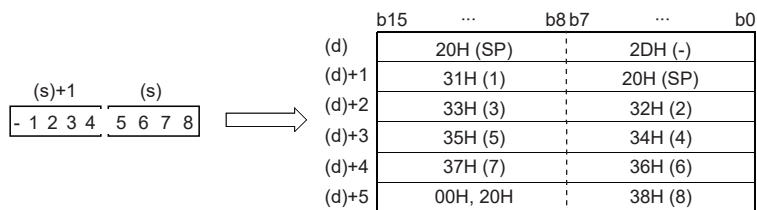
*2 數值部分為9位及其以下的情況下, 在ASCII 10⁹中儲存20H(空格)。

*3 數值部分為8位及其以下的情況下, 在ASCII 10⁸中儲存20H(空格)。

*4 數值部分為1位的情況下, 在ASCII 10¹中儲存20H(空格)。

例

在(s)中指定-12345678的情況下(指定帶符號的情況下)



■SM705 (轉換位數切換) 為ON的情況下的動作

(d) 中，10進制ASCII資料以向前填充對齊儲存。

(s) 的值與 (d) 中儲存的值的示例如下所示。

(s) 的值	(d) ~ (d)+2 的內容	(s) 的值	(d) ~ (d)+2 的內容																																																																						
<ul style="list-style-type: none"> 0 正數 (數值部分 1位) 	<ul style="list-style-type: none"> (d) 的高位字節中儲存00H。 (d)+1及其以後不變化。 <table border="1"> <tr> <td>b15</td> <td>...</td> <td>b8b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(d)</td> <td>00H</td> <td>...</td> <td>ASCII 10⁰</td> <td></td> </tr> <tr> <td>(d)+1</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>(d)+2</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>(d)+3</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>(d)+4</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>(d)+5</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	b15	...	b8b7	...	b0	(d)	00H	...	ASCII 10 ⁰		(d)+1					(d)+2					(d)+3					(d)+4					(d)+5					<ul style="list-style-type: none"> 正數 (數值部分 2位) 負數 (數值部分 1位) 	<ul style="list-style-type: none"> (d)+1中儲存00H。 (d)+2及其以後不變化。 <table border="1"> <tr> <td>b15</td> <td>...</td> <td>b8b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(d)</td> <td>ASCII 10⁰</td> <td>...</td> <td>ASCII 10¹ / 2DH (-)</td> <td></td> </tr> <tr> <td>(d)+1</td> <td></td> <td></td> <td>00H</td> <td></td> </tr> <tr> <td>(d)+2</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>(d)+3</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>(d)+4</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>(d)+5</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	b15	...	b8b7	...	b0	(d)	ASCII 10 ⁰	...	ASCII 10 ¹ / 2DH (-)		(d)+1			00H		(d)+2					(d)+3					(d)+4					(d)+5				
b15	...	b8b7	...	b0																																																																					
(d)	00H	...	ASCII 10 ⁰																																																																						
(d)+1																																																																									
(d)+2																																																																									
(d)+3																																																																									
(d)+4																																																																									
(d)+5																																																																									
b15	...	b8b7	...	b0																																																																					
(d)	ASCII 10 ⁰	...	ASCII 10 ¹ / 2DH (-)																																																																						
(d)+1			00H																																																																						
(d)+2																																																																									
(d)+3																																																																									
(d)+4																																																																									
(d)+5																																																																									
<ul style="list-style-type: none"> 正數 (數值部分 9位) 負數 (數值部分 8位) 	<ul style="list-style-type: none"> (d)+4的高位字節中儲存00H。 (d)+5及其以後不變化。 <table border="1"> <tr> <td>b15</td> <td>...</td> <td>b8b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(d)</td> <td>ASCII 10⁷</td> <td>...</td> <td>ASCII 10⁸ / 2DH (-)</td> <td></td> </tr> <tr> <td>(d)+1</td> <td>ASCII 10⁵</td> <td>...</td> <td>ASCII 10⁶</td> <td></td> </tr> <tr> <td>(d)+2</td> <td>ASCII 10³</td> <td>...</td> <td>ASCII 10⁴</td> <td></td> </tr> <tr> <td>(d)+3</td> <td>ASCII 10¹</td> <td>...</td> <td>ASCII 10²</td> <td></td> </tr> <tr> <td>(d)+4</td> <td>00H</td> <td>...</td> <td>ASCII 10⁰</td> <td></td> </tr> <tr> <td>(d)+5</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	b15	...	b8b7	...	b0	(d)	ASCII 10 ⁷	...	ASCII 10 ⁸ / 2DH (-)		(d)+1	ASCII 10 ⁵	...	ASCII 10 ⁶		(d)+2	ASCII 10 ³	...	ASCII 10 ⁴		(d)+3	ASCII 10 ¹	...	ASCII 10 ²		(d)+4	00H	...	ASCII 10 ⁰		(d)+5					<ul style="list-style-type: none"> 正數 (數值部分 10位) 負數 (數值部分 9位) 	<ul style="list-style-type: none"> (d)+5中儲存00H。 <table border="1"> <tr> <td>b15</td> <td>...</td> <td>b8b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(d)</td> <td>ASCII 10⁸</td> <td>...</td> <td>ASCII 10⁹ / 2DH (-)</td> <td></td> </tr> <tr> <td>(d)+1</td> <td>ASCII 10⁶</td> <td>...</td> <td>ASCII 10⁷</td> <td></td> </tr> <tr> <td>(d)+2</td> <td>ASCII 10⁴</td> <td>...</td> <td>ASCII 10⁵</td> <td></td> </tr> <tr> <td>(d)+3</td> <td>ASCII 10²</td> <td>...</td> <td>ASCII 10³</td> <td></td> </tr> <tr> <td>(d)+4</td> <td>ASCII 10⁰</td> <td>...</td> <td>ASCII 10¹</td> <td></td> </tr> <tr> <td>(d)+5</td> <td></td> <td></td> <td>00H</td> <td></td> </tr> </table>	b15	...	b8b7	...	b0	(d)	ASCII 10 ⁸	...	ASCII 10 ⁹ / 2DH (-)		(d)+1	ASCII 10 ⁶	...	ASCII 10 ⁷		(d)+2	ASCII 10 ⁴	...	ASCII 10 ⁵		(d)+3	ASCII 10 ²	...	ASCII 10 ³		(d)+4	ASCII 10 ⁰	...	ASCII 10 ¹		(d)+5			00H	
b15	...	b8b7	...	b0																																																																					
(d)	ASCII 10 ⁷	...	ASCII 10 ⁸ / 2DH (-)																																																																						
(d)+1	ASCII 10 ⁵	...	ASCII 10 ⁶																																																																						
(d)+2	ASCII 10 ³	...	ASCII 10 ⁴																																																																						
(d)+3	ASCII 10 ¹	...	ASCII 10 ²																																																																						
(d)+4	00H	...	ASCII 10 ⁰																																																																						
(d)+5																																																																									
b15	...	b8b7	...	b0																																																																					
(d)	ASCII 10 ⁸	...	ASCII 10 ⁹ / 2DH (-)																																																																						
(d)+1	ASCII 10 ⁶	...	ASCII 10 ⁷																																																																						
(d)+2	ASCII 10 ⁴	...	ASCII 10 ⁵																																																																						
(d)+3	ASCII 10 ²	...	ASCII 10 ³																																																																						
(d)+4	ASCII 10 ⁰	...	ASCII 10 ¹																																																																						
(d)+5			00H																																																																						
<ul style="list-style-type: none"> 負數 (數值部分 10位) 	<ul style="list-style-type: none"> (d)+5的高位字節 (1) 中，SM701 (輸出字元數切換) 為OFF時儲存00H，ON時儲存20H (空格)。 <table border="1"> <tr> <td>b15</td> <td>...</td> <td>b8b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(d)</td> <td>ASCII 10⁹</td> <td>...</td> <td>2DH (-)</td> <td></td> </tr> <tr> <td>(d)+1</td> <td>ASCII 10⁷</td> <td>...</td> <td>ASCII 10⁸</td> <td></td> </tr> <tr> <td>(d)+2</td> <td>ASCII 10⁵</td> <td>...</td> <td>ASCII 10⁶</td> <td></td> </tr> <tr> <td>(d)+3</td> <td>ASCII 10³</td> <td>...</td> <td>ASCII 10⁴</td> <td></td> </tr> <tr> <td>(d)+4</td> <td>ASCII 10¹</td> <td>...</td> <td>ASCII 10²</td> <td></td> </tr> <tr> <td>(d)+5</td> <td>00H / 20H</td> <td>...</td> <td>ASCII 10⁰</td> <td></td> </tr> </table> <p>(1)</p>	b15	...	b8b7	...	b0	(d)	ASCII 10 ⁹	...	2DH (-)		(d)+1	ASCII 10 ⁷	...	ASCII 10 ⁸		(d)+2	ASCII 10 ⁵	...	ASCII 10 ⁶		(d)+3	ASCII 10 ³	...	ASCII 10 ⁴		(d)+4	ASCII 10 ¹	...	ASCII 10 ²		(d)+5	00H / 20H	...	ASCII 10 ⁰		ASCII 10 ⁹ : 十億位的ASCII碼 ASCII 10 ⁸ : 一億位的ASCII碼 ... ASCII 10 ¹ : 十位的ASCII碼 ASCII 10 ⁰ : 個位的ASCII碼																																				
b15	...	b8b7	...	b0																																																																					
(d)	ASCII 10 ⁹	...	2DH (-)																																																																						
(d)+1	ASCII 10 ⁷	...	ASCII 10 ⁸																																																																						
(d)+2	ASCII 10 ⁵	...	ASCII 10 ⁶																																																																						
(d)+3	ASCII 10 ³	...	ASCII 10 ⁴																																																																						
(d)+4	ASCII 10 ¹	...	ASCII 10 ²																																																																						
(d)+5	00H / 20H	...	ASCII 10 ⁰																																																																						

- 運算位數未滿最大位數 (符號+10位) 的情況下，無論是SM701 (輸出字元數切換) 的ON/OFF，字元串的終端將儲存00H。字元串的終端為低位字節的情況下，高位字節中也將儲存00H。
- 運算位數為最大位數 (符號+10位) 的情況下，SM701 (輸出字元數切換) 為OFF時，(d)+5的高位字節中將儲存00H。SM701 (輸出字元數切換) 為ON時，(d)+5的高位字節中將儲存20H (空格)。

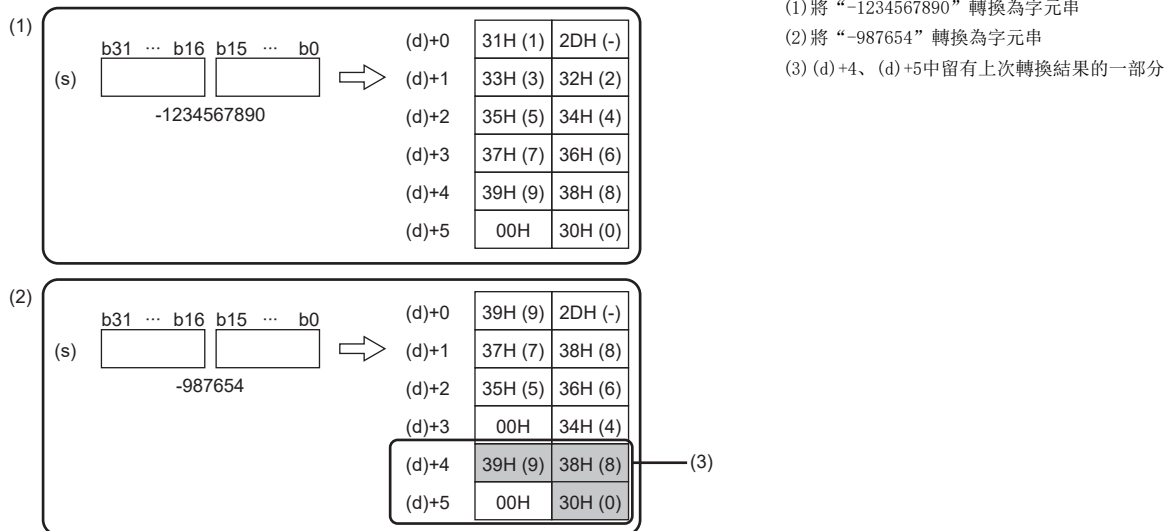
出錯

沒有運算出錯。

注意事項

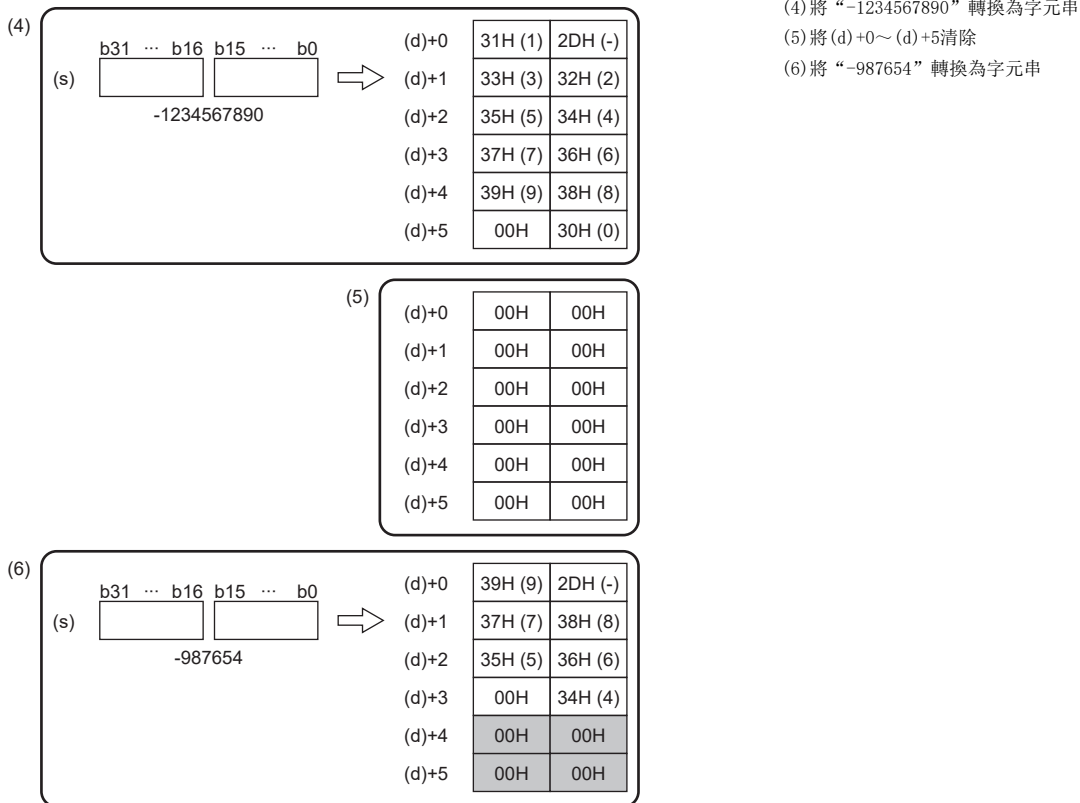
SM705(轉換位數切換)為ON時，(d)中運算結果將只儲存有效位數部分。因此，連續執行DBINDA(P) (U)指令，運算結果儲存在同一元件的情況下，(d)中的一部分中有可能上次的運算結果不被覆蓋儲存而殘留。

[例] (s)為“-1234567890”時執行DBINDA(P)指令，且(s)為“-987654”時執行DBINDA(P)指令的示例



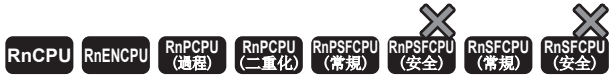
- (1) 將“-1234567890”轉換為字元串
- (2) 將“-987654”轉換為字元串
- (3) (d)+4、(d)+5中留有上次轉換結果的一部分

想避免此狀況，應建立程式使儲存區域(d)+0~(d)+5全部清除後執行DBINDA(P) (U)指令。



BIN16位元資料→16進制ASCII轉換

BINHA (P)



將BIN16位元資料轉換為16進制數ASCII碼。

梯形圖	ST
	ENO:=BINHA (EN, s, d) ; ENO:=BINHAP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
BINHA	
BINHAP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行ASCII轉換的BIN資料	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存轉換結果的起始元件	—	字元串	ANYSTRING_SINGLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將(s)中指定的BIN16位元資料，以16進制數表示時的數值轉換為ASCII碼後，儲存到(d)中指定的元件編號及其以後。
- 根據SM705(轉換位數切換)的狀態，(d)中設定的16進制ASCII資料的形式有所不同。

SM705*1的狀態	(d)的儲存形式	參閱目標
OFF	以固定位數(4位)儲存	785頁 SM705(轉換位數切換)為OFF的情況下的動作
ON	根據(s)的值各位向前填充對齊並儲存	785頁 SM705(轉換位數切換)為ON的情況下的動作

*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

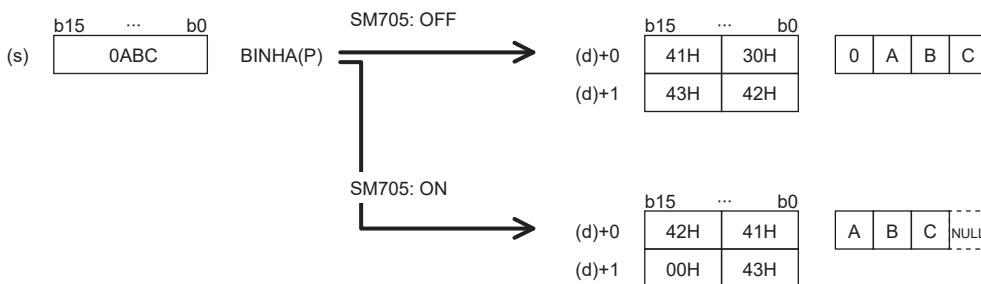
SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■動作的概要

SM705(轉換位數切換)為OFF的情況下與ON的情況下，其各自的動作如下所示。

例

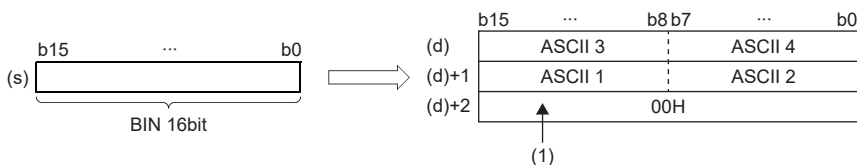
(s)中數值儲存了BIN16位元資料“0ABCH”的狀態，並執行BINHA(P)指令的情況下



- SM705(轉換位數切換)為OFF的情況下為位數固定。“0ABC”的4位被轉換為ASCII資料儲存。
- SM705(轉換位數切換)為ON的情況下為向前填充對齊。省略“0ABC”起始的0後的“ABC”被轉換為ASCII資料儲存，在終端儲存00H。

■SM705 (轉換位數切換) 為OFF的情況下的動作

(d) 中，16進制ASCII資料以固定位數(4位) 儲存。



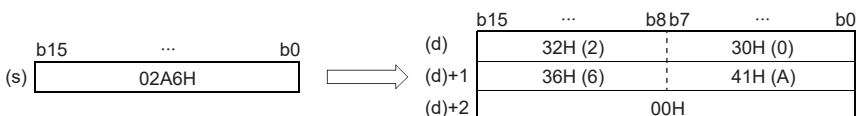
ASCII □: 第□位的ASCII碼

(1): (d)+2中，SM701(輸出字元數切換)為OFF時儲存00H，ON時不變化。

- (d) 中儲存的運算結果將被處理為4位的16進制數。因此，有效位數的左側的0將被作為“0”處理。(進行0填充。)

例

在(s) 中指定02A6H的情況下



■SM705 (轉換位數切換) 為ON的情況下的動作

(d) 中，省略有效位數的左側的0後的位數(最大4位)的16進制ASCII資料以向前填充對齊儲存。

(s) 的值與 (d) 中儲存的值的示例如下所示。

(s) 的值	(d) ~ (d)+1 的內容	(s) 的值	(d) ~ (d)+1 的內容
• 0H~FH	<ul style="list-style-type: none"> • (d) 的高位字節中儲存00H。 • (d)+1及其以後不變化。 	• 10H~FFH	<ul style="list-style-type: none"> • (d)+1中儲存00H。
• 100H~FFFH	<ul style="list-style-type: none"> • (d)+1的高位字節中儲存00H。 	• 1000H~FFFFH	<ul style="list-style-type: none"> • (1): SM701(輸出字元數切換)為OFF時(d)+2中儲存00H。SM701(輸出字元數切換)為ON時(d)+2不變化。

ASCII □: 第□位的ASCII碼

- 位數未滿最大位數(4位)的情況下，無論是SM701(輸出字元數切換)的ON/OFF，字元串的終端將儲存00H。字元串的終端為低位字節的情況下，高位字節中也將儲存00H。
- 位數為最大位數(4位)的情況下，SM701(輸出字元數切換)為OFF時，(d)+2中將儲存00H。SM701(輸出字元數切換)為ON時(d)+2不變化。

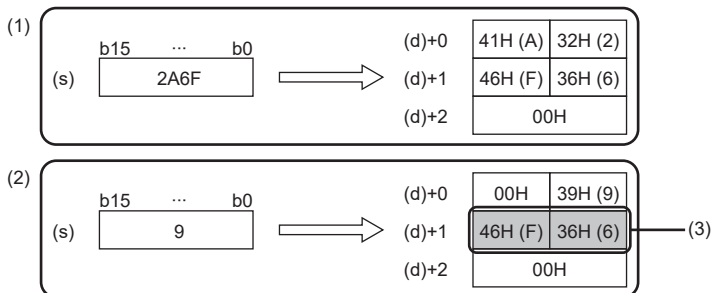
出錯

沒有運算出錯。

注意事項

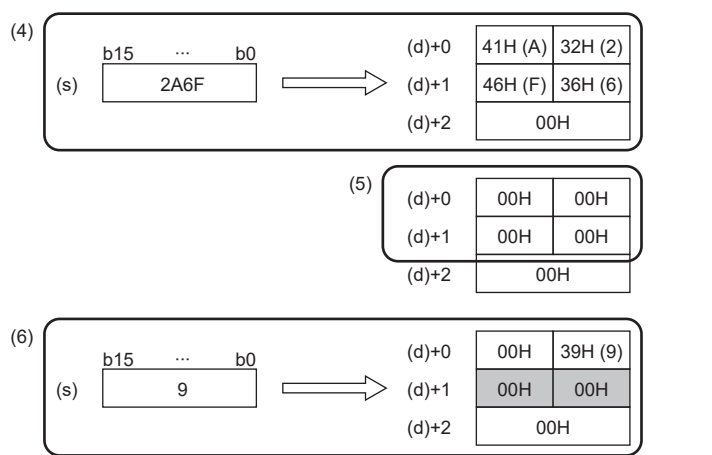
SM705(轉換位數切換)為ON時，(d)中運算結果將只儲存有效位數部分。因此，連續執行BINHA(P)指令，運算結果儲存在同一元件的情況下，(d)中的一部分中有可能上次的運算結果不被覆蓋儲存而殘留。

[例] (s)為“2A6F”時執行BINHA(P)指令，且(s)為“9”時執行BINHA(P)指令的示例



- (1) 將“2A6F”轉換為字元串
- (2) 將“9”轉換為字元串
- (3) (d)+1中留有上次轉換結果的一部分

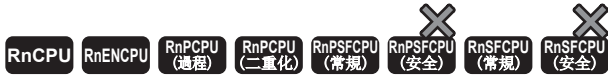
想避免此狀況，應建立程式使儲存區域(d)+0~(d)+1全部清除後執行BINHA(P)指令。



- (4) 將“2A6F”轉換為字元串
- (5) 將(d)+0~(d)+1清除
- (6) 將“9”轉換為字元串

BIN32位元資料→16進制ASCII轉換

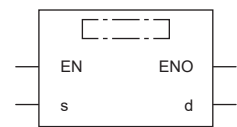
DBINHA (P)



將BIN32位元資料轉換為16進制數ASCII碼。

梯形圖	ST
	ENO:=DBINHA (EN, s, d) ; ENO:=DBINHAP (EN, s, d) ;

FBD/LD



■執行條件

指令	執行條件
DBINHA	
DBINHAP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行ASCII轉換的BIN資料	-2147483648~2147483647	帶符號BIN32位元	ANY32
(d)	儲存轉換結果的起始元件	—	字元串	ANYSTRING_SINGLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將(s)中指定的BIN32位元資料，以16進制數表示時的各位的數值轉換為ASCII碼後，儲存到(d)中指定的元件編號及其以後。
- 根據SM705(轉換位數切換)的狀態，(d)中設定的10進制ASCII資料的形式有所不同。

SM705*1的狀態	(d)的儲存形式	參閱目標
OFF	以固定位數(8位)儲存	789頁 SM705(轉換位數切換)為OFF的情況下的動作
ON	根據(s)的值各位向前填充對齊並儲存	789頁 SM705(轉換位數切換)為ON的情況下的動作

*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

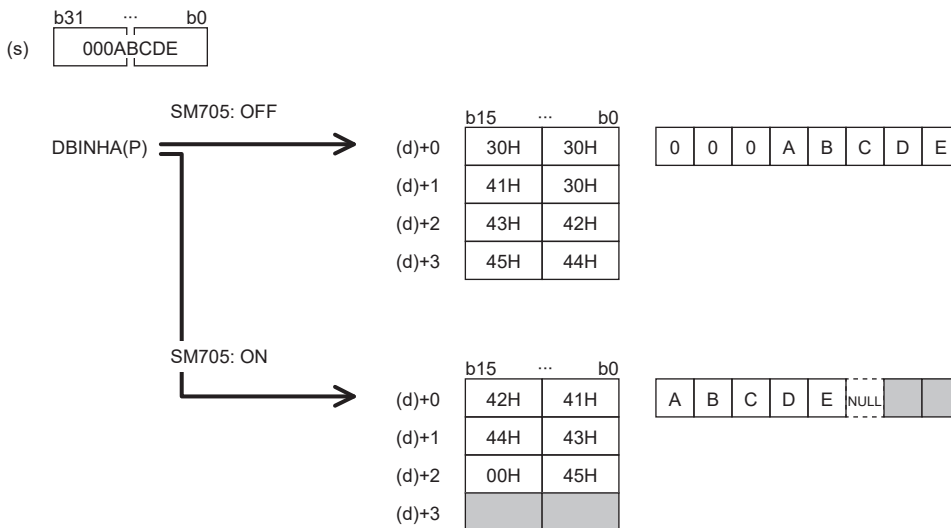
SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■動作的概要

SM705(轉換位數切換)為OFF的情況下與ON的情況下，其各自的動作如下所示。

例

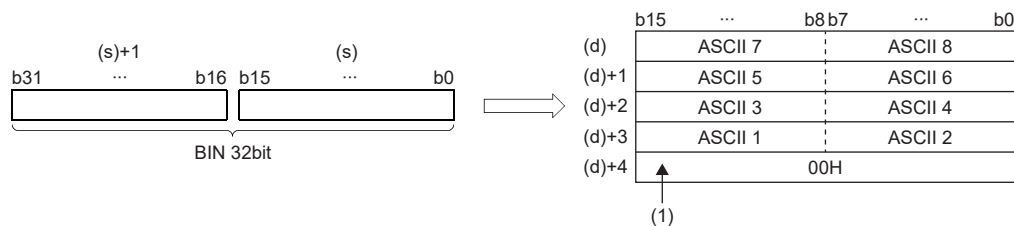
(s)中數值儲存BIN32位元資料“000ABCDEH”的狀態，並執行DBINHA(P)指令的情況下



- SM705(轉換位數切換)為OFF的情況下為位數固定。“000ABCDE”的8位被轉換為ASCII資料儲存。
- SM705(轉換位數切換)為ON的情況下為向前填充對齊。僅“000ABCDE”的有效位數部分“ABCDE”被轉換為ASCII資料儲存，在終端儲存00H。

■SM705 (轉換位數切換) 為OFF的情況下的動作

(d) ~ (d)+3中，16進制ASCII資料以固定位數(8位)儲存。



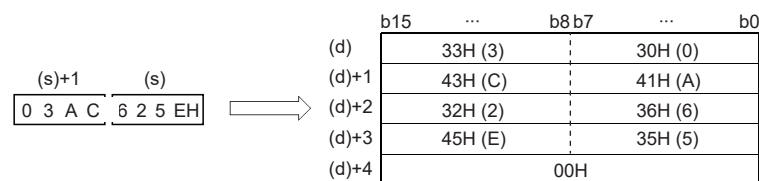
ASCII □: 第□位的ASCII碼

(1): (d)+4中，SM701(輸出字元數切換)為OFF時儲存00H，ON時不變化。

- (d)中儲存的運算結果將被處理為8位的16進制數。因此，有效位數的左側的0將被作為“0”處理。(進行0填充。)

例

在(s)中指定03AC625EH的情況下



■SM705 (轉換位數切換) 為ON的情況下的動作

(d)中，省略有效位數的左側的0後的位數(最大8位)的16進制ASCII資料以向前填充對齊儲存。

(s)的值與(d)中儲存的值的示例如下所示。

(s) 的值	(d) ~ (d)+3 的內容	(s) 的值	(d) ~ (d)+3 的內容
• 0H~FH	<ul style="list-style-type: none"> • (d) 的高位字節中儲存00H。 • (d)+1及其以後不變化。 	• 10H~FFH	<ul style="list-style-type: none"> • (d)+1中儲存00H。 • (d)+2及其以後不變化。
• 1000000H~FFFFFFFH	<ul style="list-style-type: none"> • (d)+3的高位字節中儲存00H。 	• 10000000H~FFFFFFFH	<ul style="list-style-type: none"> • (1): SM701(輸出字元數切換)為OFF時(d)+4中儲存00H。SM701(輸出字元數切換)為ON時(d)+4不變化。

ASCII □: 第□位的ASCII碼

- 位數未滿最大位數(8位)的情況下，無論是SM701(輸出字元數切換)的ON/OFF，字元串的終端將儲存00H。字元串的終端為低位字節的情況下，高位字節中也將儲存00H。
- 位數為最大位數(8位)的情況下，SM701(輸出字元數切換)為OFF時，(d)+4中將儲存00H。SM701(輸出字元數切換)為ON時(d)+4不變化。

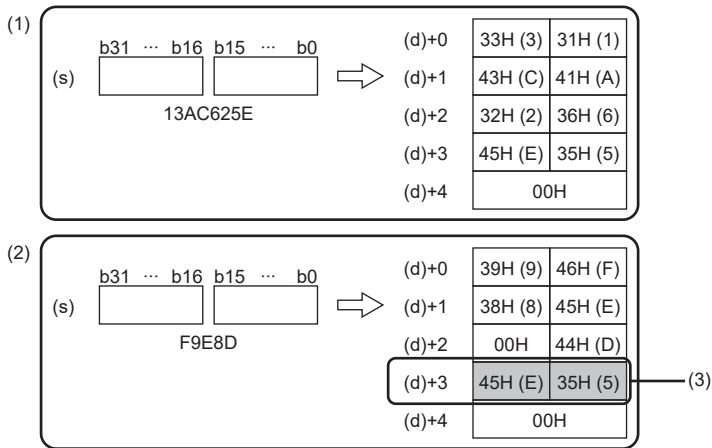
出錯

沒有運算出錯。

注意事項

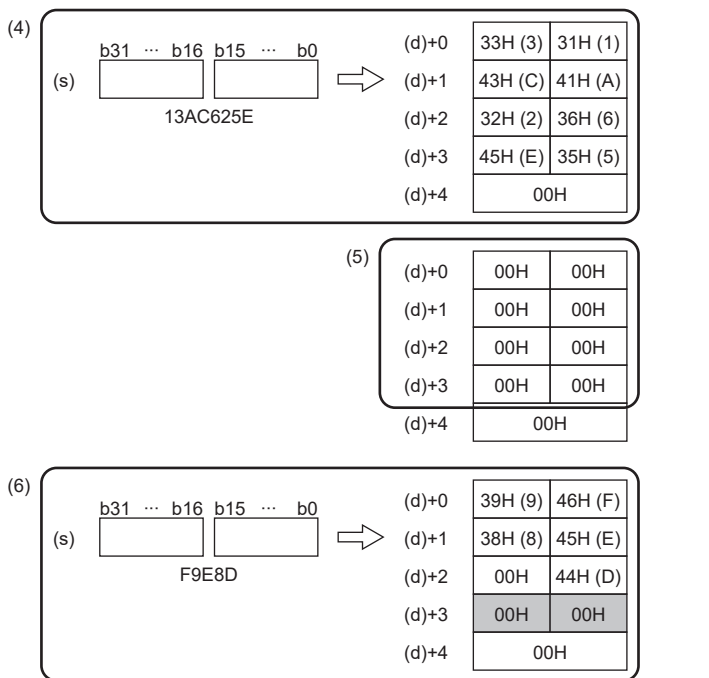
SM705(轉換位數切換)為ON時，(d)中運算結果將只儲存有效位數部分。因此，連續執行DBINHA(P)指令，運算結果儲存在同一元件的情況下，(d)中的一部分中有可能上次的運算結果不被覆蓋儲存而殘留。

[例] (s)為“13AC625E”時執行DBINHA(P)指令，且(s)為“F9E8D”時執行DBINHA(P)指令的示例



- (1) 將“13AC625E”轉換為字元串
- (2) 將“F9E8D”轉換為字元串
- (3) (d)+3中留有上次轉換結果的一部分

想避免此狀況，應建立程式使儲存區域(d)+0~(d)+3全部清除後執行DBINHA(P)指令。



- (4) 將“13AC625E”轉換為字元串
- (5) 將(d)+0~(d)+3清除
- (6) 將“F9E8D”轉換為字元串

BIN16位元資料→字元串轉換

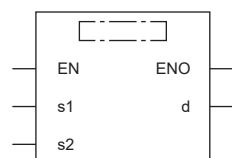
STR(P) (_U)



將BIN16位元資料在指定的位置附加小數點後轉換為字元串。

梯形圖	ST	
	ENO:=STR(EN, s1, s2, d); ENO:=STRP(EN, s1, s2, d);	ENO:=STR_U(EN, s1, s2, d); ENO:=STRP_U(EN, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
STR STR_U	
STRP STRP_U	

設置數據

■內容、範圍、資料類型

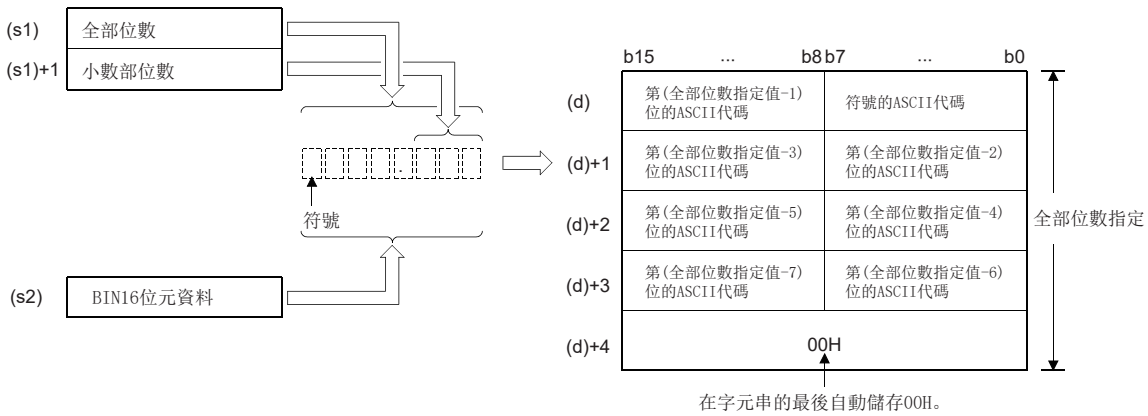
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	STR(P) STR(P)_U	儲存了轉換數值的位數的起始元件	帶符號BIN16位元 無符號BIN16位元	ANY16_S_ARRAY (要素數: 2) ANY16_U_ARRAY (要素數: 2)
(s2)	STR(P) STR(P)_U	轉換的BIN資料	-32768~32767 0~65535	帶符號BIN16位元 無符號BIN16位元
(d)		儲存轉換後的字元串的起始元件	字元串	ANYSTRING_SINGLE
EN		執行條件	位元	BOOL
ENO		執行結果	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(s1)	○	○	○	○	○	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

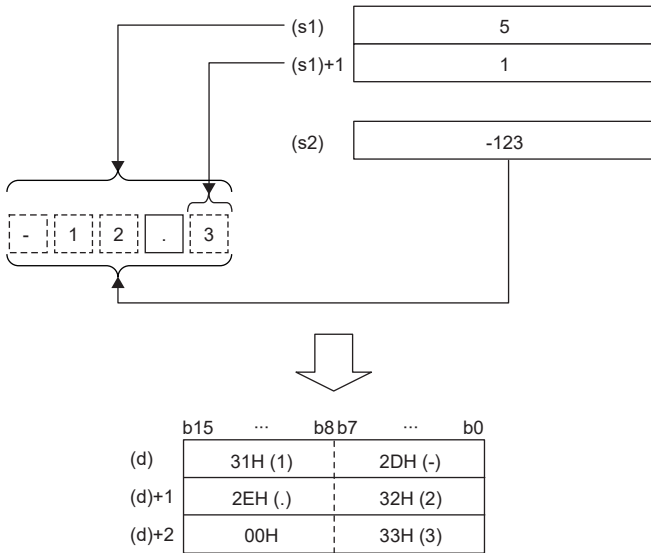
功能

- 將(s2)中指定的BIN16位元資料在(s1)中指定的位置處附加小數點後轉換為字元串，儲存到(d)中指定的元件編號及其以後。



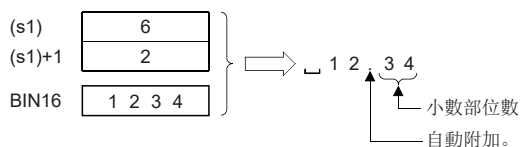
例

將(s2)的資料“-123”，作為小數部分位數為1位的值“-12.3”轉換為字元串的情況

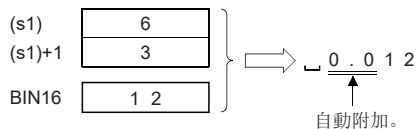


- (s1)中可指定的全部位數為2~8位。
- (s1)+1中可指定的小數部位數為0~5位。但是，設定時應滿足小數部位數≤(全部位數-3)的條件。

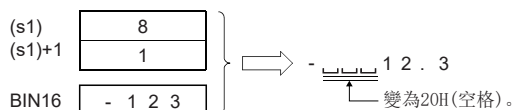
- 轉換後的字元串資料將按下述方式被儲存到(d)以後的元件編號中。
- 在符號中，BIN16位元資料為正時將儲存20H(空格)、為負時將儲存2DH(-)。
- 將小數位數設定為0以外的情況下，第指定位數+1位中將自動儲存2EH(.)。當小數位數為0時，不儲存2EH(.)。



- 在小數位數的值大於BIN16位元資料的位數的情況下，將自動附加0，向右對齊轉換為“0.□□□□”。



- 在全部位數的值中除去符號、小數點後的位數大於BIN16位元資料的位數的情況下，將於符號與數值之間儲存20H(空格)。在BIN16位元資料的位數較大的情況下，將變為出錯狀態。



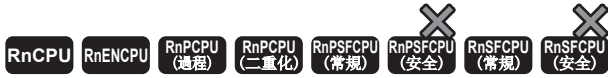
- 轉換後的字元串的最後將自動儲存00H。

出錯

出錯代碼 (SD0)	內容
3401H	<p>當 (s1) 中輸入不能轉換的不正確資料時。</p> <ul style="list-style-type: none"> 全部位數指定超出了2~8的範圍 (s1)+1的小數位數指定超出了0~5的範圍 (s1)中指定的全部位數與(s1)+1中指定的小數位數的指定值的關係不滿足以下公式 全部位數-3≥小數位數 (s1)中指定的位數小於(s2)中指定的BIN16位元資料的位數+2 [(s1)的位數]<[(s2)的不包含符號的BIN16位元資料的位數+符號(+或-)的位數+小數點(.)的位數]

BIN32位元資料→字元串轉換

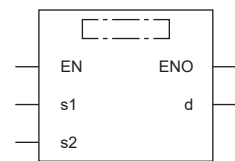
DSTR(P) (_U)



將BIN32位元資料在指定的位置處附加小數點後轉換為字元串。

梯形圖	ST	
	ENO:=DSTR(EN, s1, s2, d); ENO:=DSTRP(EN, s1, s2, d);	ENO:=DSTR_U(EN, s1, s2, d); ENO:=DSTRP_U(EN, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
DSTR DSTR_U	
DSTRP DSTRP_U	

設置數據

■內容、範圍、資料類型

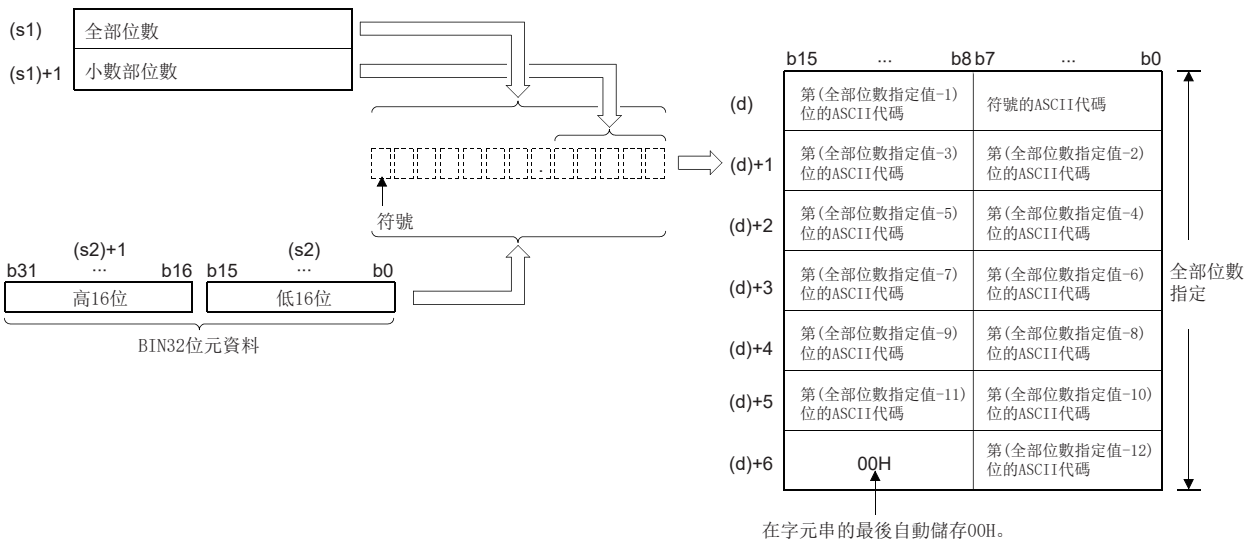
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	DSTR(P) DSTR(P)_U	儲存了轉換數值的位數的起始元件	帶符號BIN16位元 無符號BIN16位元	ANY16_S_ARRAY (要素數: 2) ANY16_U_ARRAY (要素數: 2)
(s2)	DSTR(P) DSTR(P)_U	轉換的BIN資料	-2147483648~2147483647 0~4294967295	帶符號BIN32位元 無符號BIN32位元
(d)		儲存轉換後的字元串的起始元件	字元串	ANYSTRING_SINGLE
EN		執行條件	位元	BOOL
ENO		執行結果	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(s1)	○	○	○	○	○	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

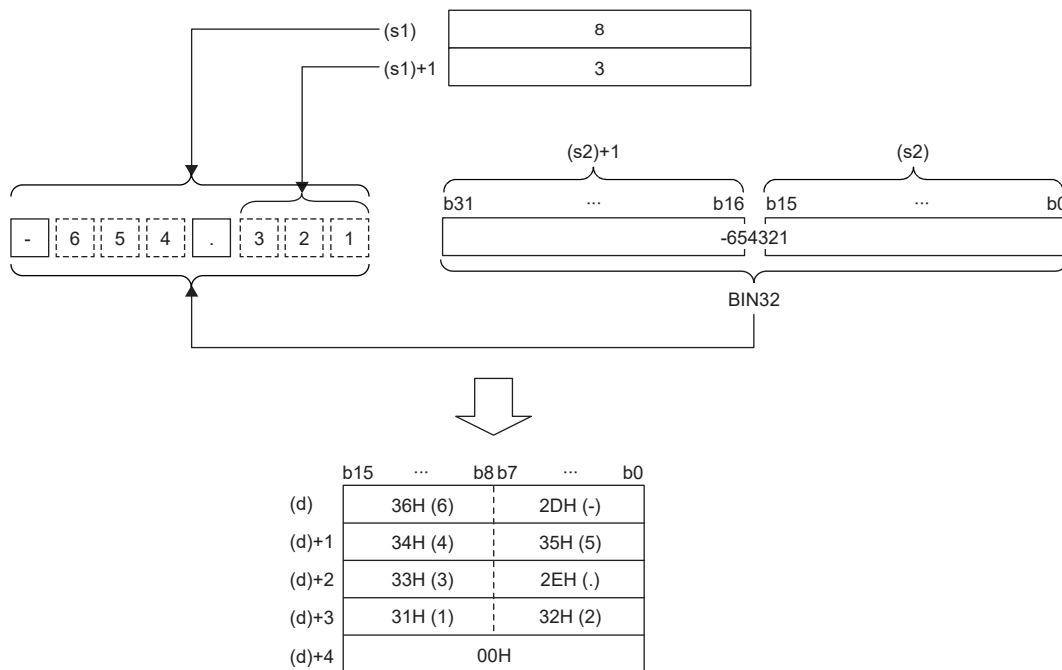
功能

- 將 (s2) 中指定的BIN32位元資料在 (s1) 中指定的位置附加小數點後轉換為字元串，儲存到 (d) 中指定的元件編號及其以後。



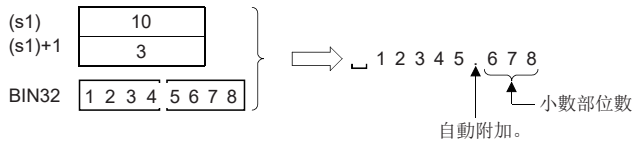
例

將 (s2) 的資料 “-654321”，作為小數部分位數為3位的值 “-654.321” 轉換為字元串的情況

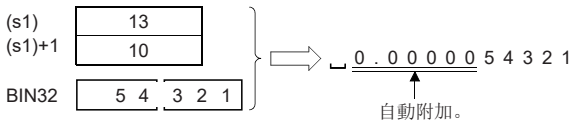


- (s1) 中可指定的全部位數為2~13位。
- (s1)+1 中可指定的小數部位數為0~10位。但是，設定時應滿足小數部位數≤(全部位數-3)的條件。

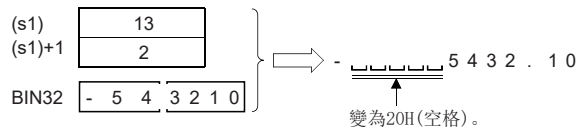
- 轉換後的字元串資料將按下述方式被儲存到(d)以後的元件編號中。
- 在符號中，BIN32位元資料為正時將儲存20H(空格)、為負時將儲存2DH(-)。
- 將小數位數設定為0以外的情況下，第指定位數+1位中將自動儲存2EH(.)。當小數位數為0時，不儲存2EH(.)。



- 在小數位數的值大於BIN32位元資料的位數的情況下將自動附加0，向右對齊轉換為“0.□□□□”。



- 在全部位數的值中除去符號、小數點後的位數大於BIN32位元資料的位數的情況下，在符號與數值之間將儲存20H(空格)。在BIN32位元資料的位數較大的情況下，將變為出錯狀態。



- 轉換後的字元串的最後將自動儲存00H。

出錯

出錯代碼(SD0)	內容
3401H	當(s1)中輸入不能轉換的不正確資料時。 <ul style="list-style-type: none"> 全部位數指定超出了2~13的範圍 (s1)+1的小數位數指定超出了0~10的範圍 (s1)中指定的全部位數與(s1)+1中指定的小數位數的指定值的關係不滿足以下公式 全部位數-3≥小數位數 (s1)中指定的位數小於(s2)中指定的BIN32位元資料的位數+2 [(s1)的位數]<[(s2)不包含符號的BIN32位元資料的位數+符號分(+或-)的位數+小數點(.)的位數]

BCD4位資料→10進制ASCII碼轉換

BCDDA (P)

RnCPU RnENCPU RnPCPU (通程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將BCD4位資料轉換為ASCII碼。

梯形圖	ST
	ENO:=BCDDA (EN, s, d) ; ENO:=BCDDAP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
BCDDA	
BCDDAP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行ASCII轉換的BCD資料	0~9999	BCD4位	ANY16
(d)	儲存轉換結果的起始元件	—	字元串	ANYSTRING_SINGLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將(s)中指定的BCD4位資料的各位的數值轉換為ASCII碼後，儲存到(d)中指定的元件編號及其以後。
- 根據SM705(轉換位數切換)的狀態，(d)中設定的16進制ASCII資料的形式有所不同。

SM705*1的狀態	(d)的儲存形式	參閱目標
OFF	以固定位數(4位)儲存	799頁 SM705(轉換位數切換)為OFF的情況下的動作
ON	根據(s)的值各位向前填充對齊並儲存	799頁 SM705(轉換位數切換)為ON的情況下的動作

*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

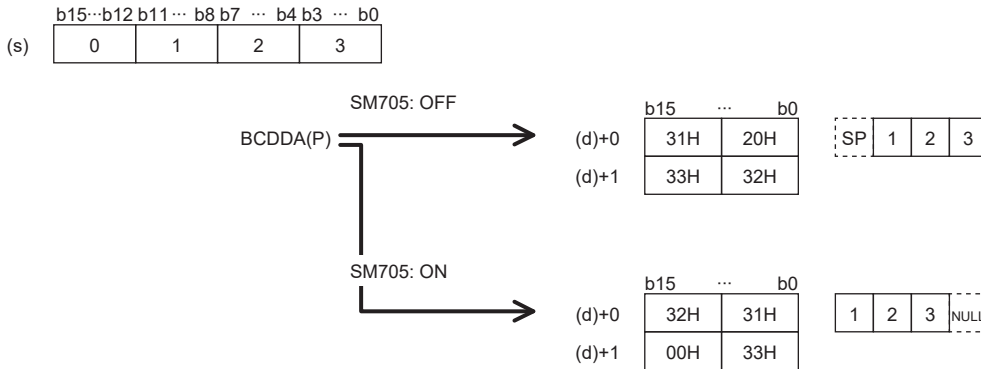
SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■動作的概要

SM705(轉換位數切換)為OFF的情況下與ON的情況下，其各自的動作如下所示。

例

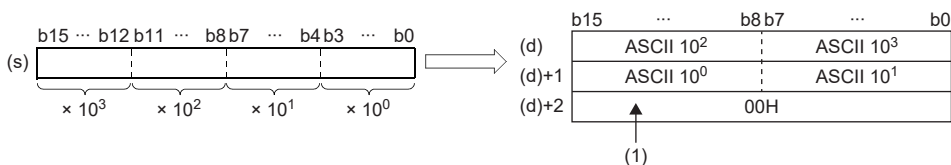
(s)中儲存BCD4位資料“0123”的狀態，並執行BCDDA(P)指令的情況下



- SM705(轉換位數切換)為OFF的情況下為位數固定。“0123”的起始0被轉換為20H(空格)並儲存。
- SM705(轉換位數切換)為ON的情況下為向前填充對齊。省略“0123”起始的0後的“123”被轉換為ASCII資料儲存，在終端儲存00H。

■SM705 (轉換位數切換) 為OFF的情況下的動作

(d) ~ (d)+1中，10進制ASCII資料以固定位數(4位)儲存。



ASCII 10³: 千位的ASCII碼

ASCII 10²: 百位的ASCII碼

ASCII 10¹: 十位的ASCII碼

ASCII 10⁰: 個位的ASCII碼

(1): (d)+2中，SM701 (輸出字元數切換) 為OFF時儲存00H，ON時不變化。

- 在(d)中儲存的運算結果中，有效位數的左側的0處將儲存20H(空格)。(進行0抑制。)例如“0050”的情況下，“00”將分別變為20H(空格)，“50”將成為有效位數。

例

在(s)中指定9105的情況下



■SM705 (轉換位數切換) 為ON的情況下的動作

(d)中，儲存省略了有效位數的左側的0後的位數(最大4位)的10進制ASCII資料。

(s)的值與(d)中儲存的值的示例如下所示。

(s) 的值	(d) ~ (d)+1 的內容	(s) 的值	(d) ~ (d)+1 的內容
• 0H~9H	<ul style="list-style-type: none"> (d) 的高位字節中儲存00H。 (d)+1及其以後不變化。 	• 10H~99H	<ul style="list-style-type: none"> (d)+1中儲存00H。
• 100H~999H	<ul style="list-style-type: none"> (d)+1的高位字節中儲存00H。 	• 1000H~9999H	<ul style="list-style-type: none"> (1): SM701 (輸出字元數切換) 為OFF時 (d)+2中儲存00H。SM701 (輸出字元數切換) 為ON時 (d)+2不變化。

ASCII 10³: 千位的ASCII碼

ASCII 10²: 百位的ASCII碼

ASCII 10¹: 十位的ASCII碼

ASCII 10⁰: 個位的ASCII碼

- 位數未滿最大位數(4位)的情況下，無論是SM701 (輸出字元數切換) 的ON/OFF，字元串的終端將儲存00H。字元串的終端為低位字節的情況下，高位字節中也將儲存00H。
- 位數為最大位數(4位)的情況下，SM701 (輸出字元數切換) 為OFF時，(d)+2中將儲存00H。SM701 (輸出字元數切換) 為ON時 (d)+2不變化。

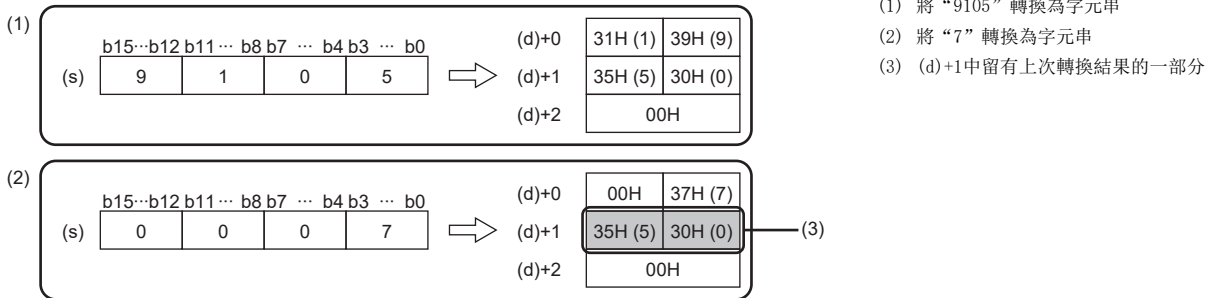
出錯

出錯代碼 (SD0)	內容
3401H	當 (s) 的資料超出了 0~9999 的範圍時。

注意事項

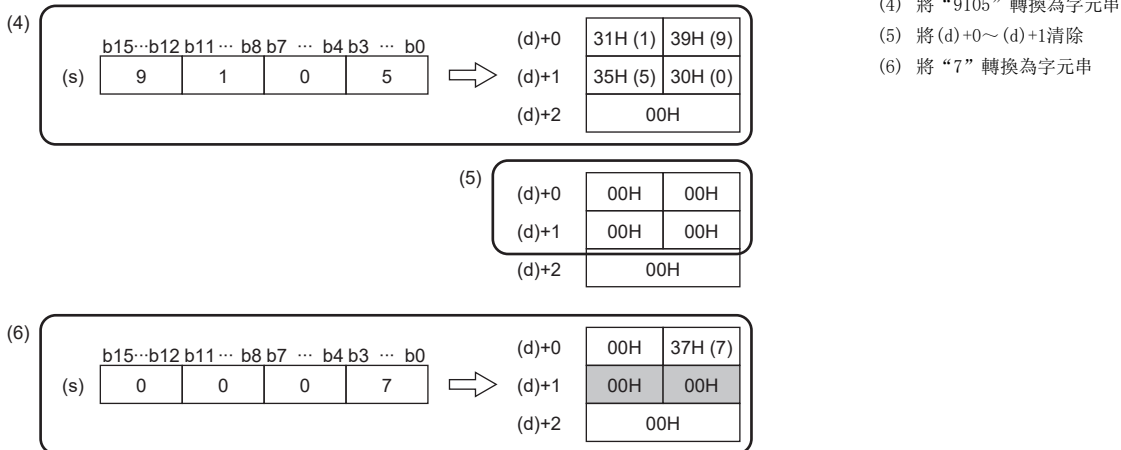
SM705 (轉換位數切換) 為 ON 時，(d) 中運算結果將只儲存有效位數部分。因此，連續執行 BCDDA (P) 指令，運算結果儲存在同一元件的情況下，(d) 中的一部分中有可能上次的運算結果不被覆蓋儲存而殘留。

[例] (s) 為 “9105H” 時執行 BCDDA (P) 指令，且 (s) 為 “0007H” 時執行 BCDDA (P) 指令的示例



- (1) 將 “9105” 轉換為字元串
- (2) 將 “7” 轉換為字元串
- (3) (d)+1 中留有上次轉換結果的一部分

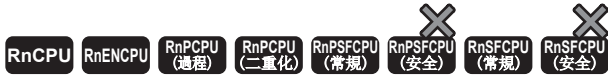
想避免此狀況，應建立程式使儲存區域 (d)+0~(d)+1 全部清除後執行 BCDDA (P) 指令。



- (4) 將 “9105” 轉換為字元串
- (5) 將 (d)+0~(d)+1 清除
- (6) 將 “7” 轉換為字元串

BCD8位資料→10進制ASCII碼轉換

DBCDDA (P)



將BCD8位資料轉換為ASCII碼。

梯形圖	ST
	ENO:=DBCDDA (EN, s, d) ; ENO:=DBCDDAP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
DBCDDA	
DBCDDAP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	進行ASCII轉換的BCD資料	0~99999999	BCD8位	ANY32
(d)	儲存轉換結果的起始元件	—	字元串	ANYSTRING_SINGLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將(s)中指定的BCD8位資料的各位的數值轉換為ASCII碼後，儲存到(d)中指定的元件編號及其以後。
- 根據SM705(轉換位數切換)的狀態，(d)中設定的10進制ASCII資料的形式有所不同。

SM705*1的狀態	(d)的儲存形式	參閱目標
OFF	以固定位數(8位)儲存	803頁 SM705(轉換位數切換)為OFF的情況下的動作
ON	根據(s)的值各位向前填充對齊並儲存	804頁 SM705(轉換位數切換)為ON的情況下的動作

*1 關於對應SM705的CPU模組的韌體版本，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

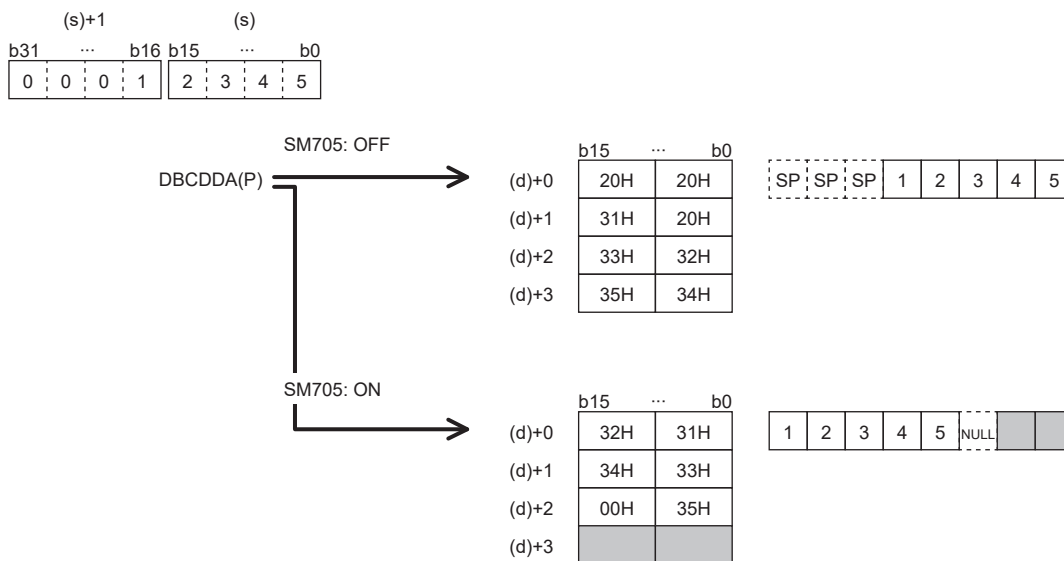
SM705未對應的CPU模組中，SM705設定為ON的情況和SM705為OFF的情況為相同的動作。

■動作的概要

SM705(轉換位數切換)為OFF的情況下與ON的情況下，其各自的動作如下所示。

例

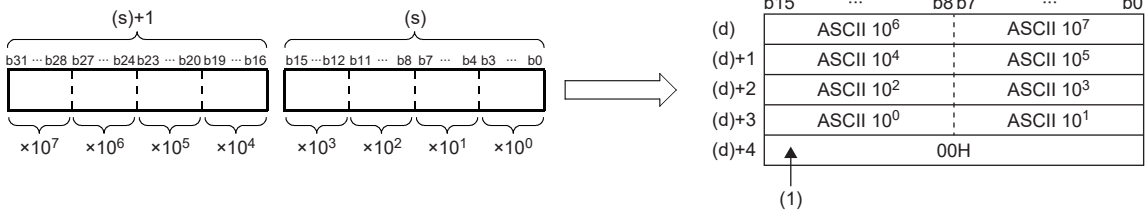
(s)中儲存BCD8位資料“00012345”的狀態，並執行DBCDDA(P)指令的情況下



- SM705(轉換位數切換)為OFF的情況下為位數固定。“00012345”的0被轉換為20H(空格)並儲存。
- SM705(轉換位數切換)為ON的情況下為向前填充對齊。省略“00012345”起始的0後的“12345”被轉換為ASCII資料儲存，在終端儲存00H。

■SM705 (轉換位數切換) 為OFF的情況下的動作

(d) ~ (d)+3中, 10進制ASCII資料以固定位數(8位)儲存。



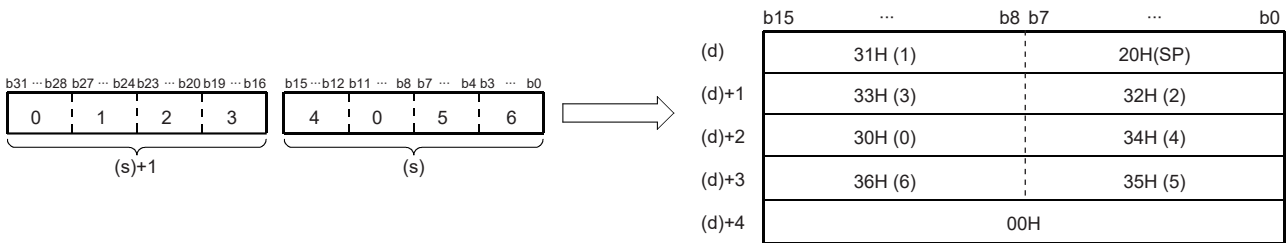
ASCII 10⁷: 千萬位的ASCII碼
 ASCII 10⁶: 百萬位的ASCII碼
 ⋮
 ASCII 10¹: 十位的ASCII碼
 ASCII 10⁰: 個位的ASCII碼

(1): (d)+4中, SM701 (輸出字元數切換) 為OFF時儲存00H, ON時不變化。

- 在(d)中儲存的運算結果中, 有效位數的左側的0處將儲存20H(空格)。(進行0抑制。)例如“00012098”的情況下, “000”將分別變為20H(空格), “12098”將成為有效位數。

例

在(s)中指定了01234056的情況下



■SM705 (轉換位數切換) 為ON的情況下的動作

(d) 中，儲存省略了有效位數的左側的0後的位數(最大8位)的10進制ASCII資料。

(s) 的值與 (d) 中儲存的值的示例如下所示。

(s) 的值	(d) ~ (d)+3 的內容	(s) 的值	(d) ~ (d)+3 的內容																																																																														
<ul style="list-style-type: none"> 0H~9H 	<ul style="list-style-type: none"> (d) 的高位字節中儲存00H。 (d)+1及其以後不變化。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(d)</td> <td>00H</td> <td>...</td> <td>ASCII 10⁰</td> <td>...</td> <td></td> </tr> <tr> <td>(d)+1</td> <td colspan="5" style="background-color: #cccccc;"></td> </tr> <tr> <td>(d)+2</td> <td colspan="5" style="background-color: #cccccc;"></td> </tr> <tr> <td>(d)+3</td> <td colspan="5" style="background-color: #cccccc;"></td> </tr> </table>		b15	...	b8 b7	...	b0	(d)	00H	...	ASCII 10 ⁰	...		(d)+1						(d)+2						(d)+3						<ul style="list-style-type: none"> 10H~99H 	<ul style="list-style-type: none"> (d)+1中儲存00H。 (d)+2及其以後不變化。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(d)</td> <td>ASCII 10⁰</td> <td>...</td> <td>ASCII 10¹</td> <td>...</td> <td></td> </tr> <tr> <td>(d)+1</td> <td colspan="5" style="background-color: #cccccc;"></td> </tr> <tr> <td>(d)+2</td> <td colspan="5" style="background-color: #cccccc;"></td> </tr> <tr> <td>(d)+3</td> <td colspan="5" style="background-color: #cccccc;"></td> </tr> </table>		b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁰	...	ASCII 10 ¹	...		(d)+1						(d)+2						(d)+3																							
	b15	...	b8 b7	...	b0																																																																												
(d)	00H	...	ASCII 10 ⁰	...																																																																													
(d)+1																																																																																	
(d)+2																																																																																	
(d)+3																																																																																	
	b15	...	b8 b7	...	b0																																																																												
(d)	ASCII 10 ⁰	...	ASCII 10 ¹	...																																																																													
(d)+1																																																																																	
(d)+2																																																																																	
(d)+3																																																																																	
⋮																																																																																	
<ul style="list-style-type: none"> 1000000H~9999999H 	<ul style="list-style-type: none"> (d)+3的高位字節中儲存00H。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(d)</td> <td>ASCII 10⁵</td> <td>...</td> <td>ASCII 10⁶</td> <td>...</td> <td></td> </tr> <tr> <td>(d)+1</td> <td>ASCII 10³</td> <td>...</td> <td>ASCII 10⁴</td> <td>...</td> <td></td> </tr> <tr> <td>(d)+2</td> <td>ASCII 10¹</td> <td>...</td> <td>ASCII 10²</td> <td>...</td> <td></td> </tr> <tr> <td>(d)+3</td> <td>00H</td> <td>...</td> <td>ASCII 10⁰</td> <td>...</td> <td></td> </tr> </table>		b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁵	...	ASCII 10 ⁶	...		(d)+1	ASCII 10 ³	...	ASCII 10 ⁴	...		(d)+2	ASCII 10 ¹	...	ASCII 10 ²	...		(d)+3	00H	...	ASCII 10 ⁰	...		<ul style="list-style-type: none"> 10000000H~99999999H 	<ul style="list-style-type: none"> (1): SM701 (輸出字元數切換) 為OFF時 (d)+4中儲存00H。SM701 (輸出字元數切換) 為ON時 (d)+4不變化。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(d)</td> <td>ASCII 10⁶</td> <td>...</td> <td>ASCII 10⁷</td> <td>...</td> <td></td> </tr> <tr> <td>(d)+1</td> <td>ASCII 10⁴</td> <td>...</td> <td>ASCII 10⁵</td> <td>...</td> <td></td> </tr> <tr> <td>(d)+2</td> <td>ASCII 10²</td> <td>...</td> <td>ASCII 10³</td> <td>...</td> <td></td> </tr> <tr> <td>(d)+3</td> <td>ASCII 10⁰</td> <td>...</td> <td>ASCII 10¹</td> <td>...</td> <td></td> </tr> <tr> <td>(d)+4</td> <td colspan="5" style="background-color: #cccccc;"></td> </tr> <tr> <td></td> <td>↑</td> <td colspan="4"></td> </tr> <tr> <td></td> <td>(1)</td> <td colspan="4"></td> </tr> </table>		b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁶	...	ASCII 10 ⁷	...		(d)+1	ASCII 10 ⁴	...	ASCII 10 ⁵	...		(d)+2	ASCII 10 ²	...	ASCII 10 ³	...		(d)+3	ASCII 10 ⁰	...	ASCII 10 ¹	...		(d)+4							↑						(1)				
	b15	...	b8 b7	...	b0																																																																												
(d)	ASCII 10 ⁵	...	ASCII 10 ⁶	...																																																																													
(d)+1	ASCII 10 ³	...	ASCII 10 ⁴	...																																																																													
(d)+2	ASCII 10 ¹	...	ASCII 10 ²	...																																																																													
(d)+3	00H	...	ASCII 10 ⁰	...																																																																													
	b15	...	b8 b7	...	b0																																																																												
(d)	ASCII 10 ⁶	...	ASCII 10 ⁷	...																																																																													
(d)+1	ASCII 10 ⁴	...	ASCII 10 ⁵	...																																																																													
(d)+2	ASCII 10 ²	...	ASCII 10 ³	...																																																																													
(d)+3	ASCII 10 ⁰	...	ASCII 10 ¹	...																																																																													
(d)+4																																																																																	
	↑																																																																																
	(1)																																																																																

ASCII 10⁷: 千萬位的ASCII碼

ASCII 10⁶: 百萬位的ASCII碼

⋮

ASCII 10¹: 十位的ASCII碼

ASCII 10⁰: 個位的ASCII碼

- 位數未滿最大位數(8位)的情況下，無論是SM701(輸出字元數切換)的ON/OFF，字元串的終端將儲存00H。字元串的終端為低位字節的情況下，高位字節中也將儲存00H。
- 位數為最大位數(8位)的情況下，SM701(輸出字元數切換)為OFF時，(d)+4中將儲存00H。SM701(輸出字元數切換)為ON時(d)+4不變化。

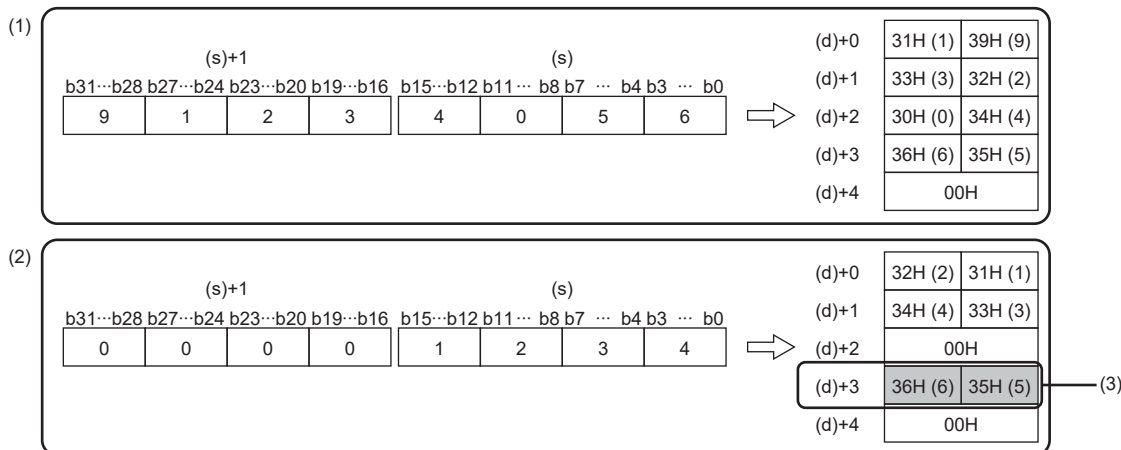
出錯

出錯代碼 (SD0)	內容
3401H	當 (s) 的資料超出了 0~99999999 的範圍時。

注意事項

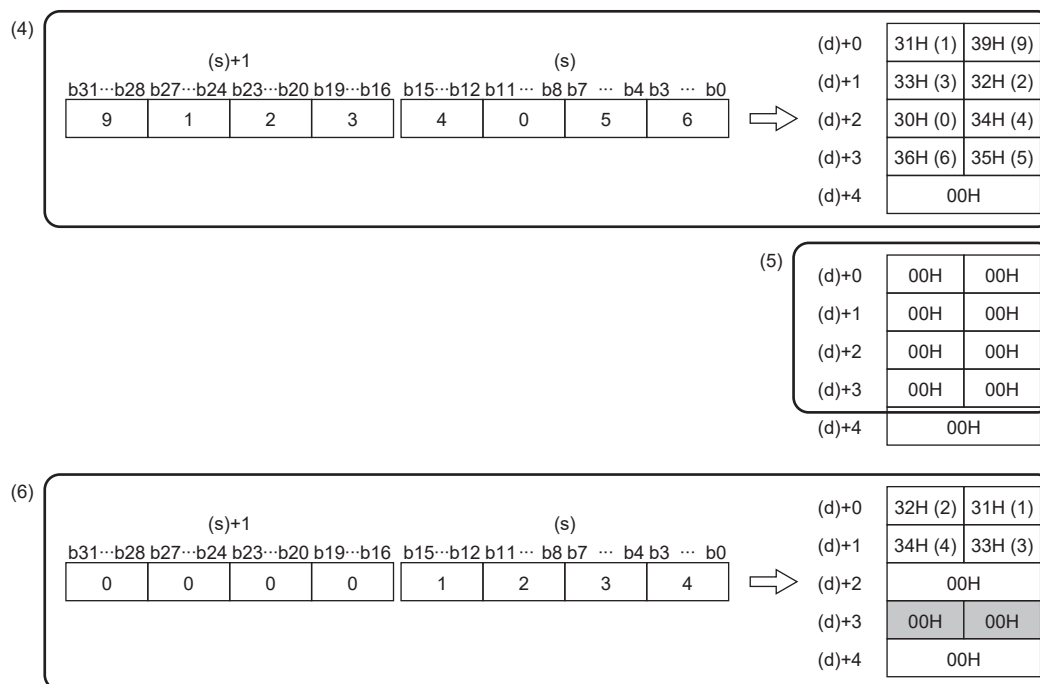
SM705(轉換位數切換)為ON時，(d)中運算結果將只儲存有效位數部分。因此，連續執行DBCDDA(P)指令，運算結果儲存在同一元件的情況下，(d)中的一部分中有可能上次的運算結果不被覆蓋儲存而殘留。

[例] (s)為“91234056H”時執行DBCDDA(P)指令，且(s)為“00001234H”時執行DBCDDA(P)指令的示例



- (1) 將“91234056”轉換為字元串
- (2) 將“1234”轉換為字元串
- (3) (d)+3中留有上次轉換結果的一部分

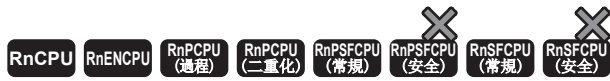
想避免此狀況，應建立程式使儲存區域(d)+0~(d)+3全部清除後執行DBCDDA(P)指令。



- (4) 將“91234056”轉換為字元串
- (5) 將(d)+0~(d)+3清除
- (6) 將“1234”轉換為字元串

單精度實數→字元串轉換

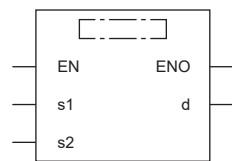
ESTR(P)



將單精度實數資料按照顯示指定轉換為字元串。

梯形圖	ST
	ENO:=ESTR (EN, s1, s2, d) ; ENO:=ESTRP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
ESTR	
ESTRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	轉換的單精度實數或儲存了資料的起始元件	$0, 2^{-126} < (s1) < 2^{128}$	單精度實數	ANYREAL_32
(s2)	儲存了轉換數值的顯示指定的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
(d)	儲存轉換後的字元串的起始元件	—	字元串	ANYSTRING_SINGLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

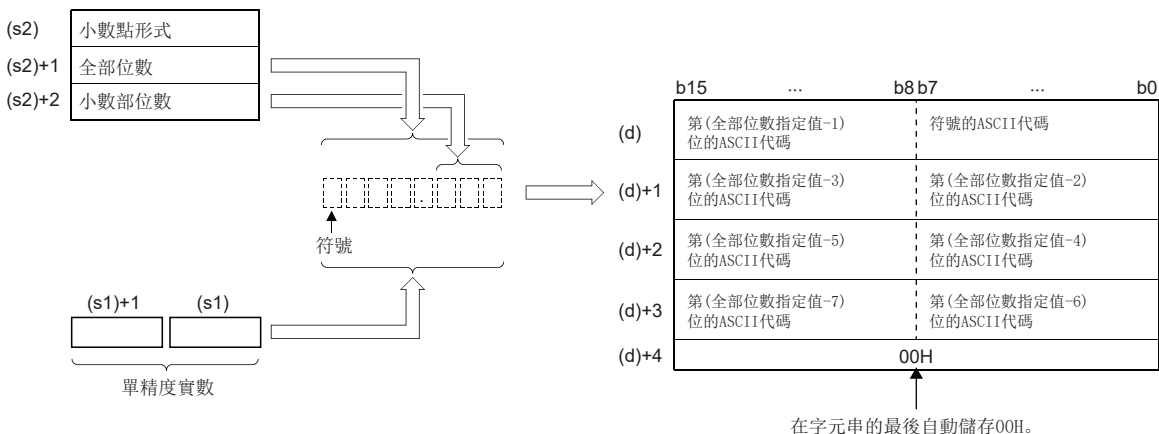
功能

- 將 (s1) 中指定的元件中儲存的單精度實數資料，按照 (s2) 中指定的元件編號以後儲存的顯示指定轉換為字元串後，儲存到 (d) 中指定的元件編號及其以後。
- 根據 (s2) 中指定的顯示指定轉換後的資料有所不同。

(s2)	0: 小數點形式 1: 指數形式
(s2)+1	全部位數
(s2)+2	小數部位數

■小數點形式

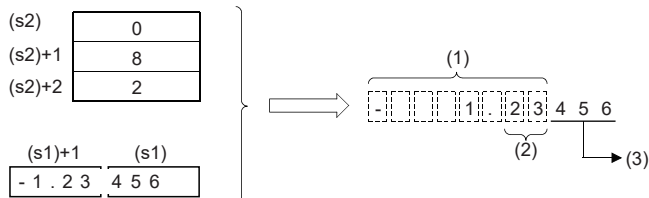
- 在 (s2) 中指定 0 的情況下，將變為小數點形式。



- 對於 (s2)+1 中可指定的全部位數，小數部位數為 0 時，變為全部位數 (最大 24) \geq 整數部位數^{*1}+1。0 以外時變為全部位數 (最大 24) \geq (整數部位數^{*1}+小數部位數+2)。

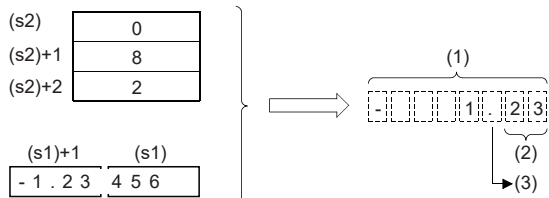
*1 (s1) 中指定的 32 位元浮點型實數資料的整數部位數如下所示。

- (s2)+2 中可指定的小數部位數為 0~7 位。但是，設定時應滿足小數部位數 \leq “全部位數-3” 的條件。
- 轉換後的字元串資料將按下述方式被儲存到 (d) 以後的元件編號中。
 - 在符號中，單精度實數資料為正時將儲存 20H (空格)、為負時將儲存 2DH (-)。
 - 在小數部位數的範圍無法容納單精度實數資料的小數部的的情況下，低位小數部將被四捨五入。



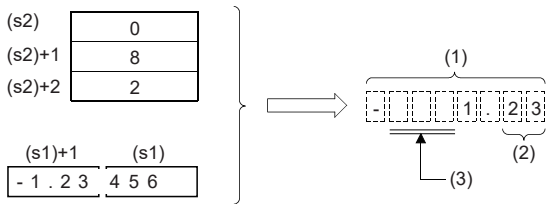
- (1): 全位數
- (2): 小數部位數
- (3): 進行四捨五入。

- 在將小數部位數設定為 0 以外的情況下，第指定小數部位數+1 位中將自動儲存 2EH (.)。當小數部位數為 0 時，不儲存 2EH (.)。



- (1): 全位數
- (2): 小數部位數
- (3): 自動附加。

- 在從全部位數中除去符號、小數點、小數部後的位數大於單精度實數資料的整數部的情況下，符號與整數部之間將儲存20H(空格)。

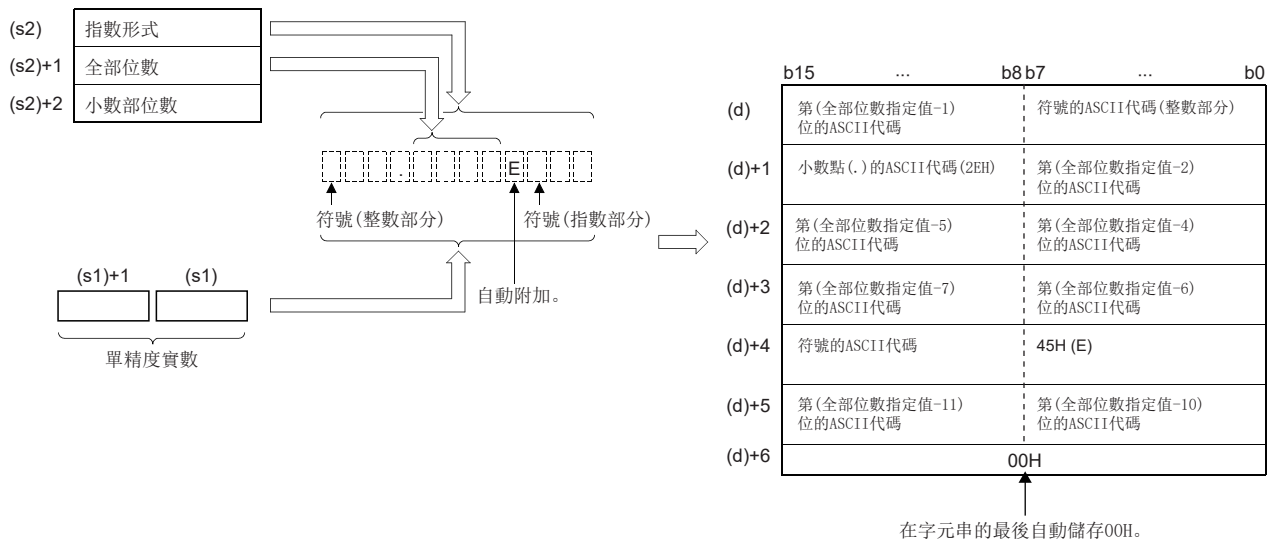


- (1): 全位數
- (2): 小數部位數
- (3): 變為20H(空格)。

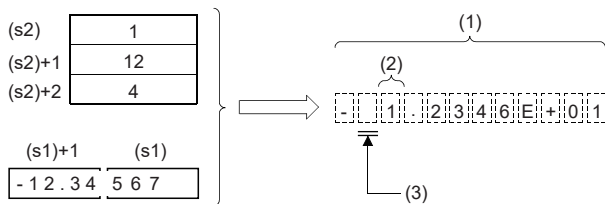
- 轉換後的字元串的最後將自動儲存00H。
- (s1)可指定的32位元浮點型實數資料的整數部位數為1~16位數。

■指數形式

- 在(s2)中指定了1的情況下，將變為指數形式。

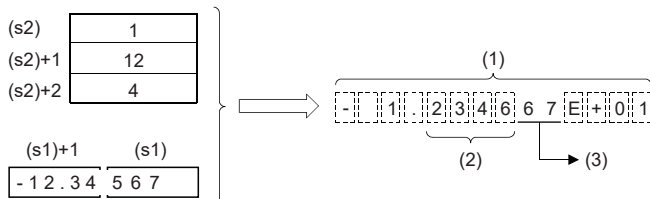


- 對於(s2)+1中可指定的全部位數，當小數部位數為0時，變為位數(最大24)≥2。0以外時變為位數(最大24)≥(小數部位數+7)。
- (s2)+2中可指定的小數部位數為0~7位。但是，設定時應滿足小數部位數≤(全部位數-7)的條件。
- 轉換後的字元串資料將按下述方式被儲存到(d)以後的元件編號中。
 - 整數部的符號中，單精度實數資料為正時將儲存20H(空格)、為負時將儲存2DH(-)。
 - 整數部固定為1位。在整數部與符號之間儲存20H(空格)。



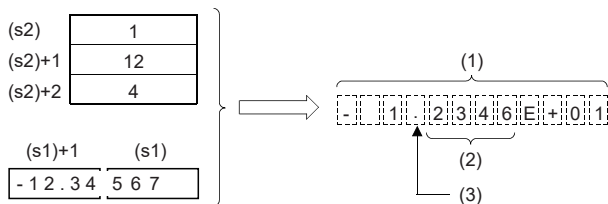
- (1): 全位數
- (2): 1位固定
- (3): 變為20H(空格)。

- 在小數部位數的範圍無法容納單精度實數資料的小數部的情況下，低位小數部將被四捨五入。



- (1): 全位數
- (2): 小數部位數
- (3): 被捨去。

- 在將小數部位數設定為0以外的情況下，第指定小數部位數+1位中將自動儲存2EH(.)。當小數部位數為0時，不儲存2EH(.)。

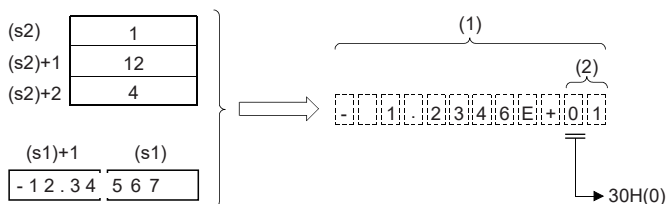


(1): 全位數

(2): 小數部位數

(3): 自動附加。

- 指數部的符號中，指數為正時將儲存2BH(+)、為負時將儲存2DH(-)。
- 指數部固定為2位。在指數部為1位的情況下，指數部的符號之間將儲存30H(0)。



(1): 全位數

(2): 2位固定

- 轉換後的字元串的最後將自動儲存00H。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3401H	<p>當 (s1) 超出下述範圍時。</p> $0, 2^{-126} \leq (s1) < 2^{128}$ <p>當 (s2) 中設定了不能轉換的不正確資料時。</p> <ul style="list-style-type: none"> 當 (s2) 中指定的形式指定為 0、1 以外時。 小數點形式中，當 (s2)+1 中指定的全部位數指定超出了下述範圍時。 小數部位數為 0 時：全部位數 \geq 整數部位數^{*1}+1 小數部位數為 0 以外時：全部位數 \geq 整數部位數^{*1}+小數點位數+2 指數形式中，當 (s2)+1 中指定的全部位數指定超出了下述範圍時。 小數部位數為 0 時：全部位數 \geq 6 小數部位數為 0 以外時：全部位數 \geq (小數部位數+7) 指數形式中，當 (s2)+2 中指定的小數部位數指定超出了下述範圍時。 小數點形式時：小數部位數 \leq (全部位數-3) 指數形式時：小數部位數 \leq (全部位數-7) 對於小數點形式，在 (s1) 中可指定的 32 位元浮點實數資料的整數部位數超過 16 位數的情況下。 (s2)+2 中指定的小數部位數超出了 0~7 位時。 <p>全部位數指定了超出 24 的值時。</p>
3402H	當 (s1) 中指定的元件的內容為 -0、非正規化數、非數、±∞ 時。

*1 (s1) 中指定的 32 位元浮點型實數資料的整數部位數如下所示。

16進制BIN資料→16進制ASCII碼轉換

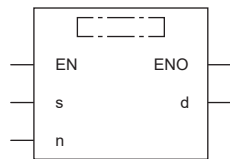
INT2ASC(P)



將BIN16位元資料轉換為16進制數ASCII碼後，以任意範圍儲存。

梯形圖	ST
	ENO:=INT2ASC (EN, s, n, d) ; ENO:=INT2ASCP (EN, s, n, d)

FBD/LD



■執行條件

指令	執行條件
INT2ASC	
INT2ASCP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	儲存了轉換為字元串的BIN資料的起始元件	—	帶符號BIN16位元	ANY16*1
(d)	儲存轉換後的字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(n)	儲存的字元數	0~16383	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

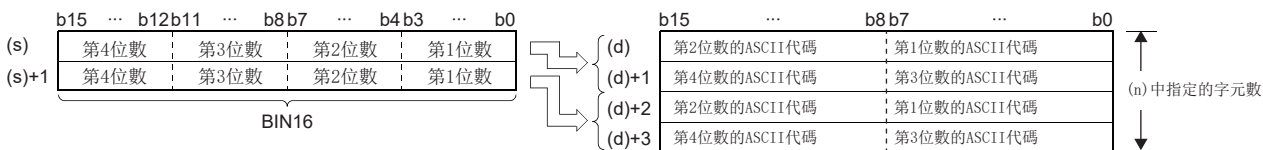
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

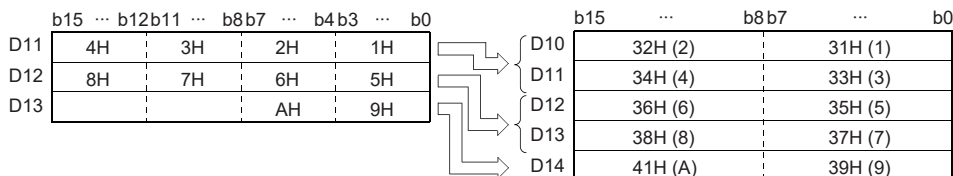
操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

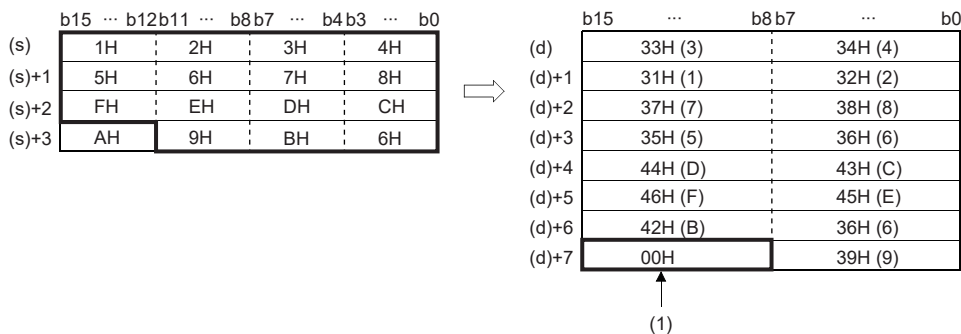
- 將 (s) 中指定的元件編號及其以後中儲存的BIN16位元資料，以16進制數處理轉換為ASCII轉換後，按照 (n) 中指定的字元數儲存到 (d) 中指定的元件編號及其以後。



- 根據 (n) 中設定的字元數，(s) 中指定的BIN資料的範圍及 (d) 中指定的字元串的儲存元件的範圍將被自動確定。
- 即使儲存了轉換的BIN資料的元件範圍與儲存轉換後的ASCII資料的元件範圍為重複的情況下，也將正常進行處理。



- 在 (n) 中指定的字元數為奇數的情況下，儲存字元串的元件範圍的最終元件編號的高位8位元中，將自動儲存00H。



(1): 自動儲存00H。

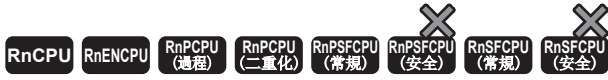
- 在 (n) 中指定的字元數為0的情況下，不進行轉換處理。

出錯

出錯代碼 (SD0)	內容
3405H	當 (n) 中輸入了超出允許指定範圍的資料時。 <ul style="list-style-type: none"> 指定的字元數超出了0~16383的範圍

Unicode字元串→移位JIS字元串轉換

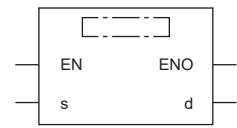
WS2SJIS (P)



將Unicode字元串轉換為移位JIS字元串。

梯形圖	ST
	<pre>ENO:=WS2SJIS (EN, s, d); ENO:=WS2SJISP (EN, s, d);</pre>

FBD/LD



■執行條件

指令	執行條件
WS2SJIS	
WS2SJISP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	儲存了轉換的字元串的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(d)	儲存轉換後的字元串的起始元件	—	字元串	ANYSTRING_SINGLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

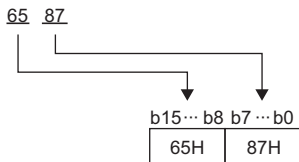
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將(s)中指定的Unicode字元串轉換為移位JIS字元串後，儲存到(d)中。
- 對於(s)的Unicode字元串，應以小端字節序進行指定。

例

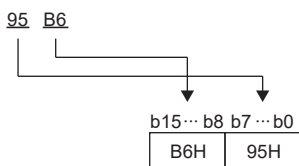
Unicode字元串的“6587H”以“6587H”進行指定。



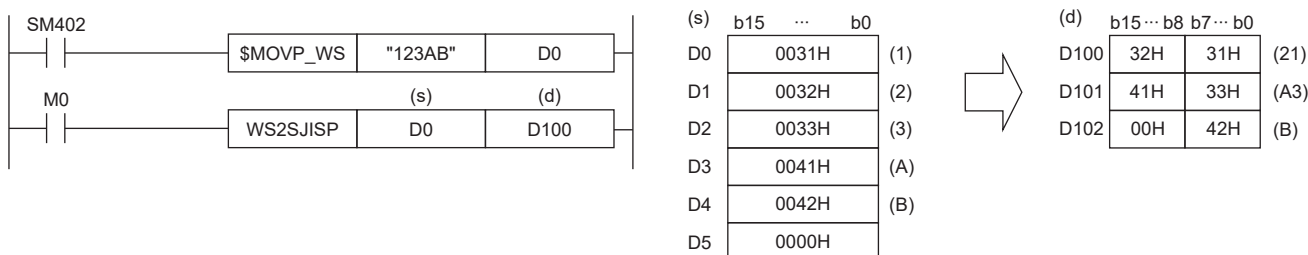
- (d)的移位JIS字元串以大端字節序儲存。

例

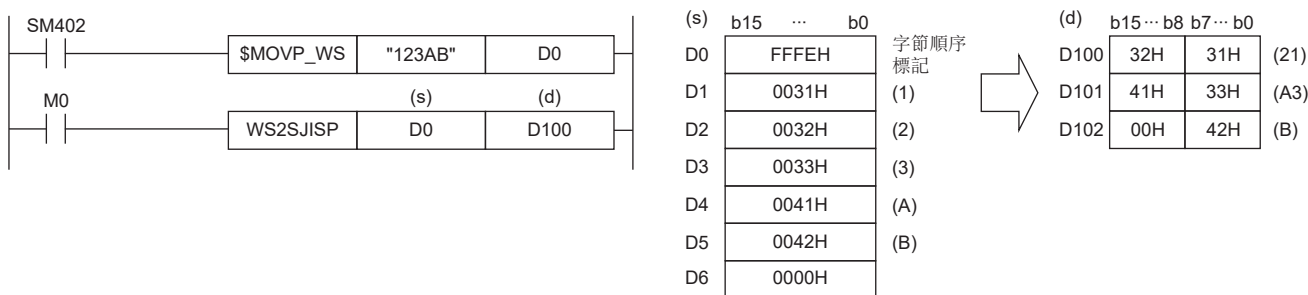
移位JIS字元串的“95B6H”以“B695H”進行指定。



- 在沒有字節順序標記情況下的Unicode→移位JIS轉換的動作如下所示。



- 在附加了字節順序標記(FE FFH)情況下的Unicode→移位JIS轉換的動作如下所示。



出錯

出錯代碼 (SD0)	內容
2821H	在(s)、(d)中指定的範圍重複的情況下。
3401H	在(s)中指定的字元串中，附加了字節順序標記FE FFH(大端字節序)的情況下。 在(s)中指定的範圍中，包含有不能轉換的字元代碼的情況下。
3405H	在(s)中指定的字元串超過16383字元的情況下。 ^{*1}

*1 對於以漢字等的移位JIS代碼表示時變為2字節資料的字元，應將每個字元以2進行計數。

移位JIS字元串→Unicode字元串轉換(無字節順序標記)

SJIS2WS(P)



將移位JIS字元串轉換為Unicode字元串。

梯形圖	ST
	<pre>ENO:=SJIS2WS(EN, s, d); ENO:=SJIS2WSP(EN, s, d);</pre>

FBD/LD

■執行條件

指令	執行條件
SJIS2WS	
SJIS2WSP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存了轉換的字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存轉換後的字元串的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

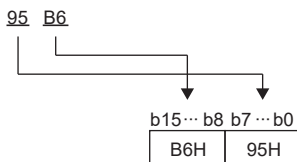
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	—	—	○	—	—	—	—	○	—	—	○	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	

功能

- 將(s)中指定的移位JIS字元串轉換為Unicode字元串後，儲存到(d)中。
- 對於(s)的移位JIS字元串，應以大端字節序進行指定。

例

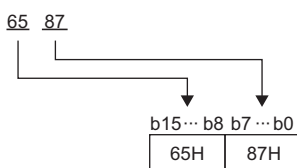
移位JIS字元串的“95B6H”以“B695H”進行指定。



- 對於(d)的Unicode字元串，以小端字節序儲存。

例

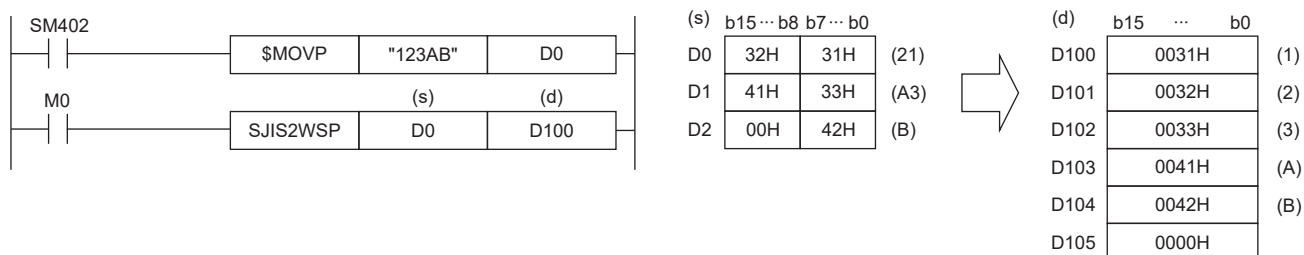
Unicode字元串的“6587H”以“6587H”進行指定。



- 在SJIS2WS(P)指令中，不在(d)的起始處附加字節順序標記。在希望附加字節順序標記的情況下，應使用SJIS2WSB(P)指令。

☞ 816頁 SJIS2WSB(P)

- 移位JIS→Unicode轉換的動作如下所示。



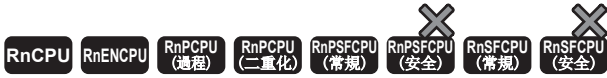
出錯

出錯代碼 (SD0)	內容
2821H	在(s)、(d)中指定的範圍重複的情況下。
3401H	在(s)中指定的範圍中，包含有不能轉換的字元代碼的情況下。
3405H	在(s)中指定的字元串超過16383字元的情況下。 ^{*1}

*1 對於以漢字等的移位JIS代碼表示時變為2字節資料的字元，應將每個字元以2進行計數。

移位JIS字元串→Unicode轉換(有字節順序標記)

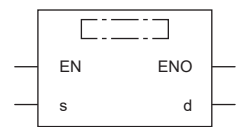
SJIS2WSB(P)



將移位JIS字元串轉換為Unicode字元串後，在起始處附加字節順序標記。

梯形圖	ST
	<pre>ENO:=SJIS2WSB(EN, s, d); ENO:=SJIS2WSBP(EN, s, d);</pre>

FBD/LD



■執行條件

指令	執行條件
SJIS2WSB	
SJIS2WSBP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存了轉換的字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存轉換後的字元串的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

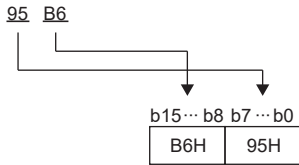
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	

功能

- 將(s)中指定的移位JIS字元串轉換為Unicode字元串後，在起始處附加字節順序標記後儲存到(d)中。
- 對於(s)的移位JIS字元串，應以大端字節序進行指定。

例

移位JIS字元串的“95B6H”以“B695H”進行指定。



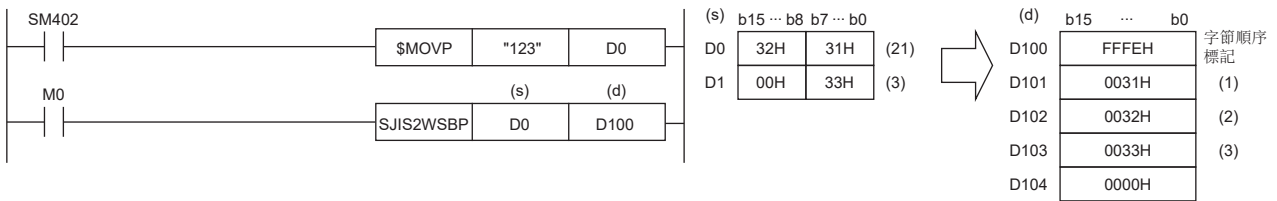
- 對於(d)的Unicode字元串，以小端字節序儲存。

例

Unicode字元串的“6587H”以“6587H”進行指定。



- 移位JIS→Unicode轉換的動作如下所示。



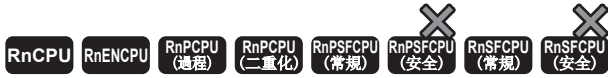
出錯

出錯代碼(SD0)	內容
2821H	在(s)、(d)中指定的範圍重複的情況下。
3401H	在(s)中指定的範圍中，包含有不能轉換的字元代碼的情況下。
3405H	在(s)中指定的字元串超過16383字元的情況下。 ^{*1}

*1 對於以漢字等的移位JIS代碼表示時變為2字節資料的字元，應將每個字元以2進行計數。

字元串的長度檢測

LEN(P)



對指定的字元串的長度進行檢測。

梯形圖	ST*1
	$ENO := LENP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援LEN指令。應使用通用函數的LEN。

☞ 1441頁 LEN(E)

■執行條件

指令	執行條件
LEN	
LENP	

設置數據

■內容、範圍、資料類型

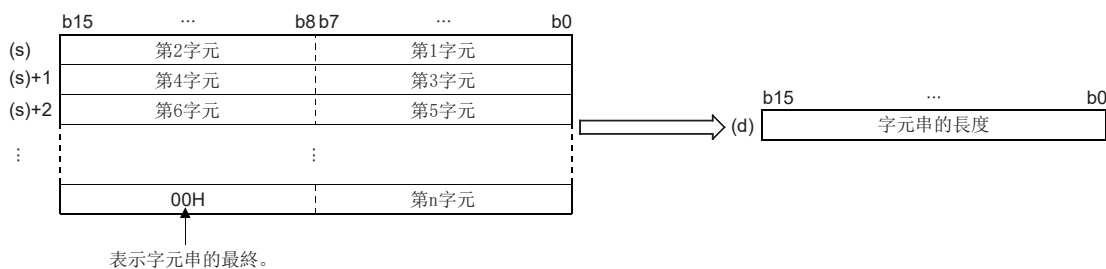
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	字元串或儲存了字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存檢測的字元串長度的元件編號	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s)	—	—	○	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	—	○	—	—	—	—

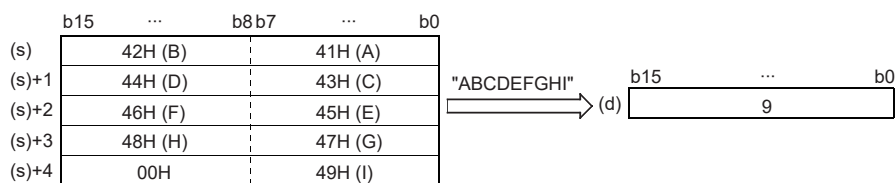
功能

- 檢測(s)中指定的字元串的長度、並儲存到(d)中指定的元件編號及其以後。將(s)中指定的元件編號開始至00H儲存的元件編號為止的資料作為字元串處理。



例

在(s)及其以後儲存了“ABCDEFGHI”的情況下



出錯

出錯代碼(SD0)	內容
2820H	(s)中指定的元件編號及其以後，元件/標籤記憶體各設定區域範圍內未設定00H時。
3405H	當(s)的字元串超過了16383字元時。

從字元串的右側開始提取

RIGHT (P)



從字元串資料的右側開始提取 (n) 字元的資料。

梯形圖	ST*1
	$ENO := RIGHTP(EN, s, n, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援RIGHT指令。應使用通用函數的RIGHT。

☞ 1443頁 LEFT(_E)、RIGHT(_E)

■執行條件

指令	執行條件
RIGHT	
RIGHTP	

設置數據

■內容、範圍、資料類型

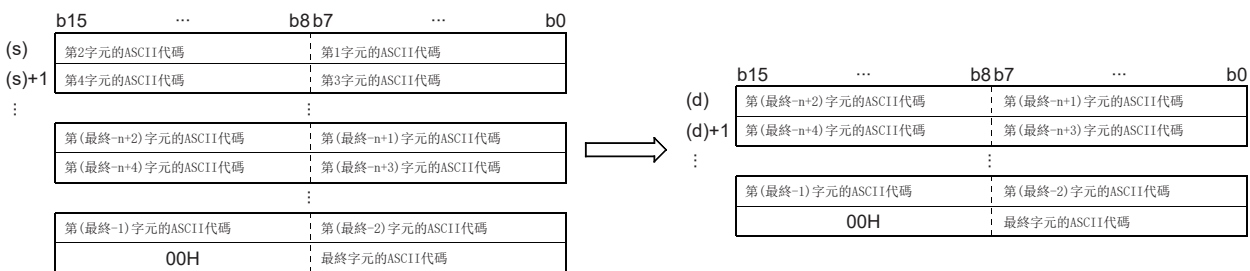
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	字元串或儲存了字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存從(s)右側開始的(n)字元的字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(n)	提取的字元數	1~16383	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

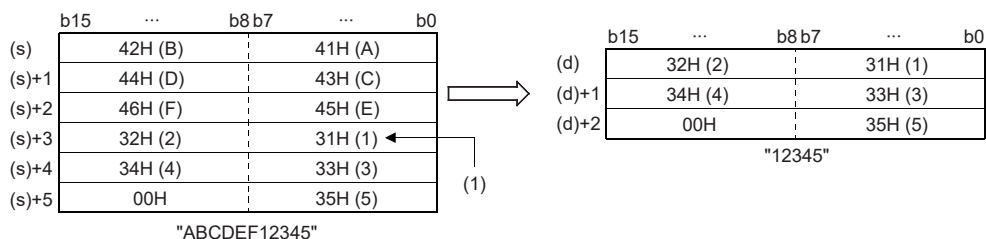
功能

- 對於(s)中指定的元件編號及其以後中儲存的字元串資料，將字元串的右側(字元串的最終)開始的(n)字元的資料，儲存到(d)中指定的元件編號及其以後。



例

在(n)=5的情況下



(1): 第5字元的ASCII碼

- 表示字元串的最終的NULL代碼(00H)將被自動附加到字元串資料的最後。
- 在(n)中指定的字元數為0的情況下，(d)中將儲存NULL代碼(00H)。

出錯

出錯代碼(SD0)	內容
2820H	(s)中指定的元件編號及其以後，元件/標籤記憶體各設定區域中未設定00H時。
3405H	當(s)中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none"> (s)的字元串超過了16383字元 (s)的字元串為0字元 (n)超過(s)中指定的字元數時。

從字元串的左側開始提取

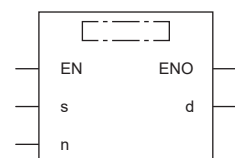
LEFT (P)



將字元串資料的左側開始的 (n) 字元的資料儲存到 (d) 中指定的元件編號及其以後。

梯形圖	ST*1
	ENO:=LEFTP (EN, s, n, d) ;

FBD/LD*1



*1 在ST、FBD/LD中不支援LEFT指令。應使用通用函數的LEFT。

☞ 1443頁 LEFT(_E)、RIGHT(_E)

■執行條件

指令	執行條件
LEFT	
LEFTP	

設置數據

■內容、範圍、資料類型

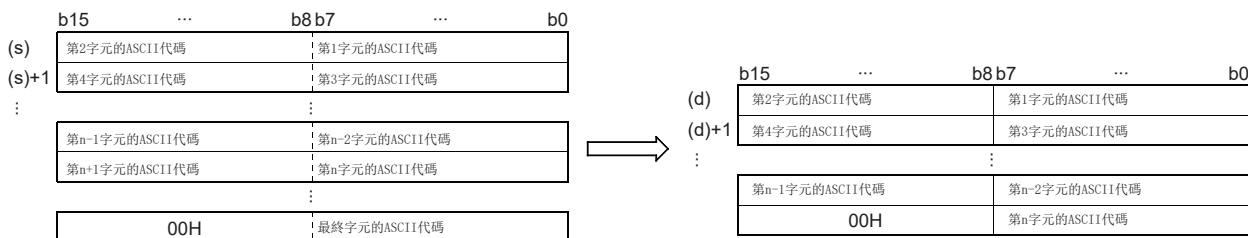
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	字元串或儲存了字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存從(s)左側開始的(n)字元的字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(n)	提取的字元數	1~16383	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\□(H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

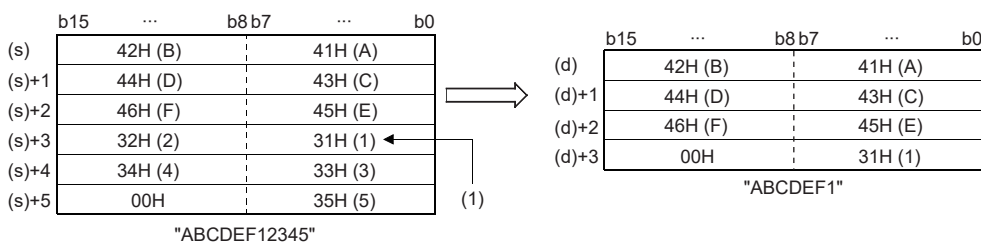
功能

- 對於(s)中指定的元件編號及其以後中儲存的字元串資料，將字元串的左側(字元串的起始)開始的(n)字元的資料，儲存到(d)中指定的元件編號及其以後。



例

在(n)=7的情況下



(1): 第7字元的ASCII碼

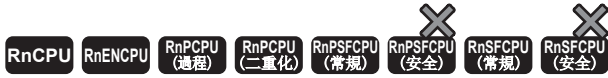
- 表示字元串的最終的NULL代碼(00H)將被自動附加到字元串資料的最後。
- 在(n)中指定的字元數為0的情況下，(d)中將儲存NULL代碼(00H)。

出錯

出錯代碼(SD0)	內容
2820H	(s)中指定的元件編號及其以後，元件/標籤記憶體各設定區域中未設定00H時。
3405H	當(s)中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none"> (s)的字元串超過了16383字元 (s)的字元串為0字元 (n)超過(s)中指定的字元數時。

字元串中的任意提取

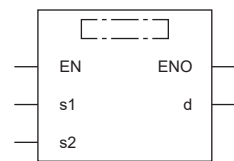
MIDR (P)



將字元串資料中的任意位置的資料儲存到 (d) 中指定的元件編號及其以後。

梯形圖	ST
	ENO:=MIDR (EN, s1, s2, d) ; ENO:=MIDRP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
MIDR	
MIDRP	

設置數據

■內容、範圍、資料類型

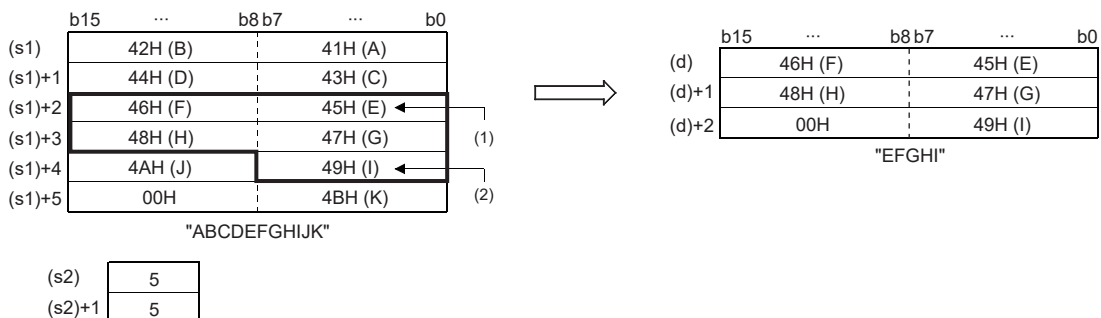
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	字元串或儲存了字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存運算結果字元串資料的起始元件	—	字元串	ANYSTRING_SINGLE
(s2)	儲存起始字元位置及字元數的起始元件 (s2): 起始字元的位置, (s2)+1: 字元數	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—	—

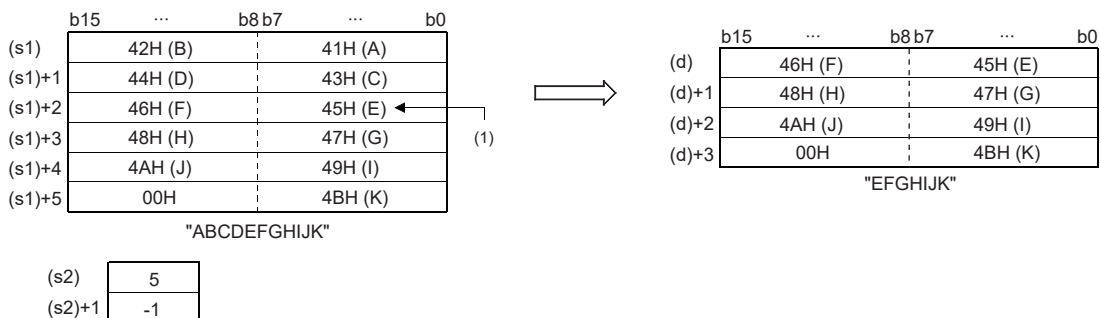
功能

- 對於(s1)中指定的元件編號及其以後中儲存的字元串資料，從(s2)中指定位置開始將(s2)+1中指定字元的資料儲存到(d)中指定的元件編號及其以後。



- (1): 第5字元的位置(s2)
- (2): 第5字元的ASCII碼(s2)+1

- 表示字元串的最終的NULL代碼(00H)將被自動附加到字元串資料的最後。
- (s2)+1中指定的字元數為0的情況下不進行處理。
- 在(s2)+1中指定的字元數為-1的情況下，將(s1)中指定的最終字元資料為止的資料儲存到(d)中指定的元件及其以後。



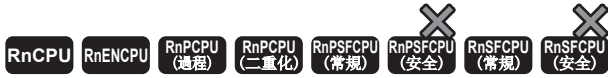
- (1): 第5字元的位置(s2)

出錯

出錯代碼(SD0)	內容
2820H	(s1)中指定的元件編號及其以後，元件/標籤記憶體各設定區域中不存在00H時。
3405H	當(s1)的字元串超過16383字元時。 當(s2)中設定超出允許指定範圍的資料時。 <ul style="list-style-type: none"> (s2)的值不超過0 (s2)+1的值為有效值(-1、0、1及其以上)以外 (s2)的值超過了(s1)的字元數 (s2)與(s2)+1的加法運算後的值超過了(s1)的字元數

字元串中的任意替換

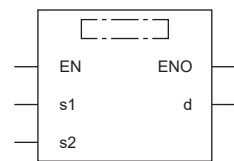
MIDW (P)



將字元串資料中任意位置的資料以任意字元串進行替換。

梯形圖	ST
	ENO:=MIDW (EN, s1, s2, d) ; ENO:=MIDWP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
MIDW	
MIDWP	

設置數據

■內容、範圍、資料類型

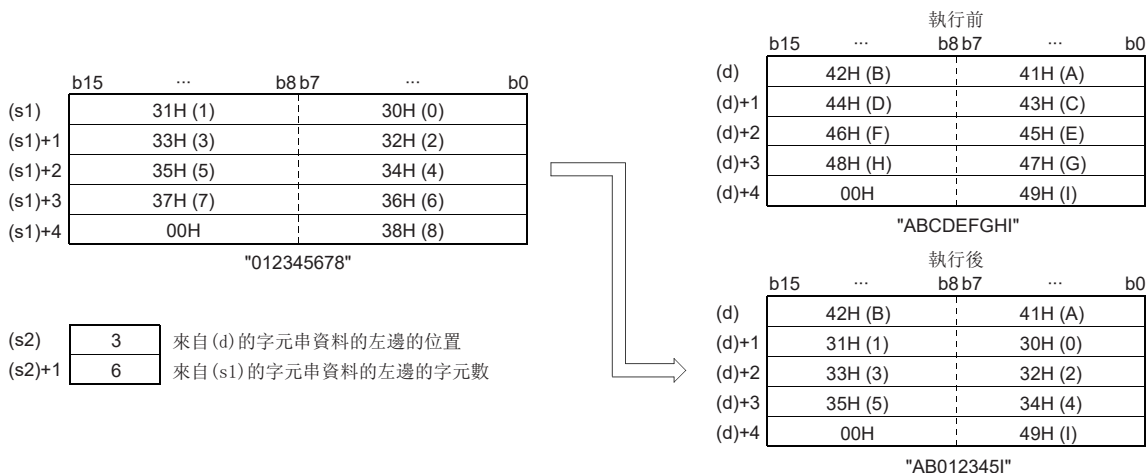
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	字元串或儲存了字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存運算結果字元串資料的起始元件	—	字元串	ANYSTRING_SINGLE
(s2)	儲存起始字元位置及字元數的起始元件 (s2): 起始字元的位置, (s2)+1: 字元數	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

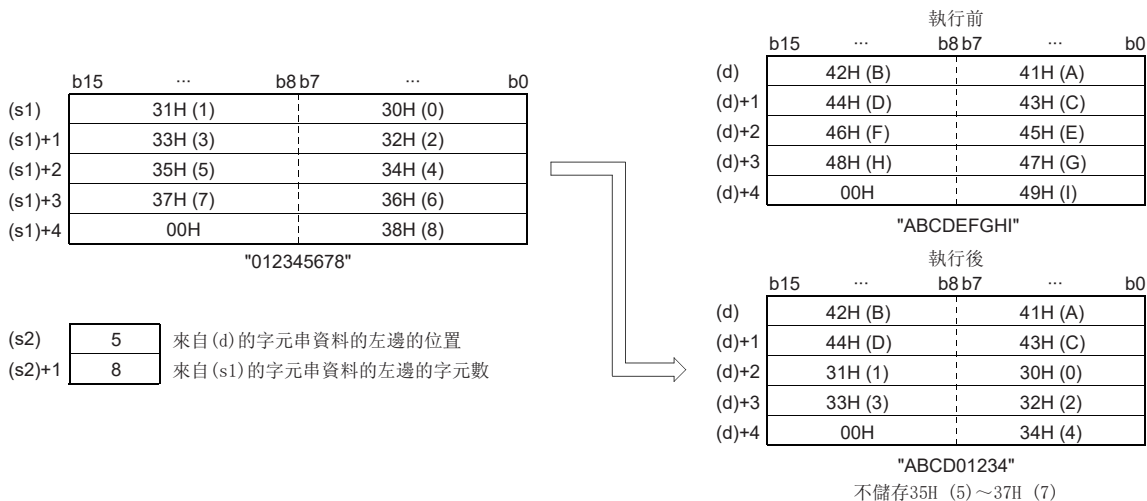
操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—	—

功能

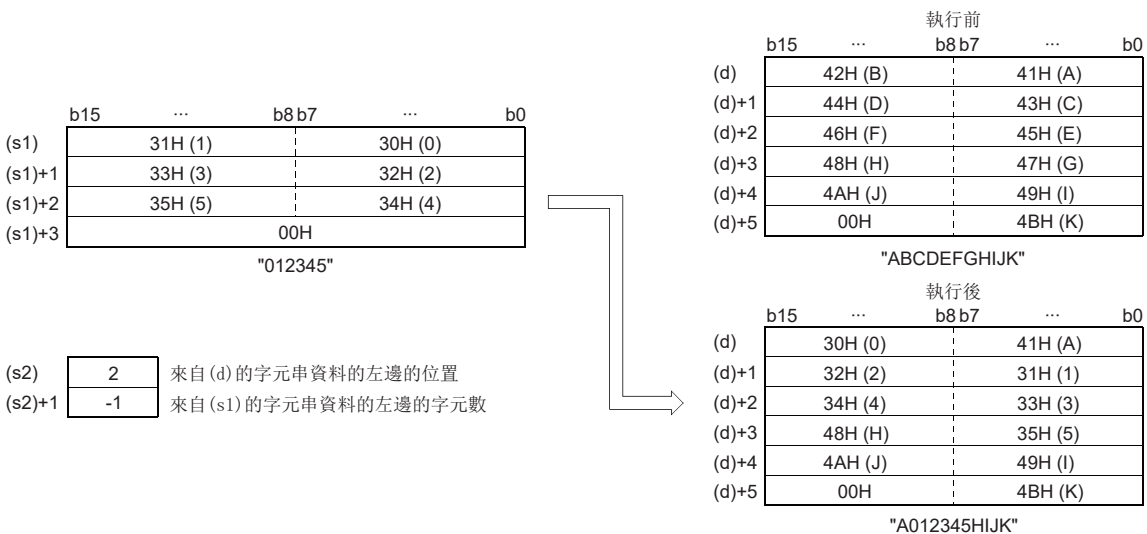
- 對於 (s1) 中指定的元件編號及其以後中儲存的字元串資料，將 (s2)+1 中指定的字元的資料儲存到 (d) 中指定的元件編號及其以後中儲存的字元串資料的 (s2) 中指定的位置及其以後。



- 表示字元串的最終的NULL代碼(00H)將被自動附加到字元串資料的最後。
- (s2)+1 中指定的字元數為0的情況下不進行處理。
- 在 (s2)+1 中指定的字元數超過了 (d) 中指定的字元串資料的最終字元的情況下，儲存 (d) 的最終字元為止的資料。



- 在 (s2)+1 中指定的字元數為-1的情況下，將 (s1) 中指定的最終字元資料為止的資料儲存到 (d) 中指定的元件及其以後。



出錯

出錯代碼 (SD0)	內容
2820H	(s1) 中指定的元件編號及其以後，元件/標籤記憶體各設定區域中不存在00H時。 (d) 中指定的元件編號及其以後，元件/標籤記憶體各設定區域中不存在00H時。
3405H	當 (s1) 的字元串超過16383字元時。 當 (d) 的字元串超過了16383字元時。 當 (s2) 中設定超出允許指定範圍的資料時。 <ul style="list-style-type: none">• (s2) 的值不超過0• (s2)+1 的值為有效值(-1、0、1及其以上)以外• (s2) 的值超過了 (d) 的字元數• 當 (s2)+1 的值超過了 (s1) 的字元數時。

字元串搜尋

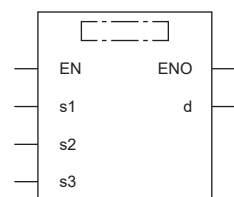
INSTR(P)



搜尋字元串資料中的指定字元串。

梯形圖	ST
	ENO:=INSTR(EN, s1, s2, s3, d); ENO:=INSTRP(EN, s1, s2, s3, d);

FBD/LD



■執行條件

指令	執行條件
INSTR	
INSTRP	

設置數據

■內容、範圍、資料類型

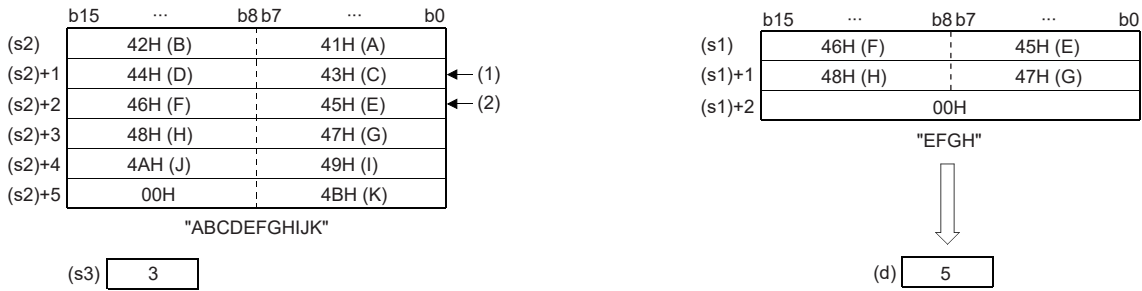
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	搜尋字元串或儲存了搜尋字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(s2)	被搜尋的字元串或儲存了被搜尋的字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存搜尋結果的元件	—	帶符號BIN16位元	ANY16
(s3)	搜尋開始位置	1~16383	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	○	—	
(s2)	—	—	○	—	—	—	○	—	—	—	○	—	
(d)	○	○	○	○	○	—	○	—	—	—	—	—	
(s3)	○	○	○	○	○	—	○	—	○	—	—	—	

功能

- 從 (s2) 中指定的元件編號及其以後中儲存的字元串資料的左側第 (s3) 字元開始，搜尋 (s1) 中指定的元件編號及其以後中儲存的字元串資料，將搜尋結果儲存到 (d) 中指定的元件中。搜尋結果將儲存 (s2) 中指定的字元串資料的起始字元開始的第幾字元。



(1): 搜尋開始位置 (s3): 第3字元

(2): 從起始字元開始第5字元

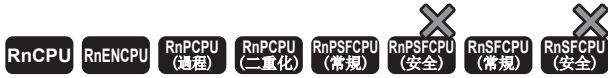
- 在沒有一致的字元串資料的情況下，(d) 中將儲存0。

出錯

出錯代碼 (SD0)	內容
2820H	(s1) 中指定的元件及其以後，元件/標籤記憶體各設定區域中沒有 NULL 代碼 (00H) 時。 (s2) 中指定的元件及其以後，元件/標籤記憶體各設定區域中沒有 NULL 代碼 (00H) 時。
3405H	當 (s1) 中設定超出允許指定範圍的資料時。 <ul style="list-style-type: none"> 設定的字元串為 0 字元 設定的字元串超過了 16383 字元 當 (s2) 中設定的字元串超過了 16383 字元時。 當 (s3) 中設定超出允許指定範圍的資料時。 <ul style="list-style-type: none"> (s3) 的值超過了 (s2) 的字元數 (s3) 的值為負的數或 0

字元串插入

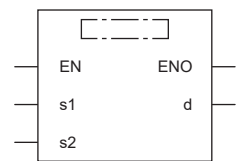
STRINS (P)



將任意字元串資料插入到字元串資料任意插入位置。

梯形圖	ST
	ENO:=STRINS (EN, s1, s2, d); ENO:=STRINSP (EN, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
STRINS	
STRINSP	

設置數據

■內容、範圍、資料類型

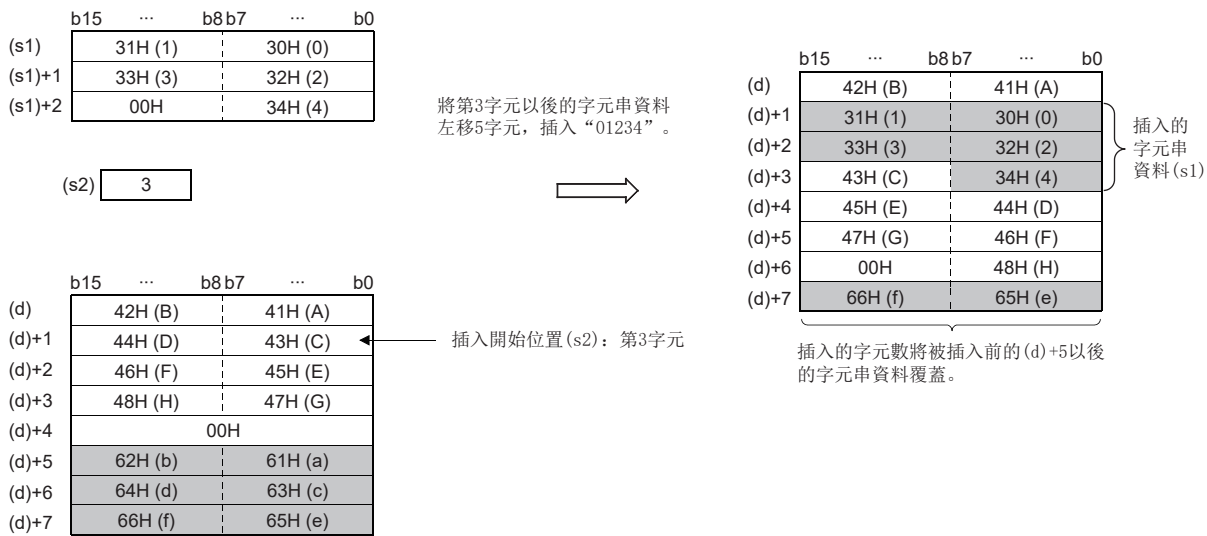
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	插入字元串或儲存了插入字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存插入字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(s2)	插入位置 (字節單位)	1~16383	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	—	—	○	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	○	○	—	—	—

功能

- 將(s1)中指定的字元串資料，插入到(d)中指定的字元串資料的起始開始的第(s2)字元(插入位置)處。



- 在插入後的字元串(s1)+(d)為偶數的情況下，字元串的最後的下一個元件(1字)中將儲存NULL代碼(00H)。
- 插入後的字元串(s1)+(d)為奇數的情況下，字元串的最後的元件(高位8位元)中將儲存NULL代碼(00H)。
- 在(s2)中指定了(d)的字元數+1的情況下，將(d)的字元串的最後與(s1)的字元串合併。

出錯

出錯代碼(SD0)	內容
2820H	(s1)中指定的元件及其以後，元件/標籤記憶體各設定區域中沒有NULL代碼(00H)時。 (d)中指定的元件及其以後，元件/標籤記憶體各設定區域中沒有NULL代碼(00H)時。
2821H	當字元串(s1)與字元串(d)的元件有部分重複時。 當插入後的字元串(s1)+(d)與(s1)的字元串儲存元件重複時。
3405H	當(s1)中指定的字元串的字元數超過了16383字元時。 當(s2)中設定超出允許指定範圍的資料時。 <ul style="list-style-type: none"> 指定的值超過了字元串(d)的字元數+1 指定的值不在下述範圍內 $1 \leq (s2) \leq 16383$ 當(d)中指定的字元串的字元數超過了16383字元時。
3406H	當插入後的字元串(s1)+(d)為超出允許輸出範圍的資料時。 <ul style="list-style-type: none"> 插入後的字元串的字元數超過了16383字元 插入後的字元串超過了指定元件/標籤記憶體各設定區域

字元串刪除

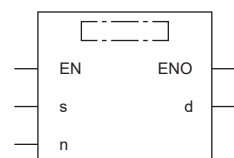
STRDEL (P)



從字元串資料的指定的位置刪除 (n) 字元的資料。

梯形圖	ST
	ENO:=STRDEL (EN, s, n, d) ; ENO:=STRDELP (EN, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
STRDEL	
STRDELP	

設置數據

■內容、範圍、資料類型

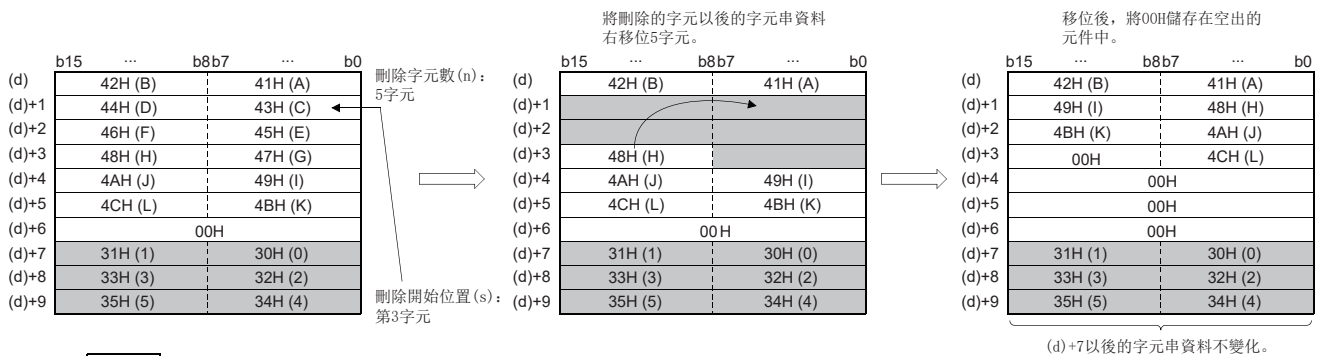
操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	儲存了刪除字元串的起始元件	—	字元串	ANYSTRING_SINGLE
(s)	刪除開始位置	1~16383	帶符號BIN16位元	ANY16
(n)	刪除字元數	0~16384-(s)	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(s)	—	○	○	○	○	—	—	○	○	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 從 (d) 中指定的字元串資料的起始開始的第 (s) 字元中指定的位置 (刪除開始位置) 開始，刪除 (n) 字元的資料。



出錯

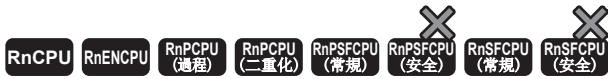
出錯代碼 (SD0)	內容
2820H	(d) 中指定的元件及其以後，元件/標籤記憶體體的各設定區域中沒有 NULL 代碼 (00H) 時。
3405H	<p>當 (d) 中指定的字元串的字元數超過了 16383 字元時。</p> <p>當 (s) 中設定了超出允許指定範圍的資料時。</p> <ul style="list-style-type: none"> 指定的值不在下述範圍內 $1 \leq (s) \leq 16383$ <ul style="list-style-type: none"> 指定的值超出了字元串 (d) 的字元數 <p>當 (n) 中設定超出允許指定範圍的資料時。</p> <ul style="list-style-type: none"> 指定的值超過了字元串 (d) 的 (s) 開始至最終字元為止的字元數 指定的值為負數

11 實數處理

11.1 浮點指令

單精度實數比較

LDE□、ANDE□、ORE□

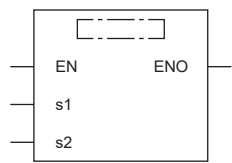


將單精度實數透過常開觸點處理進行比較運算。

梯形圖	ST*1
	ENO:=LDE_□(EN, s1, s2); ENO:=ANDE_□(EN, s1, s2); ENO:=ORE_□(EN, s1, s2); (□中放入EQ、NE、GT、LE、LT、GE。)*2

(□輸入E=、E<>、E>、E<=、E<、E>=。)

FBD/LD



(□中放入LDE_、ANDE_、ORE_與EQ、NE、GT、LE、LT、GE的組合。)*2

*1 在版本為“1.035M”及其以後的工程工具中對應。
 *2 EQ為=、NE為<>、GT為>、LE為<=、LT為<、GE為>=。

■執行條件

指令	執行條件
LDE□、ANDE□、ORE□	始終執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	比較資料或儲存了比較資料的起始元件	$0, 2^{-126} \leq (s1) < 2^{128}$	單精度實數	ANYREAL_32
(s2)	比較資料或儲存了比較資料的起始元件	$0, 2^{-126} \leq (s2) < 2^{128}$	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
EN0	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	○	○	○	○	○	—	○	—	—

功能

- 將(s1)中指定的元件的單精度實數與(s2)中指定的元件的單精度實數透過常開觸點處理進行比較運算。
- 各指令的比較運算結果如下所示。

指令符號(梯形圖、FBD/LD)	條件	比較運算結果
E=、EQ	(s1)=(s2)	導通狀態(ENO為ON)
E<>、NE	(s1)≠(s2)	
E>、GT	(s1)>(s2)	
E<=、LE	(s1)≤(s2)	
E<、LT	(s1)<(s2)	
E>=、GE	(s1)≥(s2)	
E=、EQ	(s1)≠(s2)	非導通狀態(ENO為OFF)
E<>、NE	(s1)=(s2)	
E>、GT	(s1)≤(s2)	
E<=、LE	(s1)>(s2)	
E<、LT	(s1)≥(s2)	
E>=、GE	(s1)<(s2)	

- 在(s1)、(s2)的內容超出設定資料的範圍的情況下，變為非導通(ENO為OFF)。
- 在FBD/LD中使用LDE_□指令時，應將左母線或常時ON的變數部件/常數部件指定到EN。
- 在ORE_□指令透過FBD/LD使用的情況下，將EN指定為TRUE時ENO將ON。EN不成為執行條件。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

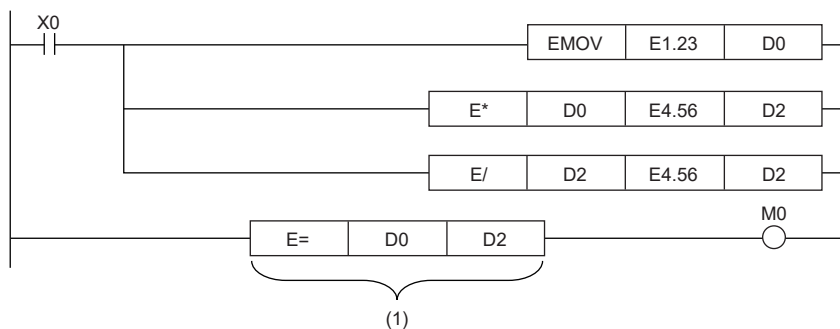
☞ 45頁 注意事項

出錯

沒有運算出錯。

要點

使用了E=指令的情況下，由於誤差等可能導致不相等，應加以注意。



(1)：可能會出現不相等的情况。

雙精度實數比較

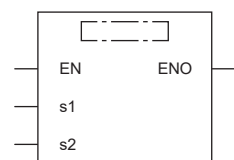
LDED□、ANDED□、ORED□



將雙精度實數透過常開觸點處理進行比較運算。

梯形圖	ST*1
<p>(□中放入ED=、ED<>、ED>、ED<=、ED<、ED>=。)</p>	<pre> ENO:=LDED_□(EN, s1, s2); ENO:=ANDED_□(EN, s1, s2); ENO:=ORED_□(EN, s1, s2); (□中放入EQ、NE、GT、LE、LT、GE。)*2 </pre>

FBD/LD



(□中放入LDED_、ANDED_、ORED_與EQ、NE、GT、LE、LT、GE的組合。)*2

*1 在版本為“1.035M”及其以後的工程工具中對應。
 *2 EQ為=、NE為<>、GT為>、LE為<=、LT為<、GE為>=。

■執行條件

指令	執行條件
LDED□、ANDED□、ORED□	始終執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	比較資料或儲存了比較資料的起始元件	$0, 2^{-1022} \leq (s1) < 2^{1024}$	雙精度實數	ANYREAL_64
(s2)	比較資料或儲存了比較資料的起始元件	$0, 2^{-1022} \leq (s2) < 2^{1024}$	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	—	—	○	—	○	—	—	—
(s2)	—	—	○	—	—	○	—	○	—	—	—

功能

- 將(s1)中指定的元件的雙精度實數與(s2)中指定的元件的雙精度實數透過常開觸點處理進行比較運算。
- 各指令的比較運算結果如下所示。

指令符號(梯形圖、FBD/LD)	條件	比較運算結果
ED=、EQ	$(s1) = (s2)$	導通狀態 (ENO為ON)
ED<>、NE	$(s1) \neq (s2)$	
ED>、GT	$(s1) > (s2)$	
ED<=、LE	$(s1) \leq (s2)$	
ED<、LT	$(s1) < (s2)$	
ED>=、GE	$(s1) \geq (s2)$	
ED=、EQ	$(s1) \neq (s2)$	非導通狀態 (ENO為OFF)
ED<>、NE	$(s1) = (s2)$	
ED>、GT	$(s1) \leq (s2)$	
ED<=、LE	$(s1) > (s2)$	
ED<、LT	$(s1) \geq (s2)$	
ED>=、GE	$(s1) < (s2)$	

- 在(s1)、(s2)的內容超出設定資料的範圍的情況下，變為非導通(ENO為OFF)。
- 在FBD/LD中使用LDED_□指令時，應將左母線或常時ON的變數部件/常數部件指定到EN。
- 在將ORED_□指令透過FBD/LD使用的情況下，將EN指定為TRUE時ENO將ON。EN不成為執行條件。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

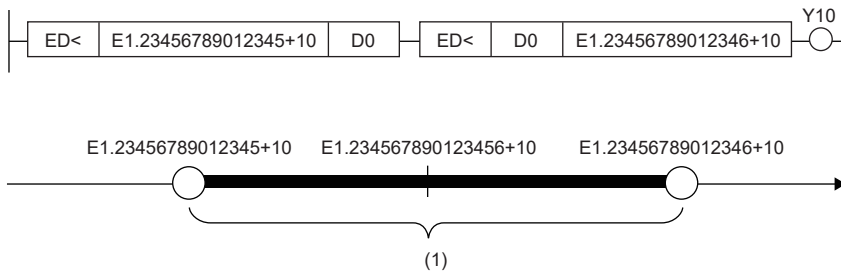
沒有運算出錯。

注意事項

因為透過工程工具可輸入的實數的位數最大為15位數，因此本項所示的指令中不能對有效位數16位數及其以上的實數進行比較。在對本項的指令中有效位數16位及其以上的實數進行一致、不一致判斷的情況下，需要透過與比較的實數前後近似值的大小比較進行判斷。

例

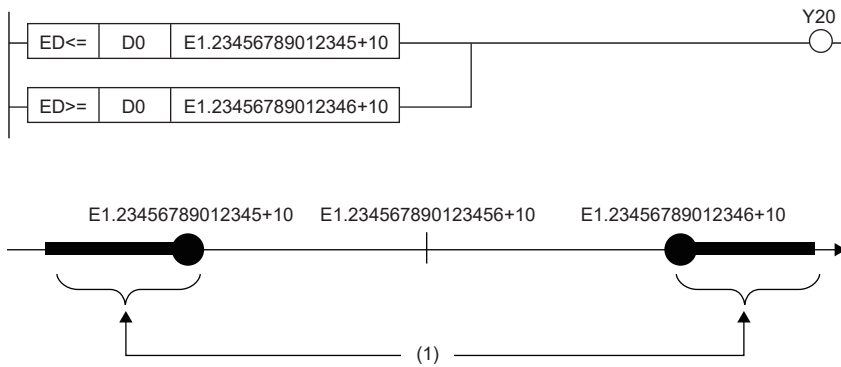
與E1.23456789012345+10(有效位數16位)進行雙精度實數的一致判斷的情況下



(1): 確認D0~D3是否在該範圍內。(邊界值不包含在範圍內。)

例

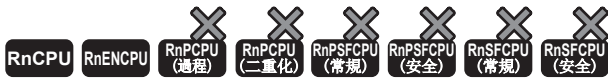
在與E1.23456789012345+10(有效位數16位)進行雙精度實數的不一致判斷的情況下



(1): 確認D0~D3是否在該範圍內。(邊界值包含在範圍內。)

單精度實數比較輸出

ECMP (P)

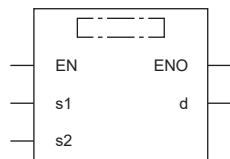


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

比較 (s1) 中指定的單精度實數資料與 (s2) 中指定的單精度實數資料，根據比較的結果 (小、一致、大)，(d)、(d)+1、(d)+2 其中之一將變為 ON。

梯形圖	ST
	ENO:=ECMP (EN, s1, s2, d) ; ENO:=ECMPP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
ECMP	
ECMPP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	比較資料或儲存了比較資料的起始元件	$0, 2^{-126} \leq s1 < 2^{128}$	單精度實數	ANYREAL_32
(s2)	比較資料或儲存了比較資料的起始元件	$0, 2^{-126} \leq s2 < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存了比較結果的起始元件	—	位元	ANYBIT_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

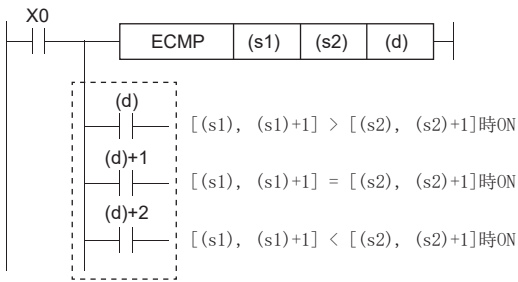
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	—	—	○	○	—	○	○	—	○	—	—
(s2)	—	—	○	○	—	○	○	—	○	—	—
(d)	○	—	○*1	—	—	—	○	—	—	—	—

*1 不能使用T、ST、C。

功能

- 比較(s1)中指定的單精度實數資料與(s2)中指定的單精度實數資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為0N。

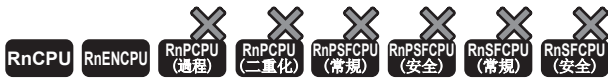


出錯

出錯代碼 (SD0)	內容
3402H	(s1)、(s2)中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。

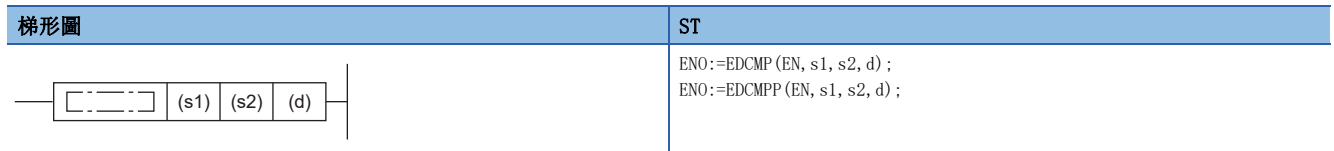
雙精度實數比較輸出

EDCMP (P)

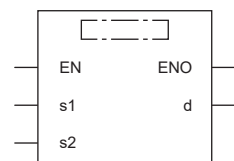


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

比較 (s1) 中指定的雙精度實數資料與 (s2) 中指定的雙精度實數資料，根據比較的結果 (小、一致、大)，(d)、(d)+1、(d)+2 其中之一將變為 ON。



FBD/LD



■執行條件

指令	執行條件
EDCMP	
EDCMP (P)	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	比較資料或儲存了比較資料的起始元件	$0, 2^{-1022} \leq (s1) < 2^{1024}$	雙精度實數	ANYREAL_64
(s2)	比較資料或儲存了比較資料的起始元件	$0, 2^{-1022} \leq (s2) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存了比較結果的起始元件	—	位元	ANYBIT_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

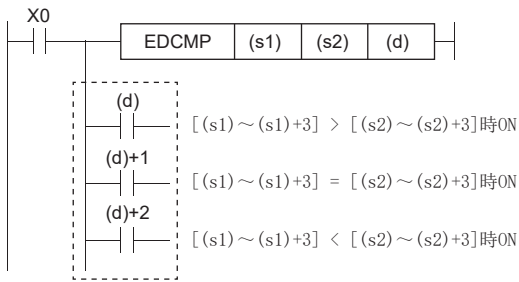
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	—	—	○	—	—	○	—	○	—	—	—
(s2)	—	—	○	—	—	○	—	○	—	—	—
(d)	○	—	○*1	—	—	—	—	○	—	—	—

*1 不能使用T、ST、C。

功能

- 比較(s1)中指定的雙精度實數資料與(s2)中指定的雙精度實數資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為ON。

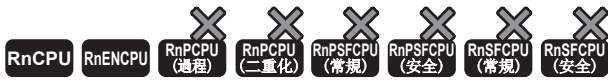


出錯

出錯代碼 (SD0)	內容
3402H	(s1)、(s2)中指定的資料為-0、非正規化數、非數、±∞時。

單精度實數區域比較輸出

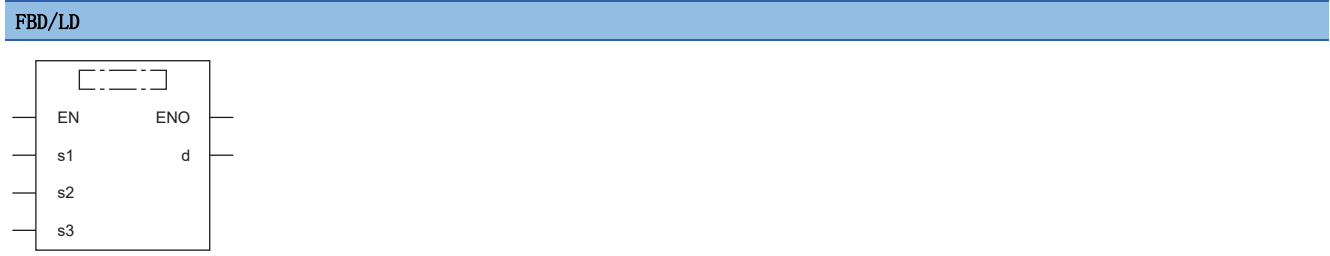
EZCP (P)



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

將下限值 (s1) 的單精度實數和上限值 (s2) 的單精度實數的區域與比較資料 (s3) 的單精度實數進行比較，根據結果 (下、區域內、上) 決定 (d)、(d)+1、(d)+2 哪個為 ON。

梯形圖	ST
	<pre>ENO:= EZCP(EN, s1, s2, s3, d); ENO:= EZCPP(EN, s1, s2, s3, d);</pre>



■執行條件

指令	執行條件
EZCP	
EZCPP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	下限值或儲存了下限值的起始元件	$0, 2^{-126} \leq s1 < 2^{128}$	單精度實數	ANYREAL_32
(s2)	上限值或儲存了上限值的起始元件	$0, 2^{-126} \leq s2 < 2^{128}$	單精度實數	ANYREAL_32
(s3)	比較資料或儲存了比較資料的起始元件	$0, 2^{-126} \leq s3 < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存了比較結果的起始元件	—	位元	ANYBIT_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

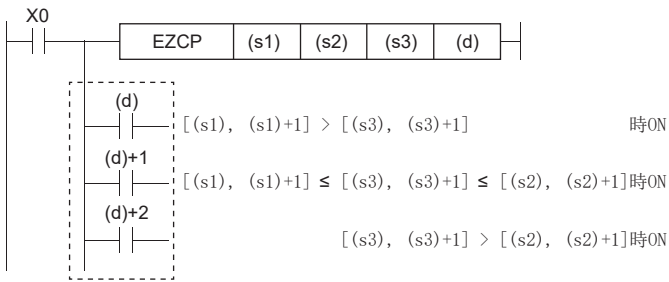
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	○	—	○	○	—	○	—	—	—
(s2)	—	—	○	○	—	○	○	—	○	—	—	—
(s3)	—	—	○	○	—	○	○	—	○	—	—	—
(d)	○	—	○*1	—	—	—	○	—	—	—	—	—

*1 不能使用T、ST、C。

功能

- 將下限值 (s1) 的單精度實數和上限值 (s2) 的單精度實數的區域與比較資料 (s3) 的單精度實數進行比較，根據結果 (下、區域內、上) 決定 (d)、(d)+1、(d)+2 哪個為 0N。



注意事項

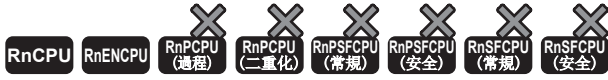
- 應將下限值 (s1) 設為比上限值 (s2) 小的值。將下限值 (s1) 設為大於上限值 (s2) 時，將上限值 (s2) 作為與下限值 (s1) 相同的值處理。

出錯

出錯代碼 (SD0)	內容
3402H	(s1) ~ (s3) 中指定的資料為 -0、非正規化數、非數、±∞ 時。

雙精度實數區域比較輸出

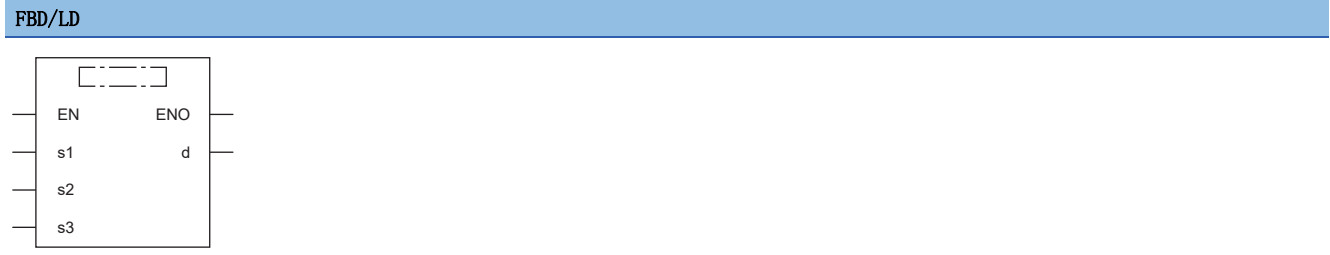
EDZCP (P)



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

將下限值 (s1) 的雙精度實數和上限值 (s2) 的雙精度實數的區域與比較資料 (s3) 的雙精度實數進行比較，根據結果 (下、區域內、上) 決定 (d)、(d)+1、(d)+2 哪個為ON。

梯形圖	ST
	<pre>ENO:= EDZCP(EN, s1, s2, s3, d); ENO:= EDZCPP(EN, s1, s2, s3, d);</pre>



■執行條件

指令	執行條件
EDZCP	
EDZCPP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	下限值或儲存了下限值的起始元件	$0, 2^{-1022} \leq s1 < 2^{1024}$	雙精度實數	ANYREAL_64
(s2)	上限值或儲存了上限值的起始元件	$0, 2^{-1022} \leq s2 < 2^{1024}$	雙精度實數	ANYREAL_64
(s3)	比較資料或儲存了比較資料的起始元件	$0, 2^{-1022} \leq s3 < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存了比較結果的起始元件	—	位元	ANYBIT_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

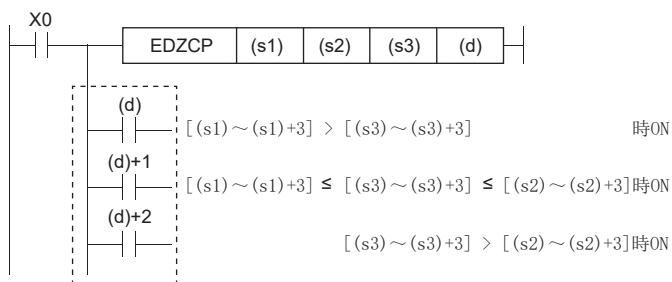
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	\$	
(s1)	—	—	○	—	—	○	—	○	—	—	—	—
(s2)	—	—	○	—	—	○	—	○	—	—	—	—
(s3)	—	—	○	—	—	○	—	○	—	—	—	—
(d)	○	—	○*1	—	—	—	—	○	—	—	—	—

*1 不能使用T、ST、C。

功能

- 將下限值 (s1) 的雙精度實數和上限值 (s2) 的雙精度實數的區域與比較資料 (s3) 的雙精度實數進行比較，根據結果 (下、區域內、上) 決定 (d)、(d)+1、(d)+2 哪個為 0N。



注意事項

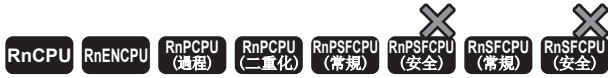
- 應將下限值 (s1) 設為比上限值 (s2) 小的值。將下限值 (s1) 設為大於上限值 (s2) 時，將上限值 (s2) 作為與下限值 (s1) 相同的值處理。

出錯

出錯代碼 (SD0)	內容
3402H	(s1) ~ (s3) 中指定的資料為 -0、非正規化數、非數、±∞ 時。

單精度實數加法運算

E+(P) [操作數為2個的情況下]



對單精度實數進行加法運算。

梯形圖	ST
	不對應。 (☞ 851頁 E+(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 851頁 E+(P) [操作數為3個的情況下])

執行條件

指令	執行條件
E+	
E+P	

設置數據

內容、範圍、資料類型

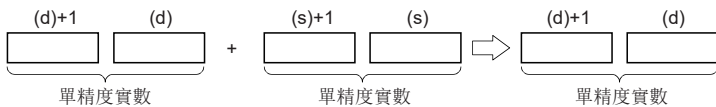
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	加法運算資料或儲存了加法運算資料的起始元件	$0, 2^{-126} \leq s < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存了被加法運算資料的起始元件	$0, 2^{-126} \leq d < 2^{128}$	單精度實數	ANYREAL_32

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—

功能

- 對(d)中指定的單精度實數與(s)中指定的單精度實數進行加法運算，將加法運算結果儲存到(d)中指定的元件中。



- (s)、(d)中可以指定及儲存 $0, 2^{-126} \leq |指定值(儲存值)| < 2^{128}$ 的值。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
	(d) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) $ d < 2^{128}$

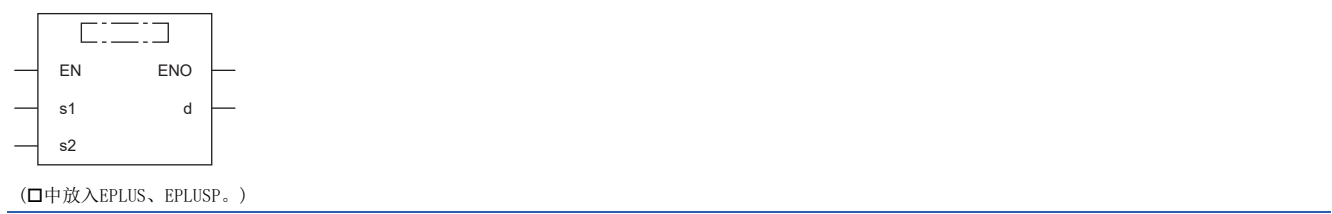
E+(P) [操作數為3個的情況下]



對單精度實數進行加法運算。

梯形圖	ST
	ENO:=EPLUS (EN, s1, s2, d) ; ENO:=EPLUSP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
E+	
E+P	

設置數據

■內容、範圍、資料類型

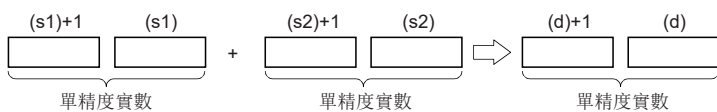
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	被加法運算資料或儲存了被加法運算資料的起始元件	$0, 2^{-126} \leq s1 < 2^{128}$	單精度實數	ANYREAL_32
(s2)	加法運算資料或儲存了加法運算資料的起始元件	$0, 2^{-126} \leq s2 < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(s1)	—	—	○	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 對(s1)中指定的單精度實數與(s2)中指定的單精度實數進行加法運算，將加法運算結果儲存到(d)中指定的元件中。



- (s1)、(s2)、(d)中可以指定及儲存 $0, 2^{-126} \leq |指定值(儲存值)| < 2^{128}$ 的值。

出錯

出錯代碼 (SD0)	內容
3402H	(s1) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
	(s2) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) $ d < 2^{128}$

單精度實數減法運算

E-(P) [操作數為2個的情況下]



對單精度實數進行減法運算。

梯形圖	ST
	不對應。 (☞ 855頁 E-(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 855頁 E-(P) [操作數為3個的情況下])

執行條件

指令	執行條件
E-	
E-P	

設置數據

內容、範圍、資料類型

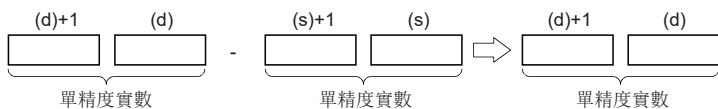
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	減法運算資料或儲存了減法運算資料的起始元件	$0, 2^{-126} \leq s < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存了被減法運算資料的起始元件	$0, 2^{-126} \leq d < 2^{128}$	單精度實數	ANYREAL_32

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s)	—	—	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—

功能

- 對 (d) 中指定的單精度實數與 (s) 中指定的單精度實數進行減法運算，將減法運算結果儲存到 (d) 中指定的元件中。



- (s)、(d) 中可以指定及儲存 $0, 2^{-126} \leq |指定值(儲存值)| < 2^{128}$ 的值。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
	(d) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) $ d < 2^{128}$

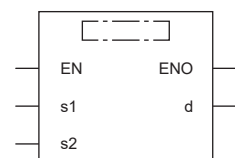
E-(P) [操作數為3個的情況下]



對單精度實數進行減法運算。

梯形圖	ST
	ENO:=EMINUS(EN, s1, s2, d); ENO:=EMINUSP(EN, s1, s2, d);

FBD/LD



(□中放入EMINUS、EMINUSP。)

■執行條件

指令	執行條件
E-	
E-P	

設置數據

■內容、範圍、資料類型

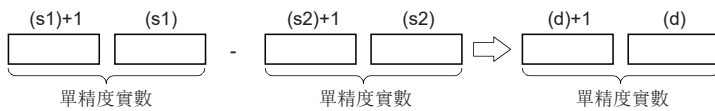
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被減法運算資料或儲存了被減法運算資料的起始元件	$0, 2^{-126} \leq (s1) < 2^{128}$	單精度實數	ANYREAL_32
(s2)	減法運算資料或儲存了減法運算資料的起始元件	$0, 2^{-126} \leq (s2) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	○	○	○	○	—	○	—	—	—
(s2)	—	—	○	○	○	○	○	—	○	—	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—	—

功能

- 對(s1)中指定的單精度實數與(s2)中指定的單精度實數進行減法運算，將減法運算結果儲存到(d)中指定的元件中。



- (s1)、(s2)、(d)中可以指定及儲存 $0, 2^{-126} \leq |\text{指定值 (儲存值)}| < 2^{128}$ 的值。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

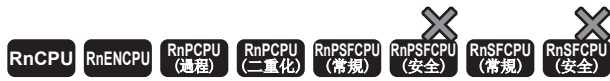
☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s1)中指定的資料為-0、非正規化數、非數、±∞時。
	(s2)中指定的資料為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生上溢時。) $ (d) < 2^{128}$

雙精度實數加法運算

ED+(P) [操作數為2個的情況下]



對雙精度實數進行加法運算。

梯形圖	ST
	不對應。 (☞ 859頁 ED+(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 859頁 ED+(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
ED+	
ED+P	

■設置數據

■內容、範圍、資料類型

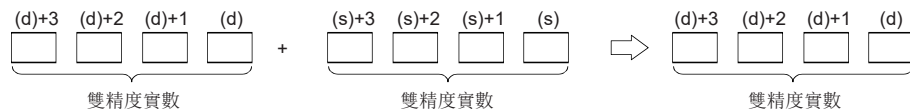
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	加法運算資料或儲存了加法運算資料的起始元件	$0, 2^{-1022} \leq s < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存了被加法運算資料的起始元件	$0, 2^{-1022} \leq d < 2^{1024}$	雙精度實數	ANYREAL_64

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

■功能

- 對(d)中指定的雙精度實數與(s)中指定的雙精度實數進行加法運算，將加法運算結果儲存到(d)中指定的元件中。



- (s)、(d)中可以指定及儲存 $0, 2^{-1022} \leq |指定值(儲存值)| < 2^{1024}$ 的值。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
	(d) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) $ d < 2^{1024}$

ED+(P) [操作數為3個的情況下]



對雙精度實數進行加法運算。

梯形圖	ST
	ENO:=EDPLUS (EN, s1, s2, d) ; ENO:=EDPLUSP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
ED+	
ED+P	

設置數據

■內容、範圍、資料類型

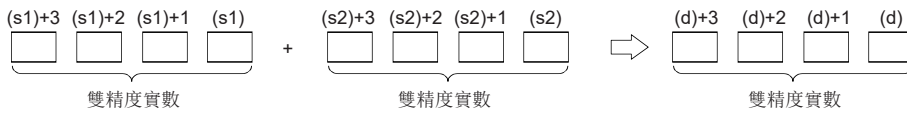
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	被加法運算資料或儲存了被加法運算資料的起始元件	$0, 2^{-1022} \leq s1 < 2^{1024}$	雙精度實數	ANYREAL_64
(s2)	加法運算資料或儲存了加法運算資料的起始元件	$0, 2^{-1022} \leq s2 < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	—	—	○	—	○	—	—	—	—
(s2)	—	—	○	—	—	○	—	○	—	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 對(s1)中指定的雙精度實數與(s2)中指定的雙精度實數進行加法運算，將加法運算結果儲存到(d)中指定的元件中。



- (s1)、(s2)、(d)中可以指定及儲存 $0, 2^{-1022} \leq | \text{指定值(儲存值)} | < 2^{1024}$ 的值。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s1)中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
	(s2)中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當(d)超出下述範圍時。(發生上溢時。) $ (d) < 2^{1024}$

雙精度實數減法運算

ED-(P) [操作數為2個的情況下]



對雙精度實數進行減法運算。

梯形圖	ST
	不對應。 (☞ 863頁 ED-(P) [操作數為3個的情況下])

FBD/LD
不對應。 (☞ 863頁 ED-(P) [操作數為3個的情況下])

■執行條件

指令	執行條件
ED-	
ED-P	

設置數據

■內容、範圍、資料類型

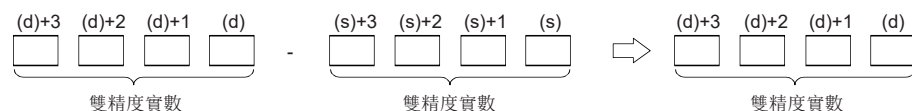
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	減法運算資料或儲存了減法運算資料的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存了被減法運算資料的起始元件	$0, 2^{-1022} \leq (d) < 2^{1024}$	雙精度實數	ANYREAL_64

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

功能

- 對(d)中指定的雙精度實數與(s)中指定的雙精度實數進行減法運算，將減法運算結果儲存到(d)中指定的元件中。



- (s)、(d)中可以指定及儲存 $0, 2^{-1022} \leq |指定值(儲存值)| < 2^{1024}$ 的值。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
	(d) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) $ d < 2^{1024}$

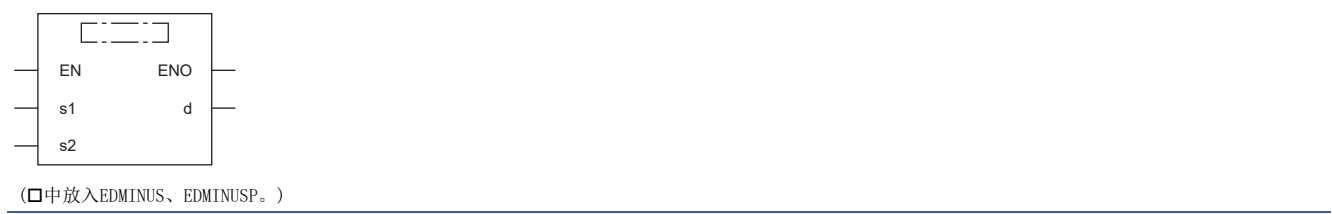
ED-(P) [操作數為3個的情況下]



對雙精度實數進行減法運算。

梯形圖	ST
	ENO:=EDMINUS (EN, s1, s2, d) ; ENO:=EDMINUSP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
ED-	
ED-P	

設置數據

■內容、範圍、資料類型

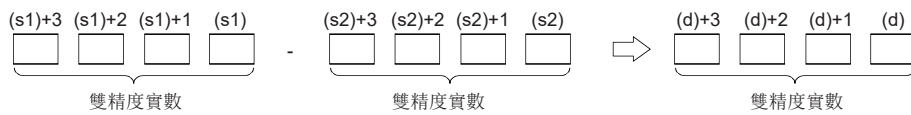
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	被減法運算資料或儲存了被減法運算資料的起始元件	$0, 2^{-1022} \leq s1 < 2^{1024}$	雙精度實數	ANYREAL_64
(s2)	減法運算資料或儲存了減法運算資料的起始元件	$0, 2^{-1022} \leq s2 < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	—	—	○	—	○	—	—	—
(s2)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

功能

- 對(s1)中指定的雙精度實數與(s2)中指定的雙精度實數進行減法運算，將減法運算結果儲存到(d)中指定的元件中。



- (s1)、(s2)、(d)中可以指定及儲存 $0, 2^{-1022} \leq | \text{指定值 (儲存值)} | < 2^{1024}$ 的值。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

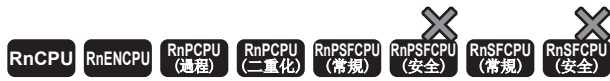
☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s1)中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
	(s2)中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當(d)超出下述範圍時。(發生上溢時。) $ (d) < 2^{1024}$

單精度實數乘法運算

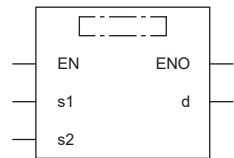
E*(P)



對單精度實數進行乘法運算。

梯形圖	ST*1
	ENO:=EMULTI (EN, s1, s2, d); ENO:=EMULTIP (EN, s1, s2, d);

FBD/LD



(□中放入EMULTI、EMULTIP。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

執行條件

指令	執行條件
E*	
E*P	

設置數據

內容、範圍、資料類型

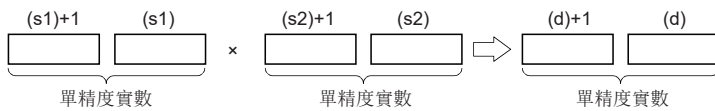
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	被乘法運算資料或儲存了被乘法運算資料的起始元件	$0, 2^{-126} \leq (s1) < 2^{128}$	單精度實數	ANYREAL_32
(s2)	乘法運算資料或儲存了乘法運算資料的起始元件	$0, 2^{-126} \leq (s2) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	○	○	○	○	—	○	—	—	—
(s2)	—	—	○	○	○	○	○	—	○	—	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—	—

功能

- 對(s1)中指定的單精度實數與(s2)中指定的單精度實數進行乘法運算，將乘法運算結果儲存到(d)中指定的元件中。



- (s1)、(s2)、(d)中可以指定及儲存 $0, 2^{-126} \leq |指定值(儲存值)| < 2^{128}$ 的值。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

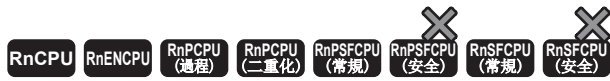
☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s1)中指定的資料為-0、非正規化數、非數、±∞時。
	(s2)中指定的資料為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生上溢時。) $ d < 2^{128}$

單精度實數除法運算

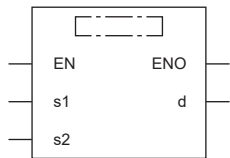
E/(P)



對單精度實數進行除法運算。

梯形圖	ST*1
	ENO:=EDIVISION(EN, s1, s2, d); ENO:=EDIVISIONP(EN, s1, s2, d);

FBD/LD



(□中放入EDIVISION、EDIVISIONP。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
E/	
E/P	

設置數據

■內容、範圍、資料類型

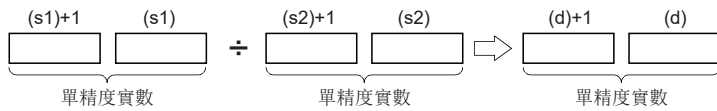
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被除法運算資料或儲存了被除法運算資料的起始元件	$0, 2^{-126} \leq (s1) < 2^{128}$	單精度實數	ANYREAL_32
(s2)	除法運算資料或儲存了除法運算資料的起始元件	$0, 2^{-126} \leq (s2) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—

功能

- 對 (s1) 中指定的單精度實數與 (s2) 中指定的單精度實數進行除法運算，將除法運算結果儲存到 (d) 中指定的元件中。



- (s1)、(s2)、(d) 中可以指定及儲存 $0, 2^{-126} \leq |指定值(儲存值)| < 2^{128}$ 的值。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3400H	當 (s2) 中指定的資料(除數)為0時。
3402H	(s1) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。 (s2) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) $ d < 2^{128}$

雙精度實數乘法運算

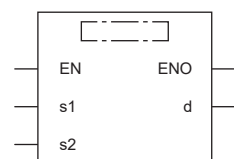
ED*(P)



對雙精度實數進行乘法運算。

梯形圖	ST*1
	ENO:=EDMULTI (EN, s1, s2, d) ; ENO:=EDMULTIP (EN, s1, s2, d) ;

FBD/LD



(□中放入EDMULTI、EDMULTIP。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
ED*	
ED*P	

設置數據

■內容、範圍、資料類型

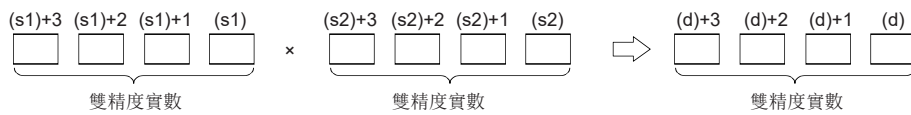
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	被乘法運算資料或儲存了被乘法運算資料的起始元件	$0, 2^{-1022} \leq (s1) < 2^{1024}$	雙精度實數	ANYREAL_64
(s2)	乘法運算資料或儲存了乘法運算資料的起始元件	$0, 2^{-1022} \leq (s2) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	—	—	○	—	○	—	—	—
(s2)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

功能

- 對(s1)中指定的雙精度實數與(s2)中指定的雙精度實數進行乘法運算，將乘法運算結果儲存到(d)中指定的元件中。



- (s1)、(s2)、(d)中可以指定及儲存 $0, 2^{-1022} \leq | \text{指定值(儲存值)} | < 2^{1024}$ 的值。
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s1)中指定的資料為-0、非正規化數、非數、±∞時。
	(s2)中指定的資料為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生上溢時。) $ (d) < 2^{1024}$

雙精度實數除法運算

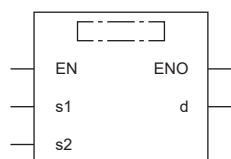
ED/(P)



對雙精度實數進行除法運算。

梯形圖	ST*1
	ENO:=EDDIVISION(EN, s1, s2, d); ENO:=EDDIVISIONP(EN, s1, s2, d);

FBD/LD



(□中放入EDDIVISION、EDDIVISIONP。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
ED/	
ED/P	

■設置數據

■內容、範圍、資料類型

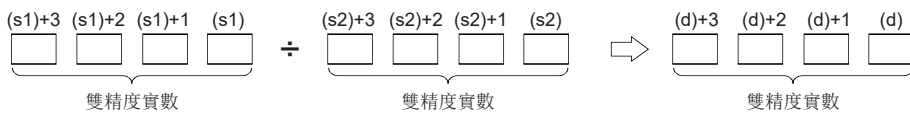
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	被除法運算資料或儲存了被除法運算資料的起始元件	$0, 2^{-1022} \leq (s1) < 2^{1024}$	雙精度實數	ANYREAL_64
(s2)	除法運算資料或儲存了除法運算資料的起始元件	$0, 2^{-1022} \leq (s2) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	—	—	○	—	○	—	—	—
(s2)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

功能

- 對(s1)中指定的雙精度實數與(s2)中指定的雙精度實數進行除法運算，將除法運算結果儲存到(d)中指定的元件中。



- (s1)、(s2)、(d)中可以指定及儲存 $0, 2^{-1022} \leq | \text{指定值(儲存值)} | < 2^{1024}$ 的值。
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

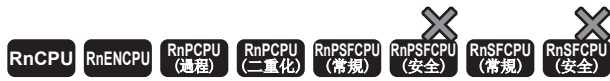
☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3400H	當(s2)中指定的資料(除數)為0時。
3402H	(s1)中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。 (s2)中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當(d)超出下述範圍時。(發生上溢時。) $ (d) < 2^{1024}$

帶符號BIN16位元資料→單精度實數轉換

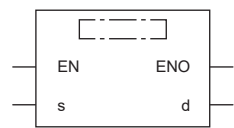
INT2FLT(P)



將帶符號BIN16位元資料轉換為單精度實數。

梯形圖	ST*1
	ENO:=INT2FLT(EN, s, d); ENO:=INT2FLTP(EN, s, d);

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
INT2FLT	
INT2FLTP	

設置數據

■內容、範圍、資料類型

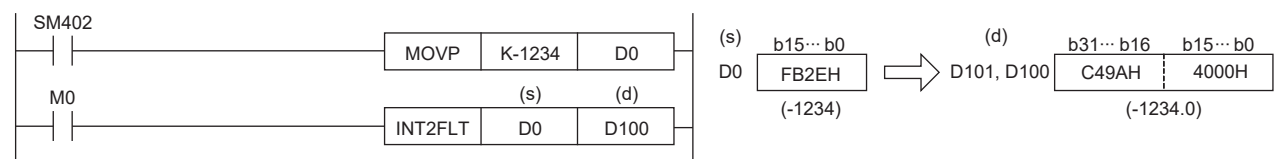
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行單精度實數轉換的整數資料或儲存了整數資料的元件	-32768~32767	帶符號BIN16位元	ANY16_S
(d)	儲存轉換後單精度實數的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	○	○	○	○	—	—	○	—	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—

功能

- 將(s)中指定的帶符號BIN16位元資料轉換為單精度實數後，儲存到(d)中。

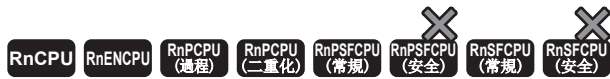


出錯

沒有運算出錯。

無符號BIN16位元資料→單精度實數轉換

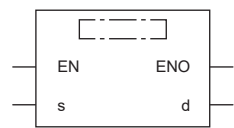
UINT2FLT (P)



將無符號BIN16位元資料轉換為單精度實數。

梯形圖	ST*1
	ENO:=UINT2FLT (EN, s, d) ; ENO:=UINT2FLTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
UINT2FLT	
UINT2FLTP	

設置數據

■內容、範圍、資料類型

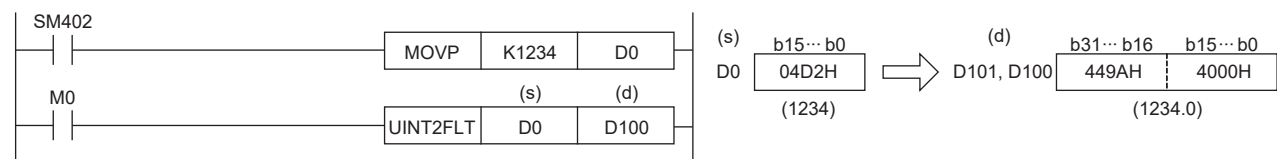
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行單精度實數轉換的整數資料或儲存了整數資料的元件	0~65535	無符號BIN16位元	ANY16_U
(d)	儲存轉換後單精度實數的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	○	○	○	○	—	—	○	—	—	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—	—

功能

- 將(s)中指定的無符號BIN16位元資料轉換為單精度實數後，儲存到(d)中。



出錯

沒有運算出錯。

帶符號BIN32位元資料→單精度實數轉換

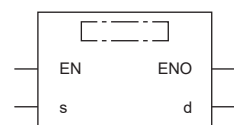
DINT2FLT (P)



將帶符號BIN32位元資料轉換為單精度實數。

梯形圖	ST*1
	ENO:=DINT2FLT (EN, s, d) ; ENO:=DINT2FLTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DINT2FLT	
DINT2FLTP	

設置數據

■內容、範圍、資料類型

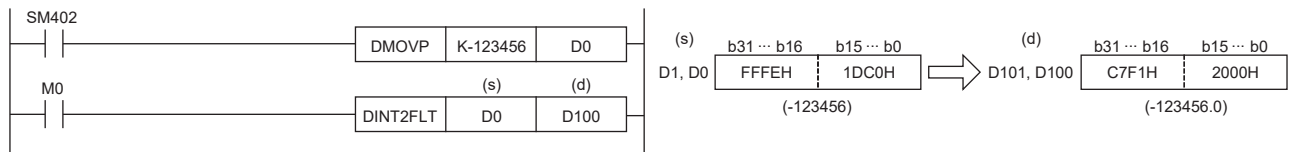
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行單精度實數轉換的整數資料或儲存了整數資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
(d)	儲存轉換後單精度實數的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

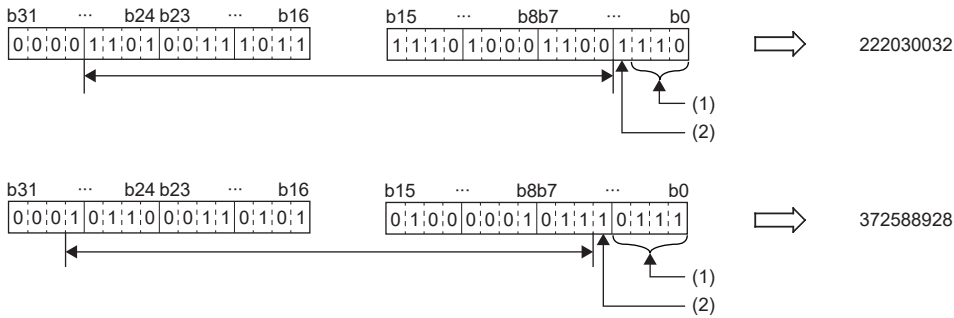
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 將(s)中指定的帶符號BIN32位元資料轉換為單精度實數後，儲存到(d)中。



- 因為單精度實數是以32位元的單精度進行處理，因此以2進制數表示時有效位數為24位元、以10進制數表示時約為7位數。因此，整數值超過了-16777216~16777215(24位BIN值)的範圍的情況下，轉換後的值將產生誤差。對於轉換結果，將整數值的高位開始的第25位元進行0捨1入、將第26位元及其以後捨去。



(1): 被捨去。

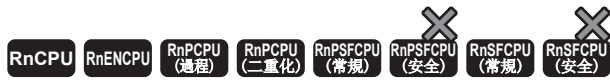
(2): 0捨1入。

出錯

沒有運算出錯。

無符號BIN32位元資料→單精度實數轉換

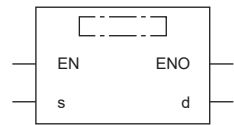
UDINT2FLT (P)



將無符號BIN32位元資料轉換為單精度實數。

梯形圖	ST*1
	ENO:=UDINT2FLT (EN, s, d) ; ENO:=UDINT2FLTP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
UDINT2FLT	
UDINT2FLTP	

設置數據

■內容、範圍、資料類型

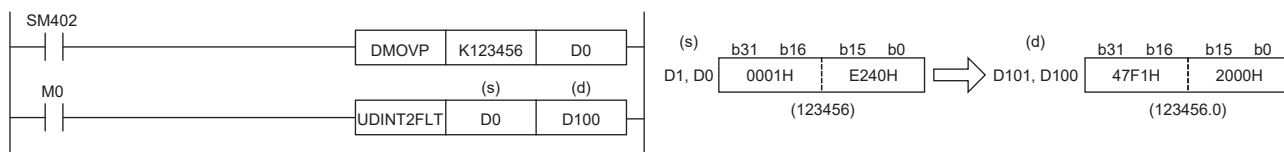
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行單精度實數轉換的整數資料或儲存了整數資料的起始元件	0~4294967295	無符號BIN32位元	ANY32_U
(d)	儲存轉換後單精度實數的起始元件	—	單精度實數	ANYREAL_32

■可使用的元件

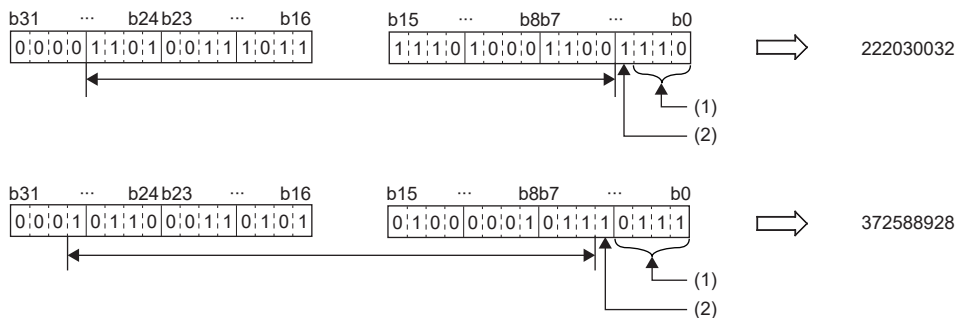
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—	—

功能

- 將(s)中指定的無符號BIN32位元資料轉換為單精度實數後，儲存到(d)中。



- 因為單精度實數是以32位元的單精度進行處理，因此以2進制數表示時有效位數為24位元、以10進制數表示時約為7位數。因此，整數值超過了0~16777215 (24位BIN值)的範圍的情況下，轉換後的值將產生誤差。對於轉換結果，將整數值的高位開始的第25位元進行0捨1入、將第26位元及其以後捨去。



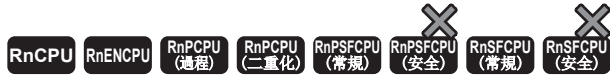
- (1): 被捨去。
(2): 0捨1入。

出錯

沒有運算出錯。

雙精度實數→單精度實數轉換

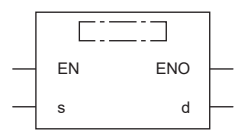
DBL2FLT(P)



將雙精度實數轉換為單精度實數。

梯形圖	ST*1
	ST^{*1} ENO:=DBL2FLT(EN, s, d); ENO:=DBL2FLTP(EN, s, d);

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

執行條件

指令	執行條件
DBL2FLT	
DBL2FLTP	

設置數據

內容、範圍、資料類型

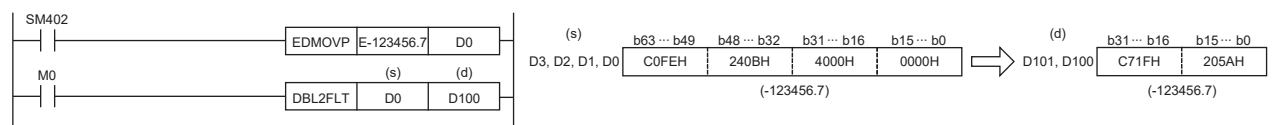
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	轉換為單精度實數的雙精度實數資料或儲存了雙精度實數資料的起始元件	$0, 2^{-1022} \leq s < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存轉換後單精度實數的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	—	—
(d)	—	—	○	—	○	○	○	○	—	—	—	—

功能

- 將(s)中指定的雙精度實數轉換為單精度實數後，儲存到(d)中。



- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

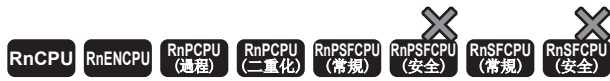
☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) $ d < 2^{128}$

帶符號BIN16位元資料→雙精度實數轉換

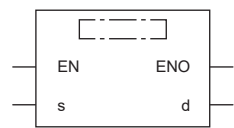
INT2DBL (P)



將帶符號BIN16位元資料轉換為雙精度實數。

梯形圖	ST*1
	ENO:=INT2DBL (EN, s, d) ; ENO:=INT2DBLP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
INT2DBL	
INT2DBLP	

設置數據

■內容、範圍、資料類型

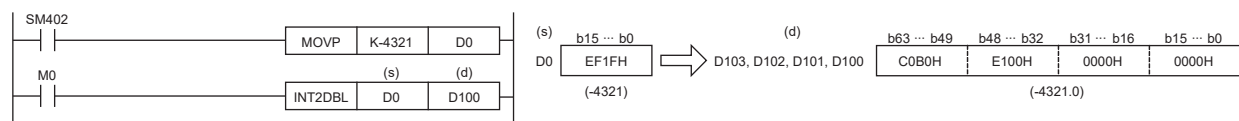
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行雙精度實數轉換的整數資料或儲存了整數資料的元件	-32768~32767	帶符號BIN16位元	ANY16_S
(d)	儲存轉換後雙精度實數的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	\$	
(s)	○	—	○	—	○	—	○	○	—	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 將(s)中指定的帶符號BIN16位元資料轉換為雙精度實數後，儲存到(d)中。

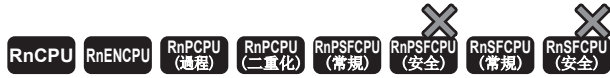


出錯

沒有運算出錯。

無符號BIN16位元資料→雙精度實數轉換

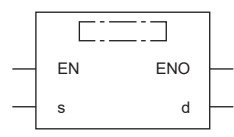
UINT2DBL (P)



將無符號BIN16位元資料轉換為雙精度實數。

梯形圖	ST*1
	ENO:=UINT2DBL (EN, s, d) ; ENO:=UINT2DBLP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
UINT2DBL	
UINT2DBLP	

設置數據

■內容、範圍、資料類型

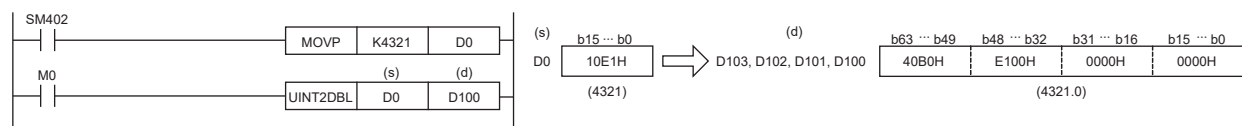
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	進行雙精度實數轉換的整數資料或儲存了整數資料的元件	0~65535	無符號BIN16位元	ANY16_U
(d)	儲存轉換後雙精度實數的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	\$	
(s)	○	—	○	—	○	—	○	○	—	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 將(s)中指定的無符號BIN16位元資料轉換為雙精度實數後，儲存到(d)中。

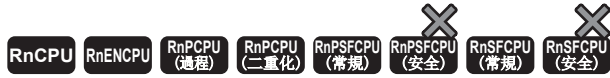


出錯

沒有運算出錯。

帶符號BIN32位元資料→雙精度實數轉換

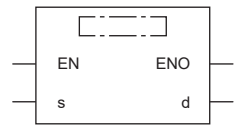
DINT2DBL (P)



將帶符號BIN32位元資料轉換為雙精度實數。

梯形圖	ST*1
	ENO:=DINT2DBL(EN, s, d); ENO:=DINT2DBLP(EN, s, d);

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DINT2DBL	
DINT2DBLP	

設置數據

■內容、範圍、資料類型

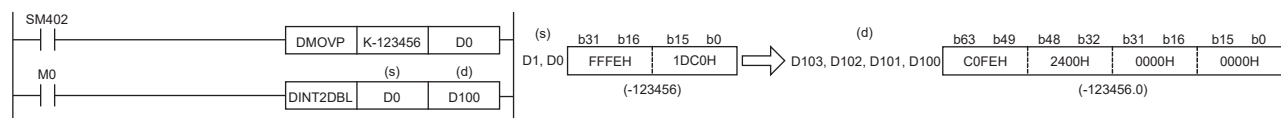
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行雙精度實數轉換的整數資料或儲存了整數資料的起始元件	-2147483648~2147483647	帶符號BIN32位元	ANY32_S
(d)	儲存轉換後雙精度實數的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	—	○	—	○	○	○	○	—	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 將(s)中指定的帶符號BIN32位元資料轉換為雙精度實數後，儲存到(d)中。

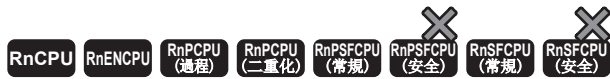


出錯

沒有運算出錯。

無符號BIN32位元資料→雙精度實數轉換

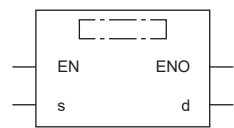
UDINT2DBL (P)



將無符號BIN32位元資料轉換為雙精度實數。

梯形圖	ST*1
	ENO:=UDINT2DBL (EN, s, d) ; ENO:=UDINT2DBLP (EN, s, d) ;

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
UDINT2DBL	
UDINT2DBLP	

設置數據

■內容、範圍、資料類型

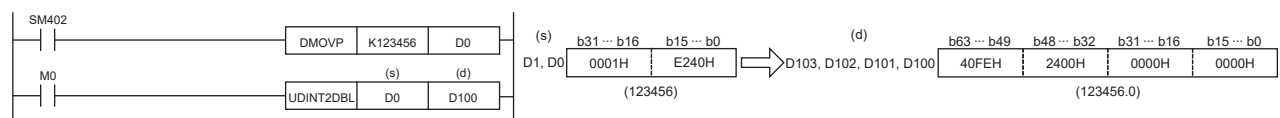
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行雙精度實數轉換的整數資料或儲存了整數資料的起始元件	0~4294967295	無符號BIN32位元	ANY32_U
(d)	儲存轉換後雙精度實數的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E		\$
(s)	○	—	○	—	○	○	○	○	○	—	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—	—

功能

- 將(s)中指定的無符號BIN32位元資料轉換為雙精度實數後，儲存到(d)中。

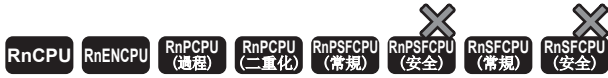


出錯

沒有運算出錯。

單精度實數→雙精度實數轉換

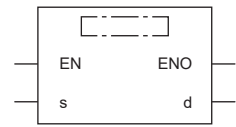
FLT2DBL (P)



將單精度實數轉換為雙精度實數。

梯形圖	ST*1
	$ENO := FLT2DBL(EN, s, d);$ $ENO := FLT2DBLP(EN, s, d);$

FBD/LD



*1 在版本為“1.035M”及其以後的工程工具中對應。

執行條件

指令	執行條件
FLT2DBL	
FLT2DBLP	

設置數據

內容、範圍、資料類型

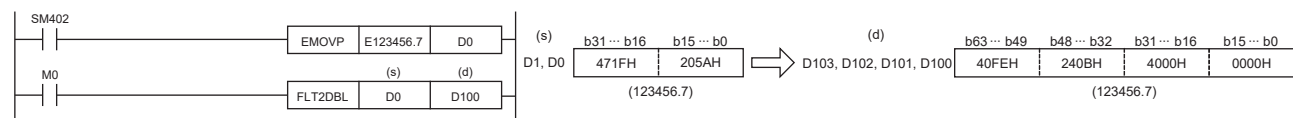
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	轉換為雙精度實數的單精度實數資料或儲存了單精度實數資料的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存轉換後雙精度實數的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	○	○	○	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—	—

功能

將(s)中指定的單精度實數轉換為雙精度實數後，儲存到(d)中。

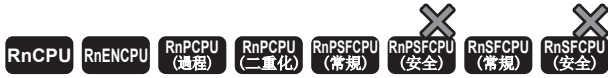


出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。

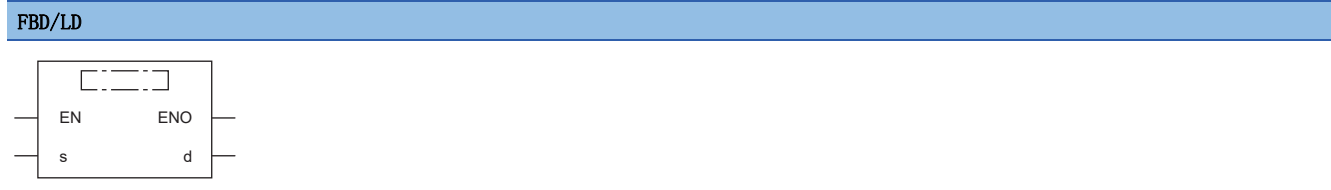
字元串→單精度實數轉換

EVAL (P)



將字元串轉換為單精度實數。

梯形圖	ST
	ENO:=EVAL (EN, s, d) ; ENO:=EVALP (EN, s, d) ;



■執行條件

指令	執行條件
EVAL	
EVALP	

設置數據

■內容、範圍、資料類型

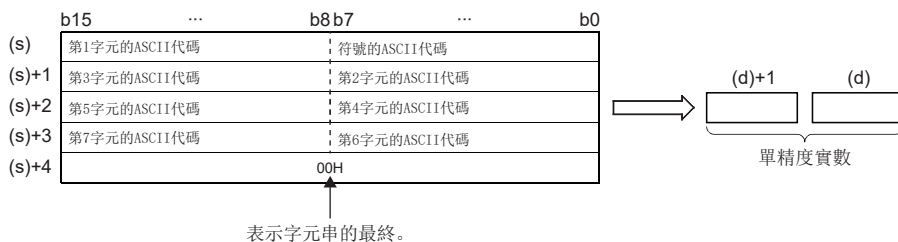
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行單精度實數資料轉換的字元元串資料或儲存了字元元串資料的起始元件	—	字元串	ANYSTRING_SINGLE
(d)	儲存轉換後單精度實數資料的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—	
(d)	—	—	○	○	○	○	○	○	—	—	—	—	

功能

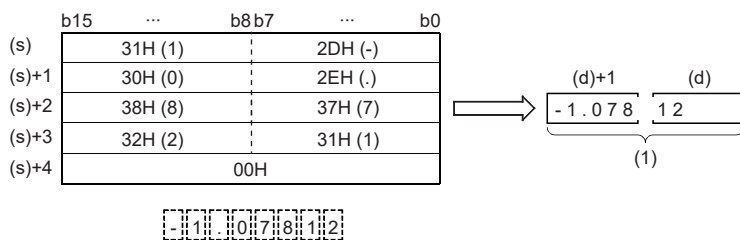
- 將(s)中指定的元件編號及其以後中儲存的字元元串轉換為單精度實數後，儲存到(d)中指定的元件中。
- 無論指定的字元串為小數點形式還是指數形式均可轉換為單精度實數。



- 字元串中包含有20H(空格)的情況下，將忽略20H進行轉換。
- 字元串最多可設定24字元。字元串中的20H(空格)、30H(0)也作為1字元計數。

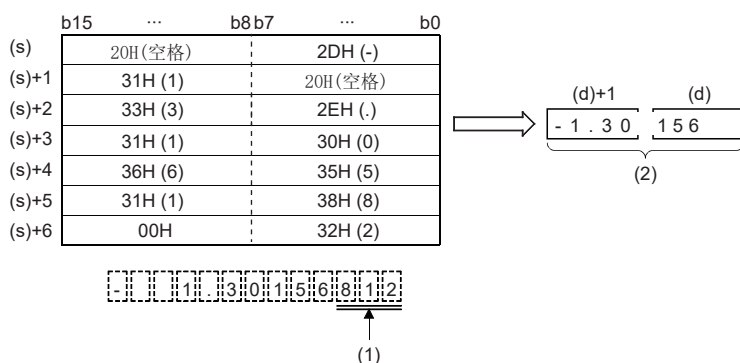
■小數點形式的情況下

- 若(s)中指定的字元串為小數點的形式時，執行情況將如下所示。



(1): 單精度實數

- (s)中指定的字元串中，對於進行單精度實數轉換的字元串，除去符號、小數點、指數部後的6位有效，第7位及其以後在轉換時將被捨去。

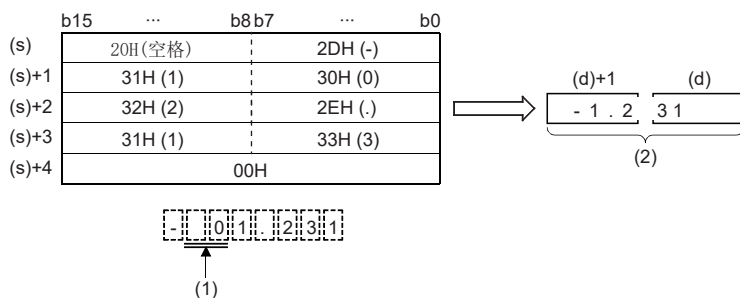


(1): 被捨去。

(2): 單精度實數

- 在小數點形式中將符號指定為2BH(+)或省略符號時，將作為正值進行轉換。此外，將符號指定為2DH(-)時，將作為負值進行轉換。

- (s)中指定的字元串中，最初的0以外的數值之間存在有20H(空格)或30H(0)的情況下，將忽略20H、30H進行轉換。

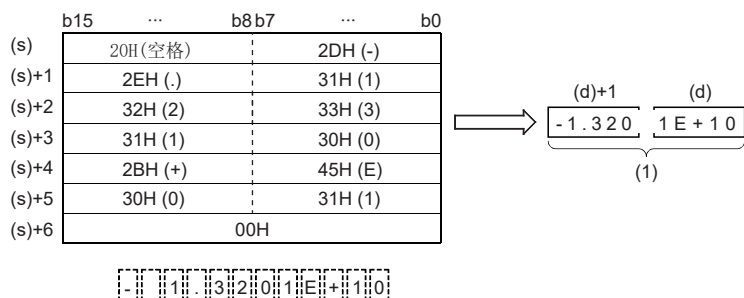


(1): 忽視。

(2): 單精度實數

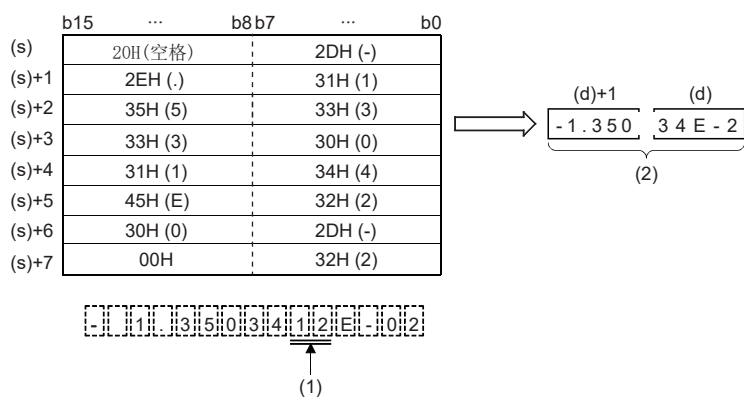
■指數形式的情況下

- (s) 中指定的字元串為指數形式的情況下，按下述方式執行。



(1): 單精度實數

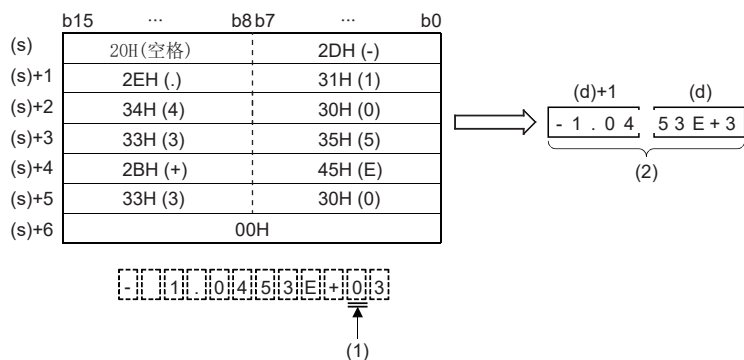
- (s) 中指定的字元串中，對於進行單精度實數轉換的字元串，除去符號、小數點、指數部後的6位有效，第7位及其以後在轉換時將被捨去。



(1): 被捨去。

(2): 單精度實數

- 在指數形式中將指數部的符號指定為2BH (+) 或省略符號時，將作為正值進行轉換。將指數部的符號指定為2DH (-) 時，將作為負值進行轉換。
- (s) 中指定的字元串中，最初的0以外的數值之間存在有20H (空格) 或30H (0) 的情況下，將忽略20H、30H進行轉換。
- 在指數形式的字元串中，“E”與數值之間儲存有30H (0) 的情況下，轉換時將忽略30H。



(1): 忽視。

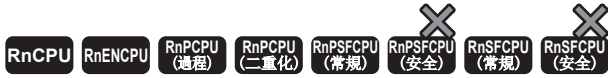
(2): 單精度實數

出錯

出錯代碼 (SD0)	內容
2820H	從 (s) 開始的相應元件範圍內沒有 00H 時。
3401H	(s) 中設定不能轉換的不正確資料時。 <ul style="list-style-type: none">• 整數部、小數部中存在有 30H (0) ~ 39H (9) 以外的字元• 指定的字元串中至少存在 2 個 2EH (.)• 指定的字元串的指數部中存在有 45H (E)、65H (e)、2BH (+)、2DH (-) 以外字元• 指定的字元串中存在有多個 45H (E)、65H (e) 指數部• 指定的字元串中，記載了 3 位或其以上的指數部數值• 指定的字元串中存在有多個 2BH (+)、2DH (-) 指數部• 指定的字元串中，在小數點形式的情況下正數部有多個 2BH (+)、2DH (-) 的符號，指數形式的情況下尾數部有多個 2BH (+)、2DH (-) 的符號• (s) 及其以後的字元數為 0 或超過 24 個字元時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) $ d < 2^{128}$

BCD格式資料→單精度實數資料轉換

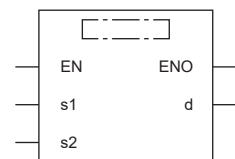
EREXP (P)



將BCD型浮點格式資料根據指定的小數位數轉換為單精度實數資料。

梯形圖	ST
	ENO:=EREXP (EN, s1, s2, d) ; ENO:=EREXPP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
EREXP	
EREXPP	

設置數據

■內容、範圍、資料類型

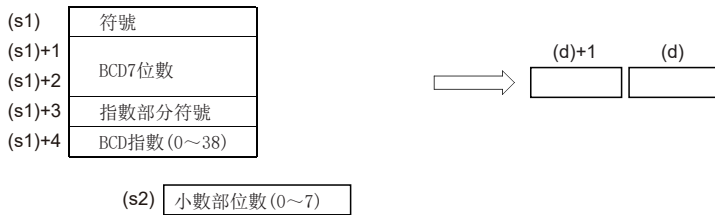
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存了BCD型浮點格式資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 5)
(s2)	小數位數資料	0~7	帶符號BIN16位元	ANY16
(d)	儲存單精度實數的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	—	—	○	○	○	○	○	○	—	—	—	—	

功能

- 將(s1)中指定的元件編號及其以後中儲存的BCD型浮點格式資料，按(s2)中指定的元件中儲存的小數部位數轉換為單精度實數資料後，儲存到(d)中指定的元件編號及其以後。

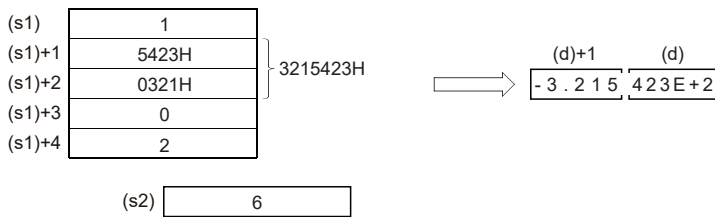


(s1): 符號 (正時0、負時1)
 (s1)+1、(s1)+2: BCD7位
 (s1)+3: 指數部符號 (正時0、負時1)
 (s1)+4: BCD指數(0~38)
 (s2): 小數部位數(0~7)
 (d)+1、(d): 單精度實數

- (s1)的符號及(s1)+3的指數部符號中，為正時設定0、為負時設定1。
- (s1)+4的BCD指數中可以設定0~38。
- (s2)的小數部位數中可以設定0~7。

例

在(s2)中設定了6的情況下

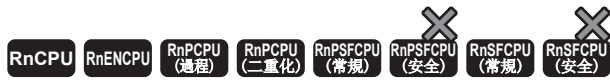


出錯

出錯代碼(SD0)	內容
3401H	(s1)中指定的形式指定為0、1以外時。
	(s1)+1、(s1)+2的各位中存在有0~9以外的值時。
	(s1)+3中指定的指數部符號為0、1以外時。
	(s1)+4中指定的指數資料為0~38以外時。
	(s2)中指定的小數部位數為0~7以外時。
3403H	當(d)超出下述範圍時。(發生上溢時。) $ d < 2^{128}$

單精度實數符號反轉

ENEG (P)



對單精度實數資料的符號進行反轉。

梯形圖	ST
	ENO:=ENEG(EN, d); ENO:=ENEGP(EN, d);

FBD/LD

■執行條件

指令	執行條件
ENEG	
ENEGP	

■設置數據

■內容、範圍、資料類型

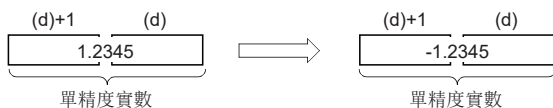
操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存了進行符號反轉的單精度實數資料的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(d)	—	—	○	○	○	○	○	○	—	—	—	—

■功能

- 對(d)中指定的元件的單精度實數的符號進行反轉後，儲存到(d)中指定的元件中。



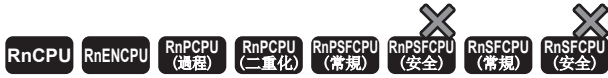
- 在對正負符號進行反轉時使用。

■出錯

出錯代碼(SD0)	內容
3402H	(d)中指定的內容為-0、非正規化數、非數、±∞時。

雙精度實數符號反轉

EDNEG (P)



對雙精度實數資料的符號進行反轉。

梯形圖	ST
	ENO:=EDNEG (EN, d) ; ENO:=EDNEGP (EN, d) ;

FBD/LD

■執行條件

指令	執行條件
EDNEG	
EDNEGP	

■設置數據

■內容、範圍、資料類型

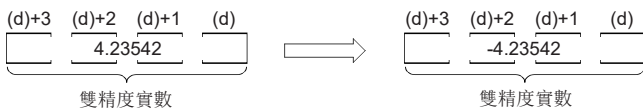
操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	儲存了進行符號反轉的雙精度實數資料的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(d)	—	—	○	—	—	○	—	○	—	—	—

■功能

- 對 (d) 中指定的元件的雙精度實數資料的符號進行反轉後，儲存到 (d) 中指定的元件中。



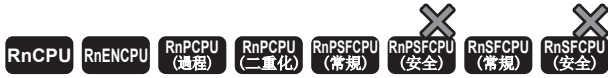
- 在對正負符號進行反轉時使用。

■出錯

出錯代碼 (SD0)	內容
3402H	(d) 中指定的內容為 -0、非正規化數、非數、±∞時。

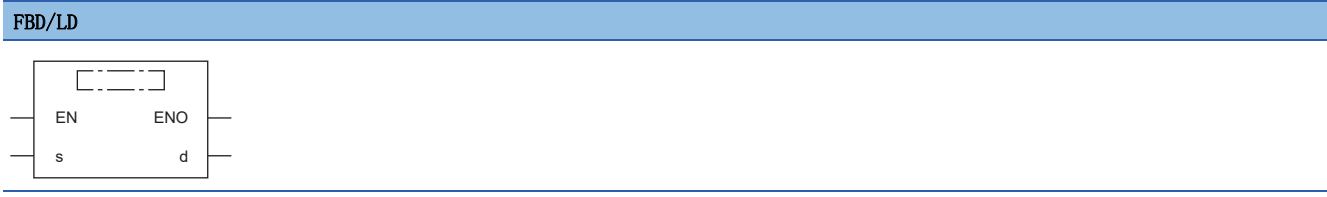
單精度實數資料傳送

EMOV (P)



將單精度實數資料傳送至指定的元件。

梯形圖	ST
	ENO:=EMOV (EN, s, d) ; ENO:=EMOVP (EN, s, d)



執行條件

指令	執行條件
EMOV	
EMOVP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	傳送資料或儲存了傳送資料的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存傳送目標資料的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—

功能

- 將 (s) 中指定的元件中儲存的單精度實數資料，傳送至 (d) 中指定的元件。

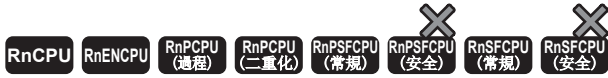


出錯

沒有運算出錯。

雙精度實數資料傳送

EDMOV (P)



將雙精度實數資料傳送至指定的元件。

梯形圖	ST
	ENO:=EDMOV (EN, s, d) ; ENO:=EDMOV (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
EDMOV	
EDMOV P	

■設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	傳送資料或儲存了傳送資料的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存傳送目標資料的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(s)	—	—	○	—	—	○	—	○	—	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

■功能

- 將 (s) 中指定的元件中儲存的雙精度實數資料，傳送至 (d) 中指定的元件。

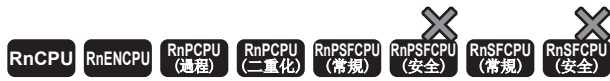


■出錯

沒有運算出錯。

單精度實數SIN運算

SIN(P)



對以單精度實數指定角度的SIN(正弦)值進行運算。

梯形圖	ST*1
	$ENO := SINP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援SIN指令。應使用通用函數的SIN。
 ↖ 1395頁 SIN(_E)

■執行條件

指令	執行條件
SIN	
SINP	

設置數據

■內容、範圍、資料類型

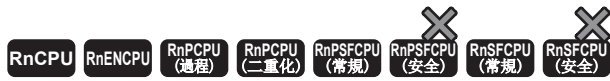
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行SIN(正弦)運算的角度資料或儲存了角度資料的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

單精度實數COS運算

COS (P)



對單精度實數中指定角度的COS(餘弦)值進行運算。

梯形圖	ST*1
	ENO:=COSP (EN, s, d) ;

FBD/LD*1

*1 在ST、FBD/LD中不支援COS指令。應使用通用函數的COS。
 ↖ 1396頁 COS (_E)

■執行條件

指令	執行條件
COS	
COSP	

設置數據

■內容、範圍、資料類型

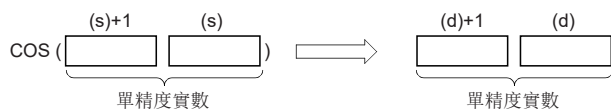
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行COS(餘弦)運算的角度資料或儲存了角度資料的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 對(s)中指定角度的COS(餘弦)值進行運算後，將運算結果儲存到(d)中指定的元件編號中。



- 對於(s)中指定的角度，以弧度單位(角度 $\times\pi\div 180$)進行設定。
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s)中指定的值為-0、非正規化數、非數、 $\pm\infty$ 時。

要點

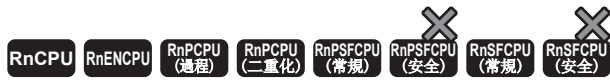
關於角度 \leftrightarrow 弧度轉換的有關內容，請參閱RAD(P)指令、DEG(P)指令。

☞ 933頁 RAD(P)

☞ 935頁 DEG(P)

單精度實數TAN運算

TAN(P)



對單精度實數中指定角度的TAN(正切)值進行運算。

梯形圖	ST*1
	$ENO := TANP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援TAN指令。應使用通用通用函數的TAN。
 ↖ 1397頁 TAN(_E)

■執行條件

指令	執行條件
TAN	
TANP	

設置數據

■內容、範圍、資料類型

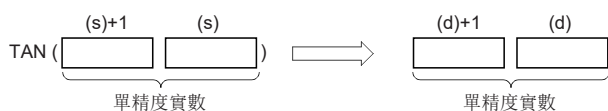
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行TAN(正切)運算的角度資料或儲存了角度資料的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s)	—	—	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—

功能

- 對(s)中指定角度的TAN(正切)值進行運算，將運算結果儲存到(d)中指定的元件編號中。



- 對於(s)中指定的角度，以弧度單位(角度 $\times\pi\div 180$)進行設定。
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

45頁 注意事項

注意事項

(s)中指定的角度為 $\pi/2$ 弧度、 $(3/2)\pi$ 弧度的情況下，因為弧度值中將產生運算誤差，因此不發生出錯。

出錯

出錯代碼(SD0)	內容
3402H	(s)中指定的值為-0、非正規化數、非數、 $\pm\infty$ 時。

要點

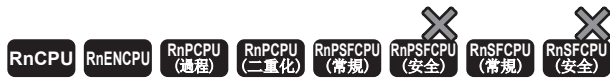
關於角度 \leftrightarrow 弧度轉換的有關內容，請參閱RAD(P)指令、DEG(P)指令。

933頁 RAD(P)

935頁 DEG(P)

單精度實數 SIN^{-1} 運算

ASIN(P)



透過單精度實數中指定的SIN值進行角度運算。

梯形圖	ST*1
	$\text{ENO} := \text{ASINP}(\text{EN}, \text{s}, \text{d});$

FBD/LD*1

*1 在ST、FBD/LD中不支援ASIN指令。應使用通用函數的ASIN。
 ↖ 1398頁 ASIN(_E)

執行條件

指令	執行條件
ASIN	
ASINP	

設置數據

內容、範圍、資料類型

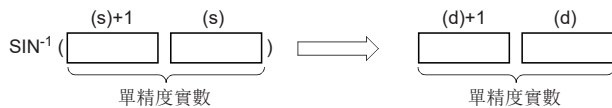
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行 SIN^{-1} (反正弦)運算的SIN值或儲存了SIN值的起始元件	-1.0~1.0	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

- 進行在 (s) 中指定角度的 SIN(正弦) 值角度運算後，將運算結果儲存到 (d) 中指定的元件編號中。



- (s) 中指定的 SIN 值可在 -1.0~1.0 的範圍內設定。
- (d) 中儲存的角度 (運算結果) 以弧度單位儲存。
- 運算結果為 -0 或發生下溢時，運算結果將變為 0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45 頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為 -0、非正規化數、非數、±∞ 時。
3405H	(s) 中指定的值超出 -1.0~1.0 的範圍時。

要點

關於弧度↔角度轉換，請參閱 RAD(P) 指令、DEG(P) 指令。

☞ 933 頁 RAD(P)

☞ 935 頁 DEG(P)

單精度實數 COS^{-1} 運算

ACOS (P)



透過單精度實數中指定的COS值進行角度運算。

梯形圖	ST*1
	$ENO := \text{ACOSP}(EN, s, d)$

FBD/LD*1

*1 在ST、FBD/LD中不支援ACOS指令。應使用通用函數的ACOS。
 ↖ 1399頁 ACOS (E)

■執行條件

指令	執行條件
ACOS	
ACOSP	

設置數據

■內容、範圍、資料類型

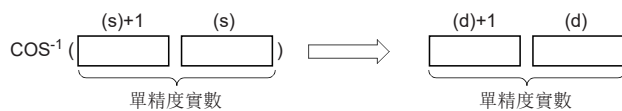
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行 COS^{-1} (反餘弦)運算的COS值或儲存了COS值的起始元件	-1.0~1.0	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	—	—	○	○	○	○	○	—	○	—	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—	—

功能

- 在進行(s)中指定的角度的COS(餘弦)值角度運算後，將運算結果儲存到(d)中指定的元件編號中。



- (s)中指定的COS值可在-1.0~1.0的範圍內設定。
- (d)中儲存的角度(運算結果)以弧度單位儲存。
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s)中指定的值為-0、非正規化數、非數、±∞時。
3405H	(s)中指定的值超出-1.0~1.0的範圍時。

要點

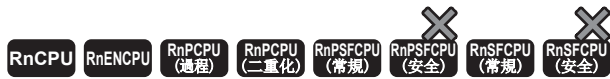
關於弧度↔角度轉換，請參閱RAD(P)指令、DEG(P)指令。

☞ 933頁 RAD(P)

☞ 935頁 DEG(P)

單精度實數 TAN^{-1} 運算

ATAN(P)



進行在單精度實數中指定的TAN值角度運算。

梯形圖	ST*1
	$ENO := ATANP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援ATAN指令。應使用通用函數的ATAN。
 ↖ 1400頁 ATAN(_E)

執行條件

指令	執行條件
ATAN	
ATANP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行 TAN^{-1} (反正切)運算的TAN值或儲存了TAN值的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	—	—	○	○	○	○	○	—	○	—	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—	—

雙精度實數SIN運算

SIND(P)



對雙精度實數中指定角度的SIN(正弦)值進行運算。

梯形圖	ST*1
	$ENO := SINDP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援SIND指令。應使用通用函數的SIN。

☞ 1395頁 SIN(_E)

■執行條件

指令	執行條件
SIND	
SINDP	

設置數據

■內容、範圍、資料類型

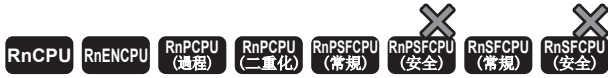
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行SIN(正弦)運算的角度資料或儲存了角度資料的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

雙精度實數COS運算

COSD(P)



對雙精度實數中指定角度的COS(餘弦)值進行運算。

梯形圖	ST*1
	$ENO := COSDP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援COSD指令。應使用通用函數的COS。
 ↖ 1396頁 COS(_E)

■執行條件

指令	執行條件
COSD	
COSDP	

設置數據

■內容、範圍、資料類型

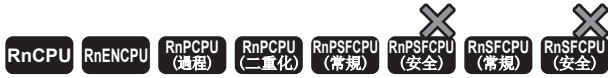
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行COS(餘弦)運算的角度資料或儲存了角度資料的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

雙精度實數TAN運算

TAND (P)



對雙精度實數中指定角度的TAN(正切)值進行運算。

梯形圖	ST*1
	$ENO := TANDP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援TAND指令。應使用通用通用函數的TAN。
 ↖ 1397頁 TAN(_E)

■執行條件

指令	執行條件
TAND	
TANDP	

設置數據

■內容、範圍、資料類型

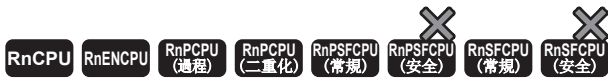
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行TAN(正切)運算的角度資料或儲存了角度資料的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

雙精度實數 SIN^{-1} 運算

ASIND(P)



進行以雙精度實數指定的SIN值角度運算。

梯形圖	ST*1
	$\text{ENO} := \text{ASINDP}(\text{EN}, \text{s}, \text{d})$

FBD/LD*1

*1 在ST、FBD/LD中不支援ASIND指令。應使用通用函數的ASIN。
 ↖ 1398頁 ASIN(_E)

■執行條件

指令	執行條件
ASIND	
ASINDP	

設置數據

■內容、範圍、資料類型

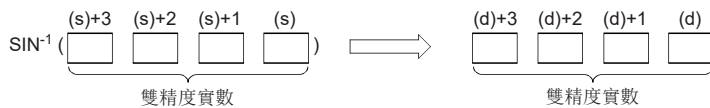
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行 SIN^{-1} (反正弦)運算的SIN值或儲存了SIN值的起始元件	-1.0~1.0	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—	
(d)	—	—	○	—	—	○	—	○	—	—	—	—	

功能

- 進行在 (s) 中指定角度的SIN(正弦)值角度運算後，將運算結果儲存到 (d) 中指定的元件編號中。



- (s) 中指定的SIN值可在-1.0~1.0的範圍內設定。
- (d) 中儲存的角度(運算結果)以弧度單位儲存。
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、±∞時。
3405H	(s) 中指定的值超出-1.0~1.0的範圍時。

要點

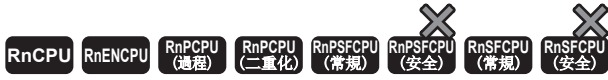
關於弧度↔角度轉換的有關內容，請參閱RADD(P)指令、DEGD(P)指令。

☞ 937頁 RADD(P)

☞ 939頁 DEGD(P)

雙精度實數 COS^{-1} 運算

ACOSD(P)



進行以雙精度實數指定的COS值角度運算。

梯形圖	ST*1
	$ENO := \text{ACOSDP}(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援ACOSD指令。應使用通用函數的ACOS。
 ↖ 1399頁 ACOS(_E)

■執行條件

指令	執行條件
ACOSD	
ACOSDP	

設置數據

■內容、範圍、資料類型

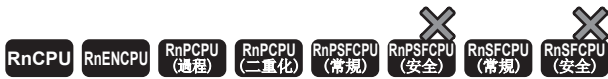
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行 COS^{-1} (反餘弦)運算的COS值或儲存了COS值的起始元件	-1.0~1.0	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—	—

雙精度實數 TAN^{-1} 運算

ATAND(P)



進行以雙精度實數指定的TAN值角度運算。

梯形圖	ST*1
	$ENO := ATANDP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援ATAND指令。應使用通用函數的ATAN。
 ↖ 1400頁 ATAN(_E)

■執行條件

指令	執行條件
ATAND	
ATANDP	

設置數據

■內容、範圍、資料類型

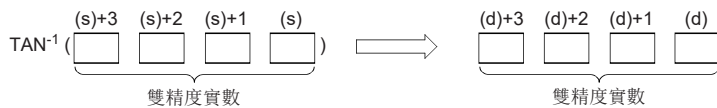
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行 TAN^{-1} (反正切)運算的TAN值或儲存了TAN值的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—	—

功能

- 進行在 (s) 中指定角度的 TAN (正切) 值角度運算後，將運算結果儲存到 (d) 中指定的元件編號中。



- (d) 中儲存的角度 (運算結果) 以弧度單位儲存。
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、±∞時。

要點

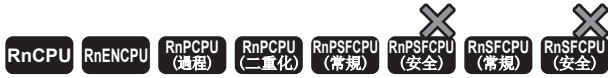
關於弧度↔角度轉換的有關內容，請參閱RADD(P)指令、DEGD(P)指令。

☞ 937頁 RADD(P)

☞ 939頁 DEGD(P)

BCD型SIN運算

BSIN(P)



對以BCD值指定的角度的SIN(正弦)值進行運算。

梯形圖	ST
	ENO:=BSIN(EN, s, d); ENO:=BSINP(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
BSIN	
BSINP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行SIN(正弦)運算的資料或儲存了資料的元件	0~9999	BCD4位	ANY16
(d)	儲存運算結果的起始元件	—	BCD4位	ANY16_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 對(s)中指定的值(角度)的SIN(正弦)值進行運算，將運算結果的符號儲存到(d)中指定的元件中，將運算結果儲存到(d)+1、(d)+2中指定的元件中。

$$\text{SIN}(s) = \overset{(d)}{\boxed{}} \overset{(d)+1}{\boxed{}} \overset{(d)+2}{\boxed{}}$$

(d): 符號

(d)+1: 整數部分

(d)+2: 小數部分

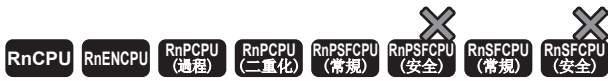
- 對於(s)中指定的值，在0~360°(DEG. 單位)的範圍內以BCD值進行設定。
- 對於(d)中儲存的運算結果的符號，運算結果為正時儲存0、為負時儲存1。
- (d)+1、(d)+2中儲存的運算結果為BCD值且在-1.000~1.000的範圍內。
- 對於運算結果，變為對小數部第5位進行了四捨五入後的值。

出錯

出錯代碼(SD0)	內容
3405H	當(s)中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">指定的資料不是BCD值指定的資料超出了0~360的範圍

BCD型COS運算

BCOS (P)



對以BCD值指定的角度的COS (餘弦) 值進行運算。

梯形圖	ST
	ENO:=BCOS (EN, s, d) ; ENO:=BCOSP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
BCOS	
BCOSP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	進行COS (餘弦) 運算的資料或儲存了資料的元件	0~9999	BCD4位	ANY16
(d)	儲存運算結果的起始元件	—	BCD4位	ANY16_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 對(s)中指定的值(角度)的COS(餘弦)值進行運算，將運算結果的符號儲存到(d)中指定的字元元件中、將運算結果儲存到(d)+1、(d)+2中指定的字元元件中。

$$\text{COS}(s) = \overset{(d)}{\boxed{}} \overset{(d)+1}{\boxed{}} \overset{(d)+2}{\boxed{}}$$

(d): 符號

(d)+1: 整數部分

(d)+2: 小數部分

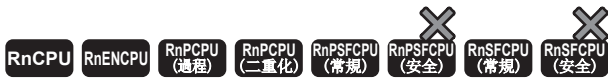
- 對於(s)中指定的值，在0~360°(DEG.單位)的範圍內以BCD值進行設定。
- 對於(d)中儲存的運算結果的符號，運算結果為正時儲存0、為負時儲存1。
- (d)+1、(d)+2中儲存的運算結果為BCD值且在-1.000~1.000的範圍內。
- 對於運算結果，變為對小數部第5位進行了四捨五入後的值。

出錯

出錯代碼(SD0)	內容
3405H	當(s)中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">指定的資料不是BCD值指定的資料超出了0~360的範圍

BCD型TAN運算

BTAN(P)



對以BCD值指定的角度的TAN(正切)值進行運算。

梯形圖	ST
	ENO:=BTAN(EN, s, d); ENO:=BTANP(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
BTAN	
BTANP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行TAN(正切)運算的資料或儲存了資料的元件	0~9999	BCD4位	ANY16
(d)	儲存運算結果的起始元件	—	BCD4位	ANY16_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 對(s)中指定的值(角度)的TAN(正切)值進行運算，將運算結果的符號儲存到(d)中指定的元件中、將運算結果儲存到(d)+1、(d)+2中指定的元件中。

$$\text{TAN}(s) = \boxed{\text{(d)}} \boxed{\text{(d)+1}} . \boxed{\text{(d)+2}}$$

(d): 符號

(d)+1: 整數部分

(d)+2: 小數部分

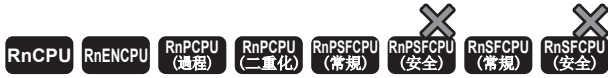
- 對於(s)中指定的值，在0~360°(DEG.單位)的範圍內以BCD值進行設定。
- 對於(d)中儲存的運算結果的符號，運算結果為正時儲存0、為負時儲存1。
- (d)+1、(d)+2中儲存的運算結果為BCD值且在-57.2901~57.2903的範圍內。
- 對於運算結果，變為對小數部第5位進行了四捨五入後的值。

出錯

出錯代碼(SD0)	內容
3405H	當(s)中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">指定的資料不是BCD值指定的資料超出了0~360的範圍指定的資料為90°、270°

BCD型 SIN^{-1} 運算

BASIN(P)



對以BCD值指定的值的 SIN^{-1} (反正弦)值進行運算。

梯形圖	ST
	ENO:=BASIN(EN, s, d); ENO:=BASINP(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
BASIN	
BASINP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存了進行 SIN^{-1} (反正弦)運算的資料的起始元件	0~9999	BCD4位	ANY16_ARRAY (要素數: 3)
(d)	儲存運算結果的元件	—	BCD4位	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	—	○	—	—	—	—

功能

• 對(s)中指定的值的 SIN^{-1} (反正弦)值進行運算，將運算結果(角度)儲存到(d)中指定的元件中。

$$SIN^{-1}(\overset{(s)}{\boxed{\quad}} \overset{(s)+1}{\boxed{\quad}} . \overset{(s)+2}{\boxed{\quad}}) = (d)$$

(s): 符號

(s)+1: 整數部分

(s)+2: 小數部分

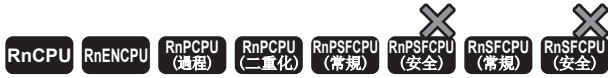
- (s)中設定進行運算的資料的符號。運算資料為正時儲存0、為負時儲存1。
- (s)+1、(s)+2中，將進行運算的資料的整數部、小數部分別以BCD值進行儲存。(可在0~1.0000的範圍內設定。)
- 對於(d)中儲存的運算結果，是以BCD值且在0~90°、270~360°(DEG.單位)的範圍內的值。
- 運算結果是對小數部進行了四捨五入後的值。

出錯

出錯代碼 (SD0)	內容
3405H	當 (s) 中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">• 指定的資料不是BCD值• 指定的資料超出了-1.0000~1.0000的範圍

BCD型 COS^{-1} 運算

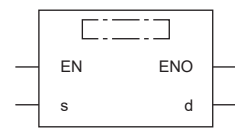
BACOS (P)



對以BCD值指定的值的 COS^{-1} (反餘弦)值進行運算。

梯形圖	ST
	ENO:=BACOS (EN, s, d) ; ENO:=BACOSP (EN, s, d) ;

FBD/LD



■執行條件

指令	執行條件
BACOS	
BACOSP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存了進行 COS^{-1} (反餘弦)運算的資料的起始元件	0~9999	BCD4位	ANY16_ARRAY (要素數: 3)
(d)	儲存運算結果的元件	—	BCD4位	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	—	○	—	—	—	—

功能

- 對(s)中指定的值的 COS^{-1} (反餘弦)值進行運算，將運算結果(角度)儲存到(d)中指定的元件中。

$$\text{COS}^{-1} \left(\begin{matrix} (s) \\ \boxed{} \\ (s)+1 \end{matrix} \begin{matrix} (s)+2 \\ \boxed{} \end{matrix} \right) = (d)$$

(s): 符號

(s)+1: 整數部分

(s)+2: 小數部分

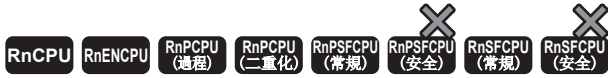
- (s)中設定進行運算的資料的符號。運算資料為正時儲存0、為負時儲存1。
- (s)+1、(s)+2中，將進行運算的資料的整數部、小數部分別以BCD值進行儲存。(可在0~1.0000的範圍內設定。)
- 對於(d)中儲存的運算結果，是以BCD值且在0~180°(DEG.單位)的範圍內的值。
- 運算結果是對小數部進行了四捨五入後的值。

出錯

出錯代碼 (SD0)	內容
3405H	當 (s) 中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">• 指定的資料不是BCD值• 指定的資料超出了-1.0000~1.0000的範圍

BCD型TAN⁻¹運算

BATAN(P)



對以BCD值指定的值的TAN⁻¹(反正切)值進行運算。

梯形圖	ST
	ENO:=BATAN(EN, s, d); ENO:=BATANP(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
BATAN	
BATANP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存了進行TAN ⁻¹ (反正切)運算的資料的起始元件	0~9999	BCD4位	ANY16_ARRAY (要素數: 3)
(d)	儲存運算結果的元件	—	BCD4位	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	—	○	—	—	—	—

功能

對(s)中指定值的TAN⁻¹(反正切)值進行運算後，將運算結果(角度)儲存到(d)中指定的元件中。

$$\text{TAN}^{-1} \left(\begin{matrix} \text{(s)} \\ \text{(s)+1} \\ \text{(s)+2} \end{matrix} \right) = \text{(d)}$$

(s): 符號

(s)+1: 整數部分

(s)+2: 小數部分

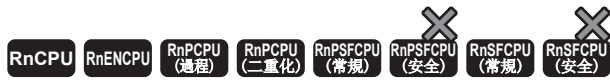
- (s)中設定進行運算的資料的符號。運算資料為正時儲存0、為負時儲存1。
- (s)+1、(s)+2中，將進行運算的資料的整數部、小數部分別以BCD值進行儲存。(可在0~9999.9999的範圍內設定。)
- 對於(d)中儲存的運算結果，是以BCD值且在0~90°、270~360°(DEG.單位)的範圍內的值。
- 運算結果是對小數部進行了四捨五入後的值。

出錯

出錯代碼 (SD0)	內容
3405H	(s) 中指定的資料不是BCD值時。

單精度實數角度→弧度轉換

RAD(P)



將角度的大小單位從以單精度實數指定的度單位轉換為弧度單位。

梯形圖	ST
	ENO:=RAD(EN, s, d); ENO:=RADP(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
RAD	
RADP	

■設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行弧度單位轉換的角度或儲存了角度的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存進行了弧度單位轉換的值的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s)	—	—	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—

功能

- 將角度的大小單位從 (s) 中指定的度單位 (DEG. 單位) 轉換為弧度單位後，儲存到 (d) 中指定的元件編號中。



- 度單位→弧度單位轉換按下述方式進行。

$$\text{弧度單位} = \text{度單位} \times \frac{\pi}{180}$$

- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

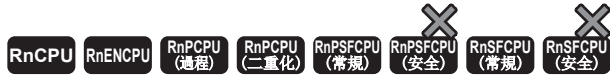
☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、±∞時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) $ (d) < 2^{128}$

單精度實數弧度→角度轉換

DEG (P)



將角度的大小單位從以單精度實數指定的弧度單位轉換為度單位。

梯形圖	ST
	ENO:=DEG (EN, s, d) ; ENO:=DEGP (EN, s, d) ;

FBD/LD

執行條件

指令	執行條件
DEG	
DEGP	

設置數據

內容、範圍、資料類型

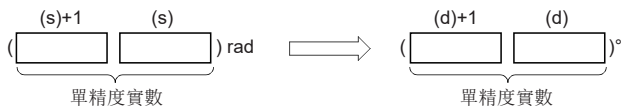
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行度單位轉換的弧度角度或儲存了弧度角度的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存進行了度單位轉換的值的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 將角度的大小單位從 (s) 中指定的弧度單位轉換為度單位 (DEG. 單位) 後，儲存到 (d) 中指定的元件編號中。



- 弧度單位→度單位轉換按下述方式進行。

$$\text{度單位} = \text{弧度單位} \times \frac{180}{\pi}$$

- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、±∞時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) $ d < 2^{128}$

雙精度實數角度→弧度轉換

RADD (P)



將角度的大小單位從以雙精度實數指定的度單位轉換為弧度單位。

梯形圖	ST
	ENO:=RADD (EN, s, d) ; ENO:=RADDP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
RADD	
RADDP	

設置數據

■內容、範圍、資料類型

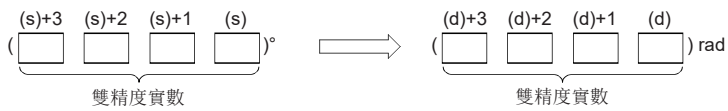
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行弧度單位轉換的角度或儲存了角度的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存進行了弧度單位轉換的值的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

功能

- 將角度的大小單位從 (s) 中指定的度單位 (DEG. 單位) 轉換為弧度單位後，儲存到 (d) 中指定的元件編號中。



- 度單位→弧度單位轉換按下述方式進行。

$$\text{弧度單位} = \text{度單位} \times \frac{\pi}{180}$$

- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

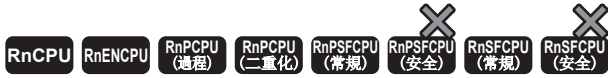
☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、±∞時。
3403H	當 (d) 超出下述範圍時。(發生上溢時。) (d) < 2 ¹⁰²⁴

雙精度實數弧度→角度轉換

DEGD (P)



將角度的大小單位從以雙精度實數指定的弧度單位轉換為度單位。

梯形圖	ST
	ENO:=DEGD (EN, s, d) ; ENO:=DEGDP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
DEGD	
DEGDP	

設置數據

■內容、範圍、資料類型

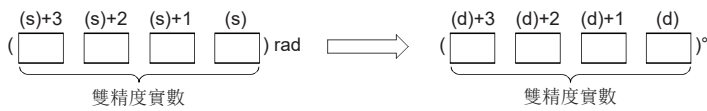
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行度單位轉換的弧度角度或儲存了弧度角度的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存進行了度單位轉換的值的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(s)	—	—	○	—	—	○	—	○	—	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 將角度的大小單位從(s)中指定的弧度單位轉換為度單位(DEG. 單位)後，儲存到(d)中指定的元件編號中。



- 弧度單位→度單位轉換按下述方式進行。

$$\text{度單位} = \text{弧度單位} \times \frac{180}{\pi}$$

- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

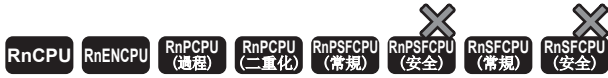
☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s)中指定的值為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生上溢時。) $ d < 2^{1024}$

單精度實數平方根

ESQRT(P)



對以單精度實數指定的值的平方根進行運算。

梯形圖	ST
	ENO:=ESQRT(EN, s, d); ENO:=ESQRTP(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
ESQRT	
ESQRTP	

■設置數據

■內容、範圍、資料類型

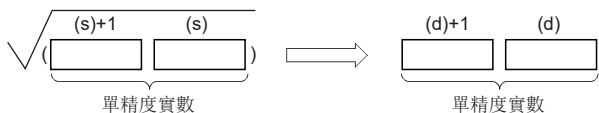
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行平方根運算的資料或儲存了資料的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

■功能

- 對(s)中指定的值的平方根進行運算後，將運算結果儲存到(d)中指定的元件編號中。



- (s)中指定的值只能設定為正的數。(不能以負的數進行運算。)
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

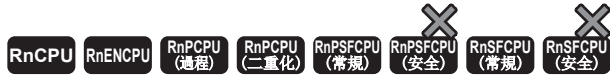
📖 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、 $\pm\infty$ 時。
3405H	(s) 中指定的值為負的數時。

雙精度實數平方根

EDSQRT (P)



對以雙精度實數指定的值的平方根進行運算。

梯形圖	ST
	ENO:=EDSQRT (EN, s, d) ; ENO:=EDSQRTP (EN, s, d) ;

FBD/LD

執行條件

指令	執行條件
EDSQRT	
EDSQRTP	

設置數據

內容、範圍、資料類型

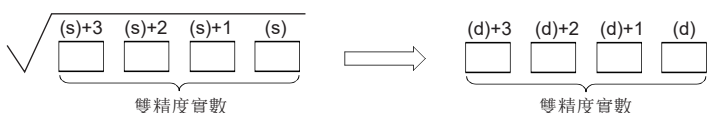
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	進行平方根運算的資料或儲存了資料的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

功能

對 (s) 中指定的值的平方根進行運算後，將運算結果儲存到 (d) 中指定的元件編號中。



- (s) 中指定的值只能設定為正的數。(不能以負的數進行運算。)
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

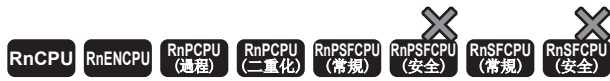
☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、 $\pm\infty$ 時。
3405H	(s) 中指定的值為負的數時。

單精度實數指數運算

EXP (P)



對以單精度實數指定的值的指數進行運算。

梯形圖	ST*1
	$ENO := EXPP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援EXP指令。應使用通用函數的EXP。
 ↖ 1394頁 EXP(_E)

■執行條件

指令	執行條件
EXP	
EXPP	

設置數據

■內容、範圍、資料類型

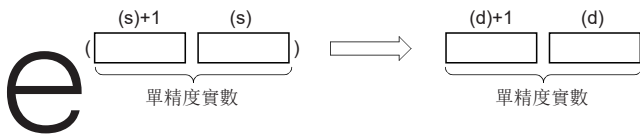
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行指數運算的資料或儲存了資料的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—	
(d)	—	—	○	○	○	○	○	○	—	—	—	—	

功能

- 進行(s)中指定的值的指數運算後，將運算結果儲存到(d)中指定的元件中。



- 在指數運算中，將底(e)作為“2.71828”進行運算。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s)中指定的值為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生上溢時。) $ (d) < 2^{128}$

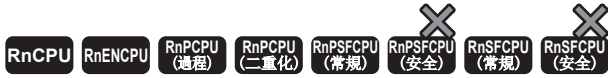
要點

- EXP(P)指令以自然對數進行運算。求出常用對數的值的狀況下，應將常用對數用0.43429相除後的值指定到(s)中。

$$10^X = e^{\frac{X}{0.43429}}$$

雙精度實數指數運算

EXPDP (P)



對以雙精度實數指定的值的指數進行運算。

梯形圖	ST*1
	$ENO := EXPDP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援EXPDP指令。應使用通用函數的EXP。
 ↖ 1394頁 EXP(_E)

■執行條件

指令	執行條件
EXPDP	
EXPDP	

設置數據

■內容、範圍、資料類型

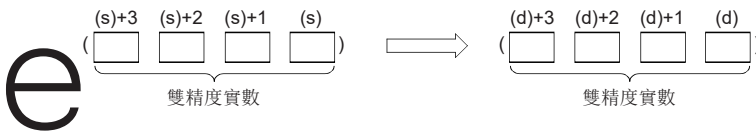
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行指數運算的資料或儲存了資料的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—	
(d)	—	—	○	—	—	○	—	○	—	—	—	—	

功能

- 進行(s)中指定的值的指數運算後，將運算結果儲存到(d)中指定的元件中。



- 在指數運算中，將底(e)作為“2.71828”進行運算。
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s)中指定的值為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生上溢時。) $ d < 2^{1024}$

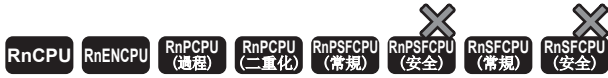
要點

- 在EXPD(P)指令中，以自然對數進行運算。求出常用對數的值的的情況下，應將常用對數用0.43429相除後的值指定到(s)中。

$$10^X = e^{\frac{X}{0.43429}}$$

單精度實數自然對數運算

LOG (P)



對以單精度實數指定的值的自然對數 (e) 為底時的對數進行運算。

梯形圖	ST*1
	$ENO := LOGP(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援LOG指令。應使用通用函數的LOG。
 ↖ 1392頁 LOG (E)

■執行條件

指令	執行條件
LOG	
LOGP	

設置數據

■內容、範圍、資料類型

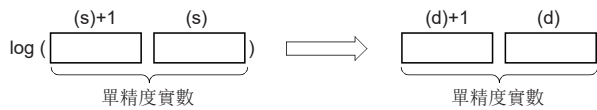
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行自然對數運算的資料或儲存了資料的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

- 對(s)中指定的值的自然對數e為底時的對數進行運算後，將運算結果儲存到(d)中指定的元件中。



- (s)中指定的值只能設定為正的數。(不能以負的數進行運算。)
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s)中指定的值為-0、非正規化數、非數、±∞時。
3405H	當(s)中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">指定的值為負數指定的值為0

雙精度實數自然對數運算

LOGD (P)



對以雙精度實數指定的值的自然對數 (e) 為底時的對數進行運算。

梯形圖	ST*1
	$ENO := LOGD(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支援LOGD指令。應使用通用函數的LOG。

☞ 1392頁 LOG (E)

執行條件

指令	執行條件
LOGD	
LOGDP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	進行自然對數運算的資料或儲存了資料的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—	—

功能

- 對(s)中指定的值的自然對數e為底時的對數進行運算後，將運算結果儲存到(d)中指定的元件中。



- (s)中指定的值只能設定為正的數。(不能以負的數進行運算。)
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

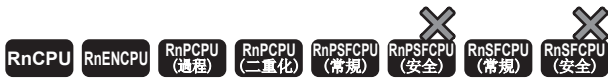
☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s)中指定的值為-0、非正規化數、非數、 $\pm\infty$ 時。
3405H	當(s)中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">指定的值為負數指定的值為0

BCD4位平方根

BSQRT (P)



對以BCD值(4位)指定的值的平方根進行運算。

梯形圖	ST
	ENO:=BSQRT (EN, s, d) ; ENO:=BSQRTP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
BSQRT	
BSQRTP	

■設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行平方根運算的資料或儲存了資料的元件	0~9999	BCD4位	ANY16
(d)	儲存運算結果的起始元件	—	BCD4位	ANY16_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

■功能

對(s)中指定的BCD4位資料的平方根進行運算後，將運算結果儲存到(d)中指定的元件中。

$$\sqrt{\boxed{(s)}} = \boxed{(d)} . \boxed{(d)+1}$$

(d): 整數部分
 (d)+1: 小數部分

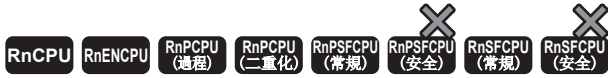
- (s)中指定的值是BCD值且最大4位(0~9999)。
- (d)的運算結果將分別以BCD值儲存0~9999.9999。
- 運算結果將變為捨去了小數部第5位後的值。

出錯

出錯代碼 (SD0)	內容
3405H	(s) 中指定的資料為BCD值以外時。

BCD8位平方根

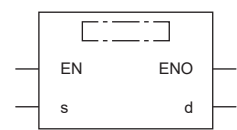
BDSQRT (P)



對以BCD值 (8位) 指定的值的平方根進行運算。

梯形圖	ST
	ENO:=BDSQRT (EN, s, d) ; ENO:=BDSQRTP (EN, s, d) ;

FBD/LD



■執行條件

指令	執行條件
BDSQRT	
BDSQRTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	進行平方根運算的資料或儲存了資料的起始元件	0~99999999	BCD8位	ANY32
(d)	儲存運算結果的起始元件	—	BCD4位	ANY16_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

功能

- 對 (s) 中指定的BCD8位資料的平方根進行運算後，將運算結果儲存到 (d) 中指定的元件中。

$$\sqrt{\left(\begin{matrix} (s)+1 \\ (s) \end{matrix} \right)} = \begin{matrix} (d) \\ (d)+1 \end{matrix}$$

(s)+1、(s)：2字資料

(d)：整數部分

(d)+1：小數部分

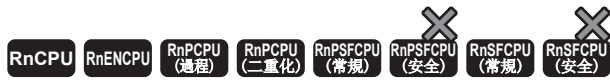
- (s) 中指定的值是BCD值且最大8位 (0~99999999)。
- (d) 的運算結果將分別以BCD值儲存0~9999.9999。
- 運算結果將變為捨去了小數部第5位後的值。

出錯

出錯代碼 (SD0)	內容
3405H	(s) 中指定的資料為BCD值以外時。

單精度實數幕運算

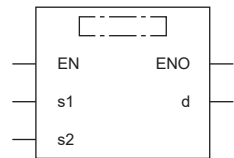
POW (P)



進行單精度實數的幕運算。

梯形圖	ST
	ENO:=POW (EN, s1, s2, d) ; ENO:=POWP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
POW	
POWP	

設置數據

■內容、範圍、資料類型

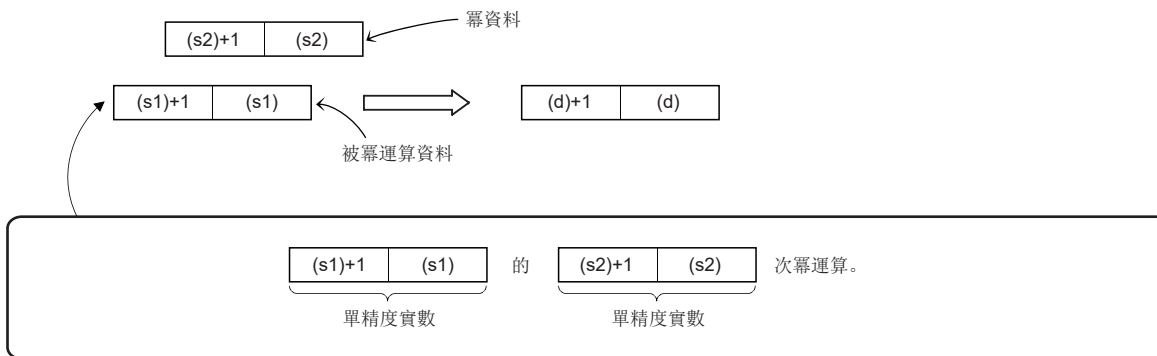
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	被幕運算資料或儲存了被幕運算資料的起始元件	$0, 2^{-126} \leq s1 < 2^{128}$	單精度實數	ANYREAL_32
(s2)	幕運算資料或儲存了資料的起始元件	$0, 2^{-126} \leq s2 < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 對(s1)中指定的單精度實數以(s2)中指定的單精度實數進行冪運算後，將運算結果儲存到(d)中指定的元件中。



- (s1)、(s2)中可指定的值及可儲存的值為0、 $2^{-126} \leq |設定值(儲存值)| < 2^{128}$ 。
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

☞ 45頁 注意事項

出錯

出錯代碼(SD0)	內容
3402H	(s1)或(s2)中指定的值為-0、非正規化數、非數、±∞時。

雙精度實數幕運算

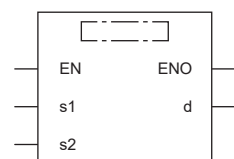
POWD (P)



進行雙精度實數的幕運算。

梯形圖	ST
	ENO:=POWD (EN, s1, s2, d) ; ENO:=POWDP (EN, s1, s2, d) ;

FBD/LD



■執行條件

指令	執行條件
POWD	
POWDP	

設置數據

■內容、範圍、資料類型

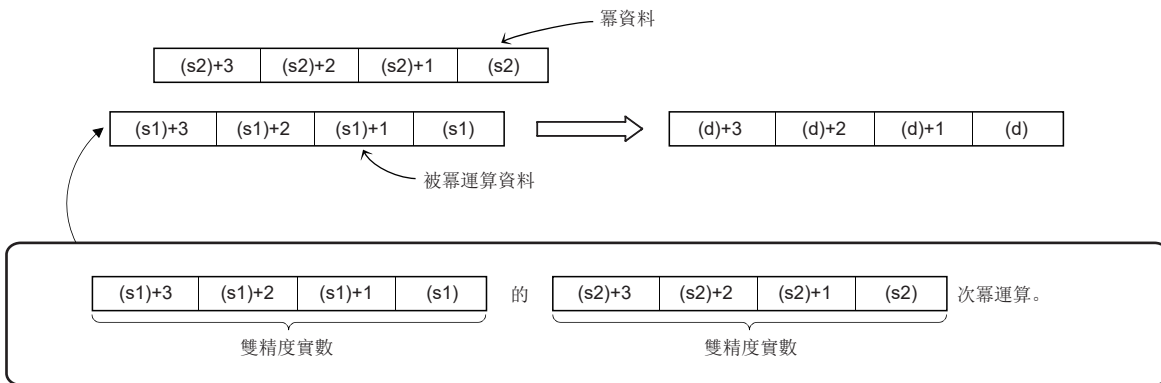
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	被幕運算資料或儲存了被幕運算資料的起始元件	$0, 2^{-1022} \leq s1 < 2^{1024}$	雙精度實數	ANYREAL_64
(s2)	幕運算資料或儲存了資料的起始元件	$0, 2^{-1022} \leq s2 < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	○	—	○	—	○	—	—
(s2)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 將 (s1) 中指定的雙精度實數以 (s2) 中指定的雙精度實數進行冪運算、將運算結果儲存到 (d) 中指定的元件中。



- (s1)、(s2) 中可指定的值及可儲存的值為 0 、 $2^{-1022} \leq |設定值(儲存值)| < 2^{1024}$ 。
- 運算結果為 -0 或發生下溢時，運算結果將變為 0 。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

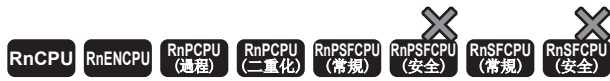
☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s1) 或 (s2) 中指定的值為 -0 、非正規化數、非數、 $\pm\infty$ 時。

單精度實數常用對數運算

LOG10(P)



對以單精度實數指定的值的常用對數(以10為底的對數)進行運算。

梯形圖	ST
	<pre>ENO:=LOG10(EN, s, d); ENO:=LOG10P(EN, s, d);</pre>

FBD/LD

■執行條件

指令	執行條件
LOG10	
LOG10P	

■設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行常用對數運算的資料或儲存了資料的起始元件	$0, 2^{-126} \leq s < 2^{128}$	單精度實數	ANYREAL_32
(d)	儲存運算結果的起始元件	—	單精度實數	ANYREAL_32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—

■功能

- 對(s)中指定的值的常用對數(以10為底的對數)進行運算，將運算結果儲存到(d)中指定的元件編號中。



- (s)中指定的值只能設定為正的數。(不能以負的數進行運算。)
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

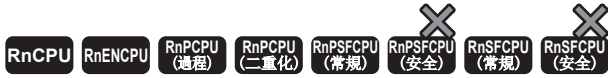
☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、 $\pm\infty$ 時。
3405H	當 (s) 中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">• 指定的值為負數• 指定的值為0

雙精度實數常用對數運算

LOG10D(P)



對以雙精度實數指定的值的常用對數(以10為底的對數)進行運算。

梯形圖	ST
	ENO:=LOG10D(EN, s, d); ENO:=LOG10DP(EN, s, d);

FBD/LD

執行條件

指令	執行條件
LOG10D	
LOG10DP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行常用對數運算的資料或儲存了資料的起始元件	$0, 2^{-1022} \leq s < 2^{1024}$	雙精度實數	ANYREAL_64
(d)	儲存運算結果的起始元件	—	雙精度實數	ANYREAL_64
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

功能

- 對(s)中指定的值的常用對數(以10為底的對數)進行運算，將運算結果儲存到(d)中指定的元件編號中。

$$\log_{10} \left(\underbrace{(s+3)_1 (s+2)_2 (s+1)_1 (s)}_{\text{雙精度實數}} \right) \implies \underbrace{((d+3)_1 (d+2)_2 (d+1)_1 (d))}_{\text{雙精度實數}}$$

- (s)中指定的值只能設定為正的數。(不能以負的數進行運算。)
- 運算結果為-0或發生下溢時，運算結果將變為0。
- 在透過工程工具設定輸入值的情況下，有可能產生化整誤差。關於透過工程工具設定輸入值情況下的注意事項，請參閱下述內容。

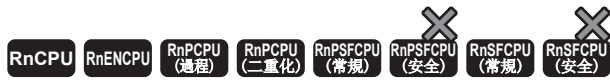
☞ 45頁 注意事項

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、 $\pm\infty$ 時。
3405H	當 (s) 中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">• 指定的值為負數• 指定的值為0

單精度實數最大值搜尋

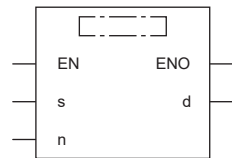
EMAX (P)



對單精度實數的塊資料的最大值進行搜尋。

梯形圖	ST*1
	$ENO := EMAX(EN, s, n, d);$

FBD/LD*1



*1 在ST、FBD/LD中不支援EMAX指令。應使用通用函數的MAX。
 1430頁 MAX(_E)、MIN(_E)

■執行條件

指令	執行條件
EMAX	
EMAXP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行最大值搜尋的單精度實數或儲存了單精度實數的起始元件	$0, 2^{-126} \leq (s) < 2^{128}$	單精度實數	ANYREAL_32*1
(d)	儲存搜尋結果的起始元件 (d)~(d)+1: 最大值 (d)+2: 位置 (d)+3: 個數	—	單精度實數	—*2 (ANY_REAL_32_ARRAY)
(n)	單精度實數塊資料的點數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

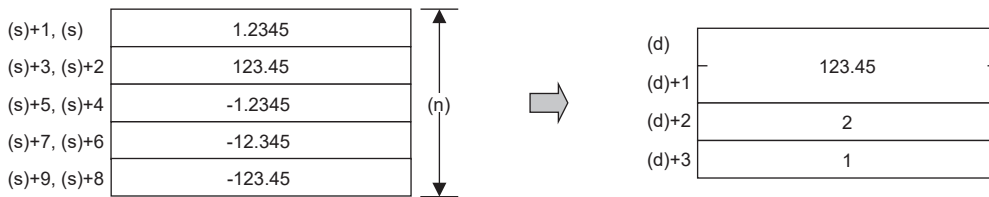
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。
 *2 無論使用的程式語言為何，透過元件指定。請勿指定標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

功能

- 對從(s)開始的(n)點的單精度實數的塊資料進行最大值搜尋後，儲存到(d)中指定的元件中。最先檢測到最大值的資料將從(s)開始的第幾點儲存到(d)+2中指定的元件中，並將與最大值相同值的個數儲存到(d)+3中指定的元件中。
- 關於搜尋結果(位置)的值，將(s)中指定的塊資料的起始作為第1點處理。



(d)、(d)+1: 最大值

(d)+2: 位置

(d)+3: 個數

- (n)中指定的值為0的情況下將變為無處理。

出錯

出錯代碼(SD0)	內容
3402H	(s)中指定的塊資料中存在有單精度實數以外的值的情況下。

雙精度實數最大值搜尋

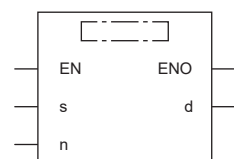
EDMAX (P)



對雙精度實數的塊資料的最大值進行搜尋。

梯形圖	ST*1
	$ENO := EDMAXP(EN, s, n, d);$

FBD/LD*1



*1 在ST、FBD/LD中不支援EDMAX指令。應使用通用函數的MAX。

☞ 1430頁 MAX(_E)、MIN(_E)

■執行條件

指令	執行條件
EDMAX	
EDMAXP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行最大值搜尋的雙精度實數或儲存了雙精度實數的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64*1
(d)	儲存搜尋結果的起始元件 (d) ~ (d)+3: 最大值 (d)+4: 位置 (d)+5: 個數	—	雙精度實數	—*2 (ANY_REAL_64_ARRAY)
(n)	雙精度實數塊資料的點數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

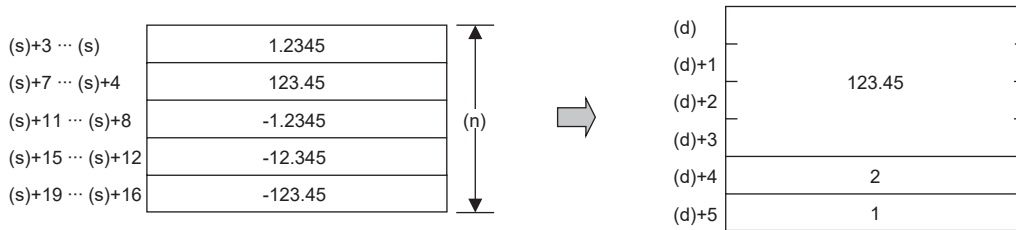
*2 無論使用的程式語言為何，透過元件指定。請勿指定標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

功能

- 對於(s)開始的(n)點的雙精度實數的塊資料，進行最大值搜尋後，儲存到(d)中指定的元件中。最先檢測到最大值的資料將從(s)開始的第幾點儲存到(d)+4中指定的元件中，並將與最大值相同值的個數儲存到(d)+5中指定的元件中。
- 關於搜尋結果(位置)的值，將(s)中指定的塊資料的起始作為第1點處理。



(d)、(d)+1、(d)+2、(d)+3: 最大值

(d)+4: 位置

(d)+5: 個數

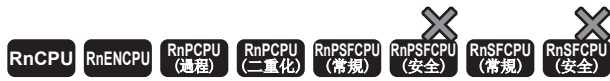
- (n)中指定的值為0的情況下將變為無處理。

出錯

出錯代碼(SD0)	內容
3402H	(s)中指定的塊資料中存在有雙精度實數以外的值的情況下。

單精度實數最小值搜尋

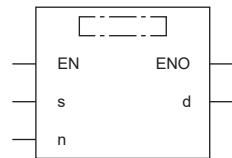
EMIN(P)



對單精度實數的塊資料的最小值進行搜尋。

梯形圖	ST*1
	$ENO := EMINP(EN, s, n, d);$

FBD/LD*1



*1 在ST、FBD/LD中不支援EMIN指令。應使用通用函數的MIN。
 1430頁 MAX(_E)、MIN(_E)

■執行條件

指令	執行條件
EMIN	
EMINP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行最小值搜尋的單精度實數或儲存了單精度實數的起始元件	$0, 2^{-126} \leq s < 2^{128}$	單精度實數	ANYREAL_32*1
(d)	儲存搜尋結果的起始元件 (d)~(d)+1: 最小值 (d)+2: 位置 (d)+3: 個數	—	單精度實數	—*2 (ANY_REAL_32_ARRAY)
(n)	單精度實數塊資料的點數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

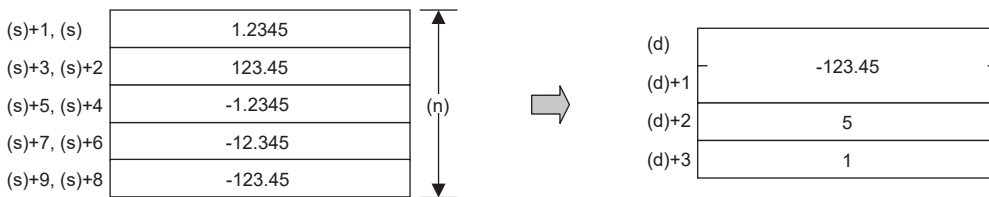
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。
 *2 無論使用的程式語言為何，透過元件指定。請勿指定標籤。

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

功能

- 對於(s)開始的(n)點的單精度實數的塊資料，進行最小值搜尋後，儲存到(d)中指定的元件中。最先檢測到最小值的資料是將從(s)開始的第幾點儲存到(d)+2中指定的元件中，並將與最小值相同值的個數儲存到(d)+3中指定的元件中。
- 關於搜尋結果(位置)的值，將(s)中指定的塊資料的起始作為第1點處理。



(d)、(d)+1: 最小值

(d)+2: 位置

(d)+3: 個數

- (n) 中指定的值為0的情況下將變為無處理。

出錯

出錯代碼(SD0)	內容
3402H	(s) 中指定的塊資料中存在有單精度實數以外的值的情況下。

雙精度實數最小值搜尋

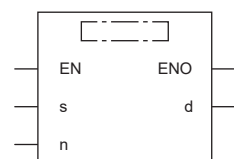
EDMIN(P)



對雙精度實數的塊資料的最小值進行搜尋。

梯形圖	ST*1
	$ENO := EDMINP(EN, s, n, d);$

FBD/LD*1



*1 在ST、FBD/LD中不支援EDMIN指令。應使用通用函數的MIN。
 1430頁 MAX(_E)、MIN(_E)

■執行條件

指令	執行條件
EDMIN	
EDMINP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	進行最小值搜尋的雙精度實數或儲存了雙精度實數的起始元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	雙精度實數	ANYREAL_64*1
(d)	儲存搜尋結果的起始元件 (d) ~ (d)+3: 最小值 (d)+4: 位置 (d)+5: 個數	—	雙精度實數	—*2 (ANY_REAL_64_ARRAY)
(n)	雙精度實數塊資料的點數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

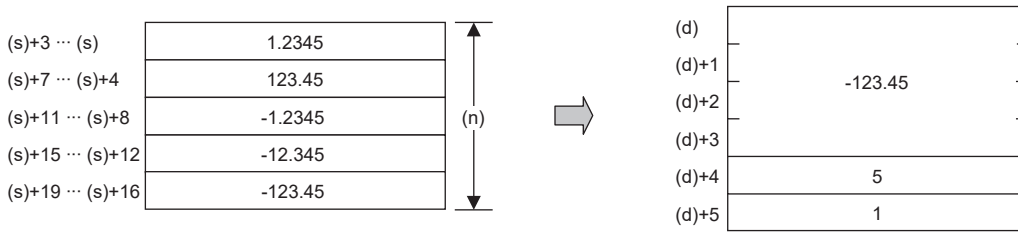
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。
 *2 無論使用的程式語言為何，透過元件指定。請勿指定標籤。

■可使用的元件

操作數	位元		字元				雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ	K、H		E	\$		
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 對於(s)開始的(n)點的雙精度實數的塊資料，進行最小值搜尋後，儲存到(d)中指定的元件中。最先檢測到最小值的資料是將從(s)開始的第幾點儲存到(d)+4中指定的元件中，並將與最小值相同值的個數儲存到(d)+5中指定的元件中。
- 關於搜尋結果(位置)的值，將(s)中指定的塊資料的起始作為第1點處理。



(d)、(d)+1、(d)+2、(d)+3: 最小值

(d)+4: 位置

(d)+5: 個數

- (n) 中指定的值為0的情況下將變為無處理。

出錯

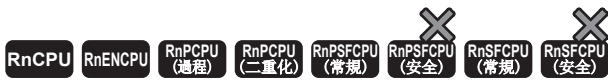
出錯代碼(SD0)	內容
3402H	(s) 中指定的塊資料中存在有雙精度實數以外的值的情況下。

12 隨機數

12.1 隨機數指令

隨機數發生

RND(P)



發生0~未滿32767的隨機數，儲存到指定的元件中。

梯形圖	ST
	ENO:=RND(EN, d); ENO:=RNDP(EN, d);

FBD/LD

■執行條件

指令	執行條件
RND	
RNDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存隨機數的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

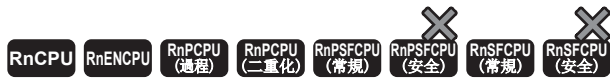
發生0~32767的隨機數，儲存到(d)中指定的元件中。RND(P)指令根據某個計算公式發生隨機數。在計算公式中，將上次的計算結果作為係數使用。

出錯

沒有運算出錯。

系列更改

SRND (P)



按照指定的元件中儲存的BIN16位元資料的內容更改隨機數系列。

梯形圖	ST
	ENO:=SRND (EN, s) ; ENO:=SRNDP (EN, s) ;

FBD/LD

■執行條件

指令	執行條件
SRND	
SRNDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	隨機數系列資料	-32768~32767	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	○	○	○	○	—	—	○	—	—	—

功能

按照 (s) 中指定的元件中儲存的BIN16位元資料的內容更改隨機數系列。SRND (P) 指令可以更改隨機數的計算模式。

出錯

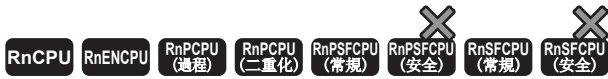
沒有運算出錯。

13 元件操作

13.1 變址寄存器指令

變址寄存器的批量儲存

ZPUSH(P)



將變址寄存器的內容儲存到指定的區域中。

梯形圖	ST
	ENO:=ZPUSH(EN, d); ENO:=ZPUSHP(EN, d);

FBD/LD

■執行條件

指令	執行條件
ZPUSH	
ZPUSHP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存變址寄存器的起始元件	—	帶符號BIN16位元	ANY16* ¹
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

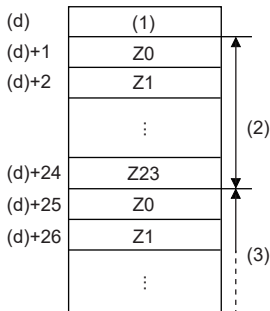
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(d)	—	—	○	—	—	—	○	—	—	—	—

功能

- 將變址寄存器的內容儲存到(d)中指定的元件或數組型標籤的要素及其以後。
- 儲存變址寄存器的內容時，儲存次數(d)將被+1。
- 與變址寄存器、超長變址寄存器點數的分配無關，儲存24字元。因此，變址寄存器為0點的情況下，將超長變址寄存器儲存12點。
- 資料恢復時，使用ZPOP(P)指令。透過成對使用ZPUSH(P)指令及ZPOP(P)指令，設定嵌套結構可以作為堆棧使用。

☞ 977頁 ZPOP(P)

- 透過將ZPUSH(P)指令的(d)中指定的區域，再次指定到ZPUSH(P)指令的(d)中，可以建立嵌套結構。每次執行ZPUSH(P)指令時，儲存次數(d)將被+1。
- 將嵌套結構以外的其它區域指定到ZPUSH(P)指令的(d)中的情況下，將變址寄存器的內容儲存到指定的其它區域中。
- 設定了嵌套結構的情況下，每次執行ZPUSH(P)指令時將添加儲存的資料，因此應預先預留出相當於使用次數的區域。
- (d)及其以後使用的區域的配置如下所示。



- (1): 儲存次數
(2): 第1嵌套(1嵌套分24字)
(3): 第2嵌套

出錯

出錯代碼(SD0)	內容
3405H	(d)+0的內容(儲存次數)為FFFF時。

變址寄存器的批量恢復

ZPOP (P)



將指定區域中儲存的資料讀取到變址寄存器中。

梯形圖	ST
	ENO:=ZPOP (EN, d) ; ENO:=ZPOPP (EN, d) ;

FBD/LD

■執行條件

指令	執行條件
ZPOP	
ZPOPP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	進行變址寄存器恢復的起始元件	—	帶符號BIN16位元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 將 (d) 中指定的元件或數組型標籤的要素及其以後儲存的資料讀取到變址寄存器中。
- 儲存的變址寄存器的內容被讀取時，儲存次數 (d) 將被-1。
- 關於 (d) 以後使用的區域配置，請參閱下述內容。

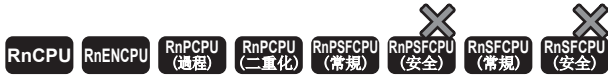
☞ 975頁 ZPUSH (P)

出錯

出錯代碼 (SD0)	內容
3405H	(d)+0的內容(儲存次數)為0時。

變址寄存器/超長變址寄存器選擇儲存

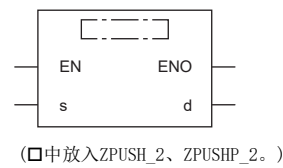
ZPUSH(P)



將變址寄存器、超長變址寄存器的內容儲存到指定的區域中。

梯形圖	ST
	<pre>ENO:=ZPUSH_2(EN, s, d); ENO:=ZPUSHHP_2(EN, s, d);</pre>

FBD/LD



■執行條件

指令	執行條件
ZPUSH	
ZPUSHHP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存的變址寄存器、超長變址寄存器類型	1~3	無符號BIN16位元	ANY16
(d)	儲存變址寄存器、超長變址寄存器的起始元件	—	帶符號BIN16位元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	—	○	—	○	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

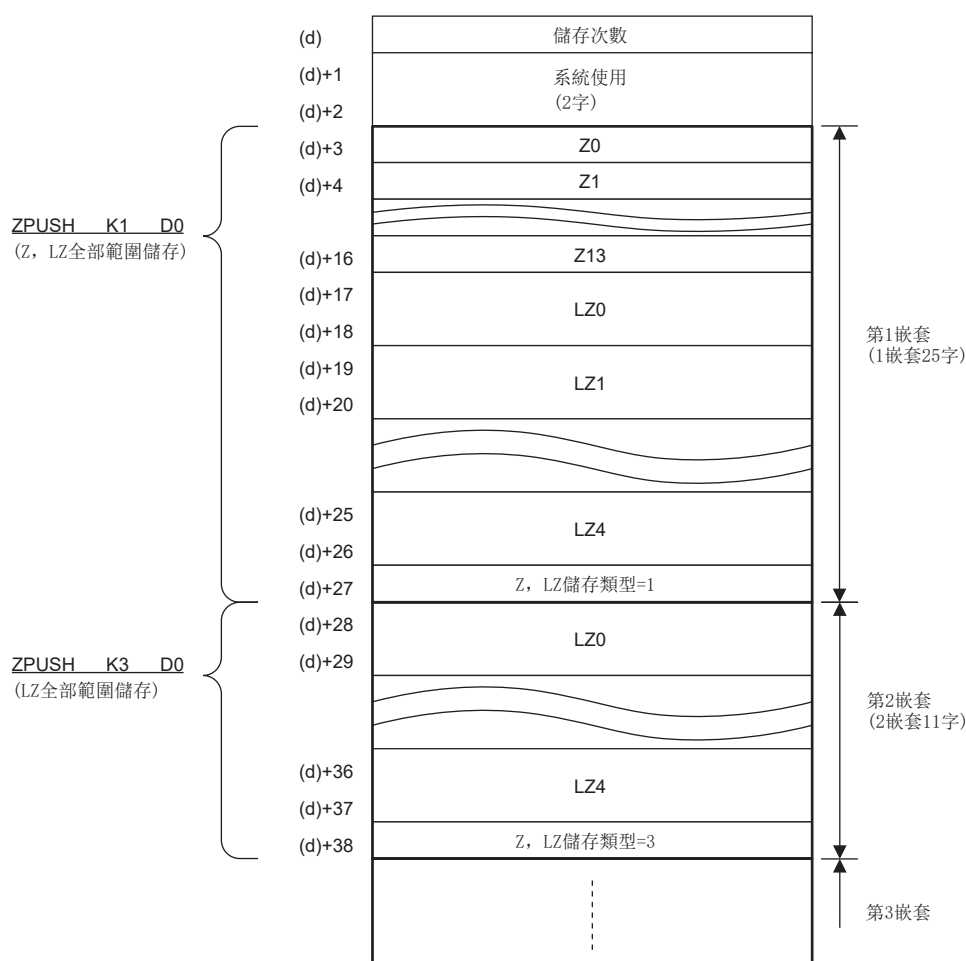
- 將(s)中指定的要儲存的變址寄存器、超長變址寄存器，儲存到(d)中指定的元件或數組型標籤的要素及其以後。此外，儲存的資料的末尾處將儲存已儲存的變址寄存器、超長變址寄存器的類型。
- 變址寄存器、超長變址寄存器的內容被儲存時儲存次數(d)將被+1。
- (s)中指定的值及儲存的變址寄存器、超長變址寄存器如下所示。

(s) 的值	儲存的Z、LZ
1	Z、LZ全範圍
2	Z全範圍
3	LZ全範圍

- 資料恢復時，使用變址寄存器/超長變址寄存器選擇恢復指令(ZPOP(P)指令)。在透過成對使用變址寄存器/超長變址寄存器選擇儲存指令(ZPUSH(P)指令)與變址寄存器/超長變址寄存器選擇恢復指令(ZPOP(P)指令)，設定嵌套結構可以作為堆棧使用。

☞ 981頁 ZPOP(P)

- 透過將變址寄存器/超長變址寄存器選擇儲存指令(ZPUSH(P)指令)的(d)中指定的區域，再次指定到變址寄存器/超長變址寄存器選擇儲存指令(ZPUSH(P)指令)的(d)中，可以建立嵌套結構。每次執行變址寄存器/超長變址寄存器選擇儲存指令(ZPUSH(P)指令)時，儲存次數(d)將被+1。
- 將嵌套結構以外的其它區域指定到變址寄存器/超長變址寄存器選擇儲存指令(ZPUSH(P)指令)的(d)中的情況下，將變址寄存器、超長變址寄存器的內容儲存到指定的其它區域中。
- 設定了嵌套結構的情況下，因為每次執行變址寄存器/超長變址寄存器選擇儲存指令(ZPUSH(P)指令)時將添加儲存的資料，因此應透過SD300、SD302確認變址寄存器、超長變址寄存器點數，預先預留出相當於使用次數的區域。
- (d)及其以後使用的區域的配置如下所示。(Z0~23、LZ0~4的情況下)



注意事項

ZPUSH(P) 指令的 (d)+1、(d)+2 為系統所用，因此應勿進行值的更改。

此外，(d) 中指定的區域內儲存的 Z、LZ 儲存類別也為系統所使用，因此應勿進行值的更改。

否則可能導致誤動作。

出錯

出錯代碼 (SD0)	內容
3405H	(s) 輸入了超出允許指定範圍的資料時。 <ul style="list-style-type: none">指定的值為 1~3 以外變址寄存器點數為 0 點的情況下，(s) 中指定了 “2”超長變址寄存器點數為 0 點的情況下，(s) 中指定了 “3”
	(d) 中指定的區域內系統所用區域中儲存的值被改寫時。

變址寄存器/超長變址寄存器選擇恢復

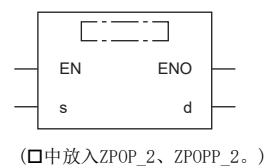
ZPOP (P)



將指定的區域中儲存的資料讀取到變址寄存器、超長變址寄存器中。

梯形圖	ST
	ENO:=ZPOP_2(EN, s, d); ENO:=ZPOPP_2(EN, s, d);

FBD/LD



■執行條件

指令	執行條件
ZPOP	
ZPOPP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	虛擬	—	無符號BIN16位元	ANY16
(d)	進行變址寄存器恢復的起始元件	—	帶符號BIN16位元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		Z	雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□		LT、LST、LC	LZ		K	H	E	\$	
(s)	○	—	○	—	○	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將(d)中指定的元件或數組型標籤的要素及其以後儲存的資料讀取到變址寄存器、超長變址寄存器中。
- 對儲存的變址寄存器、超長變址寄存器的內容進行讀取時，儲存次數(d)將被-1。
- (s)中指定的資料被作為虛擬資料處理，忽略。
- 關於(d)以後使用的區域配置，請參閱下述內容。

☞ 978頁 ZPUSH(P)

注意事項

(d)+1、(d)+2由系統使用，因此應勿進行值的更改。

此外，(d)中指定的區域中儲存的Z、LZ儲存類別也為系統所使用，因此應勿進行值的更改。否則可能導致誤動作。

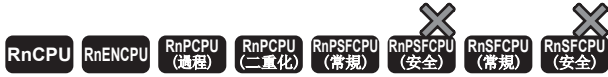
出錯

出錯代碼 (SD0)	內容
3405H	(d)+0的儲存次數為0時。
	(d)中指定的區域內系統所用區域中儲存的值被改寫時。
	Z、LZ儲存類型中設定了1~3以外的值時。

13.2 檔案寄存器操作指令

檔案寄存器的塊No. 切換

RSET (P)



更改程式中使用的檔案寄存器的塊No.。

梯形圖	ST
	<pre>ENO:=RSET(EN, s); ENO:=RSETP(EN, s);</pre>

FBD/LD

■執行條件

指令	執行條件
RSET	
RSETP	

設置數據

■內容、範圍、資料類型

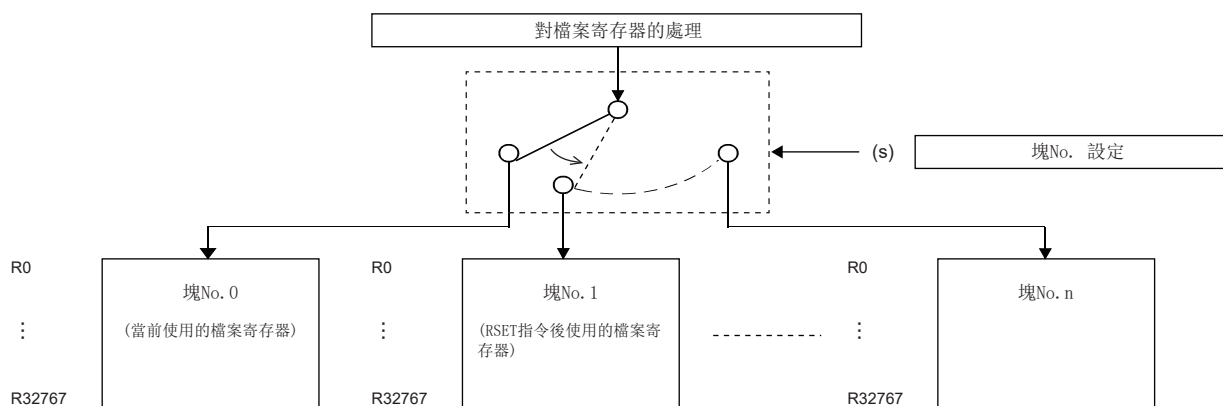
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	切換塊No. 資料或儲存了塊No. 資料的元件編號	0~32767	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 將程式中使用的檔案寄存器的塊No. 更改為(s)中指定的元件中儲存的塊No.。更改塊No. 後，順控程式中使用的所有檔案寄存器將被處理為以更改後塊No. 的檔案寄存器為對象。



注意事項

關於檔案寄存器的限制事項，請參閱下述內容。

☞ 62頁 使用檔案寄存器時的限制事項

出錯

出錯代碼 (SD0)	內容
2820H	檔案寄存器不存在時。
3405H	(s) 中指定的塊No. 不存在時。

檔案寄存器用檔案的設定

QDRSET (P)



將程式中使用的檔案寄存器的檔案名更改為(檔案名)中指定的元件中儲存的檔案名。

梯形圖	ST
<p>FILE: 檔案名</p>	ENO:=QDRSET (EN, 檔案名); ENO:=QDRSETP (EN, 檔案名);

FBD/LD
<p>FILE: 檔案名</p>

■執行條件

指令	執行條件
QDRSET	
QDRSETP	

設置數據

■內容、範圍、資料類型

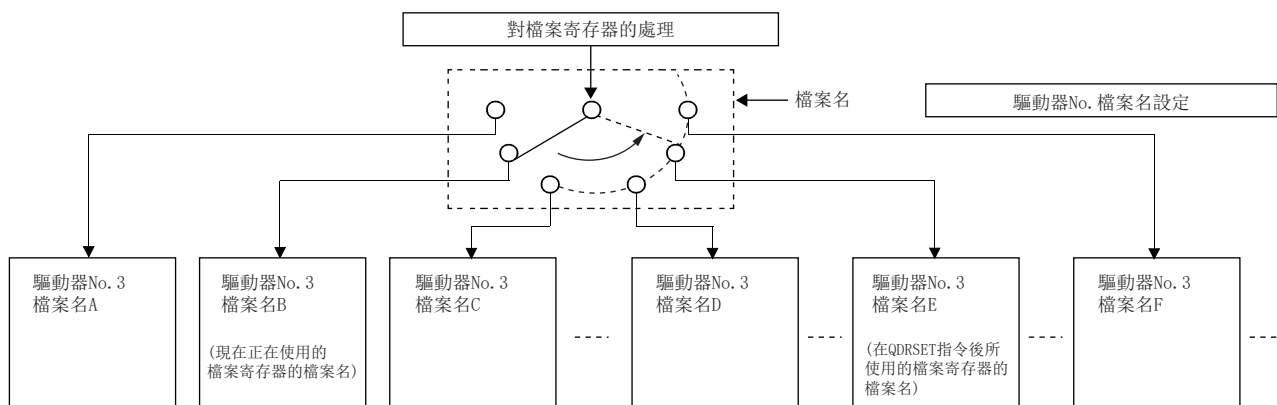
操作數	內容	範圍	資料類型	資料類型(標籤)
(檔案名)	設定的檔案寄存器的驅動器No. 檔案名的字元串資料或儲存了字元串資料的起始元件 例: "1: ABC"	—	Unicode字元串	ANYSTRING_DOUBLE
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E		\$
(檔案名)	—	—	○	—	—	—	—	○	—	—	○	—

功能

- 將程式中使用的檔案寄存器的檔案名，更改為(檔案名)中指定的元件中儲存的檔案名。更改檔案名後，程式中使用的所有檔案寄存器將以更改後的檔案名的塊No. 0的檔案寄存器為對象進行處理。塊No. 的切換是透過RSET (P)指令執行。



- 驅動器No. 可以指定1或3。
- 指定驅動器1的情況下，將作為驅動器3執行動作。此外，SD614(元件/標籤記憶體(驅動器3)使用狀況)中將反映驅動器使用狀況。不反映到SD604 (SD記憶卡(驅動器2)使用狀況)中。
- 檔案名中無需指定副檔名(.QDR)。
- 即使參數中指定了驅動器No. 檔案名的情況下，QDRSET (P)指令中指定的檔案名也將優先。

要點

- 即使透過QDRSET (P)指令更改了檔案名，CPU模組的STOP→RUN操作後，也將恢復為參數中設定的檔案名。對CPU模組進行了STOP→RUN時，也希望保持為透過QDRSET (P)指令更改的檔案名的情況下，應使用STOP→RUN時1個掃描ON的SM402 (RUN後僅1個掃描ON)來執行QDRSET (P)指令。
- 更新元件中指定了檔案寄存器的情況下，應勿透過QDRSET (P)指令更改檔案寄存器的檔案名。

注意事項

- 即使檔案名中指定了NULL代碼(0000H)，檔案名的設定也不被解除，將變為無處理。
- 關於檔案寄存器的限制事項，請參閱下述內容。

☞ 62頁 使用檔案寄存器時的限制事項

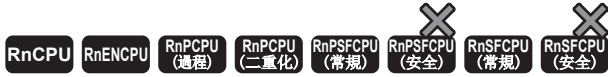
出錯

出錯代碼 (SD0)	內容
2840H	(檔案名)中指定的驅動器No./檔案名不存在時。
3405H	(檔案名)中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none"> 驅動器No. 中指定了1、3以外 僅指定了驅動器No.

13.3 檔案寄存器的1字節單位的讀取/寫入指令

檔案寄存器的1字節資料讀取

ZRRDB(P)



對指定的連號字節No. 的檔案寄存器的內容進行讀取。

梯形圖	ST
	ENO:=ZRRDB(EN, s, d); ENO:=ZRRDBP(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
ZRRDB	
ZRRDBP	

設置數據

■內容、範圍、資料類型

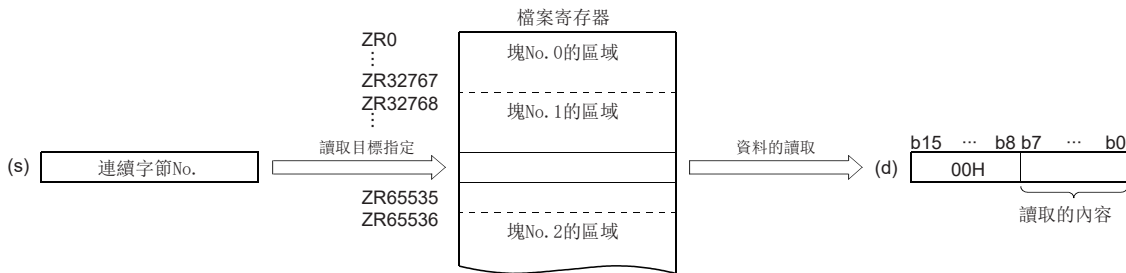
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	讀取的檔案寄存器的連號字節No.	0~4294967295	無符號BIN32位元	ANY32
(d)	儲存被讀取的資料的元件編號	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

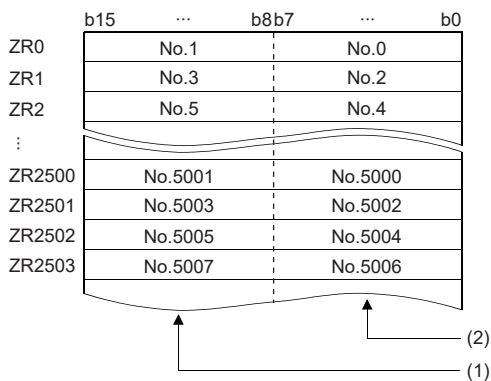
操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 在不理會塊No. 的狀況下，讀取 (s) 中指定的連號字節No. 的檔案寄存器的內容，並儲存到 (d) 中指定的元件的低位8位元中。(d) 中指定的高位8位元將變為00H。



- 連號字節No. 對應的檔案寄存器的編號如下所示。



- (1): 奇數編號指定時的資料
 (2): 偶數編號指定時的資料

例

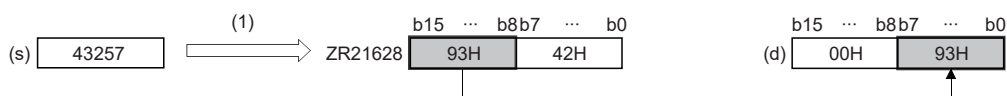
指定了 (s)=23560 的情況下，讀取 ZR11780 的低位8位元的資料。



- (1): 讀取目標指定

例

指定了 (s)=43257 的情況下，讀取 ZR21628 的高位8位元的資料。



- (1): 讀取目標指定

出錯

出錯代碼 (SD0)	內容
2820H	指定了超出允許指定範圍的元件編號(連號字節No.)時。

檔案寄存器的1字節資料寫入

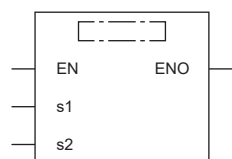
ZRWRB (P)



將指定的元件的低位的內容寫入到連號字節No. 的檔案寄存器中。

梯形圖	ST
	ENO:=ZRWRB (EN, s1, s2) ; ENO:=ZRWRBP (EN, s1, s2) ;

FBD/LD



■執行條件

指令	執行條件
ZRWRB	
ZRWRBP	

設置數據

■內容、範圍、資料類型

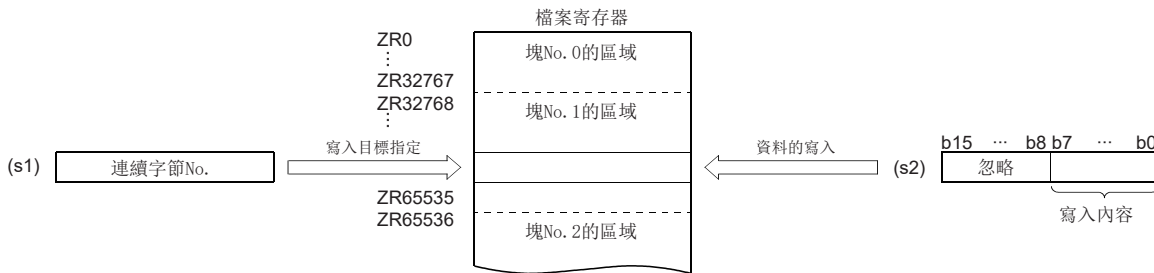
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	寫入的檔案寄存器的連號字節No.	0~4294967295	無符號BIN32位元	ANY32
(s2)	儲存了寫入資料的元件編號	-32768~32767	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

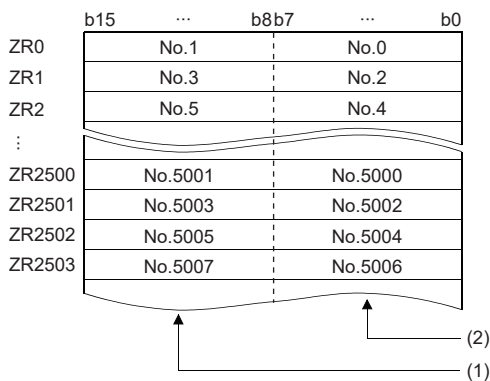
操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	○	○	○	○	○	○	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 在不理會塊No. 的狀況下，將 (s2) 中指定的元件中儲存的低位8位元的內容寫入到 (s1) 中指定的連號字節No. 的檔案寄存器中。(s2) 中指定的元件的高位8位元的資料將被忽略。



- 連號字節No. 對應的檔案寄存器的編號如下所示。



- (1): 奇數編號指定時的資料
(2): 偶數編號指定時的資料

例

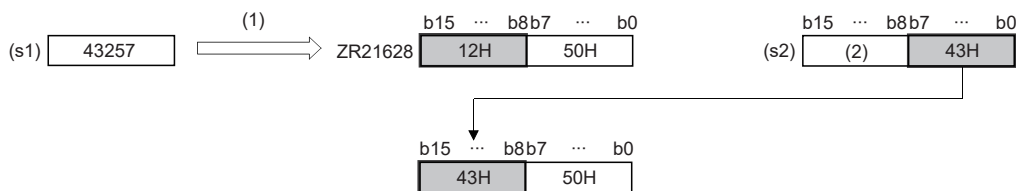
指定了 (s1)=12340 的情況下，寫入到 ZR11170 的低位8位元中。



- (1): 寫入目標指定
(2): 忽略

例

指定了 (s1)=43257 的情況下，寫入到 ZR21628 的高位8位元中。



- (1): 寫入目標指定
(2): 忽略

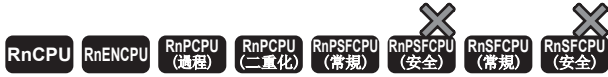
出錯

出錯代碼 (SD0)	內容
2820H	指定了超出允許指定範圍的元件編號(連號字節No.)時。

13.4 間接地址讀取指令

間接地址讀取

ADRSET (P)



讀取指定的元件的間接地址。

梯形圖	ST
	ENO:=ADRSET (EN, s, d) ; ENO:=ADRSETP (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
ADRSET	
ADRSETP	

設置數據

■內容、範圍、資料類型

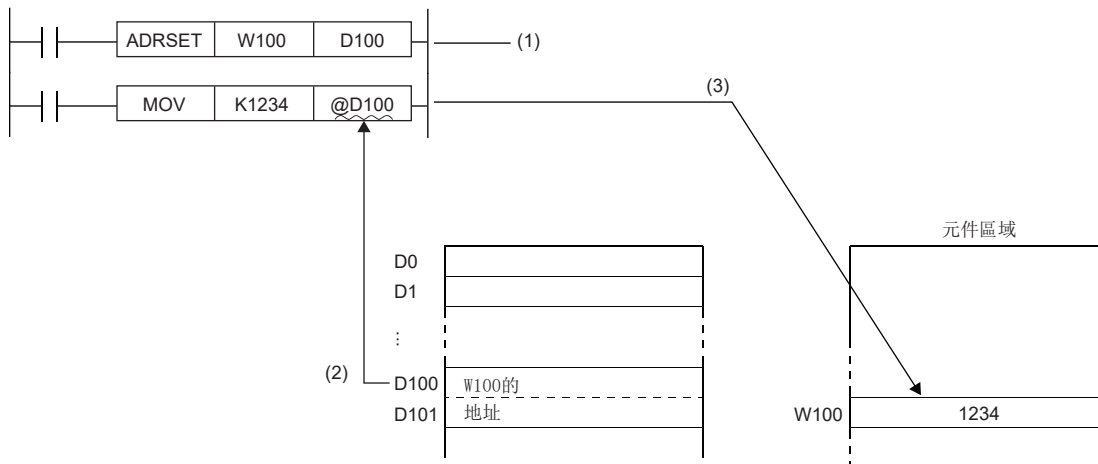
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	讀取間接地址的元件編號	—	元件名	ANY_ELEMENTARY
(d)	儲存(s)中指定的元件的間接地址的元件編號	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s)	○	—	○	—	—	○	—	○	—	—	—	—
(d)	○	—	○	—	—	—	—	○	—	—	—	—

功能

- 將 (s) 中指定的元件的間接地址儲存到 (d) 中指定的元件中。對於 (d)+0、(d)+1 的元件中儲存的地址，在透過程式進行元件的間接地址指定的情況下使用。



- 將W100的地址儲存到D100、D101中。
- 指定D100、D101的內容(W100的地址)。
- 將1234寫入到W100中。

- (s) 中不能進行位元件的位指定、字元件的位元指定。
- 關於元件的間接指定，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

- 不支援標籤的間接指定。在 (d) 中指定標籤使用的情況下，應透過以下方法使用。

標籤	內容
指定了元件的全局標籤的情況下	作為間接地址使用的情況下，應使用全局標籤中指定的元件的間接指定。 ■ST的示例 ADRSET (TRUE, intVar, gvAddr); // 將intVar的間接地址讀取到gvAddr。 INC (TRUE, @D0); // 透過指定至gvAddr的元件D0的間接指定使用。
自動分配的全局標籤/局部標籤的情況下	將間接地址傳送至元件，使用傳送目標元件的間接指定。 ■ST的示例 ADRSET (TRUE, intVar, lvAddr); // 將intVar的間接地址讀取到lvAddr。 DMOV (TRUE, lvAddr, D0); // 將讀取到lvAddr中的間接地址傳送至元件。 INC (TRUE, @D0); // 透過傳送的元件的間接指定使用。

出錯

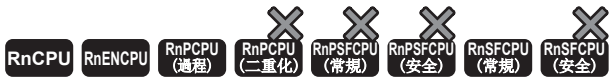
沒有運算出錯。

14 定時器、計數器

14.1 特殊計數器指令

單相輸入升值/降値計數器

UDCNT1



更新指定的計數器的當前值。

梯形圖	ST
	ENO:=UDCNT1 (EN, s1, s2, d);

FBD/LD

■執行條件

指令	執行條件
UDCNT1	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	(s1)+0: 計數輸入用的輸入編號 (s1)+1: 升值/降値設定用 OFF的情況下, 升值計數(向當前值增加方向計數) ON的情況下, 降値計數(向當前值減少方向計數)	—	位元	ANYBIT_ARRAY*1 (要素數: 2)
(d)	透過UDCNT1指令進行計數的計數器編號(元件名)	—	元件名	ANY16*2
(s2)	設定值	-32768~32767	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用元件(X)中分配的標籤。

*2 只能使用元件(C)中分配的標籤。

■可使用的元件

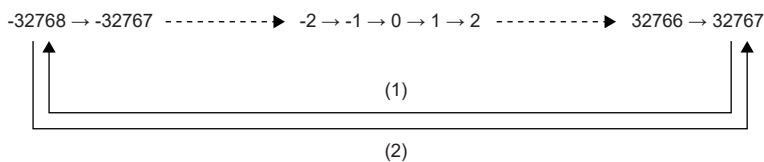
操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○*1	—	—	—	—	—	—	—	—	—	—	—
(d)	—	—	○*2	—	—	—	—	—	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—

*1 只能使用X。但是，只能在輸入輸出點數(可存取輸入輸出模組的點數)的範圍內使用。

*2 只能使用C。

功能

- (s1)中指定的輸入的OFF→ON時，對(d)中指定的計數器的當前值進行更新。
- 計數方向取決於(s1)+1中指定的輸入的ON/OFF。
 - OFF：升值計數(向當前值增加方向計數)
 - ON：降值計數(向當前值減少方向計數)
- 計數處理按以下方式進行。
 - 升值計數中，當前值等於(s2)中指定的設定值時，(d)中指定的計數器的觸點將變為ON。但是，即使(d)中指定的計數器的觸點變為ON，當前值的計數仍將繼續。
 - 降值計數中，當前值為設定值-1時，(d)中指定的計數器的觸點將變為OFF。
 - (d)中指定的計數器是環形計數器。當前值為32767時如果進行升值計數，當前值將變為-32768。此外，當前值為-32768時如果進行降值計數，當前值將變為32767。當前值的計數處理內容如下所示。



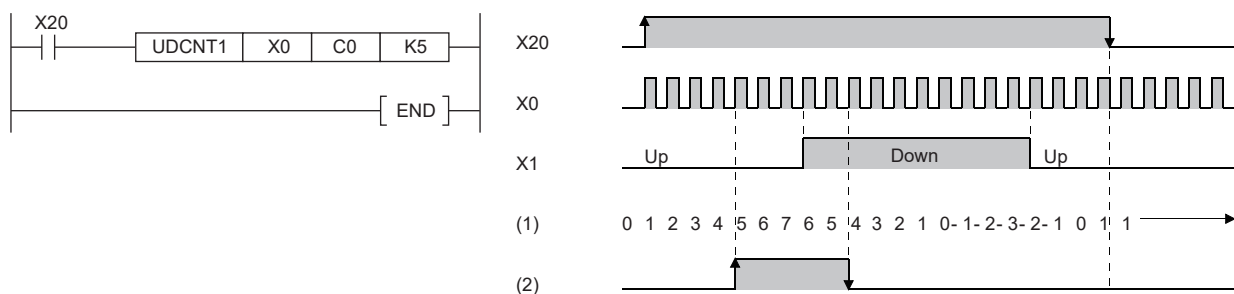
(1): 升值計數時

(2): 降值計數時

- 使用了UDCNT1指令的計數處理的動作如下所示。

例

透過C0(升值/降值計數器)，對X20為ON之後的X0的OFF→ON次數進行計數的程式和動作如下所示。



(1): C0當前值

(2): C0觸點

- 透過UDCNT1指令進行計數處理時，執行指令的OFF→ON時開始計數、ON→OFF時中止計數。再次將執行指令設為OFF→ON時，將從中止時的當前值開始再次開始計數。
- (d)中指定的計數器的當前值的清除及觸點的OFF是透過RST指令執行。

要點

- 對於UDCNT1指令，將引數的元件資料登錄到CPU模組的工作區域中後，實際的計數動作是透過系統中斷進行處理。(CPU模組的工作區域中登錄的元件資料在執行指令變為OFF、或STOP→RUN時將被清除。)因此，對於可進行計數的脈衝，其ON及OFF時間需為CPU模組的中斷間隔及其以上。CPU模組的中斷間隔為1ms。
- 透過UDCNT1指令進行的計數中(執行指令為ON中)，不能進行設定值更改。進行設定值更改的情況下，應將執行指令設為OFF之後再進行更改。
- UDCNT1指令中指定的計數器不能用於其它指令。在其它指令中使用的情況下將無法正常計數。
- UDCNT1指令在正在執行的全部程式中最多可以使用6次。從第7次及其以後將變為無處理。

出錯

沒有運算出錯。

2相輸入升值/降値計數器

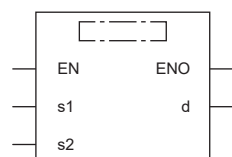
UDCNT2



根據指定的A相脈衝及B相脈衝的狀態，對計數器的當前值進行更新。

梯形圖	ST
	ENO:=UDCNT2(EN, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
UDCNT2	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	(s1)+0: 計數輸入用的輸入編號 (A相脈衝) (s1)+1: 計數輸入用的輸入編號 (B相脈衝)	—	位元	ANYBIT_ARRAY*1 (要素數: 2)
(d)	透過UDCNT2指令進行計數的計數器編號 (元件名)	—	元件名	ANY16*2
(s2)	設定值	-32768~32767	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用元件 (X) 中分配的標籤。

*2 只能使用元件 (C) 中分配的標籤。

■可使用的元件

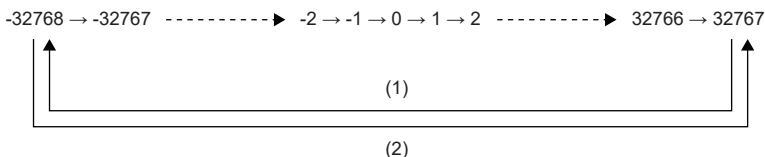
操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3B□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	○*1	—	—	—	—	—	—	—	—	—	—
(d)	—	—	○*2	—	—	—	—	—	—	—	—
(s2)	○	○	○	○	○	—	○	○	—	—	—

*1 只能使用X。但是，只能在輸入輸出點數(可存取輸入輸出模組的點數)的範圍內使用。

*2 只能使用C。

功能

- 根據(s1)中指定的輸入(A相脈衝)及(s1)+1中指定的輸入(B相脈衝)的狀態，對(d)中指定的計數器的當前值進行更新。
- 計數方向按下述方式確定。
 - (s1)為ON時(s1)+1由OFF→ON的情況下，進行升值計數(向當前值增加方向計數)。
 - (s1)為ON時(s1)+1由ON→OFF的情況下，進行降值計數(向當前值減少方向計數)。
 - (s1)為OFF時不進行計數。
- 計數處理按以下方式進行。
 - 升值計數中，當前值等於(s2)中指定的設定值時，(d)中指定的計數器的觸點將變為ON。但是，即使(d)中指定的計數器的觸點變為ON，當前值的計數仍將繼續。
 - 降值計數中，當前值為設定值-1時，(d)中指定的計數器的觸點將變為OFF。
 - (d)中指定的計數器是環形計數器。當前值為32767時如果進行升值計數，當前值將變為-32768。此外，當前值為-32768時如果進行降值計數，當前值將變為32767。當前值的計數處理內容如下所示。

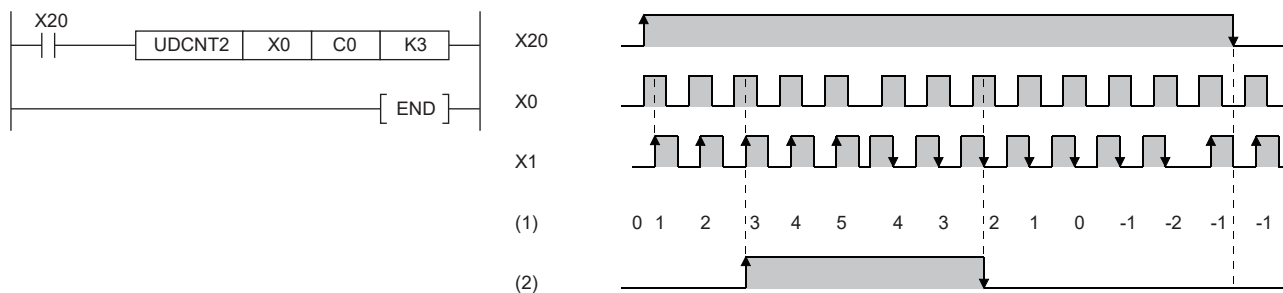


- (1): 升值計數時
- (2): 降值計數時

- 使用了UDCNT2指令的計數處理的動作如下所示。

例

透過C0(升值/降值計數器)，對X20為ON之後的X0、X1的狀態進行計數的程式和動作如下所示。



- (1): C0當前值
- (2): C0觸點

- 透過UDCNT2指令進行計數處理時，執行指令的OFF→ON時開始計數，ON→OFF時中止計數。再次將執行指令設為OFF→ON時，將從中止時的當前值開始再次開始計數。
- (d)中指定的計數器的當前值的清除及觸點的OFF是透過RST指令執行。

要點

- 對於UDCNT2指令，將引數的元件資料登錄到CPU模組的工作區域中，實際的計數動作透過系統中斷進行處理。(CPU模組的工作區域中登錄的元件資料在執行指令變為OFF、或STOP→RUN時將被清除。)因此，對於可進行計數的脈衝，其ON及OFF時間需為CPU模組的中斷間隔及其以上。CPU模組的中斷間隔為1ms。
- 透過UDCNT2指令進行的計數中(執行指令為ON中)，不能進行設定值的更改。進行設定值更改的情況下，應將執行指令設為OFF之後再進行更改。
- UDCNT2指令中指定的計數器不能用於其它指令。在其它指令中使用的情況下將無法正常計數。
- UDCNT2指令在正在執行的全部程式中最多可以使用5次。從第6次及其以後將變為無處理。

出錯

沒有運算出錯。

14.2 特殊定時器指令

示教定時器

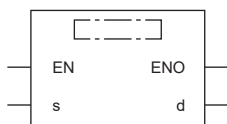
TTMR



將計測指令的ON中的時間以秒單位進行測定，對乘以乘數後的值進行儲存。

梯形圖	ST
	ENO:=TTMR (EN, s, d);

FBD/LD



■執行條件

指令	執行條件
TTMR	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	(d)+0: 測定值儲存元件 (d)+1: CPU模組的系統用	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 2)
(s)	測定值的乘數	0~2	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(d)	—	—	○	—	—	—	○	—	—	—	—	—
(s)	—	○	○	○	○	—	○	○	—	—	—	—

■控制資料

操作數: (d)			
元件	內容	設定範圍	設定方
+0	測定值儲存元件	—	系統
+1	CPU模組的系統用	—	系統

功能

- 將執行指令為ON的時間以秒單位進行測定，將乘以(s)中指定的乘數後的值儲存到(d)中指定的元件中。
- 執行指令由OFF→ON時，(d)+0、(d)+1中指定的元件將被清除。
- (s)中可指定的乘數如下所示。

(s)	乘數
0	1
1	10
2	100

- (s)中指定的值為0~2以外的情況下，將變為無處理。

要點

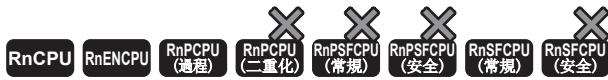
- 執行TMR指令時，進行時間計測。應勿透過JMP指令等跳過TMR指令。跳過的情況下，將無法正確進行測定。
- TMR指令執行中，應勿更改(s)中指定的乘數。更改了(s)中指定的乘數的情況下，將無法得出正確的值。
- (d)+1中指定的元件是由CPU模組的系統所使用，因此應勿對其進行值的更改。進行了值的更改的情況下，(d)中指定的元件中儲存的值將不是正確的值。

出錯

沒有運算出錯。

特殊功能定時器

STMR



進行下述4種類型的定時器輸出。

- OFF延遲定時器輸出
- OFF後單次觸發定時器輸出
- ON後單次觸發定時器輸出
- ON延遲+OFF延遲定時器輸出

梯形圖	ST
	ENO:=STMR (EN, s1, s2, d);

FBD/LD

■執行條件

指令	執行條件
STMR	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	定時器元件或定時器型標籤	—	元件名	ANY16
(s2)	設定值	0~32767	帶符號BIN16位元	ANY16
(d)	(d)+0: OFF延遲定時器輸出 (d)+1: OFF後單次觸發定時器輸出 (d)+2: ON後單次觸發定時器輸出 (d)+3: ON延遲+OFF延遲定時器輸出	—	位元	ANYBIT_ARRAY (要素數: 4)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(s1)	—	—	○*1	—	—	—	—	—	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	—	—	—	—	—	—	—	—	—	—	—

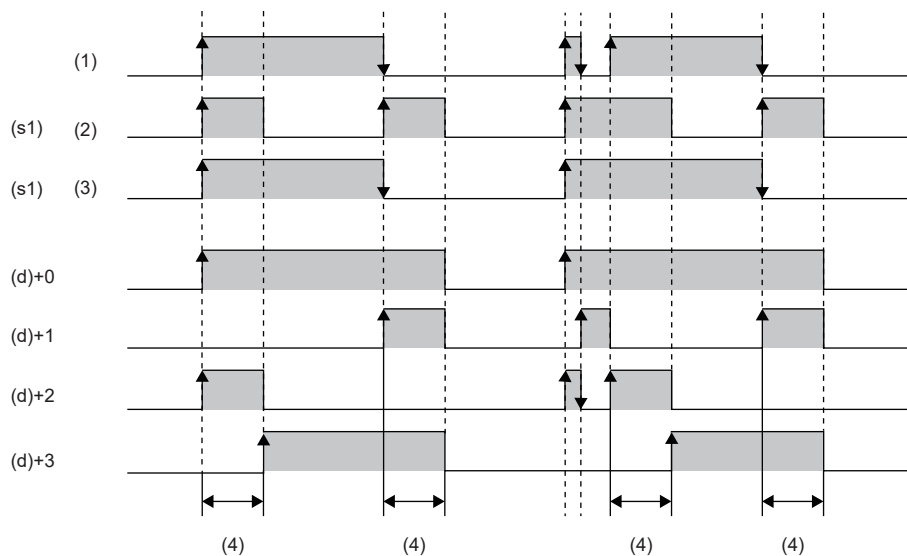
*1 只能使用T。

功能

- 使用 (d) 中指定的元件開始的4點、進行4種類型的定時器輸出。

設定資料	內容	
(d)	+0	OFF延遲定時器輸出： 透過STMR指令的指令的上升沿變為ON，指令的下降沿後經過(s2)中指定的時間時將變為OFF。
	+1	OFF後單次觸發定時器輸出： 透過STMR指令的指令的下降沿變為ON，經過(s2)中指定的時間時變為OFF。
	+2	ON後單次觸發定時器輸出： 透過元件名STMR指令的指令的上升沿變為ON，經過(s2)中指定的時間或STMR指令的指令為OFF時將變為OFF。
	+3	ON延遲+OFF延遲定時器輸出： 將定時器線圈的下降沿變為ON、STMR指令的指令的下降沿後，經過(s2)中指定的時間後將變為OFF。

- 對於(s1)中指定的定時器的線圈，透過STMR指令的指令的上升沿及下降沿變為ON，開始當前值的計測。
- 對於定時器的線圈，至(s2)中指定的設定值為止進行計測，如果時限到則變為OFF。
- 對於定時器的線圈，如果時限到之前STMR指令的指令變為OFF，則將保持為ON狀態不變。此時定時器的計測將繼續進行。STMR指令的指令再次變為ON時，將當前值設為0後，重啟計測。
- 對於定時器的觸點，透過STMR指令的指令的上升沿變為ON，定時器的線圈變為下降沿後，透過STMR指令的指令的下降沿變為OFF。因為定時器的觸點是由系統所用，因此不能使用。



- (1): STMR指令的指令
- (2): 線圈
- (3): 觸點
- (4): (s2)中指定的設定值

- 透過STMR指令指定的定時器執行當前值計測時，與STMR指令的指令的ON/OFF狀態無關。透過JMP指令等跳過了STMR指令的情況下，將無法正常進行計測。
- (d)中指定的定時器的計測單位與低速定時器的相同。
- (s2)的設定值可在0~32767的範圍內指定。0~32767以外的情況下將變為無處理。
- 對於(s1)中指定的定時器應勿使用OUT指令。STMR指令與OUT指令中使用了同一定時器元件或定時器型標籤的情況下，將無法正常動作。

注意事項

運行中的梯形圖塊更改或運行中PC寫入的寫入範圍內存在有STMR指令的情況下，STMR指令將動作。
關於詳細內容，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

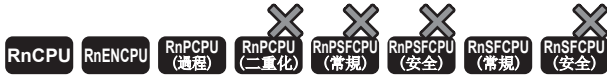
出錯

沒有運算出錯。

14.3 脈衝系統指令

脈衝密度的測定

SPD



以指定了元件輸入的時間進行計數。

梯形圖	ST
	$ENO := SPD(EN, s1, s2, d);$

FBD/LD

■執行條件

指令	執行條件
SPD	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	脈衝輸入的元件編號	—	位元	ANY_BOOL*1
(s2)	測定時間或儲存了測定時間的元件編號(單位: ms)	-32768~32767	帶符號BIN16位元	ANY16
(d)	儲存測定結果的元件	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用元件(X)中分配的標籤。

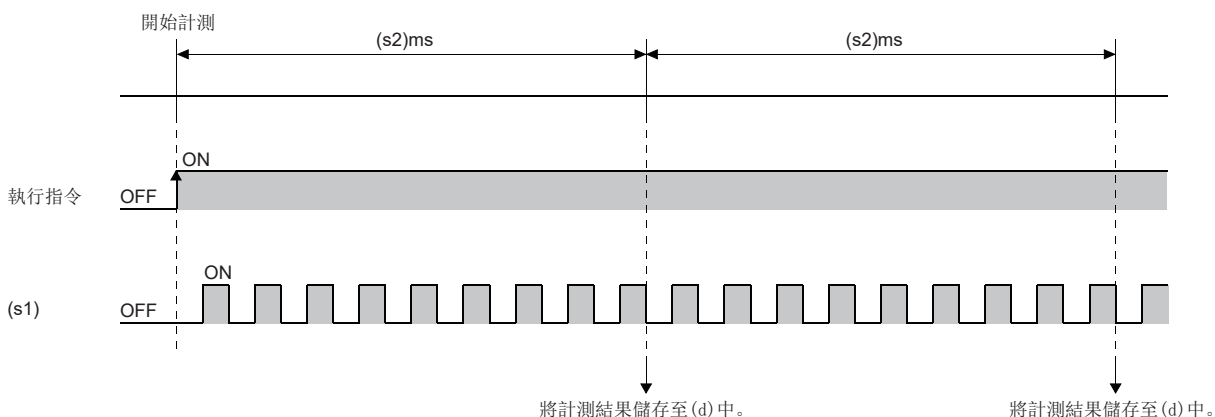
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○*1	—	—	—	—	—	—	—	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

*1 只能使用X。但是，只能在輸入輸出點數(可存取輸入輸出模組的點數)的範圍內使用。

功能

- 將 (s1) 中指定的元件輸入的OFF→ON的次數，以 (s2) 中指定的時間進行計數，將計數結果儲存到 (d) 中指定的元件中。



- 透過SPD指令進行的計測完成時，將再次從0開始進行計測。中止透過SPD指令進行的計測的情況下，應將執行指令設為OFF。
- (s2)=0的情況下將變為無處理。

出錯

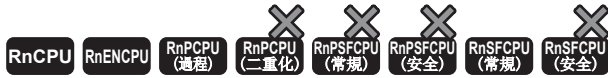
沒有運算出錯。

要點

- 對於SPD指令，將引數元件的資料登錄到CPU模組內部的工作區域後，實際的計數動作透過系統中斷進行。(對於CPU模組的工作區域中登錄的元件資料，透過將執行指令設為OFF或STOP→RUN進行清除。)因此，對於可進行計數的脈衝，其ON及OFF時間需為CPU模組的中斷間隔及其以上。CPU模組的中斷間隔為1ms。
- SPD指令在正在執行的全部程式中最多只能使用6次。從第7次及其以後將變為無處理。
- 在透過SPD指令進行的計測中(指令輸入為ON中)，不能進行設定值的更改。進行設定值更改的情況下，應將指令輸入設為OFF之後在進行更改。

恆定週期脈衝輸出

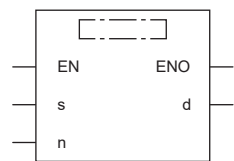
PLSY



將指定頻率的脈衝輸出到輸出模組中。

梯形圖	ST
	<pre>ENO:=PLSY(EN, s, n, d);</pre>

FBD/LD



執行條件

指令	執行條件
PLSY	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	頻率或儲存了頻率的元件編號	1~100	帶符號BIN16位元	ANY16
(n)	輸出次數或儲存了輸出次數的元件編號	0~65535	無符號BIN16位元	ANY16
(d)	脈衝輸出的元件	—	位元	ANY_BOOL*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用元件(Y)中分配的標籤。

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○*1	—	—	—	—	—	—	—	—	—	—	—

*1 只能使用Y。但是，只能在輸入輸出點數(可存取輸入輸出模組的點數)的範圍內使用。

功能

- 將(s)中指定的頻率的脈衝，以(n)中指定的次數，輸出到(d)中指定的輸出編號(Y)的輸出模組中。
- (s)的頻率可在1Hz~100Hz的範圍內設定。(s)超出1~100的範圍的情況下，將變為無處理。
- (n)的輸出次數可在0~65535(0000H~FFFFH)的範圍內指定。(n)為0的情況下，將連續輸出脈衝。
- 對於(d)中指定的脈衝輸出，只能指定輸出模組對應的輸出編號(Y)。
- 以PLSY指令的指令的上升沿開始脈衝輸出。PLSY指令的指令為OFF時，將停止脈衝輸出。

出錯

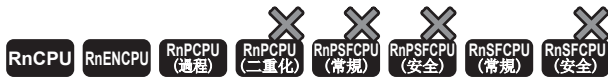
沒有運算出錯。

要點

- 對於PLSY指令，將引數的元件資料登錄到CPU模組的工作區域後，實際的輸出動作透過系統中斷進行處理。(對於CPU模組的工作區域中登錄的元件資料，透過將執行指令設為OFF或STOP→RUN進行清除。)因此，對於可輸出的脈衝，其ON及OFF時間需為CPU模組的中斷間隔及其以上。CPU模組的中斷間隔為1ms。
- 透過PLSY指令進行的脈衝輸出中(執行指令為ON中)，應勿更改PLSY指令的引數。引數的更改應將執行指令設為OFF之後再進行。
- PLSY指令在CPU模組正在執行的全部程式中只能使用1次。從第2次及其以後將變為無處理。

脈衝寬度調制

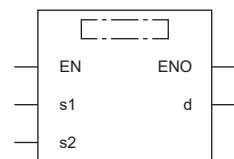
PWM



透過指定時間ON將週期脈衝輸出到輸出模組中。

梯形圖	ST
	$ENO := PWM(EN, s1, s2, d);$

FBD/LD



■執行條件

指令	執行條件
PWM	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	ON時間或儲存了ON時間的元件編號 (單位: ms)	1~65535	無符號BIN16位元	ANY16
(s2)	週期或儲存了週期的元件編號 (單位: ms)	1~65535	無符號BIN16位元	ANY16
(d)	脈衝輸出的元件編號	—	位元	ANY_BOOL*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用元件 (Y) 中分配的標籤。

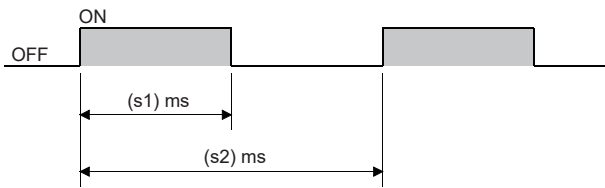
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	○	○	○	○	○	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—	—
(d)	○*1	—	—	—	—	—	—	—	—	—	—	—

*1 只能使用Y。但是，只能在輸入輸出點數(可存取輸入輸出模組的點數)的範圍內使用。

功能

- 將 (s1) 中指定的ON時間及 (s2) 中指定的週期脈衝輸出到 (d) 中指定的輸出模組中。



- (s1)、(s2) 的設定時間為1~65535 (0001H~FFFFH)。(s1) 中指定的值應小於 (s2) 中指定的值。)
- 下述情況下，將變為無處理。
 - (s1)、(s2) 為0時。
 - (s1) ≥ (s2) 時。
 - 至少執行了2次PWM指令時。

出錯

沒有運算出錯。

要點

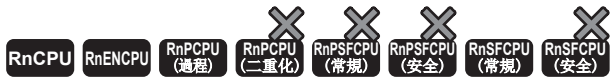
- 對於PWM指令，將引數的元件資料登錄到CPU模組的工作區域後，實際的輸出動作透過系統中斷進行處理。(對於CPU模組的工作區域中登錄的元件資料，透過將執行指令設為OFF或STOP→RUN進行清除。)CPU模組的中斷間隔為1ms。因此，PWM指令在CPU模組正在執行的全部程式中只能使用1次。
- 透過PWM指令進行的脈衝輸出中(執行指令為ON中)，應勿更改PWM指令的引數。進行引數更改的情況下，應將執行指令設為OFF之後再進行。

15 就近控制

15.1 就近控制指令

旋轉臺的就近控制

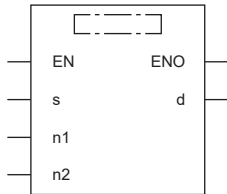
ROTC



在以指定值等分後的旋轉臺上，對旋轉臺進行控制，使其能就近旋轉。

梯形圖	ST
	ENO:=ROTC(EN, s, n1, n2, d);

FBD/LD



■執行條件

指令	執行條件
ROTC	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	(s)+0: 旋轉臺旋轉數測定用(系統用)	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
	(s)+1: 調用視窗編號		無符號BIN16位元	
	(s)+2: 調用物品編號		無符號BIN16位元	
(n1)	旋轉臺分割數	2~65535	無符號BIN16位元	ANY16
(n2)	低速區間數	不足0~(n1)	無符號BIN16位元	ANY16
(d)	(d)+0: A相輸入信號	—	位元	ANYBIT_ARRAY (要素數: 8)
	(d)+1: B相輸入信號			
	(d)+2: 0點檢測輸入信號			
	(d)+3: 高速正轉輸出信號(系統用)			
	(d)+4: 低速正轉輸出信號(系統用)			
	(d)+5: 停止輸出信號(系統用)			
	(d)+6: 低速反轉輸出信號(系統用)			
(d)+7: 高速反轉輸出信號(系統用)				
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	—	—	○	—	—	—	○	—	—	—	—	
(n1)	○	○	○	○	○	—	○	○	—	—	—	
(n2)	○	○	○	○	○	—	○	○	—	—	—	
(d)	○	—	—	—	—	—	—	—	—	—	—	

功能

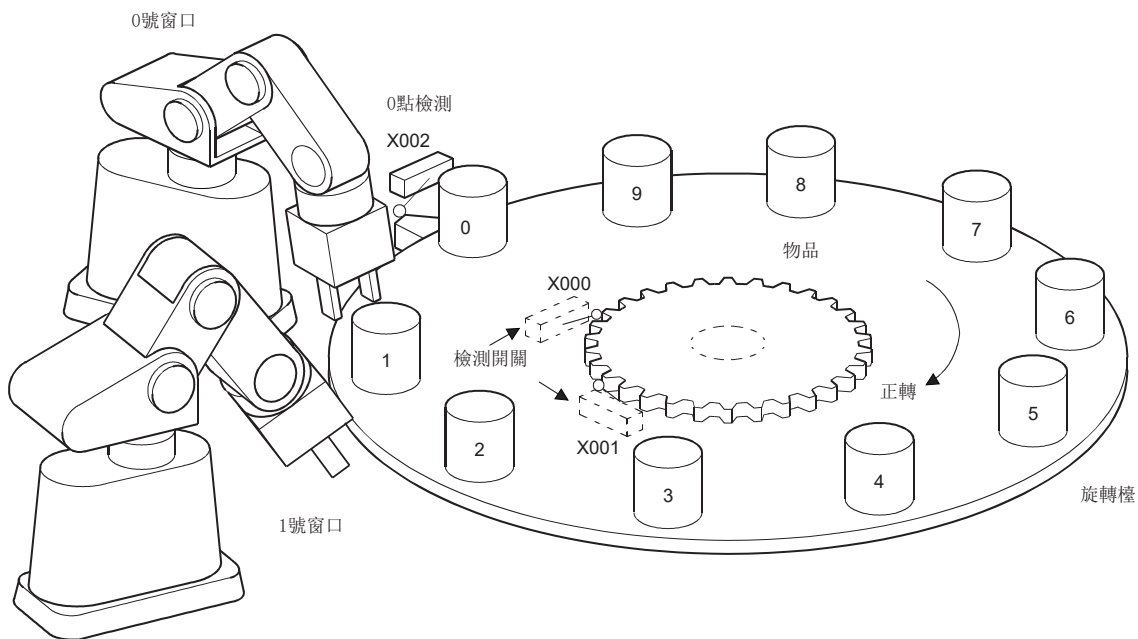
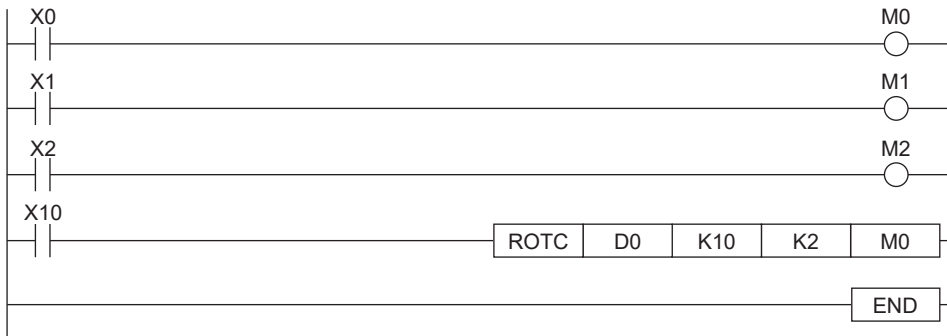
- 在以(n1)中指定的值等分後的旋轉檯上，為了將(s)+2中指定編號的物品取出放入，對旋轉檯進行控制，使其就近旋轉到(s)+1中指定的視窗編號的位置處。
- 對於物品編號及視窗編號，以逆時鐘方向分配後進行控制。
- (s)+0是由系統所用的計數器，用於對第0號的視窗中有第幾號的物品進行計數。應勿透過程式等進行資料改寫。進行了改寫的情況下，將無法正常控制。
- 對於(n2)，應設定為小於(n1)中指定的旋轉檯分割數的值。
- (d)+0及(d)+1是用於檢測旋轉檯的正轉/反轉的A相及B相輸入信號。旋轉方向的判別是透過A相為ON時的B相的上升沿/下降沿進行。
 - B相為上升沿時：正轉(順時鐘旋轉)
 - B相為下降沿時：反轉(逆時鐘旋轉)
- (d)+2是第0號的物品來到第0號的視窗時變為ON的0點檢測信號。ROTC指令執行中，(d)+2中指定的元件變為ON時，(s)+0將被清除。應在預先進行此清除操作之後，再開始透過ROTC指令進行的就近控制。
- (d)+3～(d)+7是用於對旋轉檯的動作進行控制的輸出信號。根據ROTC指令的執行結果，(d)+3～(d)+7中之一的輸出信號將變為ON。
- ROTC指令的指令為OFF的情況下，將在不進行就近控制的狀況下將(d)+3～(d)+7全部設為OFF。
- ROTC指令在正在執行的全部程式中只能使用1次。使用了2次及其以上的情況下將無法正常執行。
- (s)+0～(s)+2或(n2)的值為(n1)及其以上的情況下，將變為無處理。

■程式示例

放在分成10份的旋轉檯上的物品在視窗進出時，控制旋轉檯在周圍旋轉的程式。

物品的編號透過D2指定、視窗的編號透過D1指定。

前後2區間使檯低速旋轉。



出錯

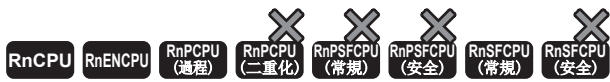
沒有運算出錯。

16 斜坡信號

16.1 斜坡信號指令

斜坡信號

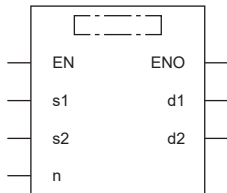
RAMPQ



從指定的值開始以 (n) 次移轉到指定的值。

梯形圖	ST
	ENO:=RAMPQ (EN, s1, s2, n, d1, d2);

FBD/LD



■執行條件

指令	執行條件
RAMPQ	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	初始值	-32768~32767	帶符號BIN16位元	ANY16
(s2)	最終值	-32768~32767	帶符號BIN16位元	ANY16
(d1)	(d1)+0: 當前值	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 2)
	(d1)+1: 執行次數			
(n)	移轉次數	1~32767	帶符號BIN16位元	ANY16
(d2)	(d2)+0: 完成元件	—	位元	ANYBIT_ARRAY (要素數: 2)
	(d2)+1: 完成時資料保持選擇位元			
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d1)	○	○	○	○	○	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	
(d2)	○	—	—	—	—	—	—	—	—	—	—	—	

功能

- 執行指令為ON時，執行下述處理。
- 從(s1)中指定的值開始向(s2)中指定的值，以(n)中指定的次數進行移轉。
- (n)中指定從(s1)移轉到(s2)的掃描數(移轉次數)。0 < (n) < 32768以外的情況下，將變為無處理。
- (d1)+1由系統使用，儲存RAMPQ指令的執行次數。
- 1次(1個掃描)的變化值由下述公式計算。

$$Cv = \frac{(s2) - (s1)}{(n)}$$

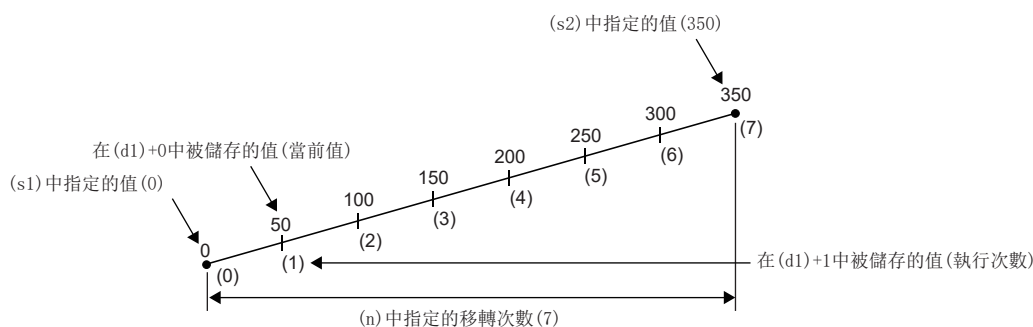
Cv: 1次的變化值

(s2): (s2)中指定的值

(s1): (s1)中指定的值

(n): (n)中指定的值

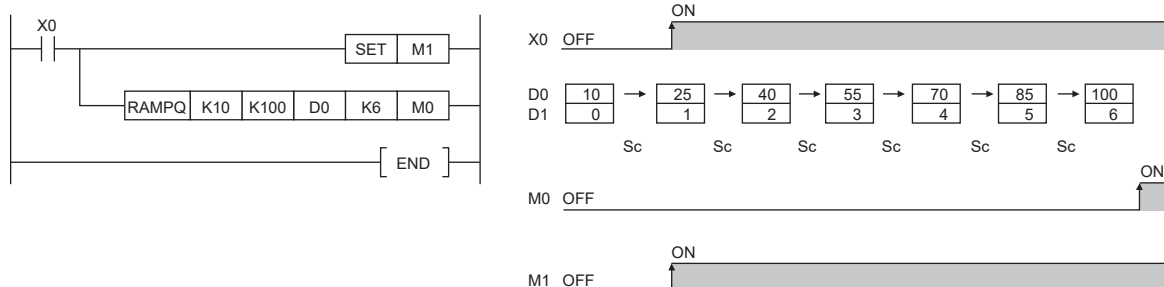
- 從0至350以7個掃描來使其變化時的情況如下所示。



- 計算1次的變化值時除不盡的情況下，以(n)中指定的移轉次數進行補償，使其變為(s2)中指定的值。因此有可能不呈直線斜坡。
- 使用了RAMPQ指令的處理的動作如下所示。

例

當X0為ON時，使D0的內容透過6個掃描使其從10變化為100、對移轉完成時D0的內容進行保持的程式和動作如下所示。



Sc: 1個掃描

- 進行(n)中指定的移轉次數的掃描時，(d2)+0中指定的完成元件將變為ON。完成元件的ON/OFF狀態及(d1)+0的內容取決於(d2)+1中指定的元件的ON/OFF。(d2)+1為OFF的情況下，在下一個掃描中將(d2)+0設為OFF，RAMPQ指令將從初始值開始再次開始移轉。(d2)+1為ON的情況下，(d2)+0將保持為ON不變，(d1)+0的內容不變化。
- RAMPQ指令的執行過程中指令變為OFF的情況下，(d1)+0的內容從此不變化。指令再次變為ON時，RAMPQ指令將從初始值開始再次開始移轉。
- (d2)+0中指定的完成元件為ON之前，應勿對(s1)及(s2)的值進行更改。因為每個掃描以同一計算公式計算(d1)+1中儲存的值，因此如果更改(s1)及(s2)有可能導致急劇變化。
- 將(d1)透過位元元件進行位指定的情況下，應將位指定以K4的形式進行指定。

注意事項

將(d1)透過位元元件進行位指定的情況下，只有將位指定指定為K4時才能使用。

出錯

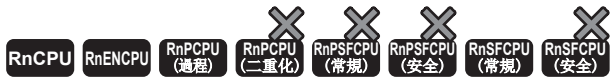
沒有運算出錯。

17 矩陣輸入

17.1 矩陣輸入指令

矩陣輸入

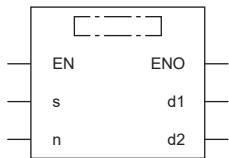
MTR



對指定的輸入編號及其以後連接的16點×n欄的輸入依次進行讀取。

梯形圖	ST
	ENO:=MTR(EN, s, n, d1, d2);

FBD/LD



■執行條件

指令	執行條件
MTR	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	輸入的起始元件	—	位元	ANY_BOOL*1*3
(d1)	輸出的起始元件	—	位元	ANY_BOOL*2*3
(d2)	儲存矩陣輸入資料的起始元件	—	位元	ANY_BOOL*3
(n)	輸入欄數	2~8	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用元件(X)中分配的標籤。

*2 只能使用元件(Y)中分配的標籤。

*3 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○*1	—	—	—	—	—	—	—	—	—	—	—
(d1)	○*2	—	—	—	—	—	—	—	—	—	—	—
(d2)	○	—	—	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

*1 只能使用X。

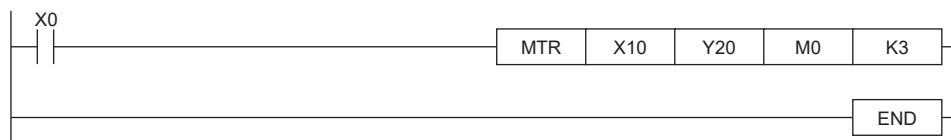
*2 只能使用Y。

功能

- 對 (s) 中指定的輸入編號以後連接的16點×(n) 欄的輸入依次進行讀取，將獲取的輸入資料儲存到 (d2) 中指定的元件及其以後。
- 在1個掃描中獲取1欄(16點)。
- 依次重複從第1欄至第 (n) 欄的獲取。
- 把從起始開始的16點為作為第1欄的資料、下一個16點為第2欄的資料儲存到 (d2) 中指定的元件及其以後。因此，從 (d2) 中指定的元件開始的16×(n) 點由MTR指令佔用。
- (d1) 是用於選擇進行獲取的欄的輸出，由系統自動設為ON/OFF。使用從 (d1) 中指定的元件開始的 (n) 點。
- (s)、(d1)、(d2) 中只能指定16的倍數的元件編號。
- (n) 中可以指定2~8。
- 下述情況下，將變為無處理。
 - (s)、(d1)、(d2) 中指定的元件No. 不是16的倍數時。
 - (s) 中指定的元件超出實際輸入的範圍時。
 - (d1) 中指定的元件超出實際輸出的範圍時。
 - (d2) 中指定的元件及其以後16×(n) 點超出相應元件的範圍時。
 - (n) 為2~8以外時。

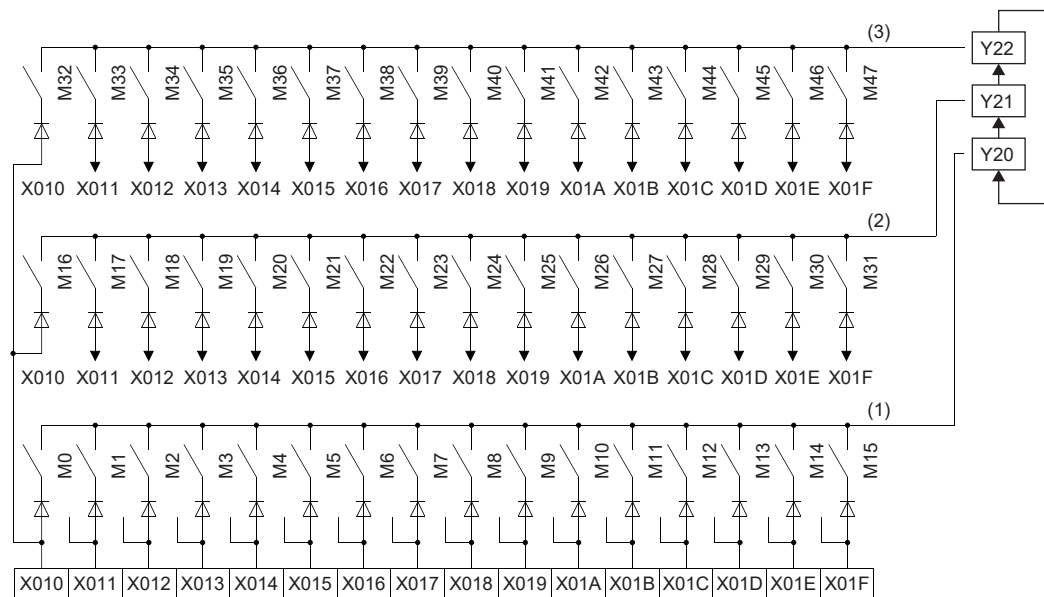
程式示例

以下為將X0設為ON時，獲取X10及其以後連接的16點×3欄的矩陣，儲存到M0以後的程式。



[動作]

- Y20為ON時，獲取第1欄的輸入信號，儲存到M0~M15中。
- Y21為ON時，獲取第2欄的輸入信號，儲存到M16~M31中。
- Y22為ON時，獲取第3欄的輸入信號，儲存到M32~M47中。



- (1): 第1欄
 (2): 第2欄
 (3): 第3欄

注意事項

- 因為MTR指令直接操作實際輸入輸出，因此應加以注意。即使MTR指令的指令為OFF，透過MTR指令設為ON的輸出也不變為OFF。應透過程式將(d1)中指定的輸出設為OFF。
- 對於MTR指令的執行間隔，應不小於將輸入模組與輸出模組的應答時間相加後的值。MTR指令的執行間隔短於上述時間的情況下，將無法正常進行輸入獲取。程式的掃描時間過短的情況下，應透過恆定掃描將掃描時間設定為長於合計的響應時間。

出錯

出錯代碼(SD0)	內容
2820H	(s)中指定了輸入(X)以外時。
	(d1)中指定了輸出(Y)以外時。

18 CPU模組內建資料庫存取功能

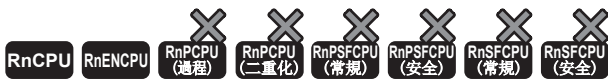
18.1 資料庫存取指令

資料庫存取指令是在可程式控制器上作為資料庫進行管理，並針對產品資訊及生產資訊等的表格形式之資料進行資料的添加/更新/提取/刪除等的指令。

在資料庫存取指令中，透過定義了表的配置等的Unicode文字檔案構建資料庫後，對構建的資料庫進行操作。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

資料庫導入

DBIMPORT (P)

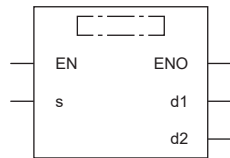


• R00CPU、R01CPU、R02CPU中不能使用。

對(s)中指定路徑的Unicode文字檔案檔案中設定的資料進行提取、構建資料庫。

梯形圖	ST
	<pre>ENO:=DBIMPORT (EN, s, d1, d2); ENO:=DBIMPORTP (EN, s, d1, d2);</pre>

FBD/LD



■執行條件

指令	執行條件
DBIMPORT	
DBIMPORTP	

設置數據

■內容、範圍、資料類型

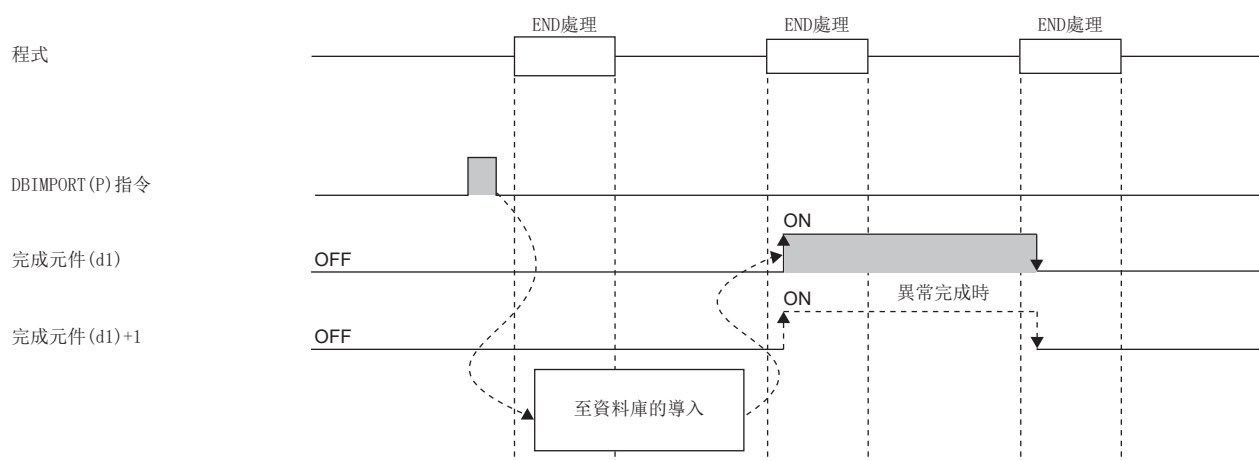
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存Unicode文字檔案名的起始元件 例：“2:\database1\recipe\recipe_db.txt” 字元數不超過255字元	—	Unicode字元串	ANYSTRING_DOUBLE
(d1)	完成元件(透過指令完成1個掃描ON的起始元件) • (d1)+0: 完成信號 • (d1)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d2)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d1)	○	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 對(s)中指定的Unicode文字檔案中設定的資料進行提取、構建資料庫。在DBIMPORT(P)指令中使用的Unicode文字檔案中預先定義表格配置等。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
- 存在有其它打開中的資料庫的情況下，如果執行DBIMPORT(P)指令將異常完成。
- 正常完成時，完成元件(d1)+0的完成信號將變為ON、(d2)的完成狀態中將儲存0。
- 執行DBIMPORT(P)指令時的完成元件的動作如下所示。



- DBIMPORT(P)指令執行中，SM753(檔案存取中)將變為ON。^{*1}SM753為ON中，無法執行DBIMPORT(P)指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
 - CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498(CPU模組內建資料庫啟動完成標誌)為OFF的期間，DBIMPORT(P)指令無法執行。^{*1}執行的情況下將變為無處理。
 - DBIMPORT(P)指令執行時，在與(s)中指定的Unicode文字檔案相同的路徑中，存在Unicode文字檔案中指定的資料庫名的情況下，刪除SD記憶卡內同名資料庫(資料庫資料夾)，並在Unicode文字檔案中新建指定的資料庫。
- ^{*1} 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述情況下，(d1)+1的異常完成信號將變為0N、(d2)的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了DBIMPORT (P) 指令的情況下
 - (s) 中指定的Unicode文字檔案不存在的情況下
 - Unicode文字檔案的欄位名行中規定的欄位數與記錄行的欄位數不一致的情況下
 - Unicode文字檔案中，沒有表格定義開始標籤及表格定義結束標籤的情況下
 - 在Unicode文字檔案的鍵限制中，設定了超出範圍的值的的情況下
 - Unicode文字檔案的資料庫名、表名、欄位名超過了32字元的情況下
 - Unicode文字檔案的表數、欄位數超過最大數的情況下
 - Unicode文字檔案的資料類型中，指定了規定以外的類型的情況下
 - Unicode文字檔案內的記錄數超過最大件數的情況下(韌體版本低於“28”的可程式控制器CPU的情況下)
 - 資料庫存取失敗的情況下
 - 資料庫名中包含禁止指定的字元的情況下
 - Unicode文字檔案中指定的資料庫名及(s) 中指定的資料夾路徑的字元數(包括驅動器路徑字元)的合計超過了128字元的情況下
 - 在存在有打開中的資料庫的狀態下執行了DBIMPORT (P) 指令的情況下
 - 註釋行的字元數超過最大字元數的情況下
 - (s) 中指定的Unicode文字檔案的資料夾名中指定了以ASCII碼能表現的字元(0020H~007EH) 以外的情況下
 - DBIMPORT (P) 指令執行時，Unicode文字檔案的資料庫名存在於SD記憶卡內，該資料庫透過其他功能被操作的情況下
- Unicode文字的格式原因中檢測出異常的情況下，DBIMPORT (P) 指令將(d1)+1的異常完成信號設為0N，在SD760、SD761 (DBIMPORT指令Unicode文字檔案異常位置) 中儲存檢測出異常的Unicode文字的行編號。

出錯

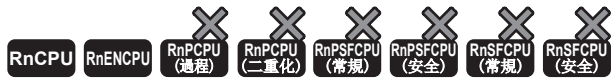
出錯代碼 (SD0)	內容
2840H	(s) 的驅動器編號中指定了2以外的數值時。
3405H	(s) 中指定的字元串(路徑)超過了255字元的情況下。

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058頁 資料庫存取指令中發生的出錯代碼

資料庫導出

DBEXPORT (P)



• R00CPU、R01CPU、R02CPU中不能使用。

將指定的資料庫中設定的資料寫入到Unicode文字檔案中。

梯形圖	ST
	<pre>ENO:=DBEXPORT (EN, s, d1, d2); ENO:=DBEXPORTP (EN, s, d1, d2);</pre>

FBD/LD



■執行條件

指令	執行條件
DBEXPORT	
DBEXPORTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存資料庫資料夾路徑的起始元件 例: "2:\database\recipe" 字元數不超過128字元	—	Unicode字元串	ANYSTRING_DOUBLE
(d1)	完成元件(透過指令完成1個掃描ON的起始元件) • (d1)+0: 完成信號 • (d1)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d2)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		Z	雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□		LT、LST、LC	LZ		K	H	E	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d1)	○	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—

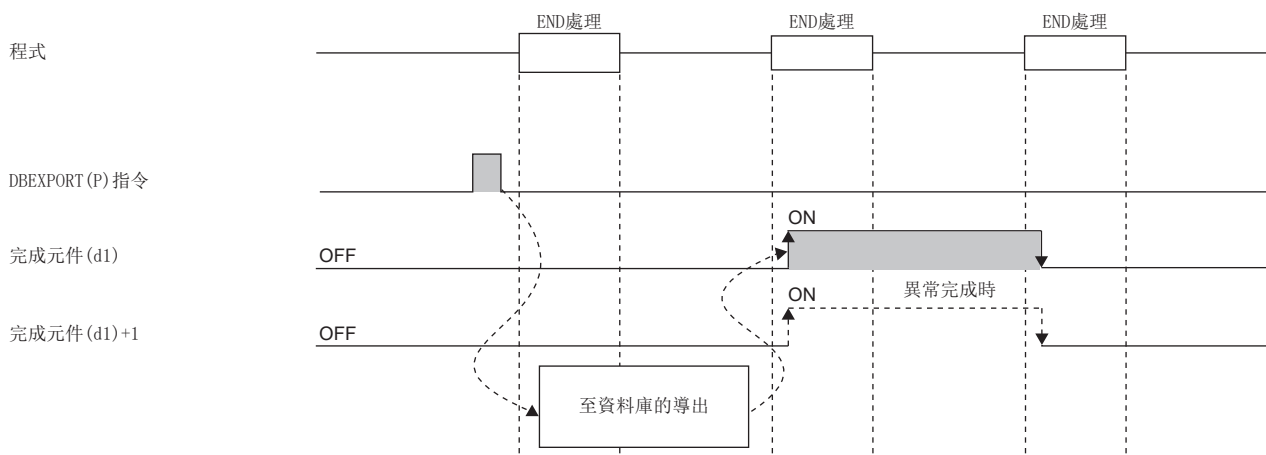
功能

- 將(s)中指定的資料庫資料夾路徑中儲存的資料庫的資料寫入到Unicode文字檔案中。
- Unicode文字檔案被建立到儲存了資料庫資料夾的資料夾中。檔案名將變為“資料庫名.txt”。已存在有Unicode文字檔案的情況下，將進行資料的覆蓋寫入。

例

資料庫的資料夾路徑為“2:\database\recipe1”的情況下，執行指令時建立Unicode文字檔案“2:\database\recipe1.txt”，並將資料寫入到該檔案中。

- 正常完成時，完成元件(d1)+0的完成信號將變為ON、(d2)的完成狀態中將儲存0。
- 執行DBEXPORT(P)指令時的完成元件的動作如下所示。



- 寫入的Unicode文字檔案的內部配置與透過DBIMPORT(P)指令導入的檔案相同。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
- DBEXPORT(P)指令執行中，SM753(檔案存取中)將變為ON。^{*1}SM753為ON中，無法執行DBEXPORT(P)指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
- CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498(CPU模組內建資料庫啟動完成標誌)為OFF的期間，DBEXPORT(P)指令無法執行。^{*1}執行的情況下將變為無處理。
- 事物處理中，不能執行DBEXPORT(P)指令。

^{*1} 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述情況下，(d1)+1的異常完成信號將變為ON、(d2)的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了DBEXPORT(P)指令的情況下
- (s)中指定的路徑不是資料庫的情況下
- 資料庫存取失敗，至Unicode文字的資料寫入失敗的情況下
- (s)中指定的路徑的字元數(包括驅動器路徑字元)超過了128字元的情況下
- 韌體版本低於“28”的可程式控制器CPU中，指定了(s)中韌體版本為“28”及其以後的可程式控制器CPU中建構或操作的資料庫的情況下
- 事物處理中，執行了DBEXPORT(P)指令的情況下

出錯

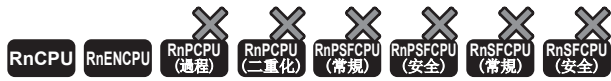
出錯代碼(SD0)	內容
2840H	(s)的驅動器編號中指定了2以外的數值時。

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058頁 資料庫存取指令中發生的出錯代碼

資料庫連接

DBOPEN (P)

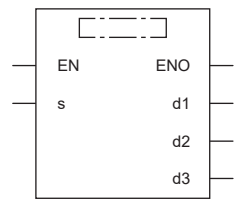


• R00CPU、R01CPU、R02CPU中不能使用。

連接 (s) 中指定的資料夾中儲存的資料庫，將資料庫設為允許使用狀態。

梯形圖	ST
	ENO:=DBOPEN (EN, s, d1, d2, d3) ; ENO:=DBOPENP (EN, s, d1, d2, d3) ;

FBD/LD



■執行條件

指令	執行條件
DBOPEN	
DBOPENP	

設置數據

■內容、範圍、資料類型

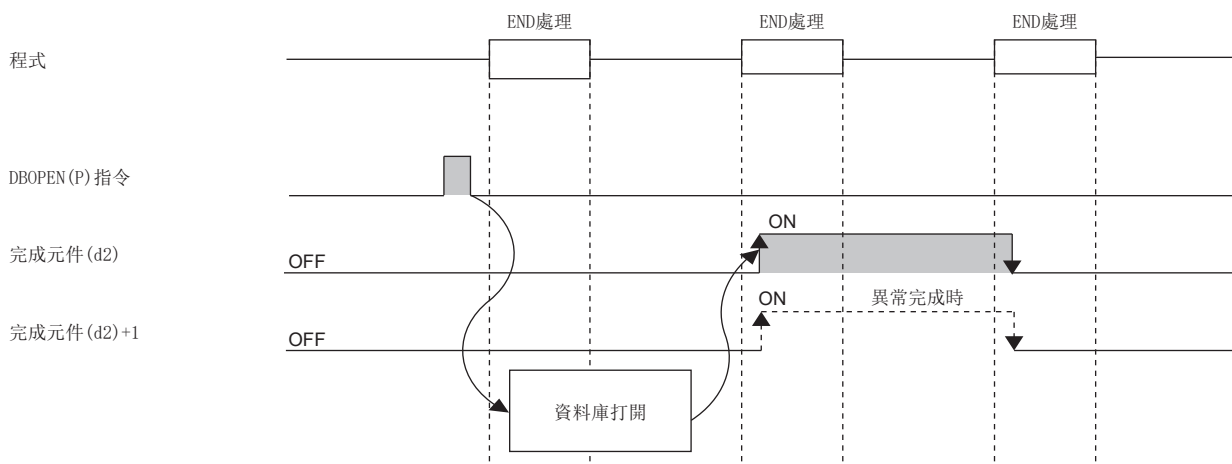
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	儲存打開對象資料庫資料夾路徑的起始元件 例: "2:\database1\recipe" 字元數不超過128字元	—	Unicode字元串	ANYSTRING_DOUBLE
(d1)	資料庫識別編號	1~4	帶符號BIN16位元	ANY16
(d2)	完成元件 (透過指令完成1個掃描ON的起始元件) • (d2)+0: 完成信號 • (d2)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d3)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成 (出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	—	—	○	—	—	○	—
(d1)	—	—	○	—	—	—	○	—	—	—	—
(d2)	○	—	○	—	—	—	○	—	—	—	—
(d3)	—	—	○	—	—	—	○	—	—	—	—

功能

- 連接(s)中指定的資料夾路徑中儲存的資料庫，將資料庫設為允許使用狀態。
- 儲存目標是在“(驅動器編號)：(資料庫資料夾的路徑)”中指定。驅動器編號固定為2(SD記憶卡)。
- 正常完成時，將連接的資料庫的識別編號儲存到(d1)中，完成元件(d2)+0的完成信號將變為ON，(d3)的完成狀態中將儲存0。
- 執行DBOPEN(P)指令時的完成元件的動作如下所示。



- 在DBOPEN(P)指令中，最多可以連接4個不同的資料庫。
- 事物處理過程中，不能打開新的資料庫。
- DBOPEN(P)指令執行中，SM753(檔案存取中)將變為ON。^{*1}SM753為ON中，無法執行DBOPEN(P)指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
- CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498(CPU模組內建資料庫啟動完成標誌)為OFF的期間，DBOPEN(P)指令無法執行。^{*1}執行的情況下將變為無處理。

*1 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述情況下，(d2)+1的異常完成信號將變為ON、(d3)的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了DBOPEN(P)指令的情況下
- (s)中指定的儲存位置不存在的情況下
- 對已連接的資料庫試圖再次進行連接的情況下
- 在超出可同時連接的資料庫個數時執行了DBOPEN(P)指令時的情況下
- (s)中指定的路徑的字元數(包括驅動器路徑字元)超過了128字元的情況下
- 事物處理中打開了資料庫的情況下

出錯

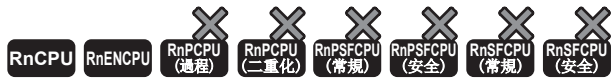
出錯代碼(SD0)	內容
2820H	(d2)中指定的範圍超出了相應元件/標籤的情況下。
2840H	(s)的驅動器編號中指定了2以外的數值時。

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058頁 資料庫存取指令中發生的出錯代碼

資料庫斷開

DBCLOSE (P)

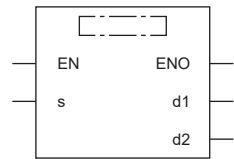


• R00CPU、R01CPU、R02CPU中不能使用。

解除與指定資料庫的連接。

梯形圖	ST
	ENO:=DBCLOSE (EN, s, d1, d2) ; ENO:=DBCLOSEP (EN, s, d1, d2) ;

FBD/LD



■執行條件

指令	執行條件
DBCLOSE	
DBCLOSEP	

設置數據

■內容、範圍、資料類型

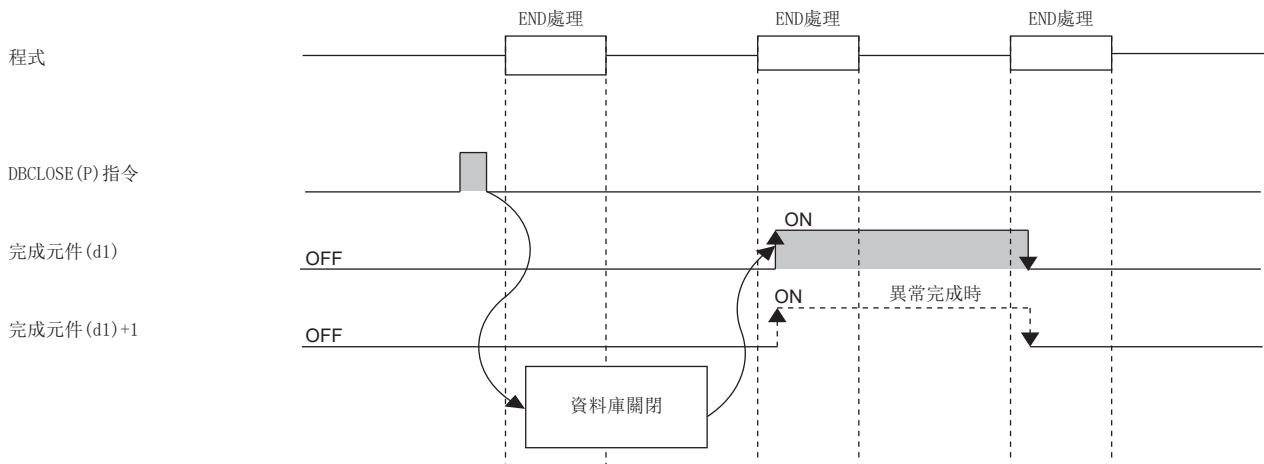
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	資料庫識別編號	1~4	帶符號BIN16位元	ANY16
(d1)	完成元件(透過指令完成1個掃描ON的起始元件) • (d1)+0: 完成信號 • (d1)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d2)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	○	—	○	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	

功能

- 斷開與(s)中指定的資料庫識別編號對應的資料庫的連接。
- 正常完成時，完成元件(d1)+0的完成信號將變為ON、(d2)的完成狀態中將儲存0。
- 執行DBCLOSE(P)指令時的完成元件的動作如下所示。



- 透過DBTRANS(P)指令開始了事物處理的狀態下，若執行DBCMMIT(P)指令或DBROLBAK(P)指令之前執行了DBCLOSE(P)指令時，在執行DBCLOSE(P)指令時的狀態下確定事物處理。
- DBCLOSE(P)指令執行中，SM753(檔案存取中)將變為ON。^{*1}SM753為ON中，無法執行DBCLOSE(P)指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
- CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498(CPU模組內建資料庫啟動完成標誌)為OFF的期間，DBCLOSE(P)指令無法執行。^{*1}執行的情況下將變為無處理。

*1 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述情況下，完成元件(d1)+1的異常完成信號將變為ON、(d2)的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了DBCLOSE(P)指令的情況下
- (s)中指定的識別編號是已斷開的資料庫的情況下
- (s)中指定了超出範圍的識別編號的情況下

出錯

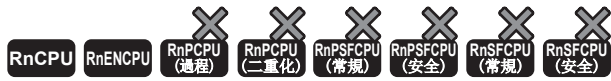
出錯代碼(SDO)	內容
2820H	(d1)中指定的範圍超出了相應元件/標籤的情況下。

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058頁 資料庫存取指令中發生的出錯代碼

資料庫記錄添加

DBINSERT (P)

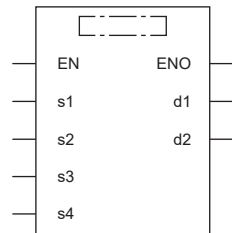


• R00CPU、R01CPU、R02CPU中不能使用。

對設定的識別編號所對應的資料庫的表格添加記錄。

梯形圖	ST
	<pre>ENO:=DBINSERT (EN, s1, s2, s3, s4, d1, d2); ENO:=DBINSERTP (EN, s1, s2, s3, s4, d1, d2);</pre>

FBD/LD



■執行條件

指令	執行條件
DBINSERT	
DBINSERTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	資料庫識別編號	1~4	帶符號BIN16位元	ANY16
(s2)	儲存資料庫的表名的起始元件*1	—	Unicode字元串	ANYSTRING_DOUBLE
(s3)	儲存資料庫的欄位名的起始元件*1	—	字元	ANY16*2
(s4)	儲存插入資料的起始元件	—	字元	ANY16*2
(d1)	完成元件(透過指令完成1個掃描ON的起始元件) • (d1)+0: 完成信號 • (d1)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d2)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 表名和欄位名區別大寫字母與小寫字母進行設定。

*2 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	○	—	
(s3)	—	—	○	—	—	—	○	—	—	—	—	
(s4)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	○	—	○	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	

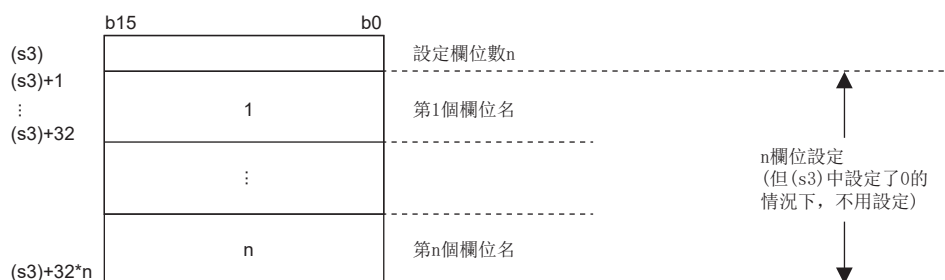
■資料庫的欄位名

操作數：(s3)				
元件	項目	內容	設定範圍	設定方
+0	欄位數	欄位數設定希望添加值的欄位的個數。 應設定為不超過(s2)中指定的表內的全部欄位數的值。 指定了0的情況下，表的全部欄位將成為輸出對象。 ^{*1}	0~128 ^{*2}	用戶
+1~+□	欄位名	設定各欄位的欄位名。欄位名固定為32字元，以Unicode字元串按欄位數進行設定。 不足32字元的情況下將向前填充對齊，以0000H填充為32字元。 根據欄位數，(s3)的最終地址□將變化。 □=32×n(n: 欄位數) (s3)+0中指定了0的情況下，不需要設定此項目。 ^{*1} (即使設定也將忽略。)	—	用戶

*1 以韌體版本為“28”及其以後的可程式控制器CPU為對象。

*2 韌體版本低於“28”的可程式控制器CPU的情況下，變為1~16。

(s3)的格式如下所示。

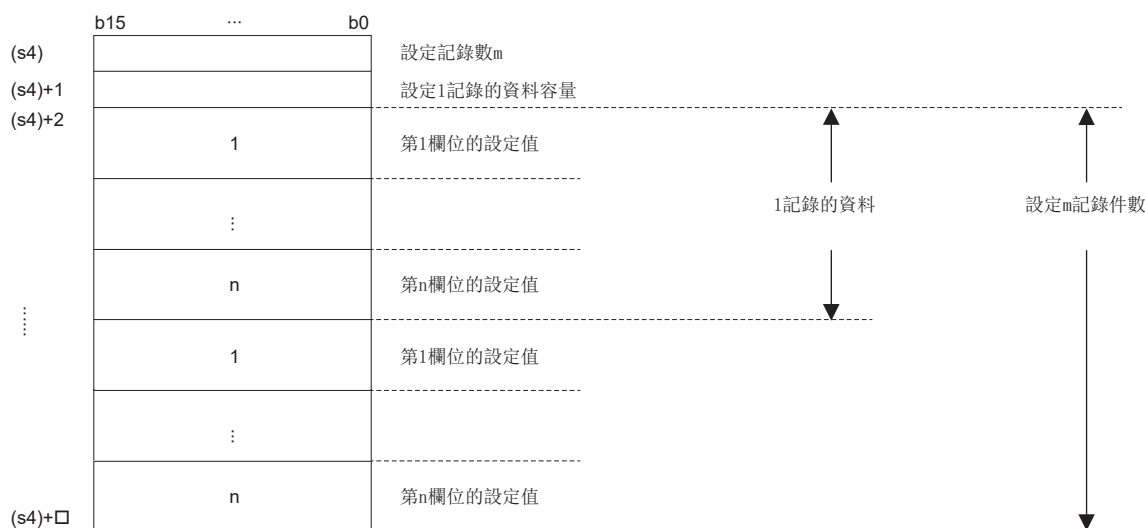


■插入資料

操作數：(s4)				
元件	項目	內容	設定範圍	設定方
+0	記錄數	設定添加的記錄個數。 添加的記錄個數，為滿足下述條件式的值。 ^{*1} 記錄個數(m) ≤ (32768/1記錄份的資料容量)	1~m ^{*1}	用戶
+1	容量	設定每個記錄的容量。	根據資料類型	用戶
+2~+□	添加的值	將(s3)中指定的欄位數的資料按照(s4)+0中指定的記錄個數(m)進行設定。	根據資料類型	用戶

*1 韌體版本低於“28”的可程式控制器CPU的情況下，變為1~16。

(s4)的格式如下所示。設定(s3)中設定的各欄位對應的設定值。



- 各值的資料容量取決於各欄位的資料類型的容量。

欄位的資料類型	欄位的資料類型資料容量(單位: 字)
BOOL	1
WORD	1
DWORD	2
INT	1
DINT	2
REAL	2
LREAL	4
STRING	<ul style="list-style-type: none"> • 字元數為偶數的情況下: $(\text{字元數} \div 2) + 1$ • 字元數為奇數的情況下: 字元數$\div 2$的進位 [例] STRING: 32的情況下: $(32 \div 2) + 1 = 17$ STRING: 15的情況下: $(15 \div 2) = 7.5$ 的進位 $\rightarrow 8$
WSTRING	字元數+1 [例] WSTRING: 32的情況下: $32 + 1 = 33$

功能

- 對(s1)中設定的識別編號所對應的資料庫的(s2)中設定的表格添加記錄。
- 在(s3)中指定添加的記錄的欄位個數、欄位名。對於添加的欄位名，不需要指定配置表的全部欄位。在不指定的欄位中儲存NULL。
- 在(s4)中指定添加的記錄的個數、每個記錄的容量及值。
- 透過DBINSERT(P)指令在表recipeA中添加1個記錄的情況如下所示。

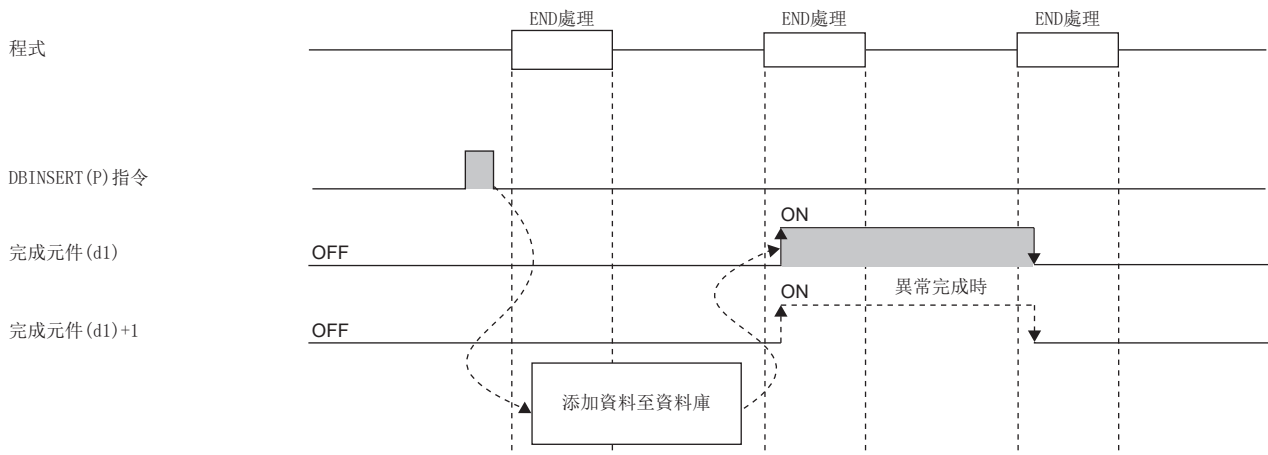
(1)	(2)	(3)
1	ProductA	100
2	ProductB	200

(1): 欄位1 WORD型

(2): 欄位2 WSTRING型(最多16字元)

(3): 欄位3 INT型

- 正常完成時，完成元件(d1)+0的完成信號將變為ON、(d2)的完成狀態中將儲存0。
- 異常完成時，完成元件(d1)+1的異常完成信號將變為ON、(d2)的完成狀態中將儲存出錯代碼。
- 執行DBINSERT(P)指令時的完成元件的動作如下所示。



- DBINSERT(P)指令執行中，SM753(檔案存取中)將變為ON。^{*2}SM753為ON中，無法執行DBINSERT(P)指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
- CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498(CPU模組內建資料庫啟動完成標誌)為OFF的期間，DBINSERT(P)指令無法執行。^{*2}執行的情況下將變為無處理。

*2 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述情況下，完成元件 (d1)+1 的異常完成信號將變為 ON、(d2) 的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了 DBINSERT (P) 指令的情況下
- (s1) 中指定了超出範圍的識別編號的情況下
- (s1) 中指定了未打開的識別編號的情況下
- 韌體版本低於“28”的可程式控制器 CPU 中，指定了 (s1) 中韌體版本為“28”及其以後的可程式控制器 CPU 中建構或操作的資料庫的情況下
- (s2) 中指定的表名不存在的情況下
- (s2) 中指定的表名的字元數超過了 32 字元的情況下
- (s3) 中添加的欄位數設定了超出範圍的值的的情況下
- (s3)+1 ~ (s3)+□ 中指定的欄位名中，設定了超出範圍的欄位名的情況下
- (s4) 中添加的記錄數設定了超出範圍的值的的情況下
- 資料庫的 INSERT 處理失敗的情況下
- (s4)+2 中設定的 1 個記錄的資料範圍與 (s4)+1 中指定的容量不一致的情況下

出錯

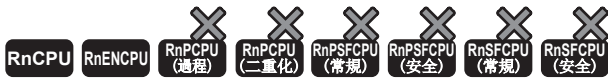
出錯代碼 (SD0)	內容
2820H	(s2)、(s3)、(s4)、(d1) 中指定的範圍超出了相應元件/標籤的情況下。

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058 頁 資料庫存取指令中發生的出錯代碼

資料庫記錄更新

DBUPDATE (P)

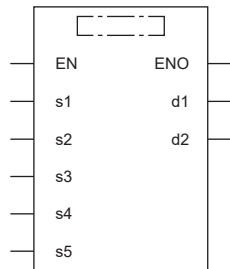


• R00CPU、R01CPU、R02CPU中不能使用。

在設定的識別編號對應的資料庫的指定表中，對符合指定條件的所有記錄進行更新。

梯形圖	ST
	<pre>ENO:=DBUPDATE (EN, s1, s2, s3, s4, s5, d1, d2); ENO:=DBUPDATEP (EN, s1, s2, s3, s4, s5, d1, d2);</pre>

FBD/LD



■執行條件

指令	執行條件
DBUPDATE	
DBUPDATEP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	資料庫識別編號	1~4	帶符號BIN16位元	ANY16
(s2)	儲存更新對象資料庫的表名的起始元件*1	—	Unicode字元串	ANYSTRING_DOUBLE
(s3)	儲存更新對象資料庫的欄位名的起始元件*1	—	字元	ANY16*2
(s4)	儲存更新後資料的起始元件	—	字元	ANY16*2
(s5)	儲存更新條件(最多2個)的起始元件	—	字元	ANY16*2
(d1)	完成元件(透過指令完成1個掃描ON的起始元件) • (d1)+0: 完成信號 • (d1)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d2)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 表名和欄位名區別大寫字母與小寫字母進行設定。

*2 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	○	—	
(s3)	—	—	○	—	—	—	○	—	—	—	—	
(s4)	—	—	○	—	—	—	○	—	—	—	—	
(s5)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	○	—	○	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	

■更新對象的欄位名

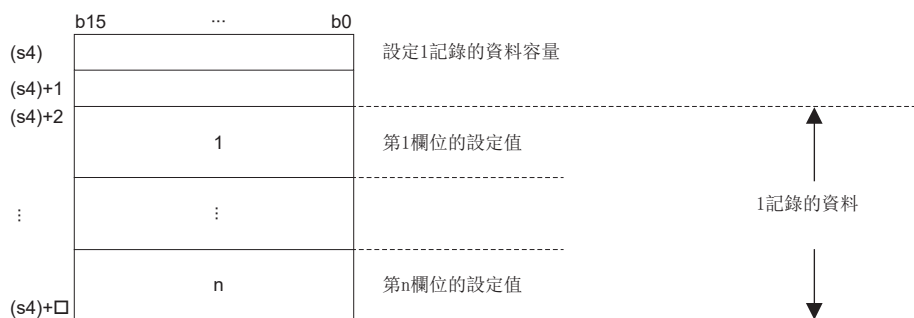
操作數：(s3)				
元件	項目	內容	設定範圍	設定方
+0	欄位數	設定更新對象的欄位數。	1~128*1	用戶
+1~+□	欄位名	設定各欄位的欄位名。欄位名固定為32字元，以Unicode字元串按欄位數進行設定。不足32字元的情況下將向前填充對齊，以0000H填充為32字元。根據欄位數，(s3)的最終地址□將變化。 □=32×n (n：欄位數)	—	用戶

*1 韌體版本低於“28”的可程式控制器CPU的情況下，變為1~16。

■更新後的資料

操作數：(s4)				
元件	項目	內容	設定範圍	設定方
+0	資料容量	設定更新對象欄位的資料容量。	—	用戶
+1	未使用	—	—	—
+2~+□	設定值	設定更新後的資料。	根據資料類型而變化。	用戶

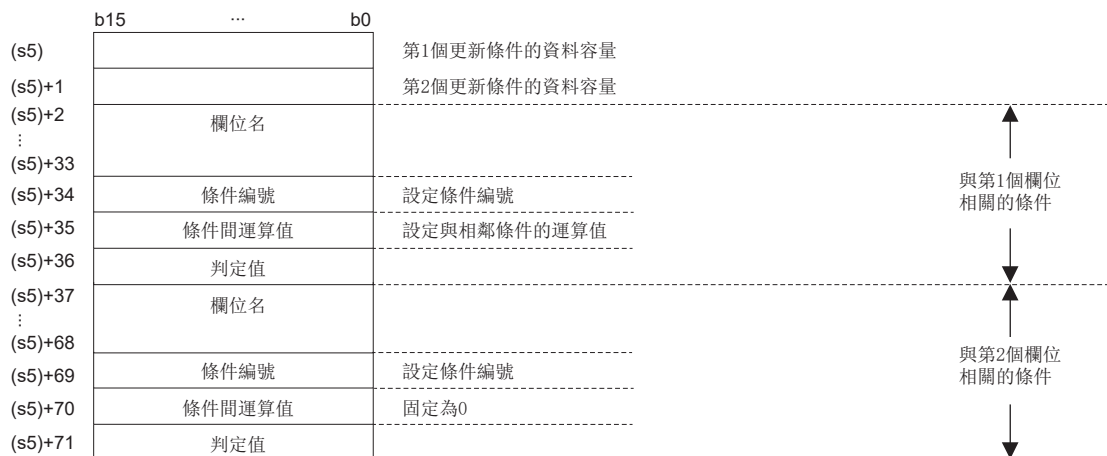
(s4) 的設定格式如下所示。設定對(s3)的各欄位的設定值。各設定值的資料容量取決於各欄位的資料類型的容量。(參閱 1027 頁 DBINSERT(P))



■更新條件

操作數：(s5)				
元件	項目	內容	設定範圍	設定方
+0	第1個條件的資料容量	設定第1個更新條件的資料容量。	1~125	用戶
+1	第2個條件的資料容量	設定第2個更新條件的資料容量。 • 0: 無條件 • 0以外: 資料容量	0~125	用戶
+2~+□	更新條件	設定更新條件。根據判定值的資料類型，(s5)的最終地址將變化。 (s5)+2 (1) ⋮ (s5)+33 (s5)+34 (2) (s5)+35 (3) (s5)+36 (4) ⋮ (s5)+□ ■欄位名(1) 固定為32字元，以Unicode字元串進行設定。 不足32字元的情況下將向前填充對齊，以0000H填充為32字元。 ■條件編號(2) 設定表示更新條件的編號。 • 1: = (等於判定值) • 2: != (判定值以外) • 3: < (小於判定值) • 4: > (大於判定值) • 5: =< (小於等於判定值) • 6: >= (大於等於判定值) • 7: is NULL (是NULL(未設定值)) • 8: is not NULL (不是NULL(設定了值)) ■條件之間運算值(3) 設定與相鄰條件的運算值。 • 0: 無設定 • 1: AND • 2: OR 更新條件為1個的情況下，將第1個更新條件的條件之間運算值設定為0。此時，即使設定第2個更新條件的內容也將被忽略。 第2個條件的條件之間運算值必須設定為0。 ■判定值(4) 設定用於更新條件判定的值。 條件編號為7或8的情況下，也準備相當於資料容量的判定值的區域。	—	用戶

(s5)的設定格式如下所示。(資料類型為WORD的情況下)按照更新條件數設定欄位名、條件編號、條件之間運算值、判定值。判定值的資料容量取決於各欄位的資料類型的容量。(☞1027頁 DBINSERT(P))



例

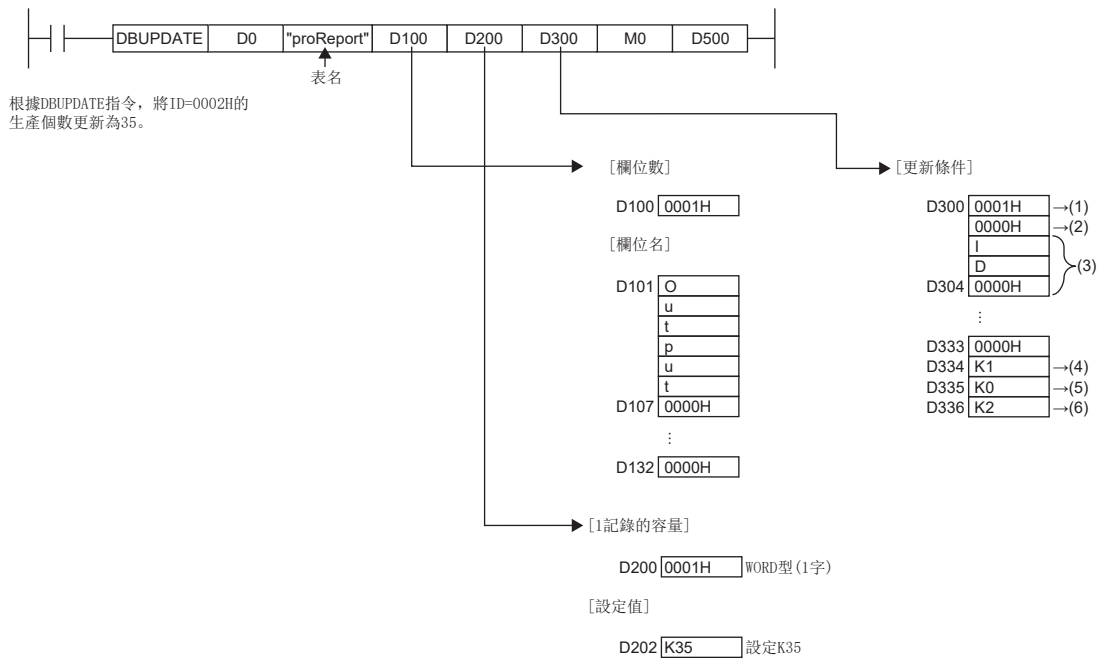
使用DBUPDATE(P)，將ID=0002H的記錄的Output更新為35的情況下

[產品生產結果表 (proReport)]

ID	Product	Output
0001H	“產品名1”	32
0002H	“產品名2”	35
0003H	“產品名3”	38
⋮		



[程式]



- (1) 第1個條件的資料容量→WORD型(1字)
- (2) 因為沒有第2個條件，因此變為0。
- (3) 更新條件(第1個)的欄位名
- (4) 因為條件編號為“=”，因此設定K1。
- (5) 因為沒有條件之間運算值，因此設定K0。
- (6) 判定值設定為K2。

功能

- 在透過 (s1) 中指定的識別編號指定的資料庫內的 (s2) 中指定的表中，對符合 (s5) 中指定的條件的所有記錄進行更新。
- 在 (s3) 中指定更新的記錄的欄位名。雖然無需指定表中的所有欄位，但最少需要指定1個。
- 在 (s4) 中指定更新的記錄的值。(s3) 中未指定的欄位的設定值將不被更新。
- 在 (s5) 中指定更新的條件。最少需要設定1個條件，最多可以設定2個。
- 根據DBUPDATE(P) 指令，更新對象的欄位名為欄位2，將更新後的資料設定為New-Product1，對符合更新條件欄位1= “2” 的表recipeA的記錄進行更新時，其情況如下所示。

表格recipeA(更新前)

欄位1 WORD型	欄位2 WSTRING型(最大16字元)	欄位3 INT型
1	ProductA	100
2	ProductB	200
3	ProductC	300

表格recipeA(更新後)

欄位1 WORD型	欄位2 WSTRING型(最大16字元)	欄位3 INT型
1	ProductA	100
2	New-Product1	200
3	ProductC	300

- 根據DBUPDATE(P) 指令，更新對象的欄位名為欄位2，將更新後的資料設定為New-Product1，對符合更新條件欄位1>= “2”、欄位1<= “3” 的表recipeA的記錄進行更新時，其情況如下所示。

表格recipeA(更新前)

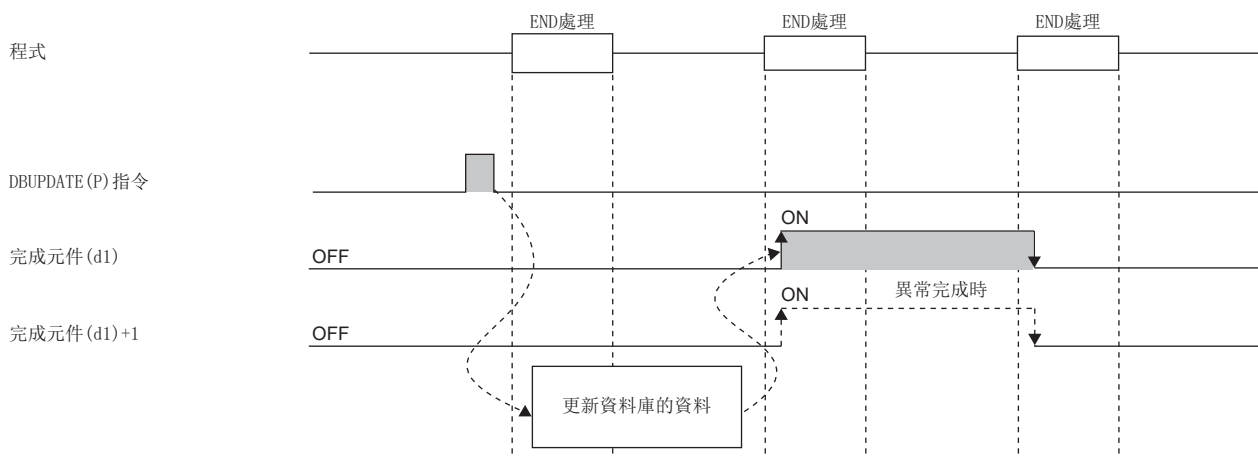
欄位1 WORD型	欄位2 WSTRING型(最大16字元)	欄位3 INT型
1	ProductA	100
2	ProductB	200
3	ProductC	300
4	ProductD	200

表格recipeA(更新後)

欄位1 WORD型	欄位2 WSTRING型(最大16字元)	欄位3 INT型
1	ProductA	100
2	New-Product1	200
3	New-Product1	300
4	ProductD	400

- 正常完成時，完成元件 (d1)+0的完成信號將變為ON、(d2) 的完成狀態中將儲存0。

• 執行DBUPDATE(P)指令時的完成元件的動作如下所示。



- DBUPDATE(P)指令執行中，SM753(檔案存取中)變為ON。^{*1}SM753為ON中，無法執行DBUPDATE(P)指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
 - CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498(CPU模組內建資料庫啟動完成標誌)為OFF的期間，DBUPDATE(P)指令無法執行。^{*1}執行的情況下將變為無處理。
- ^{*1} 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述情況下，完成元件(d1)+1的異常完成信號將變為ON、(d2)的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了DBUPDATE(P)指令的情況下
- (s1)中指定了超出範圍的識別編號的情況下
- (s1)中指定了未打開的識別編號的情況下
- 韌體版本低於“28”的可程式控制器CPU中，指定了(s1)中韌體版本為“28”及其以後的可程式控制器CPU中建構或操作的資料庫的情況下
- (s2)中指定的表名不存在的情況下
- (s2)中指定的表名的字元數超過了32字元的情況下
- (s3)中對更新的欄位數設定了超出範圍的值的的情況下
- (s3)+1~(s3)+□中指定的欄位名中，設定了超出範圍的欄位名的情況下
- (s5)中對條件容量設定了超出範圍的值的的情況下
- 將(s5)的第1個條件容量設定為0的情況下
- (s5)中對條件符號設定了超出範圍的值的的情況下
- (s5)中對條件之間運算值設定了超出範圍的值的的情況下
- 資料庫的UPDATE處理失敗的情況下
- (s5)+35中指定的條件之間運算值為1或2時，在(s5)+1中設定了0的情況下
- (s4)+2中設定的1個記錄的資料範圍與(s4)中指定的容量不一致的情況下
- (s5)+2中未設定應設定的欄位名的情況下

出錯

出錯代碼(SD0)	內容
2820H	(s2)、(s3)、(s4)、(s5)、(d1)中指定的範圍超出了相應元件/標籤的情況下。

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058頁 資料庫存取指令中發生的出錯代碼

資料庫記錄搜尋

DBSELECT (P)

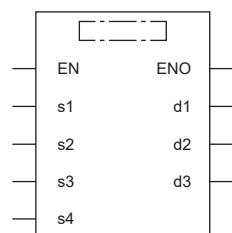


• R00CPU、R01CPU、R02CPU中不能使用。

對指定的識別編號對應的資料庫的指定表的記錄進行搜尋。

梯形圖	ST
	<pre>ENO:=DBSELECT (EN, s1, s2, s3, s4, d1, d2, d3); ENO:=DBSELECTP (EN, s1, s2, s3, s4, d1, d2, d3);</pre>

FBD/LD



■執行條件

指令	執行條件
DBSELECT	
DBSELECTP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	資料庫識別編號	1~4	帶符號BIN16位元	ANY16
(s2)	儲存搜尋對象資料庫的表名的起始元件*1	—	Unicode字元串	ANYSTRING_DOUBLE
(s3)	儲存搜尋對象資料庫的欄位名的起始元件*1	—	字元	ANY16*2
(s4)	儲存搜尋條件(最多2個)的起始元件	—	字元	ANY16*2
(d1)	搜尋結果	—	字元	ANY16*2
(d2)	完成元件(透過指令完成1個掃描ON的起始元件) • (d2)+0: 完成信號 • (d2)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d3)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 表名和欄位名區別大寫字母與小寫字母進行設定。

*2 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—
(s4)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	—	○	—	—	—	—	○	—	—	—	—
(d3)	—	—	○	—	—	—	—	○	—	—	—	—

■搜尋對象的表名

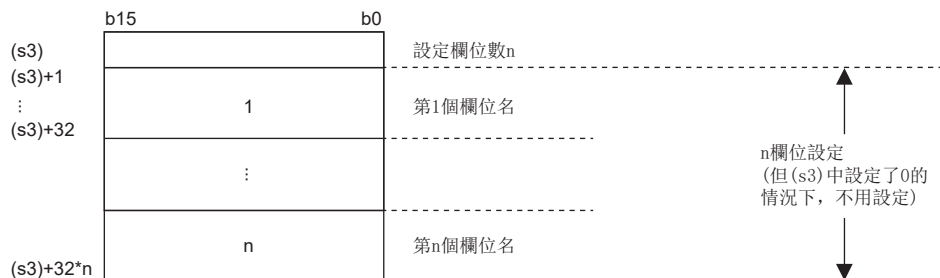
操作數：(s2)				
元件	項目	內容	設定範圍	設定方
+0~+□	表名	將搜尋對象表名設定為32字元及其以內。	—	用戶

■搜尋對象的欄位名

操作數：(s3)				
元件	項目	內容	設定範圍	設定方
+0	欄位數	設定搜尋對象的欄位數。 指定了0的情況下，表的全部欄位將成為輸出對象。	0~128*1	用戶
+1~+□	欄位名	將搜尋對象的各欄位名以32字元進行設定。根據欄位數，(s3)的最終地址將變化。 □=33×n 欄位名固定為32字元，在Unicode字元串中按欄位數進行設定。不足32字元的情況下將向前填充對齊，以0000H填充為32字元。 (s3)+0中指定了0的情況下，不需要設定此項目。(即使設定也將忽略。)	—	用戶

*1 韌體版本低於“28”的可程式控制器CPU的情況下，變為0~16。

(s3)的格式如下所示。



■搜尋條件

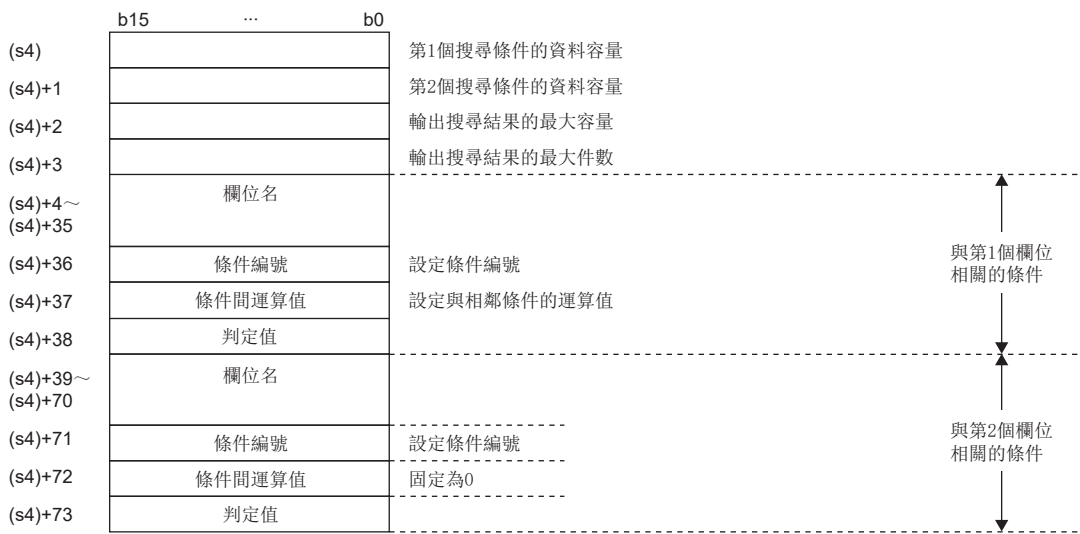
操作數：(s4)																					
元件	項目	內容	設定範圍	設定方	資料類型																
+0	第1個條件的資料容量	將(s4)+4中設定的第1個搜尋條件的資料容量以字單位進行設定。將(s4)設定為0時，表的全部記錄將成為輸出對象。 <ul style="list-style-type: none"> • 0：無條件 • 0以外：欄位的資料容量 (將(s4)設定為0的情況下，(s4)+1也設定為0。) 	0~125	用戶	WORD																
+1	第2個條件的資料容量	設定(s4)+4中設定的第2個搜尋條件的資料容量。 <ul style="list-style-type: none"> • 0：無條件 • 0以外：欄位的資料容量 	0~125	用戶																	
+2	最大輸出容量	設定搜尋結果(d1)的最大輸出容量。 <ul style="list-style-type: none"> • 0：預設值(1024字) 	0~32768*1	用戶																	
+3	最大輸出個數	最大輸出個數設定搜尋結果(d1)的最大輸出個數。 <ul style="list-style-type: none"> • 0：預設的輸出個數(16個) 	0~64	用戶																	
+4~+□	搜尋條件	設定搜尋條件。根據判定值的資料類型，(s4)的最終地址將變化。 <table border="1" style="margin-left: 20px;"> <tr><td>(s4)+4</td><td>(1)</td></tr> <tr><td>⋮</td><td></td></tr> <tr><td>(s4)+35</td><td></td></tr> <tr><td>(s4)+36</td><td>(2)</td></tr> <tr><td>(s4)+37</td><td>(3)</td></tr> <tr><td>(s4)+38</td><td>(4)</td></tr> <tr><td>⋮</td><td></td></tr> <tr><td>(s4)+□</td><td></td></tr> </table> <p>■欄位名(1) 固定為Unicode字元串32字元，按欄位數進行設定。 不足32字元的情況下將向前填充對齊，以0000H填充為32字元。</p> <p>■條件編號(2) 設定表示搜尋條件的編號。 <ul style="list-style-type: none"> • 1：= (等於判定值) • 2：!= (判定值以外) • 3：< (小於判定值) • 4：> (大於判定值) • 5：<= (小於等於判定值) • 6：>= (大於等於判定值) • 7：is NULL (是NULL(未設定值)) • 8：is not NULL (不是NULL(設定了值)) </p> <p>■條件之間運算值(3) 設定與相鄰條件的運算值。 <ul style="list-style-type: none"> • 0：無設定 • 1：AND • 2：OR 搜尋條件為1個的情況下，將第1個搜尋條件的條件之間運算值設定為0。此時，即使設定第2個搜尋條件的內容也將被忽略。 第2個條件的條件之間運算值必須設定為0。 </p> <p>■判定值(4) 設定更新條件的判定值。 條件編號為7或8的情況下，也準備相當於資料容量的判定值的區域。</p>	(s4)+4	(1)	⋮		(s4)+35		(s4)+36	(2)	(s4)+37	(3)	(s4)+38	(4)	⋮		(s4)+□		—	用戶	
(s4)+4	(1)																				
⋮																					
(s4)+35																					
(s4)+36	(2)																				
(s4)+37	(3)																				
(s4)+38	(4)																				
⋮																					
(s4)+□																					

*1 韌體版本低於“28”的可程式控制器CPU的情況下，變為0~3072。

(s4) 的設定格式如下所示。(資料類型為WORD的情況下)

按更新條件數設定欄位名、條件編號、條件之間運算值、判定值。判定值的資料容量取決於各欄位的資料類型的容量。

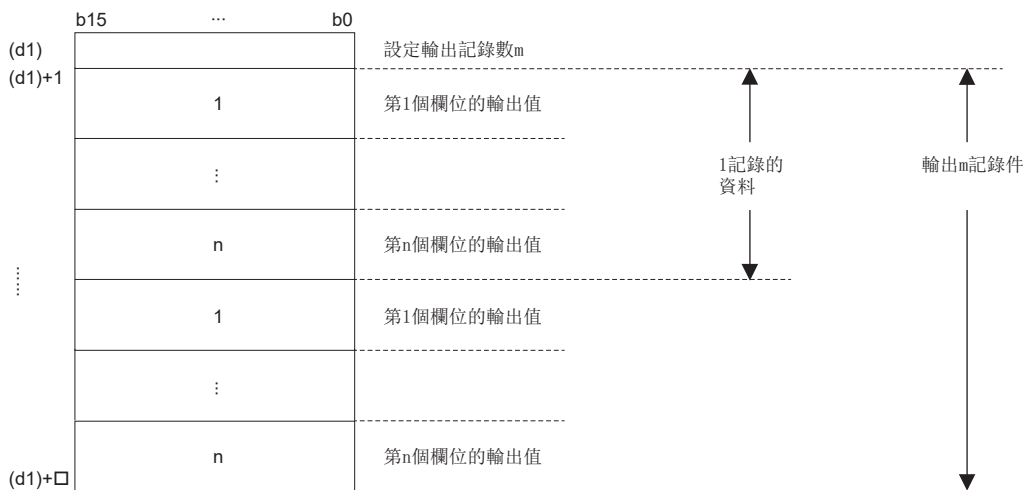
(☞ 1027頁 DBINSERT (P))



■搜尋結果

操作數: (d1)					
元件	項目	內容	設定範圍	設定方	資料類型
+0	輸出個數	符合 (s4) 中設定的條件的記錄的個數將被輸出。 輸出個數不超過 (s4)+3 中設定的最大輸出個數，且超過了最大輸出個數的記錄將不被輸出。	0~64	系統	WORD
+1~+□	輸出值	符合 (s4) 中設定的條件的記錄的值將被輸出。 對於記錄，在 (d1) ~ (d1)+(最大輸出容量) 的範圍內向前填充對齊後輸出，未容納在範圍內的記錄將不被輸出。	根據資料 類型而變 化。	系統	

(d1) 的設定格式如下所示。根據 (s3) 中指定的欄位對應的資料類型進行輸出。



例

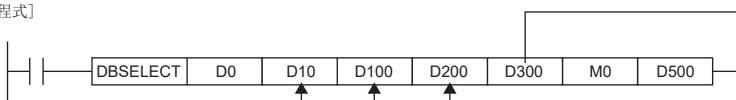
使用DBSELECT (P) 指令，獲取ID=0001H的Size(z)的情況下

[產品資訊表 (proInfo)]

ID	Product	Size(x)	Size(y)	Size(z)
0001H	“產品名1”	80	100	60
0002H	“產品名2”	40	90	40
⋮				



[程式]



透過DBSELECT指令取得ID=0001H的Size(z)的值。

[輸出結果]

D300 0001H 1件

[搜尋值]

D301 K60 →(9)

[表格名]

D10	p
	r
	o
	l
	n
	f
	o
D17	0000H
	⋮
D42	0000H

[欄位數]

D100	0001H
	⋮
D101	S
	i
	z
	e
	(
	z
)
D108	0000H
	⋮
D132	0000H

[欄位名]

[搜尋條件]

設定產品ID=0001H資料的讀取條件。

D200	0001H	→(1)
D201	0000H	→(2)
D202	K20	→(3)
D203	K0	→(4)
D204	I	} (5)
D206	D	
	0000H	
	⋮	
D235	0000H	
D236	K1	→(6)
D237	K0	→(7)
D238	K1	→(8)

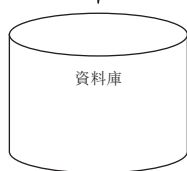
- (1) 第1個條件的資料容量→WORD型(1字)
- (2) 因為沒有第2個條件，因此變為0。
- (3) 最大輸出容量→20字
- (4) 最大輸出個數→預設(0)
- (5) 搜尋條件(第1個)的欄位名
- (6) 因為條件編號為“=”，因此設定K1。
- (7) 因為沒有條件之間運算值，因此設定K0。
- (8) 將判定值設定為K1。
- (9) 以WORD型(1字)輸出K60。

例

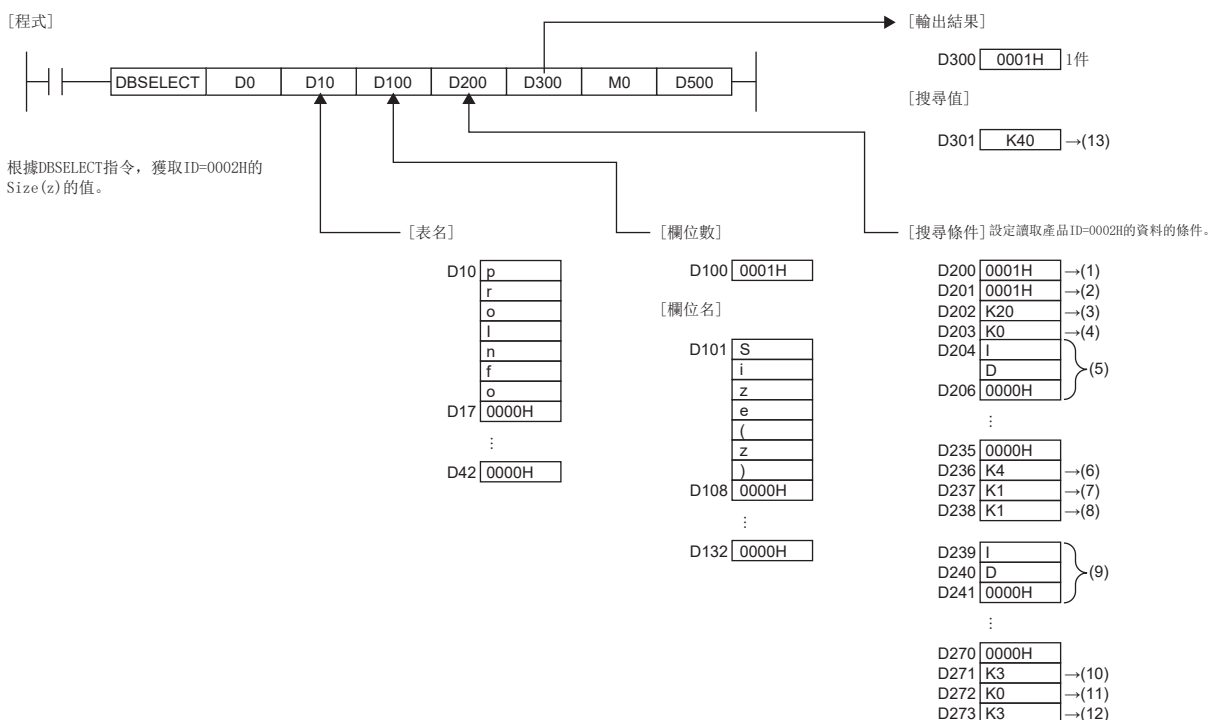
使用DBSELECT (P) 指令而獲取大於ID=0001H、小於ID=0003H的Size (z) 的值的的情況下

[產品資訊表 (proInfo)]

ID	Product	Size(x)	Size(y)	Size(z)
0001H	“產品名1”	80	100	60
0002H	“產品名2”	40	90	40
0003H	“產品名3”	60	110	50
⋮				



[程式]



- (1) 第1個條件的資料容量→WORD型 (1字)
- (2) 第2個條件的資料容量→WORD型 (1字)
- (3) 最大輸出容量→20字
- (4) 最大輸出個數→預設 (0)
- (5) 搜尋條件 (第1個) 的欄位名
- (6) 因為條件編號為 “>”，因此設定K4。
- (7) 因為條件之間運算值為 “AND”，因此設定K1。
- (8) 將判定值設定為K1。
- (9) 搜尋條件 (第2個) 的欄位名
- (10) 因為條件編號為 “<”，因此設定K3。
- (11) 因為沒有條件之間運算值，因此設定K0。
- (12) 將判定值設定為K3。
- (13) 以WORD型 (1字) 輸出K40。

功能

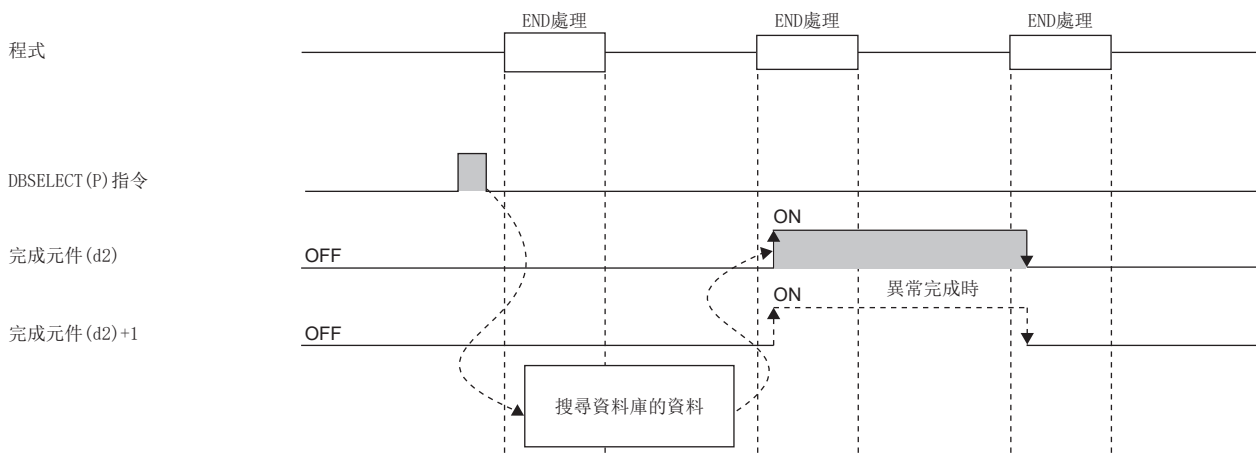
- 對(s1)中指定的識別編號對應的資料庫的(s2)中指定的表的記錄進行搜尋。最大輸出個數為64個。
- 搜尋的記錄的欄位名透過(s3)指定。
- 搜尋條件透過(s4)指定。最多可以設定2個條件。
- 搜尋結果儲存到(d1)中。
- 透過中斷程式執行的情況下也將實施記錄搜尋。
- 透過DBSELECT(P)指令，對輸出欄位為欄位2及欄位3，條件符合欄位1=2的條件的表recipeA的記錄進行搜尋並輸出時，為如下所示。

表格recipeA

欄位1 WORD型	欄位2 WSTRING型 (最大16字元)	欄位3 INT型
1	ProductA	100
2	ProductB	200
3	ProductC	300

ProductB	200
----------	-----

- 正常完成時，完成元件(d2)的完成信號將變為ON、(d3)的完成狀態中將儲存0。
- 執行DBSELECT(P)指令時的完成元件的動作如下所示。



- DBSELECT(P)指令執行中，SM753(檔案存取中)將變為ON。^{*1}SM753為ON中，無法執行DBSELECT(P)指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
- CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498(CPU模組內建資料庫啟動完成標誌)為OFF的期間，DBSELECT(P)指令無法執行。^{*1}執行的情況下將變為無處理。

^{*1} 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述的情況下，完成元件 (d2)+1 的異常完成信號將變為 0N、(d3) 的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了 DBSELECT (P) 指令的情況下
- (s1) 中指定了不正確的識別編號的情況下
- (s1) 中指定了未打開的識別編號的情況下
- 韌體版本低於“28”的可程式控制器 CPU 中，指定了 (s1) 中韌體版本為“28”及其以後的可程式控制器 CPU 中建構或操作的資料庫的情況下
- (s2) 中指定的表名不存在的情況下
- (s3) 中搜尋的欄位數超過了最大值的情況下
- (s3)+1 ~ (s3)+□ 中指定的欄位名中，設定了超出範圍的欄位名的情況下
- (s4) 中搜尋的記錄數超過了最大值的情況下
- (s4) 中搜尋的對象的欄位容量值超出了範圍的情況下
- 資料庫的 SELECT 處理失敗的情況下
- (d1) 中輸出的記錄的個數超出了 (s4) 中指定的個數的情況下
- (d1) 中輸出的記錄的容量超出了 (s4) 中指定的容量的情況下
- (s4) 中指定的條件之間運算值超出範圍的情況下
- (s4)+37 中指定的條件之間運算值為 1 或 2 時，(s4)+1 中設定了 0 的情況下
- (s4) 中設定 0、(s4)+1 中設定了 0 以外的值的情況下

出錯

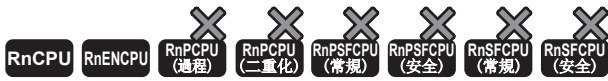
出錯代碼 (SD0)	內容
2820H	(s2)、(s3)、(s4)、(d1)、(d2) 中指定的範圍超出了相應元件/標籤的情況下。

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058 頁 資料庫存取指令中發生的出錯代碼

資料庫記錄刪除

DBDELETE (P)

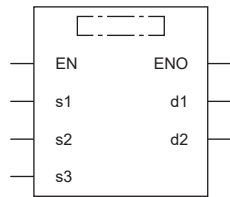


• R00CPU、R01CPU、R02CPU中不能使用。

在指定的識別編號對應的資料庫的指定表中，將符合設定的條件的記錄刪除。

梯形圖	ST
	ENO:=DBDELETE (EN, s1, s2, s3, d1, d2) ; ENO:=DBDELETEP (EN, s1, s2, s3, d1, d2) ;

FBD/LD



■執行條件

指令	執行條件
DBDELETE	
DBDELETEP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	資料庫識別編號	1~4	帶符號BIN16位元	ANY16
(s2)	儲存刪除對象資料庫的表名的起始元件*1	—	Unicode字元串	ANYSTRING_DOUBLE
(s3)	儲存刪除條件(最多2個)的起始元件	—	字元	ANY16*2
(d1)	完成元件(透過指令完成1個掃描ON的起始元件) • (d1)+0: 完成信號 • (d1)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d2)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 表名區別大寫字母與小寫字母進行設定。

*2 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	○	—	
(s3)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	○	—	○	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	

■刪除條件

操作數：(s3)																				
元件	項目	內容	設定範圍	設定方	資料類型															
+0	第1個條件的資料容量	設定第1個刪除條件的資料容量。	1~125	用戶	WORD															
+1	第2個條件的資料容量	設定第2個刪除條件的資料容量。 • 0：無條件 • 0以外：資料容量	0~125	用戶																
+2~+□	刪除條件*1	<p>設定刪除條件。對刪除條件按照刪除數進行設定。 此外，根據判定值的資料類型，(s3)的最終地址將變化。</p> <table border="1"> <tr> <td>(s3)+2</td> <td>(1)</td> </tr> <tr> <td>:</td> <td></td> </tr> <tr> <td>(s3)+33</td> <td>(2)</td> </tr> <tr> <td>(s3)+34</td> <td>(3)</td> </tr> <tr> <td>(s3)+35</td> <td>(4)</td> </tr> <tr> <td>(s3)+36</td> <td></td> </tr> <tr> <td>:</td> <td></td> </tr> <tr> <td>(s3)+□</td> <td></td> </tr> </table> <p>■欄位名(1) 固定為32字元，以Unicode字元串進行設定。不足32字元的情況下將向前填充對齊，以0000H填充為32字元。</p> <p>■條件編號(2) 設定表示刪除條件的編號。 • 1：= (等於判定值) • 2：!= (判定值以外) • 3：< (小於判定值) • 4：> (大於判定值) • 5：=< (小於等於判定值) • 6：>= (大於等於判定值) • 7：is NULL (是NULL(未設定值)) • 8：is not NULL (不是NULL(設定了值))</p> <p>■條件之間運算值(3) 設定與相鄰條件的運算值。 • 0：無設定 • 1：AND • 2：OR</p> <p>刪除條件為1個的情況下，將第1個刪除條件的條件之間運算值設定為0。此時，即使設定第2個刪除條件的內容也將被忽略。 第2個條件的條件之間運算值必須設定為0。</p> <p>■判定值(4) 設定更新條件的判定值。判定值的資料容量取決於各欄位的資料類型的容量。 (☞ 1027頁 DBINSERT(P)) 條件編號為7或8的情況下，也準備相當於資料容量的判定值的區域。</p>	(s3)+2	(1)		:		(s3)+33	(2)	(s3)+34	(3)	(s3)+35	(4)	(s3)+36		:		(s3)+□		—
(s3)+2	(1)																			
:																				
(s3)+33	(2)																			
(s3)+34	(3)																			
(s3)+35	(4)																			
(s3)+36																				
:																				
(s3)+□																				

*1 按更新條件數設定欄位名、條件編號、條件之間運算值、判定值。(☞ 1034頁 DBUPDATE(P))

功能

- 在與(s1)中指定的識別編號對應的資料庫的(s2)中指定的表中，將符合(s3)中設定的條件的記錄刪除。
- 刪除條件透過(s3)指定。刪除條件數最多為2個。
- 其它指令執行中透過中斷程式執行了動作的情況下，將變為無處理。
- 根據DBDELETE(P)指令，將條件符合欄位1=2的表recipeA的記錄刪除時，其情況如下所示。

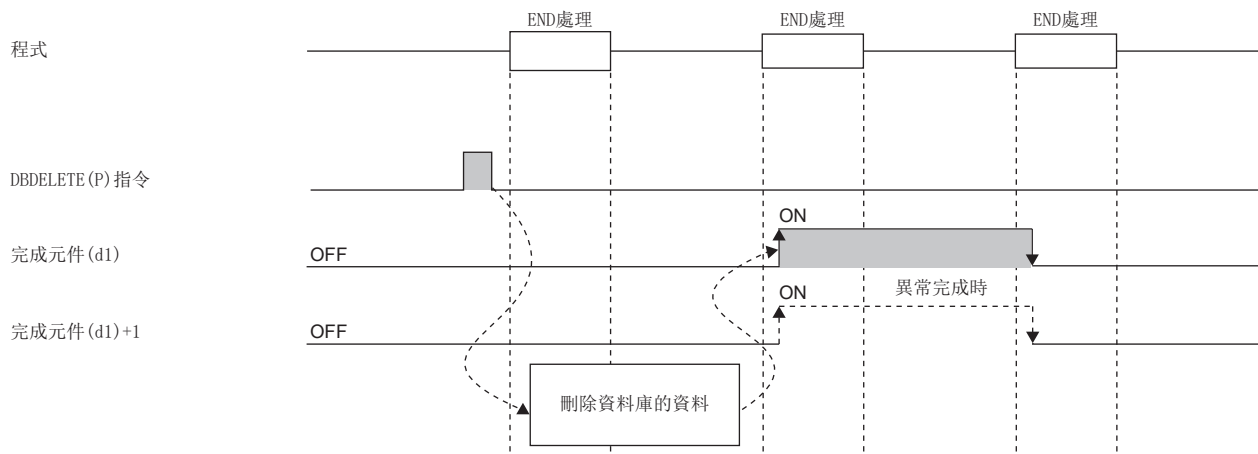
表格recipeA(刪除前)

欄位1 WORD型	欄位2 WSTRING型(最大16字元)	欄位3 INT型
1	ProductA	100
2	ProductB	200
3	ProductC	300

表格recipeA(刪除後)

欄位1 WORD型	欄位2 WSTRING型(最大16字元)	欄位3 INT型
1	ProductA	100
3	ProductC	300

- 正常完成時，完成元件(d1)+0的完成信號將變為ON、(d2)的完成狀態中將儲存0。
- 執行DBDELETE(P)指令時的完成元件的動作如下所示。



- DBDELETE(P)指令執行中，SM753(檔案存取中)將變為ON。^{*2}SM753為ON中，無法執行DBDELETE(P)指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
 - CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498(CPU模組內建資料庫啟動完成標誌)為OFF的期間，DBDELETE(P)指令無法執行。^{*2}執行的情況下將變為無處理。
- ^{*2} 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述情況下，完成元件 (d1)+1 的異常完成信號將變為 0N、(d2) 的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了本指令的情況下
- (s1) 中指定了超出範圍的識別編號的情況下
- (s1) 中指定了未打開的識別編號的情況下
- 韌體版本低於“28”的可程式控制器CPU中，指定了(s1)中韌體版本為“28”及其以後的可程式控制器CPU中建構或操作的資料庫的情況下
- (s2) 中指定的表名不存在的情況下
- (s2) 中指定的表名的字元數超過了32字元的情況下
- (s3) 中對刪除條件的容量設定了超出範圍的值的的情況下
- 資料庫的DELETE處理失敗的情況下
- (s3) 中對設定條件之間運算值設定了超出範圍的值的的情況下
- (s3)+35 中指定的條件之間運算值為1或2時，(s3)+1 設定了0的情況下
- (s3)+2 未設定應設定的欄位名的情況下
- (s3)+2~(s3)+□ 中指定的欄位名中，設定了超出範圍的欄位名的情況下

出錯

出錯代碼 (SD0)	內容
2820H	(s2)、(s3)、(d1) 中指定的範圍超出了相應元件/標籤的情況下。

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058頁 資料庫存取指令中發生的出錯代碼

資料庫事物處理開始

DBTRANS (P)



• R00CPU、R01CPU、R02CPU中不能使用。

對指定識別編號對應的資料庫中的事物處理開始進行聲明。

梯形圖	ST
	ENO:=DBTRANS (EN, s, d1, d2) ; ENO:=DBTRANSP (EN, s, d1, d2) ;

FBD/LD

■執行條件

指令	執行條件
DBTRANS	
DBTRANSP	

設置數據

■內容、範圍、資料類型

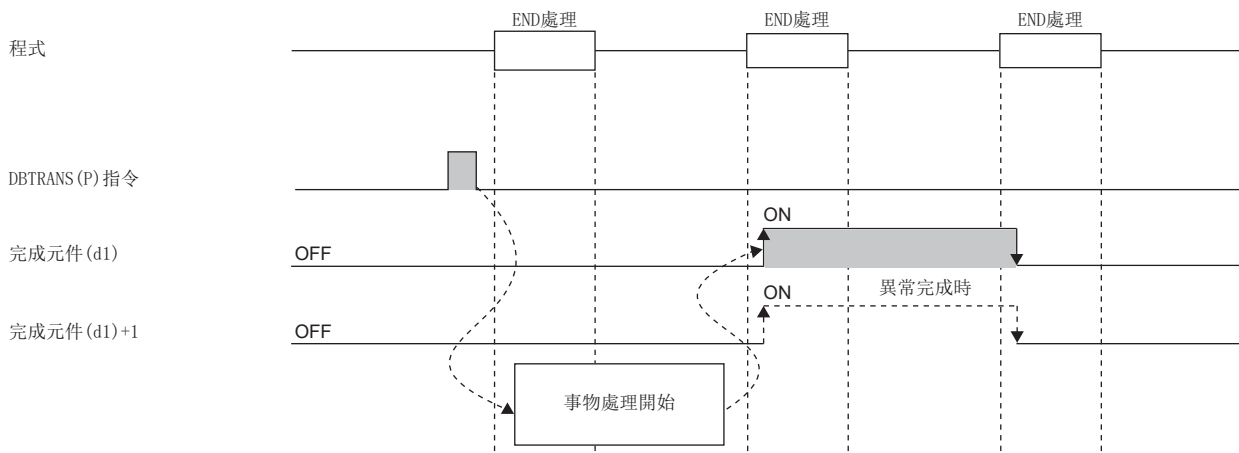
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	資料庫識別編號	1~4	帶符號BIN16位元	ANY16
(d1)	完成元件(透過指令完成1個掃描ON的起始元件) • (d1)+0: 完成信號 • (d1)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d2)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	○	—	○	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	

功能

- 對(s)中指定的識別編號對應的資料庫中的事物處理開始進行聲明。但是，(s)中指定的資料庫以外的資料庫處於打開狀態的情況下，事物處理將無法開始，DBTRANS指令將異常完成。
- 正常完成時，完成元件(d1)的完成信號將變為ON、(d2)的完成狀態中將儲存0。
- 執行DBTRANS(P)指令時的完成元件的動作如下所示。



- 欲透過DBTRANS(P)指令開始了事物處理的情況，需要透過DBCMMIT(P)指令確定事物處理、或透過DBROLBAK(P)指令返回為事物處理開始前的狀態。(執行DBCMMIT(P)指令或DBROLBAK(P)指令之前執行了DBCLOSE(P)指令的情況下，透過執行DBCLOSE(P)指令時的狀態確定事物處理。)
- DBTRANS(P)指令執行中，SM753(檔案存取中)將變為ON。^{*1}SM753為ON中，無法執行DBTRANS(P)指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
- CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498(CPU模組內建資料庫啟動完成標誌)為OFF的期間，DBTRANS(P)指令無法執行。^{*1}執行的情況下將變為無處理。

*1 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述情況下，完成元件(d1)+1的異常完成信號將變為ON、(d2)的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了DBTRANS(P)指令的情況下
- 指定了超出範圍的識別編號的情況下
- 在事物處理已開始的狀態下執行了DBTRANS(P)指令的情況下
- (s)中指定的資料庫以外的資料庫處於打開狀態的情況下
- 指定了(s)中未打開的識別編號的情況下

出錯

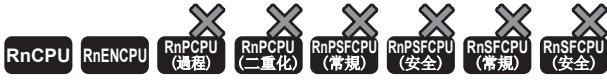
出錯代碼(SD0)	內容
2820H	(d1)中指定的範圍超出了相應元件/標籤的情況下。

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058頁 資料庫存取指令中發生的出錯代碼

資料庫提交

DBCOMMIT (P)

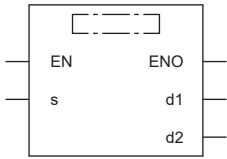


• R00CPU、R01CPU、R02CPU中不能使用。

對指定的識別編號對應的資料庫相關事物處理進行確定。

梯形圖	ST
	<pre>ENO:=DBCOMMIT (EN, s, d1, d2); ENO:=DBCOMMITP (EN, s, d1, d2);</pre>

FBD/LD



■執行條件

指令	執行條件
DBCOMMIT	
DBCOMMITP	

設置數據

■內容、範圍、資料類型

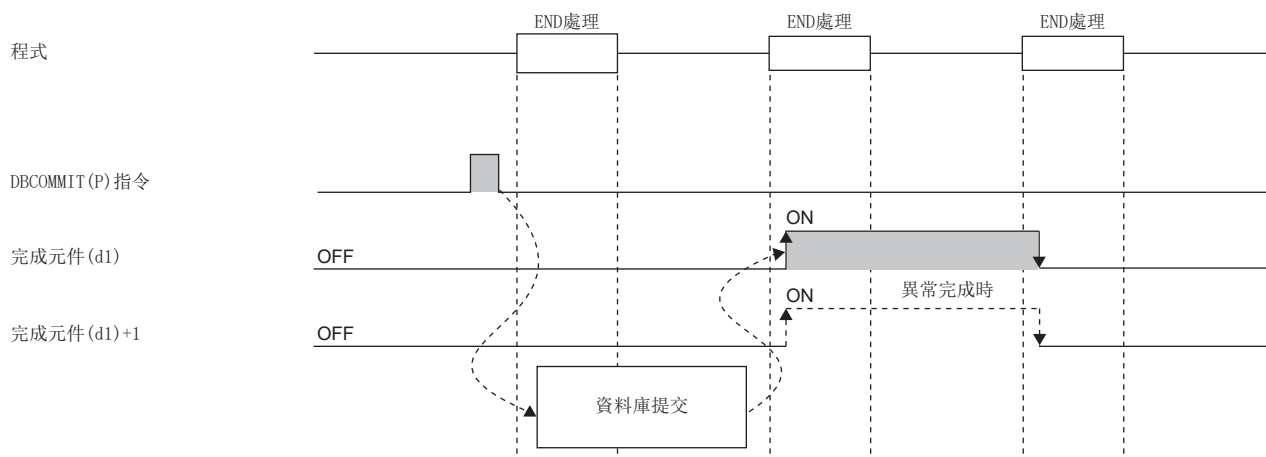
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	資料庫識別編號	1~4	帶符號BIN16位元	ANY16
(d1)	完成元件(透過指令完成1個掃描ON的起始元件) • (d1)+0: 完成信號 • (d1)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d2)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	○	—	○	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	

功能

- 對 (s) 中指定的識別編號對應的資料庫相關的事物處理進行確定。
- 正常完成時，完成元件 (d1) 的完成信號將變為ON、(d2) 的完成狀態中將儲存0。
- 執行DBCMMIT(P) 指令時的完成元件的動作如下所示。



- DBCMMIT(P) 指令執行中，SM753 (檔案存取中) 將變為ON。^{*1}SM753為ON中，無法執行DBCMMIT(P) 指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
- CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498 (CPU模組內建資料庫啟動完成標誌) 為OFF的期間，DBCMMIT(P) 指令無法執行。^{*1}執行的情況下將變為無處理。

^{*1} 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述情況下，完成元件 (d1) +1 的異常完成信號將變為ON、(d2) 的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了DBCMMIT(P) 指令的情況下
- 指定了超出範圍的識別編號的情況下
- 在沒有事物處理的狀態下執行了DBCMMIT(P) 指令的情況下
- 指定了 (s) 中未打開的識別編號的情況下

出錯

出錯代碼 (SD0)	內容
2820H	(d1) 中指定的範圍超出了相應元件/標籤的情況下。 ^{*2}

^{*2} 關於詳細內容，請參閱下述內容。

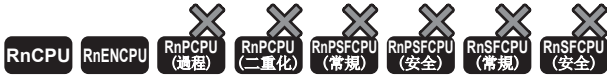
☞ 51頁 執行指令時的元件、標籤的範圍檢查

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058頁 資料庫存取指令中發生的出錯代碼

資料庫回滾

DBROLBAK (P)

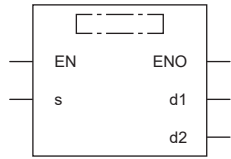


• R00CPU、R01CPU、R02CPU中不能使用。

執行指定的識別編號對應的資料庫的回滾。

梯形圖	ST
	ENO:=DBROLBAK (EN, s, d1, d2) ; ENO:=DBROLBAKP (EN, s, d1, d2) ;

FBD/LD



■執行條件

指令	執行條件
DBROLBAK	
DBROLBAKP	

設置數據

■內容、範圍、資料類型

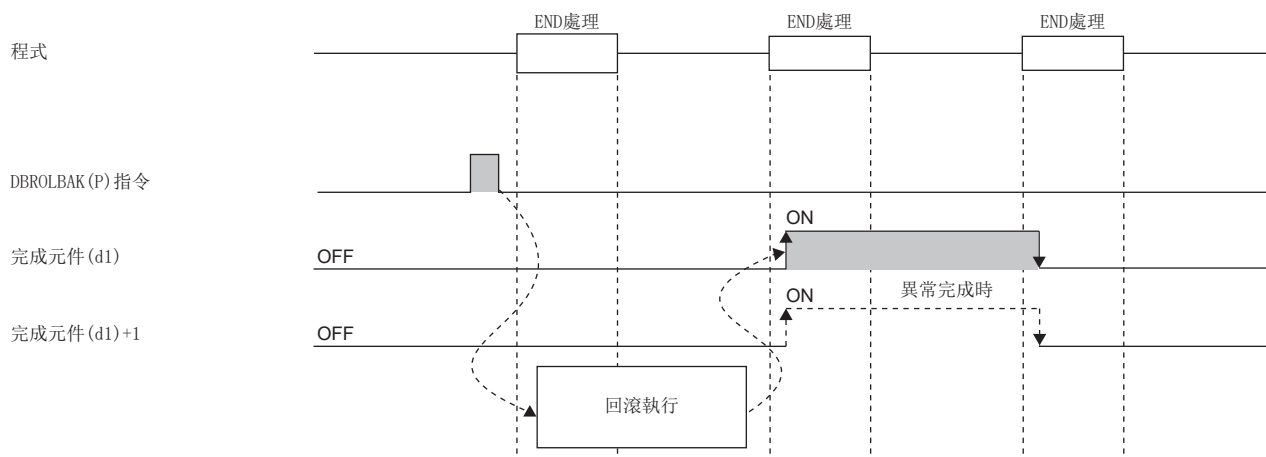
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	資料庫識別編號	1~4	帶符號BIN16位元	ANY16
(d1)	完成元件(透過指令完成1個掃描ON的起始元件) • (d1)+0: 完成信號 • (d1)+1: 異常完成信號	—	位元	ANYBIT_ARRAY (要素數: 2)
(d2)	完成狀態 • 0000: 正常完成 • 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	—	—	○	—	—	—	○	—	—	—	—	—
(d1)	○	—	○	—	—	—	○	—	—	—	—	—
(d2)	—	—	○	—	—	—	○	—	—	—	—	—

功能

- 執行(s)中指定的識別編號對應的資料庫中的回滾。
- 正常完成時，完成元件(d1)的完成信號將變為ON、(d2)的完成狀態中將儲存0。
- 執行DBROLBAK(P)指令時的完成元件的動作如下所示。



- DBROLBAK(P)指令執行中，SM753(檔案存取中)變為ON。^{*1}SM753為ON中，無法執行DBROLBAK(P)指令。執行的情況下將變為無處理。但是，資料庫存取指令執行中的情況下將異常完成。
- CPU模組的模組參數中，將“內建資料庫存取使用有無”設定為“使用”的情況下，SM1498(CPU模組內建資料庫啟動完成標誌)為OFF的期間，DBROLBAK(P)指令無法執行。^{*1}執行的情況下將變為無處理。

^{*1} 關於SM753、SM1498對應的韌體，請參閱特殊繼電器一覽。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

注意事項

下述情況下，完成元件(d1)+1的異常完成信號將變為ON、(d2)的完成狀態中將儲存出錯代碼。

- 資料庫存取指令執行中執行了DBROLBAK(P)指令的情況下
- 設定了超出範圍的識別編號的情況下
- 在沒有事物處理的狀態下執行了DBROLBAK(P)指令的情況下
- 指定了(s)中未打開的識別編號的情況下

出錯

出錯代碼(SD0)	內容
2820H	(d1)中指定的範圍超出了相應元件/標籤的情況下。

關於操作數的完成狀態中儲存的出錯代碼，請參閱下述內容。

☞ 1058頁 資料庫存取指令中發生的出錯代碼

資料庫存取指令中發生的出錯代碼

資料庫存取指令的完成狀態中儲存的出錯代碼的一覽如下所示。

出錯代碼	出錯內容	處理方法
0103H	CPU內部異常	進行電源OFF→ON或重設、確認能否再次執行。
0104H	執行中有資料庫存取指令。	確認資料庫存取指令未完成期間是否沒執行了其它的資料庫存取指令。
0105H	<ul style="list-style-type: none">指定的路徑不是資料庫資料夾。資料庫損壞。	確認路徑中顯示的資料夾中是否存在資料庫檔案，並修改成正確的路徑。
0106H	<ul style="list-style-type: none">無法打開資料庫內部的檔案。指定了不同的韌體版本中建構的資料庫。	以DBIMPORT (P) 重新建構資料庫。
0107H	指定了資料庫存取指令中不能使用的驅動器編號。	確認驅動器編號是否為2，如果為2以外則進行修改。
0108H	資料庫刪除失敗。	在其他功能中確認沒有存取至資料庫，重新以DBIMPORT (P) 指令再次建構資料庫。
07D1H	指定的表不存在於資料庫。	確認資料庫中是否有指定的表。
0940H	表名中包含有不能使用的字元。	表名從數字以外的字元開始，不能包含無法使用的字元。
0941H	表名重複。	因為資料庫不區分表名的字母大小寫，因此在Unicode文字檔案的各表名的設定中在不區分字母的大小寫下確認是否沒指定相同的字元。
0942H	指定的表不存在於資料庫。	確認資料庫中是否有指定的表。
0945H	指定的資料庫透過其他功能為使用中的狀態。	其他的功能中完成資料庫的操作後，重新執行指令。
0946H	指定的表透過其他功能為使用中的狀態。	完成其他的功能中的表的操作後，重新執行指令。
094AH	未指定資料庫名。	指定正確的資料庫名。
094BH	指定的資料夾中未儲存資料庫。	確認資料夾名是否正確，資料夾內是否有資料庫關聯檔案。
094CH	欄位名的字元串中包含有不能使用的字元。	欄位名中應避免包含不能使用的字元。
094DH	欄位名指定重複。	在Unicode文字檔案及指令的引數中確認有無重複的欄位名，如果有則修改成不重複。
094EH	指定了不存在的欄位。	確認表中是否存在指定的欄位。
0950H	索引中包含有不能使用的字元。	避免索引中包含不能使用的字元。
0951H	對主鍵或附帶有NOT NULL限制的欄位試圖設定空的資料(NULL值)。	確認表結構，確認欄位是否允許NULL，如果未允許則確認是否有設定值。
0952H	試圖在帶主鍵限制的欄位中添加同一值。	在有主鍵限制的欄位中，確認試圖添加的記錄的設定值是否與已登錄的記錄不相同。
0953H	設定索引的字段中重複設定了重複資料的情況下，索引用的資料將變大，無法確保CPU模組的記憶體。	變更發生了出錯的表定義，重新建構。
0954H	不能將設定值自動轉換為適合資料類型的形式。	確認設定值的格式是否符合規格。
0959H	導入時，資料類型與表中定義的類型不相同。	確認試圖以Unicode文字導入的設定值是否為適合欄位的資料類型的值。
095CH	進行字元串大小比較等，比較的值的資料類型有錯誤。	修改比較的值。
095DH	由於資料庫存取指令的引數中指定的設定值的資料類型轉換失敗，因此欄位中無法儲存設定值。	確認設定值是否符合資料類型的規格。
095EH	試圖在一個欄位中分配重複的索引。	確認設定了主鍵、外部鍵的欄位中是否添加分配了索引、一個欄位是否沒重複分配了索引。
095FH	索引名重複。	確認各索引的名稱是否不重複。
096CH	作為鍵限制指定了外部鍵的欄位中添加的設定值未包含在參照目標的設定值中。	確認參照目標的設定值。
096EH	其它表因為正在參照，因此無法刪除記錄。	確認要刪除的記錄的設定值中是否沒包含外部鍵參照的表的設定值。
0970H	雖然試圖添加・刪除記錄，但用於參照記錄中包含的設定值的表不存在。	確認有無參照目標的表。
0971H	試圖添加參照目標的表中未登錄的設定值。	確認參照目標的表的設定值。
0972H	透過外部鍵參照的欄位的資料類型不一致。	確認兩者中欄位的資料類型一致。
0974H	透過外部鍵參照的表不存在。	指定正確的表名。
0976H	透過外部鍵參照的表的欄位不存在。	指定正確的欄位名。
0984H	事物處理已在執行。	確認是否沒重複執行了事物處理。

出錯代碼	出錯內容	處理方法
09C2H	在使用的CPU模組中，對非對應的資料庫進行操作。	<ul style="list-style-type: none"> ■韌體版本低於“28”的可程式控制器CPU的情況下 在韌體版本為“28”及其以後的可程式控制器CPU中執行DBEXPORT (P) 指令後，使用獲取的Unicode文字檔案，並在韌體版本低於“28”的可程式控制器CPU中執行DBIMPORT (P) 指令。 ■韌體版本為“28”及其以後的可程式控制器CPU的情況下 在韌體版本低於“28”的可程式控制器CPU中執行DBEXPORT (P) 指令後，使用獲取的Unicode文字檔案，並在韌體版本為“28”及其以後的可程式控制器CPU中執行DBIMPORT (P) 指令。
09C3H	韌體版本低於“28”的可程式控制器CPU的DBIMPORT (P) 指令中建立的資料庫在韌體版本為“28”及其以後的可程式控制器CPU中操作後，在韌體版本低於“28”的可程式控制器CPU中操作。	<ul style="list-style-type: none"> • 變為出錯的資料庫在韌體版本為“28”及其以後的可程式控制器CPU中操作。 • 希望在韌體版本低於“28”的可程式控制器CPU中操作的情況下，在韌體版本為“28”及其以後的可程式控制器CPU中執行DBEXPORT (P) 指令，將輸出的Unicode文字檔案在韌體版本低於“28”的可程式控制器CPU中執行DBIMPORT (P) 指令後操作。
09D7H	未指定BOOL型的資料。	設定BOOL型的資料。
09F9H	添加記錄時，指定了不存在的欄位。	確認表配置。
0A0AH	指定了禁止的欄位名。	指定其它欄位名。
0A16H	試圖在設定了主鍵的欄位中設定了空值狀態下添加記錄。	確認Unicode文字中設定了主鍵的欄位的設定值中是否沒設定了空值。
0A17H	試圖在設定了主鍵的欄位中包含有相同值的狀態下添加記錄。	確認Unicode文字中設定了主鍵的欄位的設定值中是否沒設定了相同的值。
0A2BH	試圖設定多個主鍵。	確認Unicode文字的鍵設定行中，是否沒設定了多個主鍵。
0A2FH	參照目標的表中不存在主鍵。	確認試圖以Unicode文字導入的設定值是否為適合欄位的資料類型的值。
0A30H	設定了外部鍵的欄位的設定值未包含在設定了主鍵的欄位的設定值中。	雖然是作為外部鍵而試圖添加的欄位的設定值，但應確認作為外部鍵添加的欄位的設定值是否包含在設定了主鍵的欄位的設定值中。
0A31H	試圖更改或删除正在由外部鍵參照的欄位的設定值。	確認各表的參照關係。
0A50H	試圖在同一表中分配重複的索引名。	確認Unicode文字的內容，避免索引名重複。
0A5DH	合併的欄位的資料類型不相同。	確認合併的欄位的資料類型是否一致。
0A5FH	同一表中指定了重複的欄位名。	確認同一表中欄位名是否重複，如果重複則修改欄位名。
0AAAH	無符號整數資料類型的欄位中指定了負值。	確認欄位的資料類型是否不為不能受理負值的類型。
0AB1H	試圖在整數型的資料類型中指定字元串。	確認欄位的資料類型為整數型的情況下，是否沒指定了字元串。
0AB9H	試圖對未設定設定值的欄位分配索引。	確認指定索引的欄位的記錄中設定值是否已全部設定。
0AECH	表打開失敗。	進行電源OFF→ON或重設，確認能否打開。
1000H	導入時指定的Unicode文字檔案不存在。	確認資料夾路徑中所示的檔案是否為Unicode文字檔案，如果錯誤則修改為正確的資料夾路徑。
1002H	沒有可分配的識別編號。	因為可打開的資料庫最多為4個，因此將打開的資料庫個數限制調整為4個。
1003H	指定了未打開的識別編號。	透過DBOPEN (P) 指令預先打開資料庫、獲取識別編號。
1004H	指令中指定的識別編號超出了範圍。	指定透過DBOPEN (P) 指令獲取的識別編號。
1007H	未指定指令中指定的欄位數。	將引數的欄位數設定為1及其以上。
1008H	<ul style="list-style-type: none"> • 指令中指定的欄位數超出了範圍。 • 指令中指定的欄位名為超出了範圍的欄位名。 	<ul style="list-style-type: none"> • 將引數的欄位數設定為128及其以內。^{*1} • 將引數的欄位名設定在可以指定的範圍內(第128號及其以內)的欄位名。
1009H	未指定指令中指定的記錄數。	將引數的記錄數設定為1及其以上。
100AH	指令中指定的記錄數超出了範圍。	將引數的記錄數設定在範圍內。
100CH	Unicode文字的表及設定值的分割方法不符合格式。	確認表的分割符及設定值的分割方法是否符合格式。
100DH	資料庫發生了異常。	進行電源OFF→ON或重設，確認是否再次發生相同現象。
100EH	指令中指定的條件編號的設定超出了範圍。	確認條件編號的範圍是否正確。
100FH	指令中指定2個條件的情況下，條件之間運算值為0。	確認引數的條件之間運算值是否為1或2。
1010H	超過了可事物處理的資料庫數。	確認是否沒正在對2個及其以上的資料庫執行事物處理，或沒在已開始事物處理的狀態下執行了事物處理。
1011H	在未執行事物處理開始的狀況下，執行了提交或回滾。	確認是否在無事物處理的狀態下執行了提交、回滾。
1012H	添加、更新中使用的每個記錄的容量值不正確。	確認各欄位的資料容量與要添加、更新的資料的容量是否匹配。
1013H	設定了不支援的資料類型。	確認Unicode文字的資料類型。
1014H	<ul style="list-style-type: none"> • 在事物處理執行過程中打開了其它資料庫。 • 事物處理中，執行了DBEXPORT (P) 指令的情況下。 	<ul style="list-style-type: none"> • 事物處理執行過程中應避免打開新的資料庫。 • 事物處理結束後，執行DBEXPORT (P) 指令。
1015H	對多個已打開的資料庫執行了事物處理。	關閉事物處理對象外的資料庫。
101BH	各指令的引數中指定的條件中，未設定第1個條件。	確認引數的第1個條件的欄位容量是否為0以外的值，欄位名是否不為空白。

出錯代碼	出錯內容	處理方法
101CH	在多個條件設定中，未設定欄位的資料容量。	確認條件中指定的欄位的資料容量是否不為0。
101DH	在多個條件設定中，未設定欄位名。	確認條件中指定的欄位名是否不為空白。
101EH	條件之間運算符的值超出了範圍。	確認條件之間運算符的值是否正確。
101FH	第2個條件設定中的條件之間運算符的值超出了範圍。	確認第2個條件的“條件之間運算符”中是否沒設定了0以外的值。
1020H	設定值超出了資料類型中規定的範圍。	確認設定值是否在允許範圍內。
1021H	試圖打開已被打開的資料庫。	確認DBOPEN(P)指令的(s)中是否沒重複指定了對應的路徑。
1022H	在存在有已打開的資料庫的狀態下，執行了DBIMPORT(P)指令。	關閉已打開的資料庫之後，再執行DBIMPORT(P)指令。
1023H	指定了超出範圍的表編號。	確認表編號是否在允許範圍內。
102EH	條件中指定的欄位容量超出了範圍。	確認條件的欄位容量是否在允許範圍內。
1030H	輸出容量超過了設定值。	調整輸出容量。
1031H	輸出個數超過了設定值。	調整輸出個數。
1032H	在DBOPEN(P)指令中資料庫為打開中或是資料庫存取指令執行中時，由於SD記憶卡被取出，而無法執行資料庫存取指令或從外部設備的CPU模組內建資料庫存取功能。	進行CPU模組的電源OFF→ON或重設。
1040H	超過添加的記錄的合計容量。	調整添加的記錄的合計容量。
2000H	Unicode文字檔案的格式不正確。	<ul style="list-style-type: none"> • 確認Unicode文字檔案的格式是否正確，如果錯誤則進行修改。 • 確認註釋行的字元數是否在範圍內。
2001H	Unicode文字檔案的鍵設定不正確。	確認在Unicode文字檔案的鍵限制中，是否沒設定了超出範圍的值。
2002H	資料庫名、表或欄位名的字元數超出了範圍。	<ul style="list-style-type: none"> • 確認Unicode文字檔案的資料庫名、表名、欄位名是否沒超過了32字元。 • 確認是否沒在1行中使用了128個及其以上的標籤。
2003H	表數超出了範圍。	確認Unicode文字檔案的表數是否沒超過了32個。 ^{*1}
2004H	在Unicode文字檔案的記錄中，設定值的格式不正確。	確認Unicode文字檔案的設定值的格式是否正確。
2005H	超過了可導入的最大記錄數。	確認文字檔案內的記錄數是否沒超過了最大個數(100000個)。
2006H	資料庫名中包含有不能使用的字元。	確認文字檔案的資料庫名中是否沒包含有不能使用的字元。
2007H	索引行的格式有錯誤。	確認索引行的格式是否正確。
2009H	Unicode文字檔案的讀取失敗。	<ul style="list-style-type: none"> • 確認SD記憶卡的狀態。 • 確認Unicode文字檔案的內容。
200AH	Unicode文字檔案的寫入失敗。	確認SD記憶卡的狀態。
200BH	Unicode文字檔案內的欄位數超過了最大數。	確認Unicode文字檔案內的欄位數是否沒超過了128個。 ^{*1}
200CH	Unicode文字檔案中，欄位行中設定的列數與其它行中設定的列數不一致。	確認在Unicode文字檔案中，欄位行中設定的列數(欄位數)與其它行的列數是否匹配。
200DH	Unicode文字檔案內的索引數超出了範圍。	確認文字檔案內的索引數是否沒超過了表的欄位數。
200EH	Unicode文字檔案內的主鍵、外部鍵的鍵名或索引名的字元數超出了範圍。	確認鍵名、索引名是否設定為16字元及其以內。
200FH	因為SD記憶卡處於寫保護狀態，因此無法執行資料庫存取指令。或是從外部設備的CPU模組內建資料庫存取功能的啟動失敗。	解除SD記憶卡的寫保護。
2010H	由於SD記憶卡中沒有空的容量(至少10M字節)，因此無法執行資料庫存取指令或從外部設備的CPU模組內建資料庫存取功能。	確保SD記憶卡的空餘容量。
2011H	資料庫的路徑的字元數超過了128字元。	將路徑的字元數設定為128字元及其以內。
2020H	CPUDB資料夾內的配置不正確。	<ul style="list-style-type: none"> • 將CPUDB資料夾的資料庫減少為不超過32個。 • 將CPUDB資料夾內的非資料庫的資料夾刪除。 • 參閱異常資料庫確認檔案(2:\CPUDB\ErrorDB.txt)，刪除出錯原因的資料夾。
2021H	資料庫損壞。	<ul style="list-style-type: none"> • 重新建構資料庫。 • 參閱異常資料庫確認檔案(2:\CPUDB\ErrorDB.txt)，刪除出錯原因的資料夾。
2023H	資料庫啟動失敗。	<ul style="list-style-type: none"> • 進行CPU模組的電源OFF→ON或重設。 • 確認資料記憶體的容量，建立至少1K字節的空白。
2024H	資料庫停止失敗。或是拆卸SD記憶卡，資料庫停止。	<ul style="list-style-type: none"> • 進行CPU模組的電源OFF→ON或重設。 • 重新將SD記憶卡插入後，進行CPU模組的電源OFF→ON或重設。

出錯代碼	出錯內容	處理方法
2025H	資料庫無法識別。或是修復失敗。	<ul style="list-style-type: none"> • 確認SD記憶卡的狀態。 • 存在異常資料庫確認檔案 (2:\CPUDB\ErrorDB.txt)，異常資料庫確認檔案中記載的資料庫為CPUDB資料夾及其以下的情況下，在各資料夾刪除資料庫後重新建立資料夾。 • 進行了CPU模組的電源OFF→ON或重設後，重新執行仍發生同樣狀況的情況下，或存在異常資料庫確認檔案 (2:\CPUDB\ErrorDB.txt)，異常資料庫確認檔案中記載的資料庫為CPUDB資料夾以外的情況下，刪除SD記憶卡內的\$MELPRJ\$\\$DBASYS\$資料夾中儲存的dbmaintainpath.txt檔案，在各資料夾中刪除所有的資料庫資料夾(除了CPUDB資料夾內)，以DBIMPORT (P)指令重新建立資料庫。
上述以外	<ul style="list-style-type: none"> • Unicode文字檔案的字元代碼有錯誤。 • 存取對象資料庫處於不正確狀態。 	<ul style="list-style-type: none"> • 確認Unicode文字檔案的字元代碼。 • 在再次執行電源OFF→ON或重設時仍發生相同現象的情況下，將存取對象資料庫資料夾刪除後，透過DBIMPORT (P)指令重新建立資料庫。

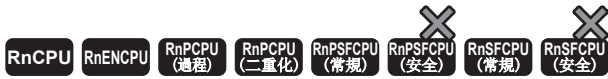
*1 韌體版本低於“28”的可程式控制器CPU的情況下，應設為不超過16。

19 時鐘

19.1 時鐘用指令

時鐘資料的讀取

DATERD (P)



從CPU模組的時鐘單元中讀取“年、月、日、時、分、秒、星期”。

梯形圖	ST
	ENO:=DATERD (EN, d) ; ENO:=DATERDP (EN, d) ;

FBD/LD

■執行條件

指令	執行條件
DATERD	
DATERDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存讀取的時鐘資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 7)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 從CPU模組的時鐘單元中讀取“年、月、日、時、分、秒、星期”，以BIN值儲存到(d)中指定的元件及其以後。

(資料)	(d)	(d)+1	(d)+2	(d)+3	(d)+4	(d)+5	(d)+6
(內容)	年(公曆)	月	日	時(24小時制)	分	秒	星期
(設定範圍)	1980~2079	1~12	1~31	0~23	0~59	0~59	0~6

- (d)的“年”以公曆4位儲存。
- 對於(d)+6的“星期”，將“日~六”以“0~6”儲存。

星期	日	一	二	三	四	五	六
儲存資料	0	1	2	3	4	5	6

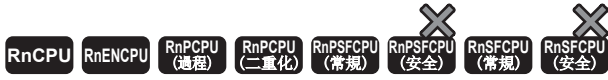
- 閏年將被自動補償。

出錯

沒有運算出錯。

時鐘資料的寫入

DATEWR (P)



將指定的元件及其以後儲存的時鐘資料寫入到CPU模組的時鐘單元中。

梯形圖	ST
	ENO:=DATEWR (EN, s) ; ENO:=DATEWRP (EN, s) ;

FBD/LD

■執行條件

指令	執行條件
DATEWR	
DATEWRP	

■設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存了寫入時鐘單元的時鐘資料的起始元件	功能參照	帶符號BIN16位元	ANY16_ARRAY (要素數: 7)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將(s)中指定的元件及其以後儲存的時鐘資料寫入到CPU模組的時鐘單元中。

(資料)	(d)	(d)+1	(d)+2	(d)+3	(d)+4	(d)+5	(d)+6
(內容)	年(公曆)	月	日	時(24小時制)	分	秒	星期
(設定範圍)	1980~2079	1~12	1~31	0~23	0~59	0~59	0~6

- 各項目的設定以BIN值進行設定。
- 對於(s)的“年”以公曆4位在1980~2079的範圍內設定。
- 對於(s)+1的“月”在1~12(1月~12月)的範圍內設定。
- 對於(s)+2的“日”在1~31(1日~31日)的範圍內設定。
- 對於(s)+3的“時”在0~23(0時~23時)的範圍內設定。(以24小時制進行設定。)
- 對於(s)+4的“分”在0~59(0分~59分)的範圍內設定。
- 對於(s)+5的“秒”在0~59(0秒~59秒)的範圍內設定。
- 對於(s)+6的“星期”將“日~六”以“0~6”進行設定。

星期	日	一	二	三	四	五	六
儲存資料	0	1	2	3	4	5	6

出錯

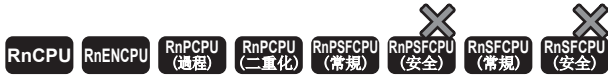
出錯代碼(SD0)	內容
3405H	(s)~(s)+6的各項目中設定了超出設定範圍的資料時。
3425H	將(s)~(s)+6設定為夏季時間開始時間前1小時以內(低於1小時)的情況。

要點

更改了時鐘資料時，事件履歷中將儲存“時鐘設定”(事件代碼：24000)。因此，執行了本指令的情況下，事件履歷中將儲存“時鐘設定”。

時鐘資料的加法運算

DATE+ (P)



對時間資料進行加法運算。

梯形圖	ST*1
	ENO:=DATEPLUS (EN, s1, s2, d) ; ENO:=DATEPLUSP (EN, s1, s2, d) ;

FBD/LD



(□中放入DATEPLUS、DATEPLUSP。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DATE+	
DATE+P	

設置數據

■內容、範圍、資料類型

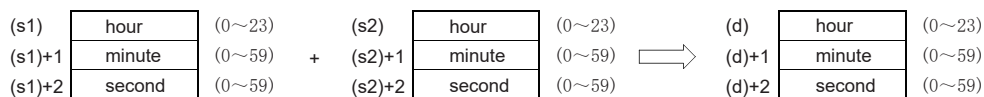
操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	儲存加上時間資料的起始元件	功能參照	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
(s2)	儲存加法運算時間(時刻)資料的起始元件	功能參照	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
(d)	儲存加法運算結果時刻(時間)資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—	—

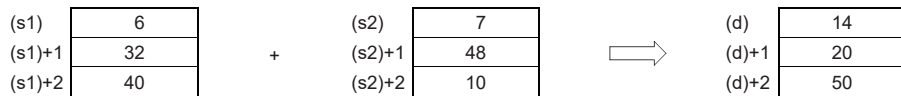
功能

- 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，將加法運算結果儲存到 (d) 中指定的元件編號及其以後。

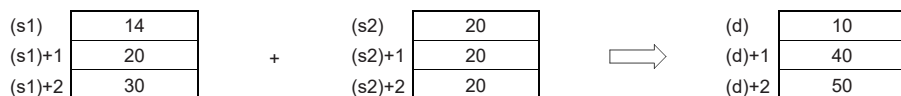


例

將6時32分40秒與7時48分10秒進行了加法運算的情況下



- 運算結果的時間超過了24時的情況下，減去24小時後的值將成為運算結果。例如，將14時20分30秒與20時20分20秒進行了加法運算的情況下，其結果不是34時40分50秒，而是10時40分50秒。

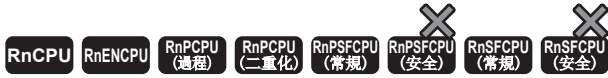


出錯

出錯代碼 (SD0)	內容
3405H	(s1)、(s2)的資料超出範圍時。

時鐘資料的減法運算

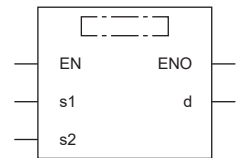
DATE-(P)



對時間資料進行減法運算。

梯形圖	ST*1
	ENO:=DATEMINUS (EN, s1, s2, d) ; ENO:=DATEMINUSP (EN, s1, s2, d) ;

FBD/LD



(□中放入DATEMINUS、DATEMINUSP。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
DATE-	
DATE-P	

設置數據

■內容、範圍、資料類型

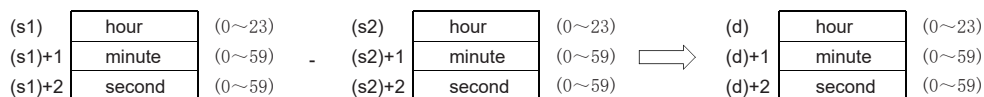
操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存被減法運算時間資料的起始元件	功能參照	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
(s2)	儲存減法運算時間(時刻)資料的起始元件	功能參照	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
(d)	儲存減法運算結果時刻(時間)資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—	—

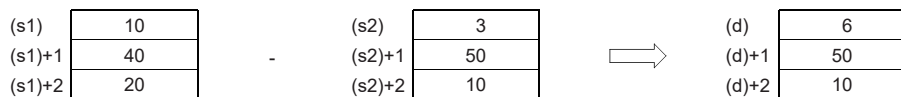
功能

- (s1) 中指定的時間資料減去 (s2) 中指定的時間資料，將減法運算結果儲存到 (d) 中指定的元件編號及其以後。

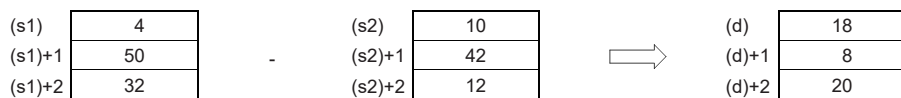


例

將10時40分20秒中與3時50分10秒進行減法運算的情況下



- 運算結果時間為負數的情況下，將該資料進行了+24的值將成為運算結果。例如，將4時50分32秒與10時42分12秒進行減法運算的情況下，其結果不是-6時8分20秒，而是18時8分20秒。

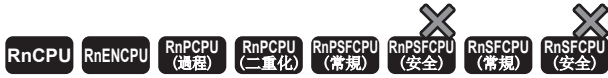


出錯

出錯代碼 (SD0)	內容
3405H	(s1)、(s2)的資料超出範圍時。

時間資料的轉換(時分秒→秒)

TIME2SEC(P)



將時間資料(時、分、秒)換算為秒資料。

梯形圖	ST
	ENO:=TIME2SEC(EN, s, d); ENO:=TIME2SECP(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
TIME2SEC	
TIME2SECP	

■設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存轉換前時鐘資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
(d)	儲存轉換後時鐘資料的起始元件	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

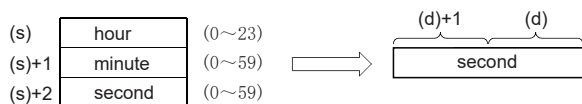
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	○*1	○	○	○	○	○	○	○	—	—	—	—	—

*1 不能使用FX、FY。

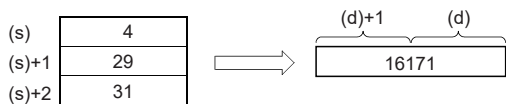
功能

- 將 (s) 中指定的元件編號及其以後儲存的時間資料換算為秒，並將換算結果儲存到 (d) 中指定的元件中。



例

(s) 中指定了4小時29分31秒的情況下

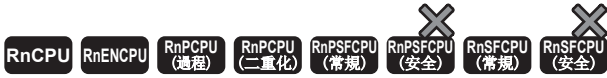


出錯

出錯代碼 (SD0)	內容
3405H	(s) 的資料超出範圍時。

時間資料的轉換(秒→時分秒)

SEC2TIME (P)



將秒資料轉換為時間資料(時、分、秒)。

梯形圖	ST
	ENO:=SEC2TIME (EN, s, d); ENO:=SEC2TIMEP (EN, s, d);

FBD/LD

■執行條件

指令	執行條件
SEC2TIME	
SEC2TIMEP	

■設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存轉換前時鐘資料的起始元件	0~86399	帶符號BIN32位元	ANY32
(d)	儲存轉換後時鐘資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

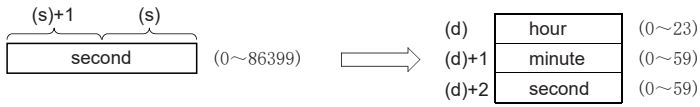
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○*1	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

*1 不能使用FX、FY。

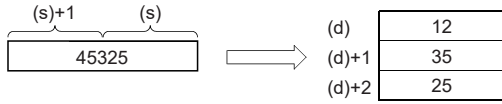
功能

- 將 (s) 中指定的元件編號以後儲存的秒資料換算為時、分、秒，將換算結果儲存到 (d) 中指定的元件及其以後。



例

(s) 中指定了 45325 秒的情況下

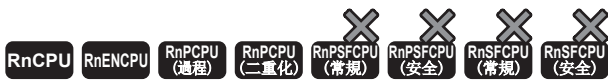


出錯

出錯代碼 (SD0)	內容
3405H	(s) 的資料超出範圍時。

日期時間資料的轉換(日期時間→秒)

DATE2SEC(P) (_U)



- R00CPU、R01CPU、R02CPU可以在韌體版本為“06”及其以後的版本使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。
- RnPCPU可以在韌體版本為“21”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

將日期時間資料換算為秒。

梯形圖	ST
	ENO:=DATE2SEC(EN, s, d); ENO:=DATE2SECP(EN, s, d); ENO:=DATE2SEC_U(EN, s, d); ENO:=DATE2SECP_U(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
DATE2SEC DATE2SEC_U	
DATE2SECP DATE2SECP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存更改前的日期時間資料的起始元件	參閱詳細功能。	帶符號BIN16位元	ANY16_ARRAY (要素數: 6)
(d)	DATE2SEC(P)	0~2145916799	帶符號BIN32位元	ANY32
	DATE2SEC(P)_U	0~3155759999	無符號BIN32位元	ANY32_U
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

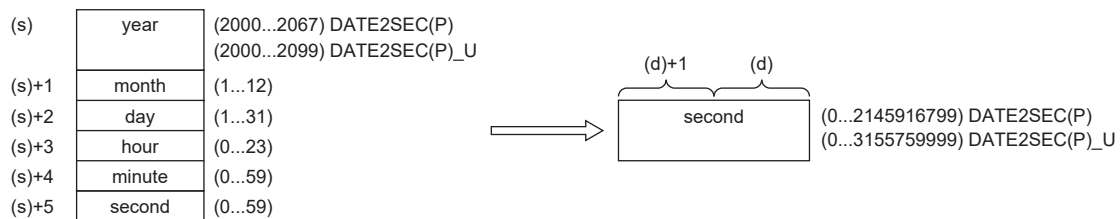
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	○*1	○	○	○	○	○	○	○	—	—	—	—	—

*1 不能使用FX、FY。

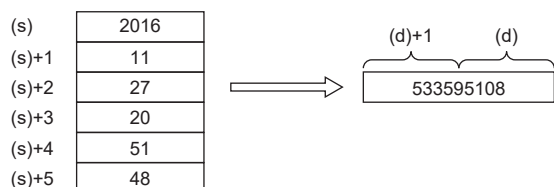
功能

- 將 (s) 中指定的元件編號及其以後儲存的日期時間資料換算為秒，並將換算結果儲存到 (d) 中指定的元件中。相當於起點 (0秒) 為2000年1月1日0時0分0秒。



- 年的資料範圍在DATE2SEC(P)中為2000~2067，DATE2SEC(P)_U時為2000~2099。

例如，指定了2016年11月27日20時51分48秒的情況下，將變為如下所示。



- 本指令的動作不影響時鐘資料的更改(包含時區、夏季時間的時間補償)。例如，夏季時間期間中的換算結果和夏季時間期間以外的換算結果為相同。

出錯

出錯代碼 (SD0)	內容
3405H	(s)的資料超出範圍時。
	指定了實際不存在的日期時間時。(例如：2016年2月30日)

日期時間資料的轉換(秒→日期時間)

SEC2DATE (P) (_U)



- R00CPU、R01CPU、R02CPU可以在韌體版本為“06”及其以後的版本使用。工程工具可以在版本為“1.047Z”及其以後的版本使用。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“31”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。
- RnPCPU可以在韌體版本為“21”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

將秒資料轉換為日期時間。

梯形圖	ST
	ENO:=SEC2DATE (EN, s, d) ; ENO:=SEC2DATEP (EN, s, d) ; ENO:=SEC2DATE_U (EN, s, d) ; ENO:=SEC2DATEP_U (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
SEC2DATE SEC2DATE_U	
SEC2DATEP SEC2DATEP_U	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	SEC2DATE (P) SEC2DATE (P)_U	0~2145916799 0~3155759999	帶符號BIN32位元 無符號BIN32位元	ANY32 ANY32_U
(d)	儲存更改後的日期時間資料的起始元件	參閱詳細功能。	帶符號BIN16位元	ANY16_ARRAY (要素數: 7)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

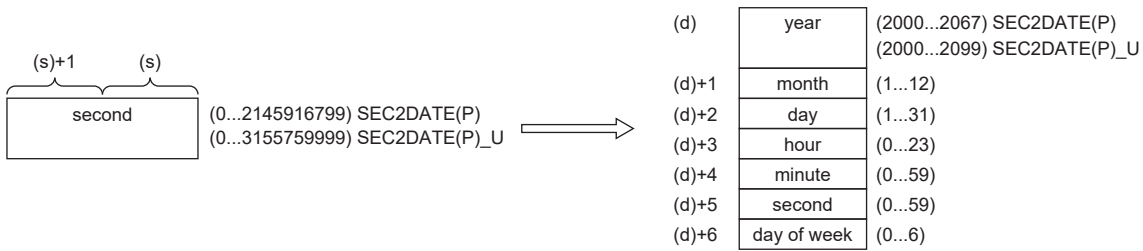
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○*1	○	○	○	○	○	○	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—	—

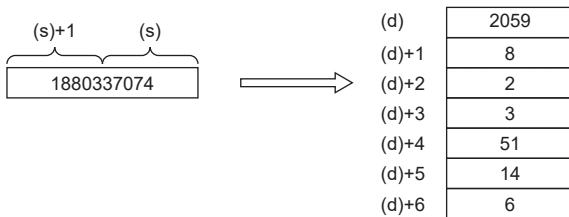
*1 不能使用FX、FY。

功能

- 將 (s) 中指定的元件編號及其以後儲存的秒資料換算為日期時間，並將換算結果儲存到 (d) 中指定的元件及其以後。相當於起點 (0秒) 為2000年1月1日0時0分0秒。



例如，指定了1880337074秒的情況下，將變為如下所示。



- (d)+6的星期，將“日~六”以“0~6”儲存。

星期	日	一	二	三	四	五	六
儲存資料	0	1	2	3	4	5	6

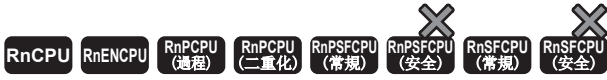
- 本指令的動作不影響時鐘資料的更改 (包含時區、夏季時間的時間補償)。例如，夏季時間期間中的換算結果和夏季時間期間以外的換算結果為相同。

出錯

出錯代碼 (SD0)	內容
3405H	(s) 的資料超出範圍時。

日期比較

LDDT□、ANDDT□、ORDT□



進行指定日期資料的比較，或將當前日期與日期資料進行比較。

梯形圖	ST*1
	ENO:=LDDT_□(EN, s1, s2, s3); ENO:=ANDDT_□(EN, s1, s2, s3); ENO:=ORDT_□(EN, s1, s2, s3); (□中放入EQ、NE、GT、LE、LT、GE。)*2
(□中輸入DT=、DT<>、DT>、DT<=、DT<、DT>=。)	

FBD/LD
(□中放入LDDT_、ANDDT_、ORDT_與EQ、NE、GT、LE、LT、GE的組合。)*2

*1 在版本為“1.035M”及其以後的工程工具中對應。
 *2 EQ為=、NE為<>、GT為>、LE為<=、LT為<、GE為>=。

■執行條件

指令	執行條件
LDDT□、ANDDT□、ORDT□	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存被比較資料的起始元件	—	帶符號BIN16位元	ANY_DT*1
(s2)	儲存被比較資料的起始元件	—	帶符號BIN16位元	ANY_DT*1
(s3)	表示比較對象的值或儲存了比較對象的資料數	0001H~0007H、8001H~8007H	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

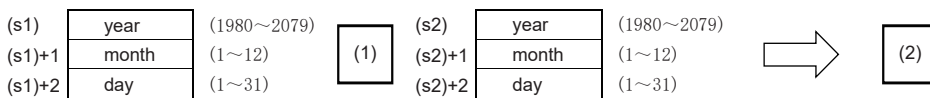
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s3)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 進行 (s1)、(s2) 中指定的日期資料的比較，或將當前的日期與 (s1) 中指定的日期資料進行比較。透過 (s3) 可以選擇比較對象。

- 與任意日期資料的比較

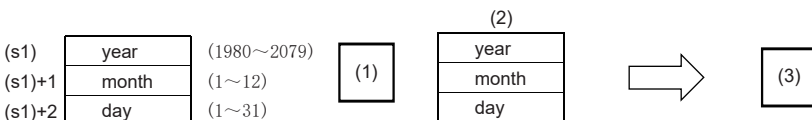
將 (s1) 中指定的日期資料與 (s2) 中指定的日期資料的比較按照 (s3) 的條件進行常開觸點處理。



- (1) 比較運算符
- (2) 比較運算結果

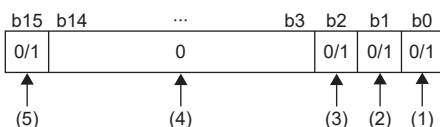
- 與當前的日期資料的比較

將 (s1) 中指定的日期資料與當前的日期資料的比較按照 (s3) 的條件進行常開觸點處理。(s2) 中指定的日期資料被作為虛擬資料處理而忽略。



- (1) 比較運算符
- (2) 當前的日期資料
- (3) 比較運算結果

- 各項目的設定以BIN值進行設定。
- 對於 (s1)、(s2) 的“年”，以公曆4位在1980~2079內設定。
- 對於 (s1)+1、(s2)+1 的“月”，在1~12(1月~12月)內設定。
- 對於 (s1)+2、(s2)+2 的“日”，在1~31(1日~31日)內設定。
- 透過在 (s3) 中指定下述值，可以對比較對象進行詳細設定。(s3) 的位元配置如下所示。



- (1) 在比較對象中添加“日”。
- (2) 在比較對象中添加“月”。
- (3) 在比較對象中添加“年”。
- (4) 指定為0。指定為0以外時，與運算結果無關，將變為非導通。
- (5) 將第15位元設為ON(1)時，將 (s1) 與當前的日期以0~第2位中指定的條件進行比較。

- 比較對象日期(0~第2位)為0的情況下，不進行比較對象的日期資料(年、月、日)的比較。為1的情況下，對比較對象的日期資料(年、月、日)進行比較。
- 比較運算對象(第15位元)為0的情況下，進行 (s1) 中指定的日期資料與 (s2) 中指定的日期資料的比較。為1的情況下，進行 (s1) 中指定的日期資料與當前的日期資料的比較。(s2) 中指定的日期資料將被忽略。
- 比較對象位元的處理內容如下所示。

與任意日期資料比較時的 (s3) 值	與當前日期資料比較時的 (s3) 值	比較對象日期	處理內容
0001H	8001H	日	僅對 (s1)+2 進行比較。
0002H	8002H	月	僅對 (s1)+1 進行比較。
0003H	8003H	月、日	對 (s1)+1、(s1)+2 進行比較。
0004H	8004H	年	僅對 (s1) 進行比較。
0005H	8005H	年、日	對 (s1)、(s1)+2 進行比較。
0006H	8006H	年、月	對 (s1)、(s1)+1 進行比較。
0007H	8007H	年、月、日	對 (s1)、(s1)+1、(s1)+2 均進行比較。
0001H~0007H、8001H~8007H以外		無	對 (s1)、(s1)+1、(s1)+2 均不進行比較。(變為非導通。)

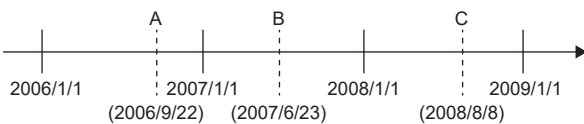
- 比較對象元件中儲存的資料不能被識別為日期資料的情況下，執行指令後SM709(DT、TM指令不正確資料檢測標誌)將ON，變為非導通。即使不能被識別為日期資料的情況下，只要是在設定範圍內，則SM709不變為ON。(s1)~(s1)+2或(s2)~(s2)+2 超出了元件/標籤記憶體各設定區域的情況下，SM709也將變為ON，變為非導通。對於SM709，一旦變為ON，因為在重設/電源OFF時之前將保持為ON狀態，因此應根據需要將其設為OFF。

• 各指令的比較運算結果如下所示。

指令符號(梯形圖、FBD/LD)	條件	比較運算結果
DT=、EQ	$(s1) = (s2)$	導通狀態 (ENO為ON)
DT<>、NE	$(s1) \neq (s2)$	
DT>、GT	$(s1) > (s2)$	
DT<=、LE	$(s1) \leq (s2)$	
DT<、LT	$(s1) < (s2)$	
DT>=、GE	$(s1) \geq (s2)$	
DT=、EQ	$(s1) \neq (s2)$	非導通狀態 (ENO為OFF)
DT<>、NE	$(s1) = (s2)$	
DT>、GT	$(s1) \leq (s2)$	
DT<=、LE	$(s1) > (s2)$	
DT<、LT	$(s1) \geq (s2)$	
DT>=、GE	$(s1) < (s2)$	

例

A、B、C各個日期的比較如下所示。



• 上述日期A、B、C的比較運算結果如下所示。即使以相同條件進行了比較的情況下，根據選擇的比較對象其比較運算結果有所不同。

比較對象	比較條件*1		
	A<B	B<C	A<C
日	導通	非導通	非導通
月	非導通	導通	非導通
月、日	非導通	導通	非導通
年	導通	導通	導通
年、日	導通	導通	導通
年、月	導通	導通	導通
年、月、日	導通	導通	導通
無	非導通	非導通	非導通

*1 FBD/LD的情況下，導通時ENO顯示為ON、非導通時ENO顯示為OFF。

- 即使比較的日期不存在的情況下，只要是在設定範圍內的日期，將按照下述條件進行比較運算。
- 日期A: 2006/02/30 (不存在，但可進行日期的設定。)
- 日期B: 2007/03/29
- 日期C: 2008/02/31 (不存在，但可進行日期的設定。)

比較對象	比較條件*2		
	A<B	B<C	A<C
日	非導通	非導通	導通
月	非導通	非導通	非導通
月、日	導通	非導通	導通
年	導通	導通	導通
年、日	導通	導通	導通
年、月	導通	導通	導通
年、月、日	導通	導通	導通
無	非導通	非導通	非導通

*2 FBD/LD的情況下，導通時ENO顯示為ON、非導通時ENO顯示為OFF。

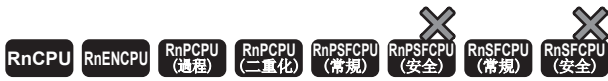
- 在FBD/LD中使用LDDT_□指令時，應將左母線或常時ON的變數部件/常數部件指定到EN。
- 在FBD/LD中使用ORDT_□指令的情況下，將EN指定為TRUE時ENO將ON。EN不成為執行條件。

出錯

沒有運算出錯。

時間比較

LDTM□、ANDTM□、ORTM□



進行指定的時間資料的比較，或進行指定的時間資料與當前時間的比較。

梯形圖	ST*1
	ENO:=LDTM_□(EN, s1, s2, s3); ENO:=ANDTM_□(EN, s1, s2, s3); ENO:=ORTM_□(EN, s1, s2, s3); (□中放入EQ、NE、GT、LE、LT、GE。)*2

(□中輸入TM=、TM<>、TM>、TM<=、TM<、TM>=。)

FBD/LD

(□中放入LDTM_、ANDTM_、ORTM_與EQ、NE、GT、LE、LT、GE的組合。)*2

*1 在版本為“1.035M”及其以後的工程工具中對應。
 *2 EQ為=、NE為<>、GT為>、LE為<=、LT為<、GE為>=。

執行條件

指令	執行條件
LDTM□、ANDTM□、ORTM□	常時執行

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存被比較資料的起始元件	—	帶符號BIN16位元	ANY_TM*1
(s2)	儲存被比較資料的起始元件	—	帶符號BIN16位元	ANY_TM*1
(s3)	表示比較對象的值或儲存了比較對象的資料數	0001H~0007H、 8001H~8007H	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

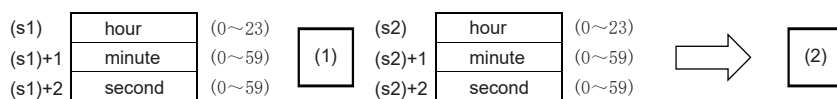
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(s3)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

• 對(s1)、(s2)中指定的時間資料進行比較，或對(s1)中指定的時間資料與當前的時間進行比較。透過(s3)可以選擇比較對象。

• 與任意時間資料的比較

將(s1)中指定的時間資料與(s2)中指定的時間資料的比較按照(s3)的條件進行常開觸點處理。

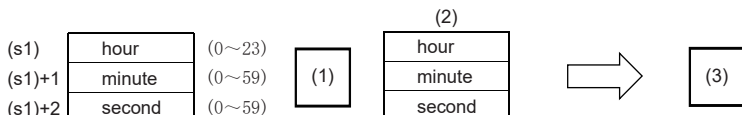


(1) 比較運算符

(2) 比較運算結果

• 與當前的時間資料的比較

將(s1)中指定的時間資料與當前的時間資料的比較按照(s3)的條件進行常開觸點處理。(s2)中指定的時間資料將被作為虛擬資料處理而忽略。



(1) 比較運算符

(2) 當前的時間資料

(3) 比較運算結果

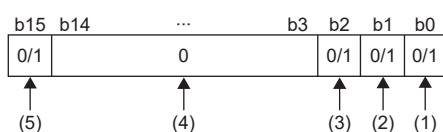
• 各項目的設定以BIN值進行設定。

• 對於(s1)、(s2)的“時”，以0~23(0時~23時)的24小時制進行設定。

• 對於(s1)+1、(s2)+1的“分”，以0~59(0分~59分)進行設定。

• 對於(s1)+2、(s2)+2的“秒”，以0~59(0秒~59秒)進行設定。

• 透過在(s3)中指定下述值，可以對比較對象進行詳細設定。(s3)的位元配置如下所示。



(1) 在比較對象中添加“秒”。

(2) 在比較對象中添加“分”。

(3) 在比較對象中添加“時”。

(4) 指定為0。指定為0以外時，與運算結果無關，將變為非導通。

(5) 將第15位元設為0N(1)時，將(s1)與當前的時間以0~第2位中指定的條件進行比較。

• 比較對象時間(0~第2位元)為0的情況下，不進行比較對象的時間資料(時、分、秒)的比較。為1的情況下，進行比較對象的時間資料(時、分、秒)之比較。

• 比較運算對象(第15位元)為0的情況下，對(s1)中指定的時間資料與(s2)中指定的時間資料進行比較。為1的情況下，對(s1)中指定的時間資料與當前的時間資料進行比較。(s2)中指定的時間資料將被忽略。

• 比較對象位元的處理內容如下所示。

與任意時間資料比較時的(s3)值	與當前時間資料比較時的(s3)值	比較對象時間	處理內容
0001H	8001H	秒	僅對(s1)+2進行比較。
0002H	8002H	分	僅對(s1)+1進行比較。
0003H	8003H	分、秒	對(s1)+1、(s1)+2進行比較。
0004H	8004H	時	僅對(s1)進行比較。
0005H	8005H	時、秒	對(s1)、(s1)+2進行比較。
0006H	8006H	時、分	對(s1)、(s1)+1進行比較。
0007H	8007H	時、分、秒	對(s1)、(s1)+1、(s1)+2均進行比較。
0001H~0007H、8001H~8007H以外		無	對(s1)、(s1)+1、(s1)+2均不進行比較。(變為非導通。)

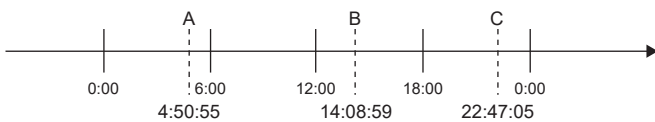
• 比較對象元件中儲存的資料在不能識別為時間資料的情況下，執行指令後SM709(DT、TM指令不正確資料檢測標誌)將ON，變為非導通(ENO為OFF)。(s1)~(s1)+2或(s2)~(s2)+2超出了元件/標籤記憶體的各設定區域的情況下，SM709也將變為ON，變為非導通(ENO為OFF)。對於SM709，一旦變為ON，因為在重設/電源OFF時之前將保持為ON狀態，因此應根據需要將其設為OFF。

• 各指令的比較運算結果如下所示。

指令符號(梯形圖、FBD/LD)	條件	比較運算結果
TM=、EQ	$(s1) = (s2)$	導通狀態(ENO為ON)
TM<>、NE	$(s1) \neq (s2)$	
TM>、GT	$(s1) > (s2)$	
TM<=、LE	$(s1) \leq (s2)$	
TM<、LT	$(s1) < (s2)$	
TM>=、GE	$(s1) \geq (s2)$	
TM=、EQ	$(s1) \neq (s2)$	非導通狀態(ENO為OFF)
TM<>、NE	$(s1) = (s2)$	
TM>、GT	$(s1) \leq (s2)$	
TM<=、LE	$(s1) > (s2)$	
TM<、LT	$(s1) \geq (s2)$	
TM>=、GE	$(s1) < (s2)$	

例

A、B、C各個時間的比較如下所示。



• 上述時間A、B、C的比較運算結果如下所示。即使以相同條件進行了比較的情況下，根據選擇的比較對象其比較運算結果有所不同。

比較對象	比較條件*1		
	A<B	B<C	A<C
秒	導通	非導通	非導通
分	非導通	導通	非導通
分、秒	非導通	導通	非導通
時	導通	導通	導通
時、秒	導通	導通	導通
時、分	導通	導通	導通
時、分、秒	導通	導通	導通
無	非導通	非導通	非導通

*1 FBD/LD的情況下，導通時ENO顯示為ON、非導通時ENO顯示為OFF。

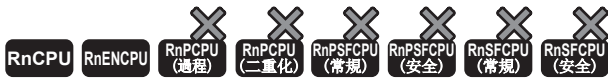
- 在FBD/LD中使用LDTM_□指令時，應將左母線或常時ON的變數部件/常數部件指定到EN。
- 透過FBD/LD使用ORTM_□指令的情況下，將EN指定為TRUE時ENO將ON。EN不成為執行條件。

出錯

沒有運算出錯。

時間比較輸出

TCMP (P)

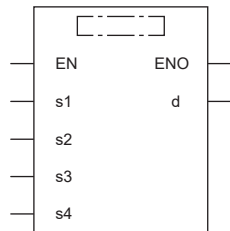


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

比較 (s1)、(s2)、(s3) 中指定的比較時間資料與 (s4) 中指定的時間資料，根據比較的結果 (小、一致、大)，(d)、(d)+1、(d)+2 其中之一將變為 ON。

梯形圖	ST
	<pre>ENO:=TCMP (EN, s1, s2, s3, s4, d); ENO:=TCMPP (EN, s1, s2, s3, s4, d);</pre>

FBD/LD



■執行條件

指令	執行條件
TCMP	
TCMPP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	儲存比較時間資料 (時) 的起始元件	0~23	帶符號BIN16位元	ANY16
(s2)	儲存比較時間資料 (分) 的起始元件	0~59	帶符號BIN16位元	ANY16
(s3)	儲存比較時間資料 (秒) 的起始元件	0~59	帶符號BIN16位元	ANY16
(s4)	儲存比較時間資料 (時、分、秒) 的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
(d)	儲存比較結果的起始元件	—	位元	ANYBIT_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

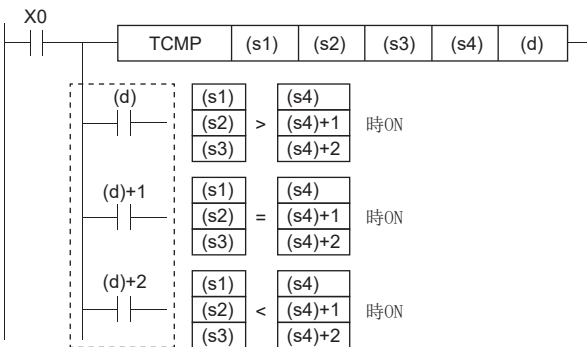
操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s1)	○*1	○	○	○	○	—	—	○	○	—	—	—
(s2)	○*1	○	○	○	○	—	—	○	○	—	—	—
(s3)	○*1	○	○	○	○	—	—	○	○	—	—	—
(s4)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	○	—	○*2	—	—	—	—	○	—	—	—	—

*1 不能使用FX、FY。

*2 不能使用T、ST、C。

功能

- 比較(s1)、(s2)、(s3)中指定的比較時間資料與(s4)中指定的時間資料，根據比較的結果(小、一致、大)，(d)、(d)+1、(d)+2其中之一將變為ON。



• (s1): 時、(s2): 分、(s3): 秒

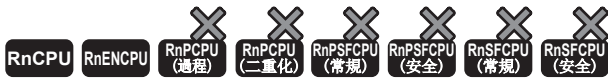
• (s4): 時、(s4)+1: 分、(s4)+2: 秒

出錯

出錯代碼 (SD0)	內容
3405H	(s1)、(s4) 中指定的值超出下述範圍時。 0~23
	(s2)、(s3)、(s4)+1、(s4)+2 中指定的值超出下述範圍時。 0~59

時間區域比較輸出

TZCP (P)

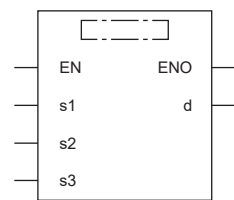


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

將下限值 (s1) 的時間資料和上限值 (s2) 的時間資料的區域與比較時間資料 (s3) 進行比較，根據結果 (下、區域內、上) 決定 (d)、(d)+1、(d)+2 哪個為ON。

梯形圖	ST
	ENO:=TZCP (EN, s1, s2, s3, d) ; ENO:=TZCPP (EN, s1, s2, s3, d) ;

FBD/LD



■執行條件

指令	執行條件
TZCP	
TZCPP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	儲存比較時間資料 (時、分、秒) 的下限值的起始元件	參閱詳細功能	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
(s2)	儲存比較時間資料 (時、分、秒) 的上限值的起始元件	參閱詳細功能	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
(s3)	儲存比較時間資料 (時、分、秒) 的起始元件	參閱詳細功能	帶符號BIN16位元	ANY16_ARRAY (要素數: 3)
(d)	儲存比較結果的起始元件	—	位元	ANYBIT_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

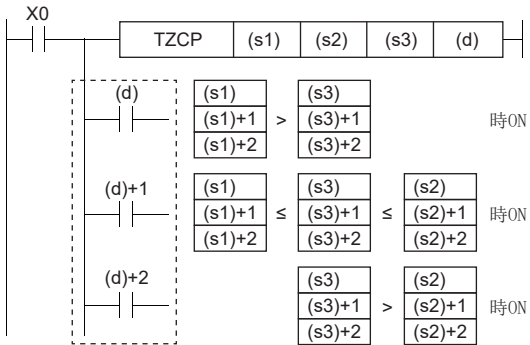
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—	—
(s3)	—	—	○	—	—	—	○	—	—	—	—	—
(d)	○	—	○*1	—	—	—	○	—	—	—	—	—

*1 不能使用T、ST、C。

功能

- 將下限值 (s1) 的時間資料和上限值 (s2) 的時間資料的區域與比較時間資料 (s3) 進行比較，根據結果 (下、區域內、上) 決定 (d)、(d)+1、(d)+2 哪個為 ON。



元件	時鐘資料	資料範圍
(s1)、(s2)、(s3)	時	0~23
(s1)+1、(s2)+1、(s3)+1	分	0~59
(s1)+2、(s2)+2、(s3)+2	秒	0~59

注意事項

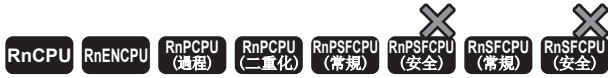
- 應將下限值 (s1) 設為比上限值 (s2) 小的值。將下限值 (s1) 設為大於上限值 (s2) 時，將上限值 (s2) 作為與下限值 (s1) 相同的值處理。

出錯

出錯代碼 (SD0)	內容
3405H	(s1)、(s2)、(s3) 中指定的值超出下述範圍時。 0~23
	(s1)+1、(s2)+1、(s3)+1、(s1)+2、(s2)+2、(s3)+2 中指定的值超出下述範圍時。 0~59

擴展時鐘資料的讀取

S(P). DATERD



從CPU內部的時鐘單元中讀取包含有1/1000秒的時鐘資料。

梯形圖	ST
	<pre>ENO:=S_DATERD(EN, d); ENO:=SP_DATERD(EN, d);</pre>

FBD/LD

■執行條件

指令	執行條件
S. DATERD	
SP. DATERD	

■設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(d)	儲存讀取的時鐘資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 8)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	
(d)	—	—	○	—	—	—	○	—	—	—	—

功能

- 從CPU模組的時鐘單元中讀取“年、月、日、時、分、秒、星期、1/1000秒”，以BIN值儲存到(d)中指定的元件及其以後。

(資料)	(d)	(d)+1	(d)+2	(d)+3	(d)+4	(d)+5	(d)+6	(d)+7
(內容)	年(公曆)	月	日	時(24小時制)	分	秒	星期	1/1000秒
(設定範圍)	1980~2079	1~12	1~31	0~23	0~59	0~59	0~6	0~999

- 對(d)的“年”，以公曆4位儲存。
- 對於(d)+6的“星期”，將“日~六”以“0~6”儲存。

星期	日	一	二	三	四	五	六
儲存資料	0	1	2	3	4	5	6

- 閏年將被自動補償。

注意事項

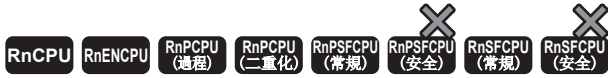
- 對於S(P).DATERD指令，即使CPU模組中設定了錯誤的時鐘資料的情況下，也將讀取時鐘資料並儲存到元件中。(例：2月30日)。透過DATEWR(P)指令及工程工具設定時鐘資料時，應設定正確的時鐘資料。
- 1/1000秒的時鐘資料讀取時的誤差最大為2ms。(CPU模組內部時鐘單元中儲存的資料與透過S(P).DATERD指令讀取的資料的誤差。)

出錯

沒有運算出錯。

擴展時鐘資料的加法運算

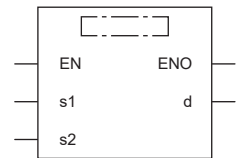
S(P). DATE+



對時間資料進行加法運算。

梯形圖	ST*1
	<pre>ENO:=S_DATEPLUS(EN, s1, s2, d); ENO:=SP_DATEPLUS(EN, s1, s2, d);</pre>

FBD/LD



(□中放入S_DATEPLUS、SP_DATEPLUS。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
S. DATE+	
SP. DATE+	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存加上時間資料的起始元件	功能參照	帶符號BIN16位元	ANY16_ARRAY (要素數: 5)
(s2)	儲存加法運算時間(時刻)資料的起始元件	功能參照	帶符號BIN16位元	ANY16_ARRAY (要素數: 5)
(d)	儲存加法運算結果時刻(時間)資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 5)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—	—

功能

- 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，將加法運算結果儲存到(d)中指定的元件編號及其以後。

(s1)	hour	(0~23)	(s2)	hour	(0~23)	(d)	hour	(0~23)
(s1)+1	minute	(0~59)	(s2)+1	minute	(0~59)	(d)+1	minute	(0~59)
(s1)+2	second	(0~59)	(s2)+2	second	(0~59)	(d)+2	second	(0~59)
(s1)+3	-		(s2)+3	-		(d)+3	-	
(s1)+4	1/1000second	(0~999)	(s2)+4	1/1000second	(0~999)	(d)+4	1/1000second	(0~999)

例

將6時32分40秒875與7時48分10秒500進行加法運算的情況下

(s1)	6	(s2)	7	(d)	14
(s1)+1	32	(s2)+1	48	(d)+1	20
(s1)+2	40	(s2)+2	10	(d)+2	51
(s1)+3	-	(s2)+3	-	(d)+3	-
(s1)+4	875	(s2)+4	500	(d)+4	375

- 運算結果的時間超過了24小時的情況下，減去24小時後的值將成為運算結果。例如，對14時20分30秒875與20時20分20秒500進行了加法運算的情況下，其結果不是34時40分51秒375，而是10時40分51秒375。

(s1)	14	(s2)	20	(d)	10
(s1)+1	20	(s2)+1	20	(d)+1	40
(s1)+2	30	(s2)+2	20	(d)+2	51
(s1)+3	-	(s2)+3	-	(d)+3	-
(s1)+4	875	(s2)+4	500	(d)+4	375

要點

- (s1)+3、(s2)+3、(d)+3的元件在運算中不使用。
- 可以將透過S(P).DATERD指令讀取的時鐘資料直接進行加法運算。

(d)	時
(d)+1	分
(d)+2	秒
(d)+3	星期
(d)+4	1/1000秒

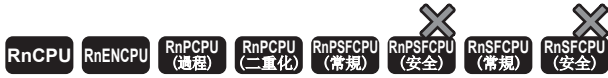
透過S(P).DATERD指令讀取的情況下，在“秒”與“1/1000秒”之間放入“星期”。
在S(P).DATE+指令中，因為不進行運算，因此可以直接進行加法運算。

出錯

出錯代碼(SD0)	內容
3405H	(s1)、(s2)的設定資料超出範圍時。

擴展時鐘資料的減法運算

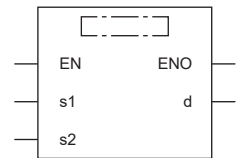
S(P). DATE-



對時間資料進行減法運算。

梯形圖	ST*1
	<pre>ENO:=S_DATEMINUS(EN, s1, s2, d); ENO:=SP_DATEMINUS(EN, s1, s2, d);</pre>

FBD/LD



(□中放入S_DATEMINUS、SP_DATEMINUS。)

*1 在版本為“1.035M”及其以後的工程工具中對應。

■執行條件

指令	執行條件
S. DATE-	
SP. DATE-	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存被減法運算時間資料的起始元件	功能參照	帶符號BIN16位元	ANY16_ARRAY (要素數: 5)
(s2)	儲存減法運算時間(時刻)資料的起始元件	功能參照	帶符號BIN16位元	ANY16_ARRAY (要素數: 5)
(d)	儲存減法運算結果時刻(時間)資料的起始元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 5)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—	—

功能

- (s1) 中指定的時間資料減去 (s2) 中指定的時間資料，將減法運算結果儲存到 (d) 中指定的元件編號及其以後。

(s1)	hour	(0~23)	(s2)	hour	(0~23)	(d)	hour	(0~23)
(s1)+1	minute	(0~59)	(s2)+1	minute	(0~59)	(d)+1	minute	(0~59)
(s1)+2	second	(0~59)	(s2)+2	second	(0~59)	(d)+2	second	(0~59)
(s1)+3	-		(s2)+3	-		(d)+3	-	
(s1)+4	1/1000second	(0~999)	(s2)+4	1/1000second	(0~999)	(d)+4	1/1000second	(0~999)

例

將10時40分20秒875與3時50分10秒500進行減法運算的情況下

(s1)	10	(s2)	3	(d)	6
(s1)+1	40	(s2)+1	50	(d)+1	50
(s1)+2	20	(s2)+2	10	(d)+2	10
(s1)+3	-	(s2)+3	-	(d)+3	-
(s1)+4	875	(s2)+4	500	(d)+4	375

- 運算結果時間為負數的情況下，將該資料進行了+24的值將成為運算結果。例如，將4時50分32秒875與10時42分12秒500進行減法運算的情況下，其結果不是-6時8分20秒375，而是18時8分20秒375。

(s1)	4	(s2)	10	(d)	18
(s1)+1	50	(s2)+1	42	(d)+1	8
(s1)+2	32	(s2)+2	12	(d)+2	20
(s1)+3	-	(s2)+3	-	(d)+3	-
(s1)+4	875	(s2)+4	500	(d)+4	375

要點

- (s1)+3、(s2)+3、(d)+3的元件在運算中不使用。
- 可以將透過S(P). DATERD指令讀取的時鐘資料直接進行減法運算。

(d)	時
(d)+1	分
(d)+2	秒
(d)+3	星期
(d)+4	1/1000秒

透過S(P). DATERD指令讀取的情況下，在“秒”與“1/1000秒”之間放入“星期”。
因為在S(P). DATE-指令中不進行運算，因此可以直接進行減法運算。

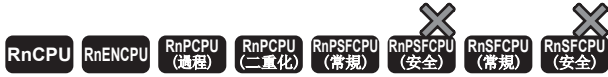
出錯

出錯代碼 (SD0)	內容
3405H	(s1)、(s2)的設定資料超出範圍時。

19.2 時機計測指令

時機脈衝發生

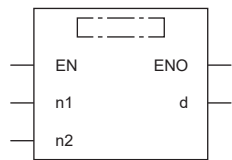
DUTY



將用戶用時機時鐘按指定的掃描數設為ON、按指定的掃描數設為OFF。

梯形圖	ST
	ENO:=DUTY (EN, n1, n2, d);

FBD/LD



執行條件

指令	執行條件
DUTY	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(n1)	設為ON的掃描數	0~65535	無符號BIN16位元	ANY16
(n2)	設為OFF的掃描數	0~65535	無符號BIN16位元	ANY16
(d)	動作的用戶用時機時鐘特殊繼電器元件編號	SM420~SM424	位元	ANY_BOOL*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用SM420~SM424中分配的標籤。

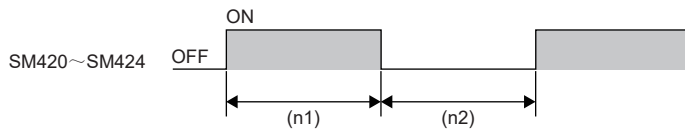
可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(n1)	○	○	○	○	○	—	—	○	○	—	—	—	
(n2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○*1	—	—	—	—	—	—	○	—	—	—	—	

*1 只能使用SM420~SM424。

功能

- 將 (d) 中指定的SM420~SM424 (用戶用時機時鐘) 按照 (n1) 中指定的掃描數設為ON、按照 (n2) 中指定的掃描數設為OFF。



(n1): (n1) 掃描

(n2): (n2) 掃描

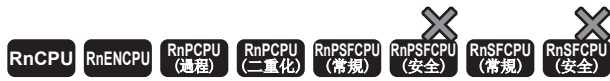
- 在掃描執行類型程式中使用SM420~SM424。
- (n1)、(n2) 被設定為0的情況下，(n1)=0、(n2)≥0時SM420~SM424將保持為OFF不變，(n1)>0、(n2)=0時SM420~SM424將保持為ON不變。
- 執行DUTY指令時將(n1)、(n2)、(d)中指定的資料登錄到系統中，時機脈衝的ON/OFF透過END處理進行。

出錯

出錯代碼 (SD0)	內容
2820H	(d) 中指定的元件為SM420~SM424以外時。

指定資料的時間計測

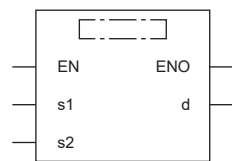
TIMCHK



對元件的ON時間進行計測，如果連續ON的時間不小於設定時間，則將指定元件設為ON。

梯形圖	ST
	$ENO := TIMCHK(EN, s1, s2, d);$

FBD/LD



■執行條件

指令	執行條件
TIMCHK	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存計測的當前值的元件(單位: 100ms)	—	帶符號BIN16位元	ANY16
(s2)	計測的設定值或儲存計測的設定值的元件(單位: 100ms)	0~32767	帶符號BIN16位元	ANY16
(d)	時限到時設為ON的元件	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	—	○	—	—	—	—	○	—	—	—	—	—

功能

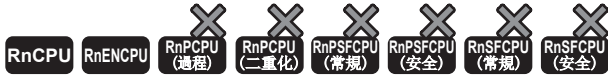
- 對條件元件的ON時間進行計測，如果連續ON的時間不小於(s2)中指定的元件中所設定的時間，則將(d)中指定的元件設為ON。
- 在執行指令的上升沿時，將執行(s1)中指定的元件的當前值的歸零及(d)中指定的元件的OFF。對於(s1)中指定的元件的當前值與(d)中指定的元件的ON狀態，即使執行指令為OFF其狀態也將被保持。
- 計測的當前值以100ms為單位被儲存。此外，計測的設定值以100ms為單位進行設定。
- (s2)中指定了0的情況下，在執行指令的上升沿時將進行(s1)中指定的元件的當前值的歸零及(d)中指定的元件的OFF。
- 將(s2)指定為0~32767範圍外時，執行指令變為ON後的下一個掃描時(d)將變為ON。

出錯

沒有運算出錯。

計時器

HOURL



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

啟動觸點為ON的時間是以小時為單位進行計測，ON的時間累計達到(s)中指定的時間(BIN16位元資料)時，將(d2)中指定的元件設為ON。

梯形圖	ST
	ENO:=HOURL (EN, s, d1, d2) ;

FBD/LD

■執行條件

指令	執行條件
HOURL	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	將報警(d2)設為ON的時間(以小時為單位進行設定)	0~32767	帶符號BIN16位元	ANY16
(d1)	儲存計測的當前值的元件	—	帶符號BIN16位元	ANY16_ARRAY (要素數: 2)
(d2)	時限到時設為ON的元件(報警輸出)	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○*1	○	○	○	○	—	—	○	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d2)	○	—	○*2	—	—	—	—	—	—	—	—	—	—

*1 不能使用FX、FY。

*2 不能使用T、ST、C、FD。

功能

- 啟動觸點為ON的時間是以小時為單位進行計測，ON的時間累計達到(s)中指定的時間(BIN16位元資料)時，將(d2)中指定的元件設為ON。
- 以小時為單位設定使(s)中的(d2)設為ON所需的時間。
- (d1)中儲存以小時為單位計測的當前值。
- (d1)+1中儲存不足1小時的計測的當前值(1秒單位)。
- 即使(d2)中指定的報警輸出變為ON後仍繼續計測。
- 在(d1)達到最大值(32767)且(d1)+1也到達最大值(3599)的時刻時，停止計測。
- 即使(d1)中設定負值也會動作。
- HOURM指令的動作不影響時鐘資料的更改(包含夏季時間功能補償)。

注意事項

- 從初始值開始計測時和當前值達到16位的最大值後仍繼續計測時，若是透過元件指定(d1)的情況下，應將(d1)~(d1)+1清除；若是透過標籤指定(d1)的情況下，應將2要素部分清除。
- 由於定時器未能正常計測，應勿使用於初始執行類型程式、中斷程式、恆定週期執行類型程式、事件執行類型程式。

出錯

出錯代碼(SD0)	內容
3405H	(s)中指定的範圍超出下述範圍時。 0~32767

DHOURM



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。

啟動觸點為ON的時間是以小時為單位進行計測，ON的時間累計達到(s)中指定的時間(BIN32位元資料)時，將(d2)中指定的元件設為ON。

梯形圖	ST
	ENO:=DHOURM(EN, s, d1, d2);

FBD/LD

■執行條件

指令	執行條件
DHOURM	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	將報警(d2)設為ON的時間(以小時為單位進行設定)	0~2147483647	帶符號BIN32位元	ANY32
(d1)	儲存計測的當前值的元件	—	帶符號BIN32位元	ANY32_ARRAY (要素數: 2)
(d2)	時限到時設為ON的元件(報警輸出)	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H	E	
(s)	○*1	○	○	○	○	○	○	○	—	—	—	—
(d1)	—	—	○	—	—	—	○	—	—	—	—	—
(d2)	○	—	○*2	—	—	—	—	—	—	—	—	—

*1 不能使用FX、FY。

*2 不能使用T、ST、C、FD。

功能

- 啟動觸點為ON的時間是以小時為單位進行計測，ON的時間累計達到(s)中指定的時間(BIN32位元資料)時，將(d2)中指定的元件設為ON。
- 以小時為單位設定使(s)+1、(s)中的(d2)設為ON所需的時間。
- (d1)+1、(d1)中儲存以小時為單位計測的當前值。((d1)+1：高位、d1)：低位)
- (d1)+2中儲存不足1小時的計測的當前值(1秒單位)。
- 值不儲存在(d1)+3中。
- 即使(d2)中指定的報警輸出變為ON後仍繼續計測。
- 在(d1)+1、(d1)達到最大值(2147483647)且(d1)+2也到達最大值(3599)的時刻時停止計測。
- 即使(d1)中設定負值也會動作。
- DHOORM指令的動作不影響時鐘資料的更改(包含夏季時間功能補償)。

注意事項

- 從初始值開始計測時和當前值達到32位的最大值後仍繼續計測時，若是透過元件指定(d1)的情況下，應將(d1)~(d1)+2清除；若是透過標籤指定(d1)的情況下，應將2要素部分清除。
- 由於定時器未能正常計測，應勿使用於初始執行類型程式、中斷程式、恆定週期執行類型程式、事件執行類型程式。

出錯

出錯代碼(SD0)	內容
3405H	(s)中指定的範圍超出下述範圍時。 0~2147483647

20 模組存取

20.1 模組存取指令

I/O更新

RFS (P)



對指定的元件開始的n點進行更新，進行外部輸入的獲取或至輸出模組的輸出。

梯形圖	ST
	ENO:=RFS (EN, s, n); ENO:=RFSP (EN, s, n);

FBD/LD

■執行條件

指令	執行條件
RFS	
RFSP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	更新的起始元件	—	位元	ANY_BOOL*1
(n)	更新點數	1~4096	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 只能使用元件 (X、Y) 中分配的標籤。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○*1	—	—	—	—	—	—	—	—	—	—	—	—
(n)	—	○	○	○	○	—	—	○	○	—	—	—	—

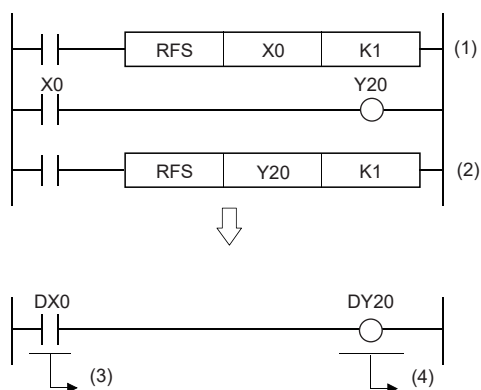
*1 只能使用X、Y。

功能

- 是1個掃描過程中僅對相應的元件進行更新，進行外部輸入的獲取或至輸出模組的輸出的功能。
- 因為輸入的獲取及至外部的輸出僅在執行程式的END指令後批量地進行，因此1個掃描中不能向外部輸出脈衝信號。執行I/O更新指令時，因為程式執行過程中相應的輸入(X)或輸出(Y)將被強制更新，因此1個掃描中可以向外部輸出脈衝信號。
- 將輸入(X)或輸出(Y)以1點單位進行更新的情況下，應使用直接存取輸入(DX)、直接存取輸出(DY)。
- 在過程CPU(二重化模式)中指定了由待機系統分配至擴展基板上的模組的輸入(X)及輸出(Y)的情況下，將變為無處理。

例

將RFS指令的程式設定為直接存取程式時



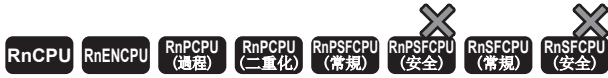
- (1) 更新X0。
- (2) 更新Y20。
- (3) 直接存取輸入
- (4) 直接存取輸出

出錯

出錯代碼(SD0)	內容
2820H	(s) 中指定的元件開始的(n)點的範圍超出了相鄰I/O的範圍時。

選擇更新

COM(P)



進行I/O更新及網路模組的連結更新、元件/標籤存取服務處理等。

梯形圖	ST
	ENO:=COM(EN) ; ENO:=COMP(EN) ;

FBD/LD

■執行條件

指令	執行條件
COM	
COMP	

功能

- 對於COM(P)指令，以順程式執行過程中的任意時機，在實施I/O更新等的情況下使用。
- 執行COM(P)指令時，將進行下述處理。
 - I/O更新
 - CC-Link模組的連結更新
 - CC-Link IE控制網路模組的連結更新
 - CC-Link IE現場網路模組的連結更新*1
 - CC-Link IE現場網路Basic的連結更新
 - MELSECNET/H模組的連結更新
 - 智能功能模組的更新*2
 - 使用了多CPU系統的CPU緩衝記憶體更新(END時)
 - 多CPU系統的組外的輸入/輸出的獲取
 - 元件/標籤存取服務處理(與工程工具、GOT或其它外部設備的通訊)
- 將SM775(執行COM指令時更新處理選擇)設為OFF時，進行除I/O更新以外的處理。

處理內容	SM775 OFF時	SM775 ON時
I/O更新、多CPU系統的組外的輸入/輸出的獲取	非執行	可以選擇執行/非執行
CC-Link模組的連結更新	執行	
CC-Link IE控制網路模組的連結更新		
CC-Link IE現場網路模組的連結更新*1		
CC-Link IE現場網路Basic的連結更新		
MELSECNET/H模組的連結更新		
智能功能模組的更新*2		
使用了多CPU系統的CPU緩衝記憶體的更新(END時)		
元件/標籤存取服務處理(與工程工具、GOT或其它外部設備的通訊)		

*1 也包括簡單運動控制模組(RD77GF)的連結更新。
 *2 也包括簡單運動控制模組(RD77GF)、運動控制模組的緩衝記憶體更新、RJ71GN11-EIP(EtherNet/IP部)的更新。

- 於選擇處理的情況下，在SD775(執行COM指令時更新處理選擇)的b0~b4、b6、b13、b15中選擇執行/非執行後，將SM775設為ON。

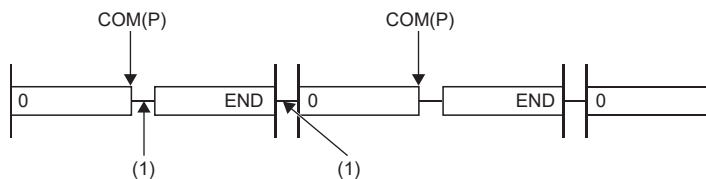
	b15	b14	b13	b12...b7	b6	b5	b4	b3	b2	b1	b0
SD775	1/0	0	1/0	0	1/0	0	1/0	1/0	1/0	1/0	1/0
	↑		↑		↑		↑	↑	↑	↑	↑
	(8)		(7)		(6)		(5)	(4)	(3)	(2)	(1)

- (1) I/O更新、多CPU系統的組外的輸入/輸出的獲取
- (2) CC-Link模組的連結更新
- (3) CC-Link IE控制網路模組、MELSECNET/H模組的連結更新
- (4) 智能功能模組*2
- (5) 使用了多CPU系統的CPU緩衝記憶體更新(END時)
- (6) CC-Link IE現場網路模組的連結更新*1
- (7) CC-Link IE現場網路Basic的連結更新
- (8) 元件/標籤存取服務處理(與工程工具、GOT或其它外部設備的通訊)

*1 也包括簡單運動控制模組(RD77GF)的連結更新。

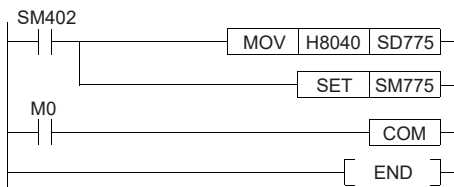
*2 也包括簡單運動控制模組(RD77GF)、運動控制模組的緩衝記憶體更新、RJ71GN11-EIP(EtherNet/IP部)的更新。

- 執行COM(P)指令時，進行指定的更新處理。



(1): 指定的處理

- 在下述程式示例中，M0變為ON時，執行CC-Link IE現場網路模組的連結更新。



- 在過程CPU(二重化模式)中由待機系統對擴展基板上的模組執行下述處理時將變為無處理。
- I/O更新
- CC-Link模組的連結更新
- 智能功能模組的更新

注意事項

- COM(P)指令在程式中的使用次數無限制。但是，根據SD775中選擇的處理時間，程式的掃描時間將相應延長。
- COM(P)指令的執行中，將變為中斷允許狀態。在中斷程式等中使用更新資料等的情況下，有可能發生資料背離。
- 在乙太網路埠上連接乙太網路時，透過COM(P)指令進行了元件/標籤存取服務處理的情況下，COM(P)指令的處理時間有可能延長。

要點

COM(P)指令不能在中斷程式中使用。

出錯

沒有運算出錯。

模組更新

S(P).ZCOM



進行指定模組的更新處理。

梯形圖	ST
	<pre>ENO:=S_ZCOM(EN, J); ENO:=S_ZCOM(EN, U); ENO:=SP_ZCOM(EN, J); ENO:=SP_ZCOM(EN, U);</pre>

FBD/LD

■執行條件

指令	執行條件
S.ZCOM	
SP.ZCOM	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(J)*1*2	本站的網路No.	1~239	元件名	ANY16
(U)*1	模組的起始輸入輸出編號(以16進制數4位表示時的高3位)	0H~FFH		
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 CC-Link IE TSN Plus主站・本地站模組中，(U)指定時RJ71GN11-EIP(CC-Link IE TSN部)的連結更新(更新設定時)及RJ71GN11-EIP(EtherNet/IP部)的更新(更新設定時)動作，(J)指定時僅RJ71GN11-EIP(CC-Link IE TSN部)的連結更新(更新設定時)動作。

*2 簡單運動控制模組(RD77GF)、運動控制模組的緩衝記憶體更新也執行動作。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其它 (J/U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(J/U)	—	—	—	—	—	—	—	—	—	—	—	—	○

• 對於S(P).ZCOM指令，在順控程式的執行途中，以任意時機進行更新的情況下使用。透過S(P).ZCOM指令進行更新的對象如下所示。

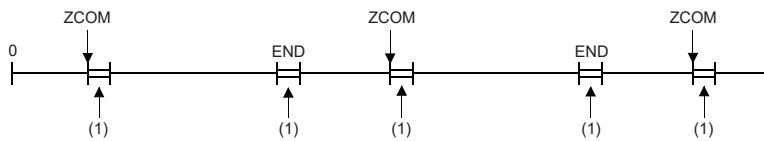
- CC-Link IE TSN主站・本地站模組的連結更新(更新設定時)
- CC-Link IE控制網路模組的連結更新(更新設定時)
- CC-Link IE現場網路主站・本地站模組的連結更新(更新設定時)*1
- MELSECNET/H模組的連結更新(更新設定時)
- CC-Link模組的連結更新(更新設定時)
- 智能功能模組的更新(更新設定時)*2

*1 也包括簡單運動控制模組(RD77GF)的連結更新。

*2 也包括簡單運動控制模組(RD77GF)、運動控制模組的緩衝記憶體更新、RJ71GN11-EIP(EtherNet/IP部)的更新。

功能

- 執行S(P).ZCOM指令時，CPU模組將暫時中斷順控程式的處理，進行(J/U)中指定的模組的更新處理。



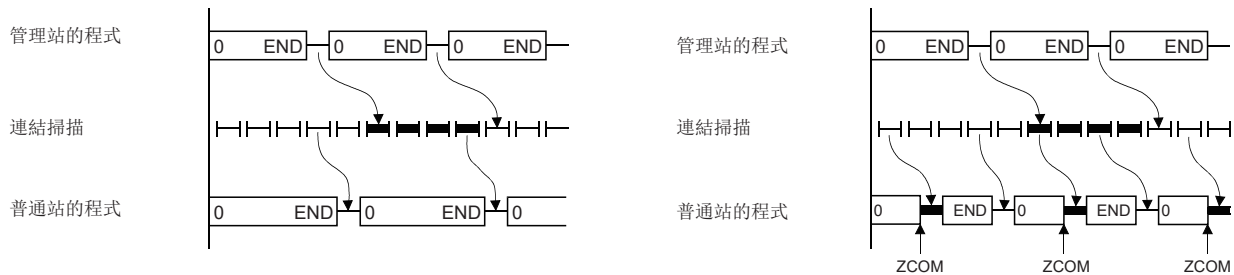
(1): 更新處理

- 進行CC-Link IE控制網路模組及MELSECNET/H模組(可程式控制器間網路)更新處理的情況，如下所示。

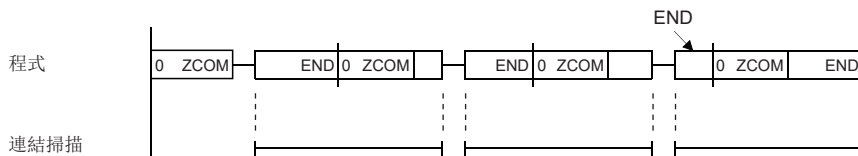
- 對於S(P).ZCOM指令，在本站的順控程式的掃描時間長於其它站的掃描時間的情況下，可靠其獲取來自其它站的資料。

[不使用S(P).ZCOM指令的情況下]

[使用了S(P).ZCOM指令的情況下]



- 順控程式的掃描時間短於連結掃描時間掃描時間的情況下，即使使用S(P).ZCOM指令，資料的發送接收也不會變快。



- 在二重化擴展基板配置中執行了S(P).ZCOM指令的情況下，應參閱下述內容。

☞ 1105頁 COM(P)

注意事項

- S(P).ZCOM指令在程式中的使用次數無限制。但是，根據更新時間，程式的掃描時間將相應延長。
- S(P).ZCOM指令執行中，將變為中斷允許狀態。在中斷程式等中使用更新資料的情況下，有可能發生資料背離。

出錯

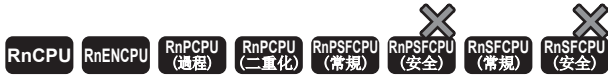
出錯代碼(SD0)	內容
2800H	指定的起始輸入輸出編號為0~FFH以外時。
2801H	起始輸入輸出編號中指定的位置不存在模組時。
2804H	指定的網路No. 超出1~239的範圍時。
2820H	指定的網路No. 未連接到本站時。

要點

- S(P).ZCOM指令不能在中斷程式中使用。
- 希望僅執行與外部設備的通訊的情況下，應使用COM(P)指令。

從模組中的1字/2字資料讀取(16位指定)

FROM(P)、DFROM(P)



• FROM(P) :

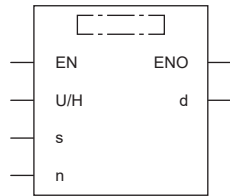
從指定的模組/其它機號CPU模組內的緩衝記憶體地址開始讀取n字元的資料。

• DFROM(P) :

從指定的模組/其它機號CPU模組內的緩衝記憶體地址開始讀取n×2字元的資料。

梯形圖	ST
	ENO:=FROM(EN, U/H, s, n, d); ENO:=FROMP(EN, U/H, s, n, d); ENO:=DFROM(EN, U/H, s, n, d); ENO:=DFROMP(EN, U/H, s, n, d);

FBD/LD



■執行條件

指令	執行條件
FROM DFROM	
FROMP DFROMP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U/H)	模組及CPU模組的起始輸入輸出編號(以16進制數4位表示時的高3位)	0H~FFH、3E0H~3E3H	無符號BIN16位元	ANY16
(s)	儲存讀取資料的緩衝記憶體及CPU記憶體的起始地址	0~65535	無符號BIN16位元	ANY16
(d)	FROM(P)	—	帶符號BIN16位元	ANY16*1
	DFROM(P)		帶符號BIN32位元	ANY32*1
(n)	讀取資料數	0~65535	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

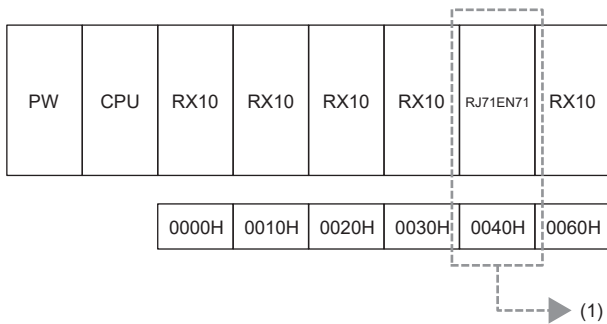
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它 (U)	
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H)G□	Z	LT、 LST、LC		LZ	K、H	E		\$
(U/H)	○	○	○	○	○	—	—	○	○	—	—	○
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	—	○	—	—	○*1	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

*1 只能使用DFROM(P)指令。

功能

- 對於(U/H)，將模組及CPU模組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。



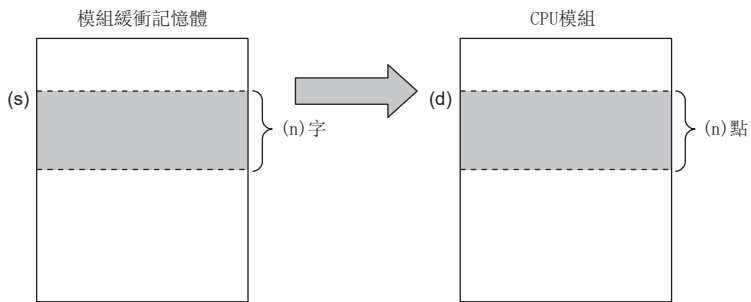
(1): 讀取模組的起始輸入輸出編號指定K4或H4

CPU模組的起始輸入輸出編號，按下表所示指定。

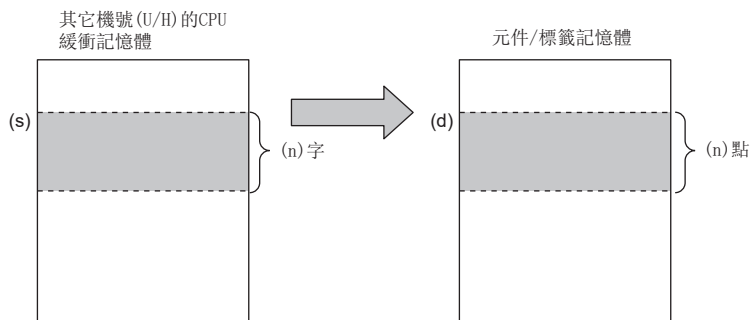
CPU模組	起始輸入輸出編號
1號機	3E0H
2號機	3E1H
3號機	3E2H
4號機	3E3H

■FROM(P)

- 從(U/H)中指定的模組或其它機號CPU模組內的緩衝記憶體(s)中指定的地址開始讀取(n)字元的資料，並儲存到(d)中指定的元件以後。
- 從模組的字元資料讀取



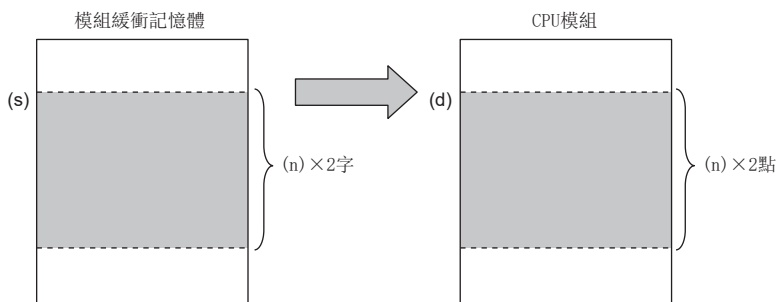
- 從其它機號CPU模組的字元資料讀取



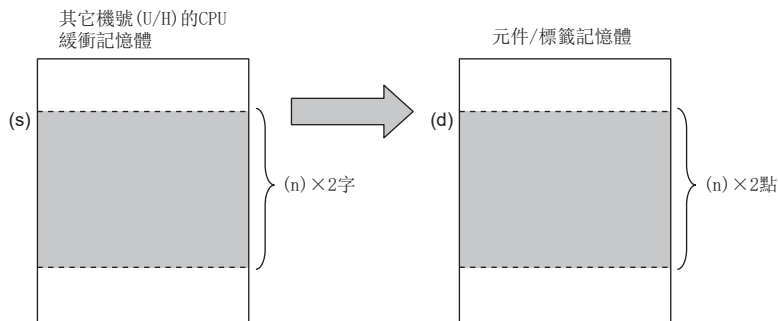
- 讀取資料(n)為0時，將變為無處理。
- 執行指令時，對象模組的異常及處理中等無法存取的情況下，將變為無處理。

■DFROM(P)

- 從(U/H)中指定的模組或其它機號CPU模組內的緩衝記憶體(s)中指定的地址開始讀取(n)×2字元的資料，並儲存到(d)中指定的元件以後。
- 從模組的雙字元資料讀取



- 從其它機號CPU模組的雙字元資料讀取



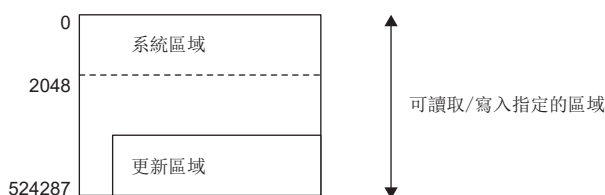
- 讀取資料(n)為0時，將變為無處理。
- 執行指令時，對象模組的異常及處理中等無法存取的情況下，將變為無處理。

出錯

出錯代碼 (SD0)	內容
2820H	(U) 中指定的輸入輸出編號不具有緩衝記憶體模組時。
2823H	(H) 中指定的輸入輸出編號不具有緩衝記憶體的模組時。
	(s) 中指定的地址超出緩衝記憶體/CPU緩衝記憶體的範圍時。
	(s) 中指定的地址開始的 (n) 點超出緩衝記憶體/CPU緩衝記憶體的範圍時。(FROM(P) 指令的情況下)
	(s) 中指定的地址開始的 2×(n) 點超出緩衝記憶體/CPU緩衝記憶體的範圍時。(DFROM(P) 指令的情況下)
3461H	嘗試由待機系統存取安裝在擴展基板上的模組的緩衝記憶體時。

要點

- 模組的資料讀取也可使用模組存取元件進行。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
- 對於CPU緩衝記憶體的可讀取/寫入區域的更新區域，不進行更新設定的情況下，可以作為可讀取/寫入指定的區域使用。此外，即使進行了更新設定的情況下，更新發送範圍以後也可作為可讀取/寫入指定的區域使用。



- 從CPU緩衝記憶體的資料讀取時，可以使用CPU緩衝記憶體存取元件。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
- 在FROM(P) 指令、DFROM(P) 指令中，可以從不超過64K的緩衝記憶體地址中讀取資料。從超過64K的緩衝記憶體地址中讀取資料的情況下，使用FROMD(P) 指令、DFROMD(P) 指令。(1118頁 FROMD(P)、DFROMD(P))

至模組的1字元/2字元資料寫入(16位元指定)

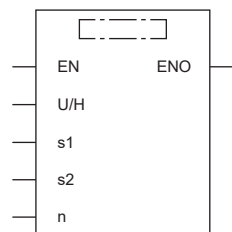
TO(P)、DTO(P)



- TO(P)：將指定的元件開始的n點的資料寫入到模組/本機CPU模組內的緩衝記憶體中。
- DTO(P)：將指定的元件開始的n×2點的資料寫入到模組/本機CPU模組內的緩衝記憶體中。

梯形圖	ST
	ENO:=TO(EN, U/H, s1, s2, n); ENO:=TOP(EN, U/H, s1, s2, n); ENO:=DTO(EN, U/H, s1, s2, n); ENO:=DTOP(EN, U/H, s1, s2, n);

FBD/LD



■執行條件

指令	執行條件
TO DTO	
TOP DTOP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)	
(U/H)	模組及CPU模組的起始輸入輸出編號(以16進制數4位表示時的高3位)	0H~FFH、3E0H~3E3H	無符號BIN16位元	ANY16	
(s1)	寫入資料的緩衝記憶體及CPU記憶體的起始地址	0~65535	無符號BIN16位元	ANY16	
(s2)	寫入資料或儲存寫入資料的起始元件	TO(P)	-32768~32767	帶符號BIN16位元	ANY16*1
		DTO(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32*1
(n)	寫入資料數	0~65535	無符號BIN16位元	ANY16	
EN	執行條件	—	位元	BOOL	
ENO	執行結果	—	位元	BOOL	

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

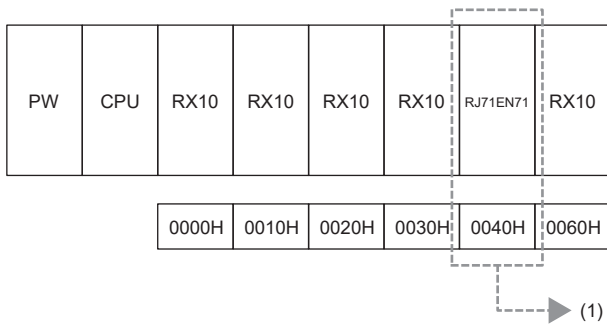
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它 (U)	
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H)G□	Z	LT、 LST、LC		LZ	K、H	E		\$
(U/H)	○	○	○	○	○	—	—	○	○	—	—	○
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	—	○	—	—	○*1	—	○	○	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

*1 只能使用DT0(P)指令。

功能

- 對於(U/H)，將模組及CPU模組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。



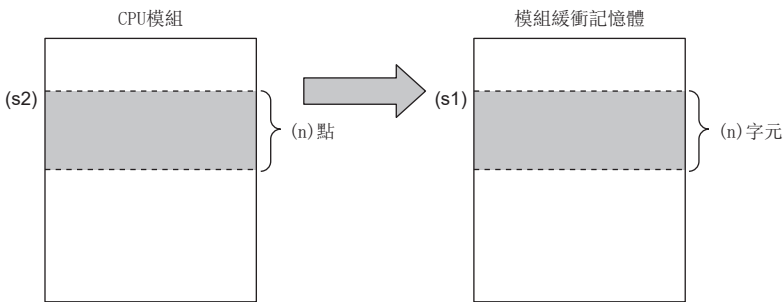
(1): 寫入模組的起始輸入輸出編號指定K4或H4

CPU模組的起始輸入輸出編號，按下表所示指定。

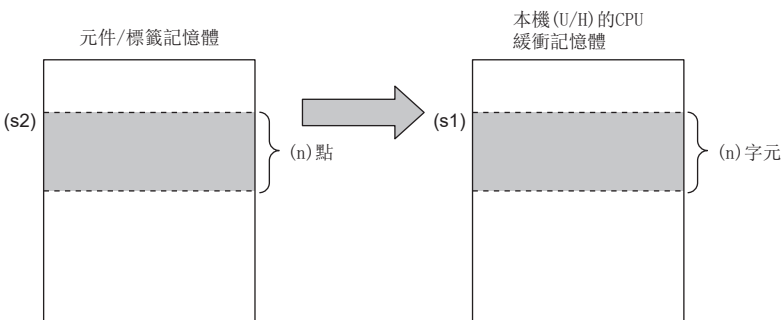
CPU模組	起始輸入輸出編號
1號機	3E0H
2號機	3E1H
3號機	3E2H
4號機	3E3H

■T0(P)

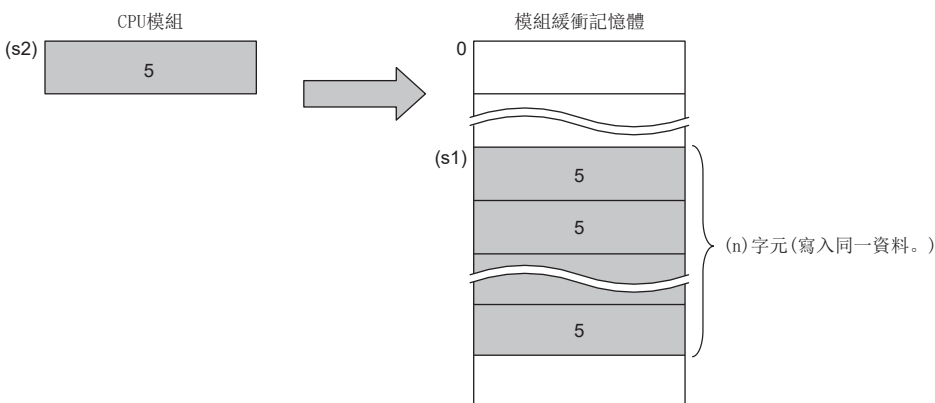
- 將(s2)中指定的元件開始的(n)點的資料，寫入到(U/H)中指定的模組或本機CPU模組內的緩衝記憶體(s1)中指定的地址及其以後。
- 至模組的字元資料寫入



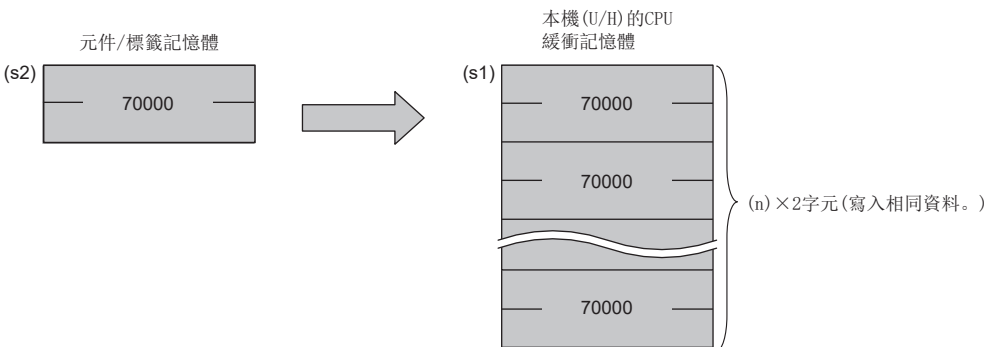
- 至本機CPU模組的字元資料寫入



- (s2)中指定了常數的情況下，將同一資料((s2)中指定的值)，寫入到指定的緩衝記憶體地址開始的(n)字元中。
- 至模組的字元資料寫入



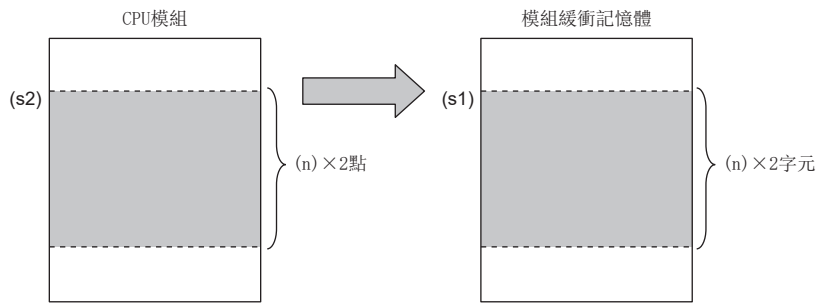
- 至本機CPU模組的字元資料寫入



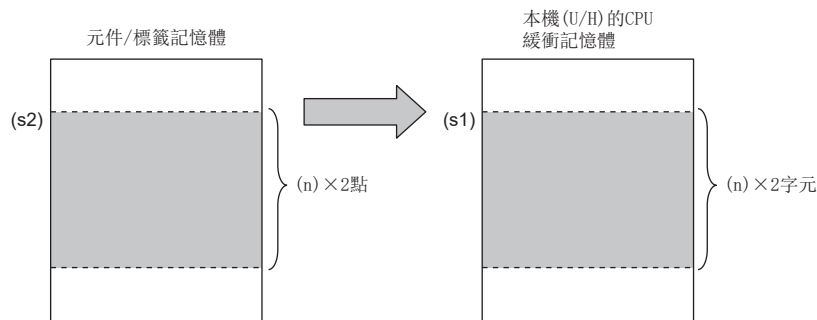
- 執行指令時，對象模組的異常及處理中等無法存取的情況下，將變為無處理。

■DT0(P)

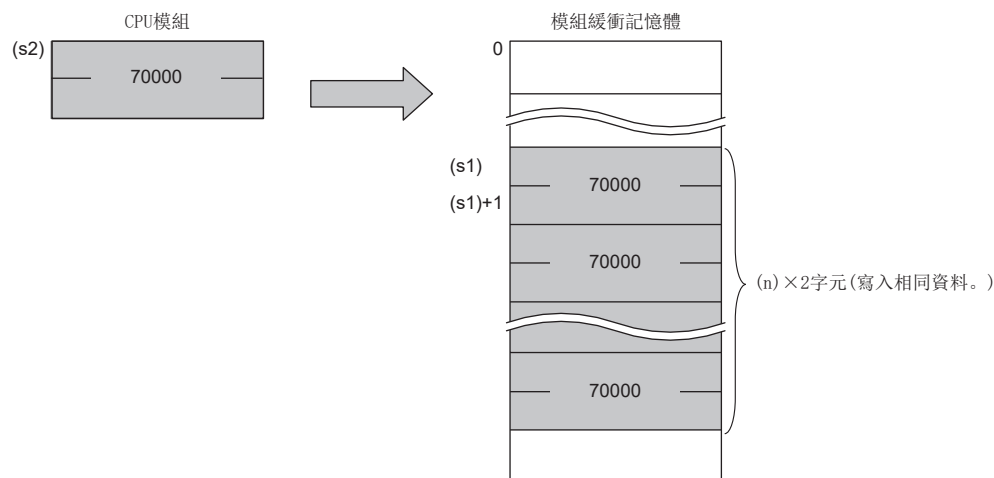
- 將(s2)中指定的元件開始的(n)×2點的資料，寫入到(U/H)中指定的模組或本機CPU模組內的緩衝記憶體的(s1)中指定的地址及其以後。
- 至模組的雙字元資料寫入



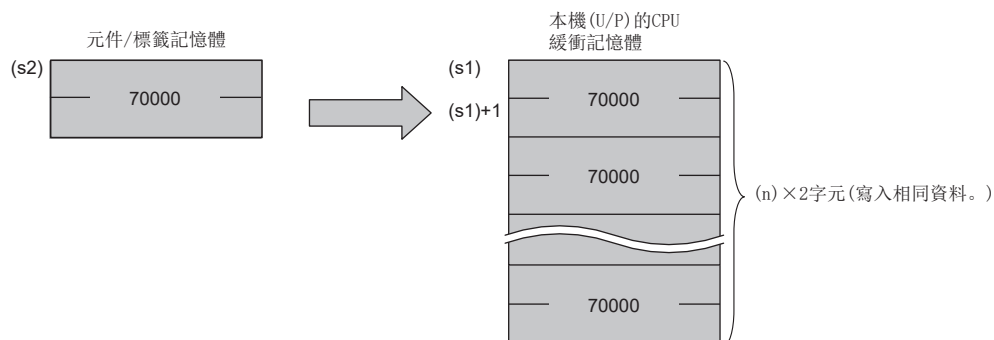
- 至本機CPU模組的雙字元資料寫入



- (s2)中指定了常數的情況下，將同一資料((s2)中指定的值)，寫入到指定的緩衝記憶體地址開始的(n)×2字元中。
- 至模組的雙字元資料寫入



- 至本機CPU模組的雙字元資料寫入



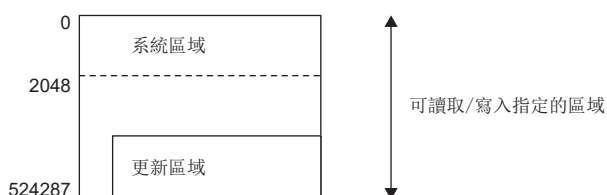
- 執行指令時，對象模組的異常及處理中等無法存取的情況下，將變為無處理。

出錯

出錯代碼 (SD0)	內容
2820H	(U) 中指定的輸入輸出編號不具有緩衝記憶體模組時。
2823H	(H) 中指定的輸入輸出編號不具有緩衝記憶體的模組時。
	(s1) 中指定的地址超出緩衝記憶體/CPU緩衝記憶體的範圍時。
	(s1) 中指定的地址開始的(n)點超出緩衝記憶體/CPU緩衝記憶體的範圍時。(T0(P)指令的情況下)
	(s1) 中指定的地址開始的2×(n)點超出緩衝記憶體/CPU緩衝記憶體的範圍時。(DT0(P)指令的情況下)
3461H	嘗試由待機系統存取安裝在擴展基板上的模組的緩衝記憶體時。

要點

- 對於CPU緩衝記憶體的可讀取/寫入區域的更新區域，不進行更新設定的情況下，可以作為可讀取/寫入指定的區域使用。此外，即使進行了更新設定的情況下，更新發送範圍以後也可作為可讀取/寫入指定的區域使用。



- 至CPU緩衝記憶體的資料的寫入時，可以使用CPU緩衝記憶體存取元件。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
- 在T0(P)指令、DT0(P)指令中，可以將資料寫入不超過64K的緩衝記憶體地址中。將資料寫入超過64K的緩衝記憶體地址中的情況下，使用TOD(P)指令、DTOD(P)指令。(1122頁 TOD(P)、DTOD(P))

從模組中的1字元/2字元資料讀取 (32位元指定)

FROMD (P)、DFROMD (P)



• FROMD (P) :

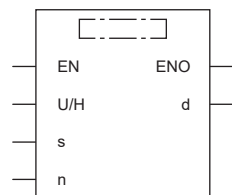
從指定的模組/其它機號CPU模組內的緩衝記憶體地址開始讀取n字元的資料。

• DFROMD (P) :

從指定的模組/其它機號CPU模組內的緩衝記憶體地址開始讀取n×2字元的資料。

梯形圖	ST
	ENO:=FROMD (EN, U/H, s, n, d) ; ENO:=FROMDP (EN, U/H, s, n, d) ; ENO:=DFROMD (EN, U/H, s, n, d) ; ENO:=DFROMDP (EN, U/H, s, n, d) ;

FBD/LD



■執行條件

指令	執行條件
FROMD DFROMD	
FROMDP DFROMDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(U/H)	模組及CPU模組的起始輸入輸出編號 (以16進制數4位表示時的高3位)	0H~FFH、3E0H~3E3H	無符號BIN16位元	ANY16
(s)	儲存讀取資料的緩衝記憶體及儲存CPU記憶體的起始地址的起始元件	0~4294967295	無符號BIN32位元	ANY32
(d)	FROMD (P)	—	帶符號BIN16位元	ANY16* ¹
	DFROMD (P)		帶符號BIN32位元	ANY32* ¹
(n)	讀取資料數	0~4294967295	無符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

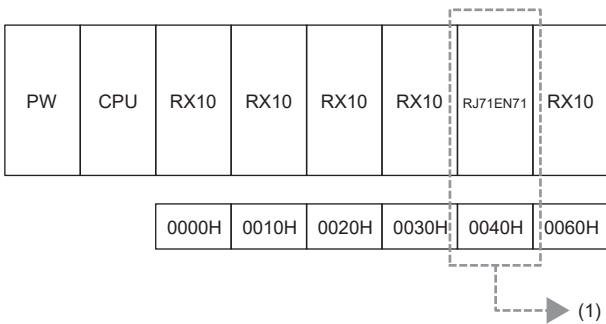
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它 (U)	
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H)G□	Z	LT、 LST、LC		LZ	K、H	E		\$
(U/H)	○	○	○	○	○	—	—	○	○	—	—	○
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	—	○	—	—	○*1	—	○	○	—	—	—
(n)	○	○	○	○	○	○	○	○	○	—	—	—

*1 只能使用DFROMD(P)指令。

功能

- 對於(U/H)，將模組及CPU模組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。



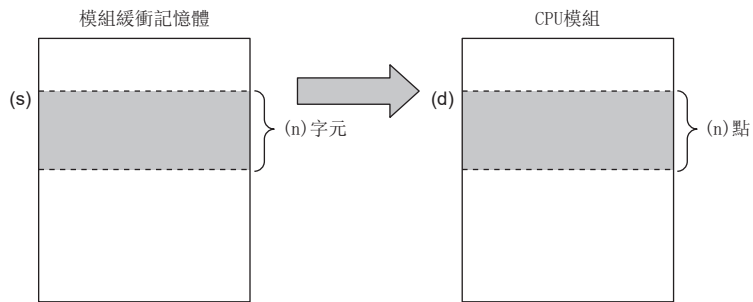
(1): 讀取模組的起始輸入輸出編號指定K4或H4

CPU模組的起始輸入輸出編號，按下表所示指定。

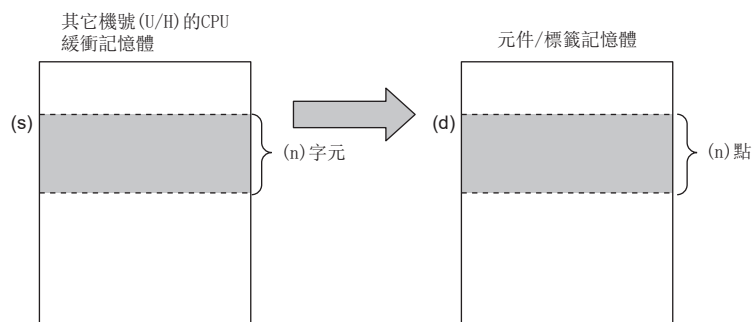
CPU模組	起始輸入輸出編號
1號機	3E0H
2號機	3E1H
3號機	3E2H
4號機	3E3H

FROMD (P)

- 從 (U/H) 中指定的模組或其它機號CPU模組內的緩衝記憶體 (s) 中指定的地址開始讀取 (n) 字元的資料，並儲存到 (d) 中指定的元件以後。
- 從模組的字元資料讀取



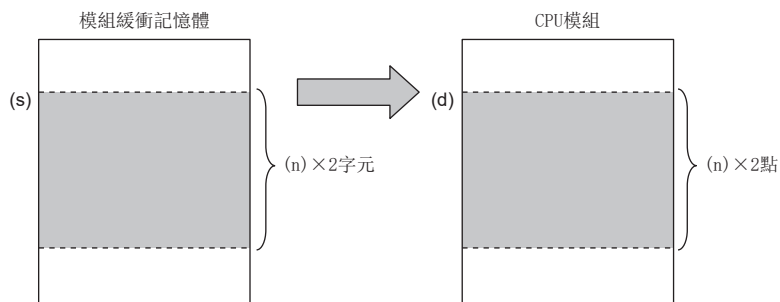
- 從其它機號CPU模組的字元資料讀取



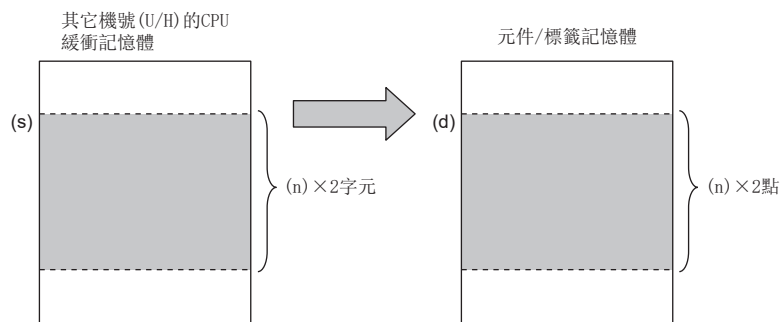
- 讀取資料 (n) 為 0 時，將變為無處理。
- 執行指令時，對象模組的異常及處理中等無法存取的情況下，將變為無處理。

DFROMD (P)

- 從 (U/H) 中指定的模組或其它機號CPU模組內的緩衝記憶體的 (s) 中指定的地址開始讀取 (n) × 2 字元的資料，並儲存到 (d) 中指定的元件以後。
- 從模組的雙字元資料讀取



- 從其它機號CPU模組的雙字元資料讀取



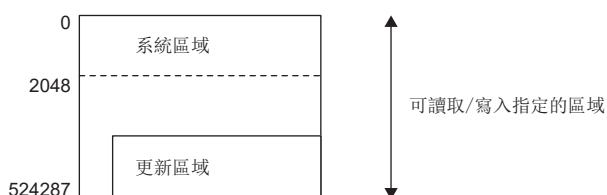
- 讀取資料 (n) 為 0 時，將變為無處理。
- 執行指令時，對象模組的異常及處理中等無法存取的情況下，將變為無處理。

出錯

出錯代碼 (SD0)	內容
2820H	(U) 中指定的輸入輸出編號不具有緩衝記憶體模組時。
2823H	(H) 中指定的輸入輸出編號不具有緩衝記憶體的模組時。
	(s) 中指定的地址超出緩衝記憶體/CPU緩衝記憶體的範圍時。
	(s) 中指定的地址開始的 (n) 點超出緩衝記憶體/CPU緩衝記憶體的範圍時。(FROMD (P) 指令的情況下)
	(s) 中指定的地址開始的 2×(n) 點超出緩衝記憶體/CPU緩衝記憶體的範圍時。(DFROMD (P) 指令的情況下)
3461H	嘗試由待機系統存取安裝在擴展基板上的模組的緩衝記憶體時。

要點

- 對於CPU緩衝記憶體的可讀取/寫入區域的更新區域，不進行更新設定的情況下，可以作為可讀取/寫入指定的區域使用。此外，即使進行了更新設定的情況下，更新發送範圍以後也可作為可讀取/寫入指定的區域使用。



- 從CPU緩衝記憶體的資料讀取時，可以使用CPU緩衝記憶體存取元件。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
- 在FROMD (P) 指令、DFROMD (P) 指令中，可以從超過64K的緩衝記憶體地址中讀取資料。

至模組的1字元/2字元資料寫入(32位元指定)

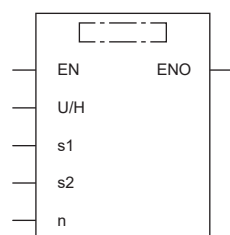
TOD(P)、DTOD(P)



- TOD(P)：將指定的元件開始的n點的資料寫入到模組/本機CPU模組內的緩衝記憶體中。
- DTOD(P)：將指定的元件開始的n×2點的資料寫入到模組/本機CPU模組內的緩衝記憶體中。

梯形圖	ST
	ENO:=TOD(EN, U/H, s1, s2, n); ENO:=TODP(EN, U/H, s1, s2, n); ENO:=DTOD(EN, U/H, s1, s2, n); ENO:=DTODP(EN, U/H, s1, s2, n);

FBD/LD



■執行條件

指令	執行條件
TOD DTOD	
TODP DTODP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U/H)	模組及CPU模組的起始輸入輸出編號(以16進制數4位表示時的高3位)	0H~FFH、3E0H~3E3H	無符號BIN16位元	ANY16
(s1)	寫入資料的緩衝記憶體及儲存CPU記憶體的起始地址的起始元件	0~4294967295	無符號BIN32位元	ANY32
(s2)	TOD(P)	-32768~32767	帶符號BIN16位元	ANY16*1
	DTOD(P)	-2147483648~2147483647	帶符號BIN32位元	ANY32*1
(n)	寫入資料數	0~4294967295	無符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

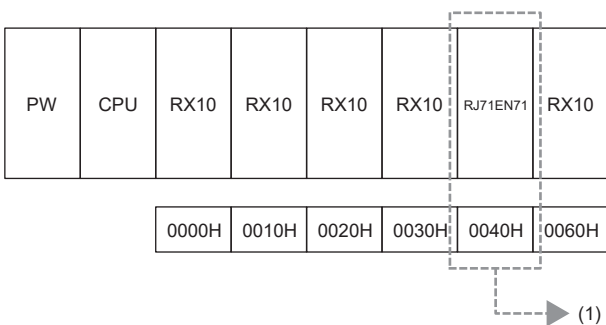
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H)G□	Z	LT、 LST、LC	LZ		K、H	E	\$	
(U/H)	○	○	○	○	○	—	—	○	○	—	—	○
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	—	○	—	—	○*1	—	○	○	—	—	—
(n)	○	○	○	○	○	○	○	○	○	—	—	—

*1 只能使用DTOD(P)指令。

功能

- 對於(U/H)，將模組及CPU模組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。



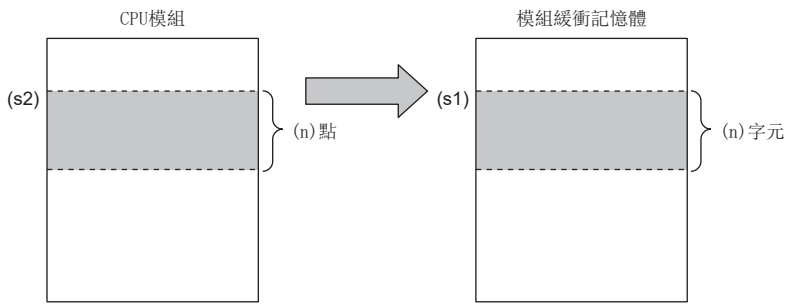
(1): 寫入模組的起始輸入輸出編號指定K4或H4

CPU模組的起始輸入輸出編號，按下表所示指定。

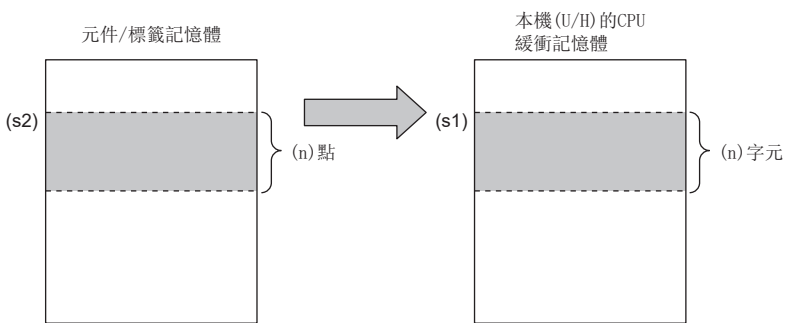
CPU模組	起始輸入輸出編號
1號機	3E0H
2號機	3E1H
3號機	3E2H
4號機	3E3H

■TOD(P)

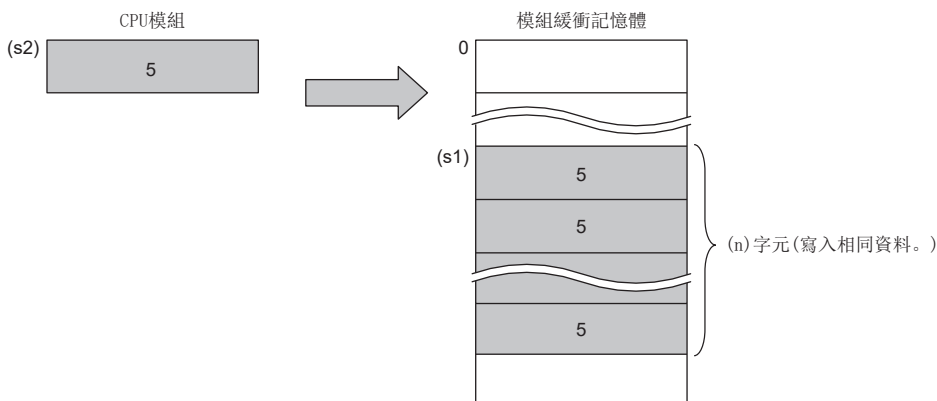
- 將 (s2) 中指定的元件開始的 (n) 點的資料，寫入到 (U/H) 中指定的模組或本機CPU模組內的緩衝記憶體 (s1) 中指定的地址及其以後。
- 至模組的字元資料寫入



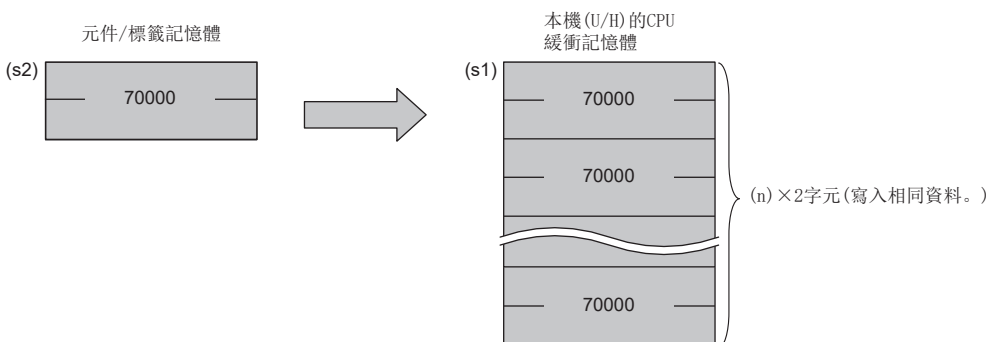
- 至本機CPU模組的字元資料寫入



- (s2) 中指定了常數的情況下，將同一資料 ((s2) 中指定的值) 寫入到指定的緩衝記憶體地址開始的 (n) 字元中
- 至模組的字元資料寫入



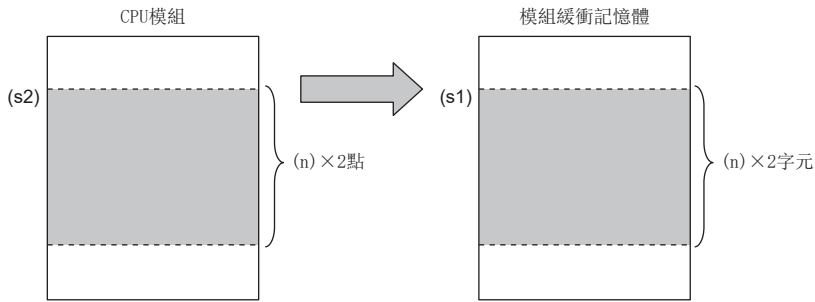
- 至本機CPU模組的字元資料寫入



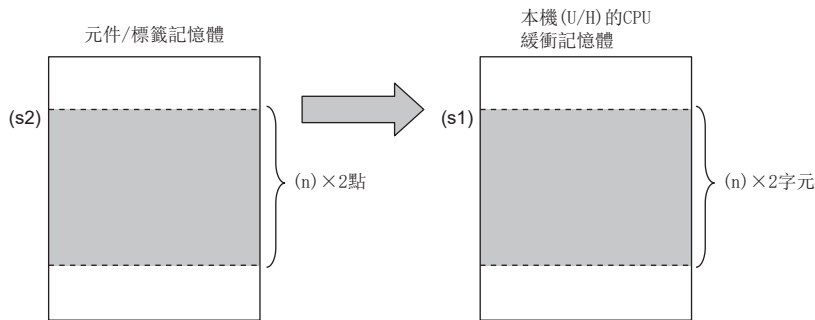
- 執行指令時，對象模組的異常及處理中等無法存取的情況下，將變為無處理。

■DTOD(P)

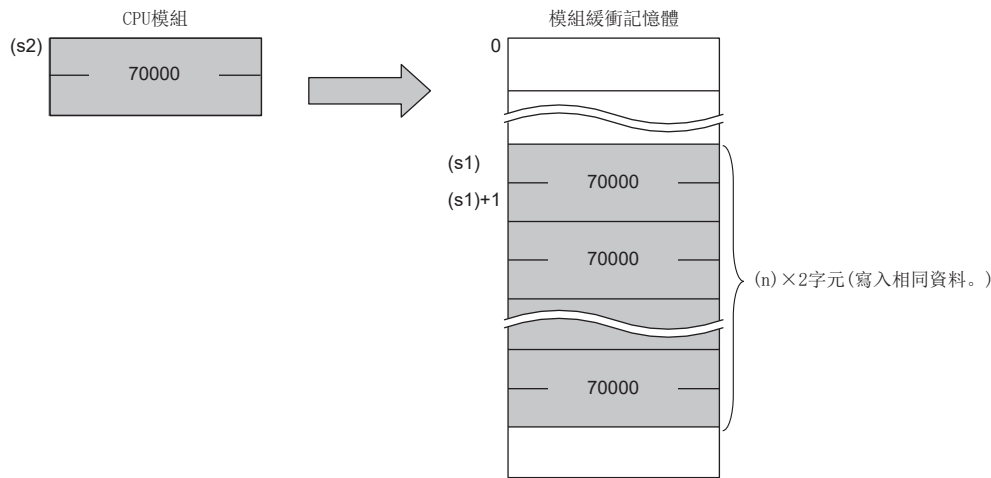
- 將(s2)中指定的元件開始的(n)×2點的資料，寫入到(U/H)中指定的模組或本機CPU模組內的緩衝記憶體的(s1)中指定的地址及其以後。
- 至模組的雙字元資料寫入



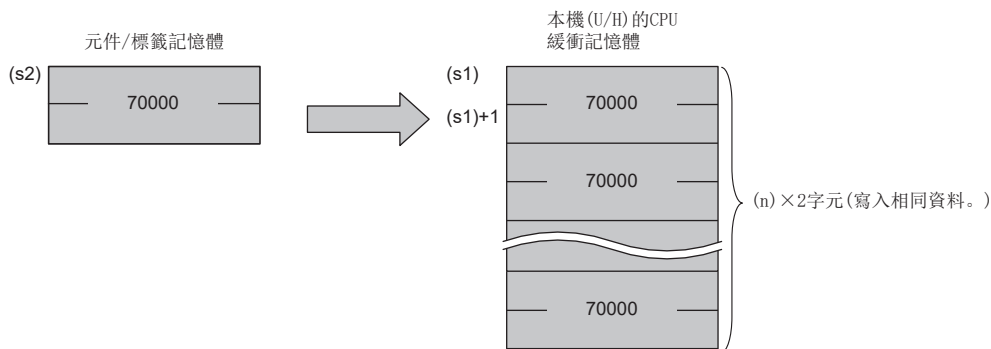
- 至本機CPU模組的雙字元資料寫入



- (s2)中指定了常數的情況下，將同一資料((s2)中指定的值)寫入到指定的緩衝記憶體地址開始的(n)×2字元中。
- 至模組的雙字元資料寫入



- 至本機CPU模組的雙字元資料寫入



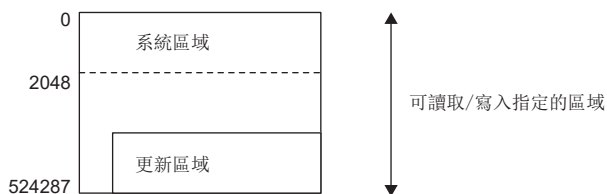
- 執行指令時，對象模組的異常及處理中等無法存取的情況下，將變為無處理。

出錯

出錯代碼 (SD0)	內容
2820H	(U) 中指定的輸入輸出編號不具有緩衝記憶體模組時。
2823H	(H) 中指定的輸入輸出編號不具有緩衝記憶體的模組時。
	(s1) 中指定的地址超出緩衝記憶體/CPU緩衝記憶體的範圍時。
	(s1) 中指定的地址開始的(n)點超出緩衝記憶體/CPU緩衝記憶體的範圍時。(TOD(P)指令的情況下)
	(s1) 中指定的地址開始的 $2 \times (n)$ 點超出緩衝記憶體/CPU緩衝記憶體的範圍時。(DTOD(P)指令的情況下)
3461H	嘗試由待機系統存取安裝在擴展基板上的模組的緩衝記憶體時。

要點

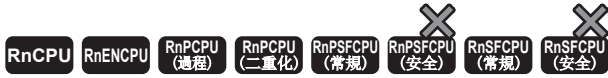
- 對於CPU緩衝記憶體的可讀取/寫入區域的更新區域，不進行更新設定的情況下，可以作為可讀取/寫入指定的區域使用。此外，即使進行了更新設定的情況下，更新發送範圍以後也可作為可讀取/寫入指定的區域使用。



- 至CPU緩衝記憶體的資料的寫入時，可以使用CPU緩衝記憶體存取元件。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
- 在TOD(P)指令、DTOD(P)指令中，可以將資料寫入到超過64K的緩衝記憶體地址中。

模組型號讀取

TYPERD (P)



讀取指定的插槽的模組型號。

梯形圖	ST
	ENO:=TYPERD (EN, H, d) ; ENO:=TYPERDP (EN, H, d) ;

FBD/LD

■執行條件

指令	執行條件
TYPERD	
TYPERDP	

設置數據

■內容、範圍、資料類型

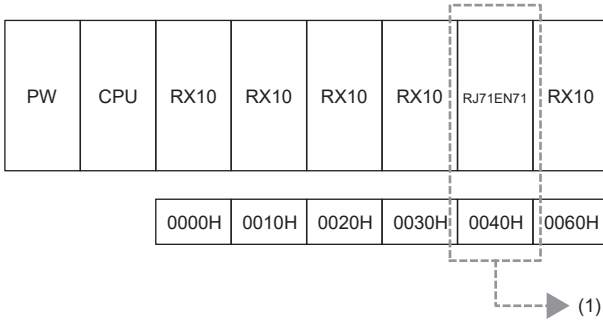
操作數	內容	範圍	資料類型	資料類型(標籤)
(H)	讀取模組型號的模組的起始輸入輸出編號(以16進制數4位表示時的高3位)	0H~FFH、3E0H~3E3H	無符號BIN16位元	ANY16
(d)	(d)+0: 指令執行結果, (d)+1~(d)+9: 模組型號	—	字元	ANY16_ARRAY (要素數: 10)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(H)	○	—	○	—	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 讀取 (H) 中指定的插槽的模組型號，並儲存到 (d) 中指定的元件及其以後。對象模組如下所示。
 - CPU 模組
 - 輸入模組
 - 輸出模組
 - 輸入輸出混合模組
 - 智能功能模組
- 在 (H) 中，以將讀取對象模組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。



(1): 讀取模組的起始輸入輸出編號指定K4或H4

- 指定佔用2插槽的模組的情況下，關於作為讀取對象指定的插槽，請參閱各模組的手冊中記載的輸入輸出佔用點數。
- CPU 模組的起始輸入輸出編號，按下表所示指定。

CPU 模組	起始輸入輸出編號
1號機	3E0H
2號機	3E1H
3號機	3E2H
4號機	3E3H

- (d)+0 中儲存指令執行結果、(d)+1~(d)+9 中儲存模組型號。(d) 中儲存的值如下所示。

條件	(d)+0	(d)+1~(d)+9
讀取對象模組保持有型號的情況下	0	模組保持的型號
讀取對象模組未保持型號的情況下	1	由模組類型及點數組合的字元串
讀取對象插槽為空餘插槽的情況下	-1	0000H
讀取對象模組處於在線更換中的情況下		
(H) 中指定的輸入輸出編號不是模組的起始編號的情況下		
由二重化擴展基板配置的待機系統指定了擴展基板上的模組的情況下		
讀取對象模組處於重新啟動中的情況下		

- 讀取對象模組保持有型號的情況下，(d)+1及其以後儲存的模組型號如下所示。
 - 使用9字元。
 - 以ASCII字元儲存。
 - 第18字元儲存00H。
 - 字元數不足17字元的情況下，剩餘的字元中將儲存00H。
 - 儲存模組保持的型號。(可能與額定銘牌上記載的型號不同，應加以注意)

例

儲存的型號的示例如下所示。

對象模組	儲存的型號的示例
CPU模組	R04CPU
輸入輸出模組	INPUT_16
網路模組	RJ71GP21-SX

要點

I/O分配設定中設定的型號與實際安裝的模組型號不相同的情況下，將儲存實際安裝的模組保持的型號。

- 讀取對象模組未保持型號的情況下，(d)+1及其以後儲存的字元串如下所示。
 - 使用9字元。
 - 以ASCII字元儲存。
 - 第18字元儲存00H。
 - 字元數不足17字元的情況下，剩餘的字元中將儲存00H。
 - 儲存由“表示模組類型的字元串”及“表示點數的字元串”所組合的字元串。

例

儲存的字元串的示例如下所示。

對象模組	儲存的字元串的示例
輸入模組	INPUT_16
輸出模組	OUTPUT_32
輸入輸出混合模組	MIXED_64
智能功能模組	INTELLIGENT_128

表示點數的字元串的示例如下所示。

點數	表示點數的字元串的示例
16點	_16
32點	_32
48點	_48
64點	_64
128點	_128
256點	_256
512點	_512
1024點	_1024

要點

I/O分配設定中設定的點數與實際安裝的模組的點數不相同的情況下，將儲存實際安裝的模組的點數。

- CPU模組為RnENCPU的情況下，指定輸入輸出編號且讀取模組型號時的儲存的字元串如下所示。

例

儲存的字元串的示例如下所示。

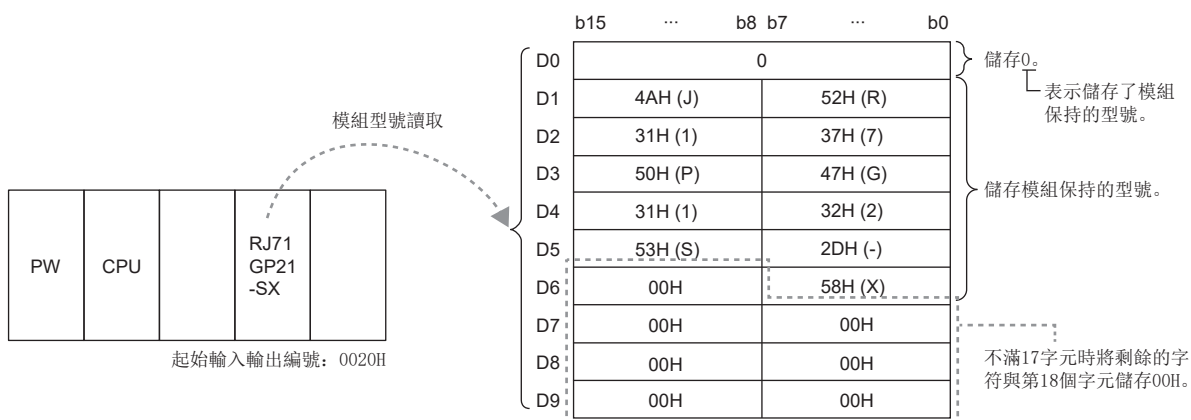
對象模組的輸入輸出編號	儲存的字元串的示例
3E00H	R120ENCPU
0000H	_RJ71EN71 (E+IEF)*1

*1 RnENCPU(網路部分)的儲存的字元串，將變為透過工程工具的模組配置圖的部件選擇視窗中所選擇的模組。

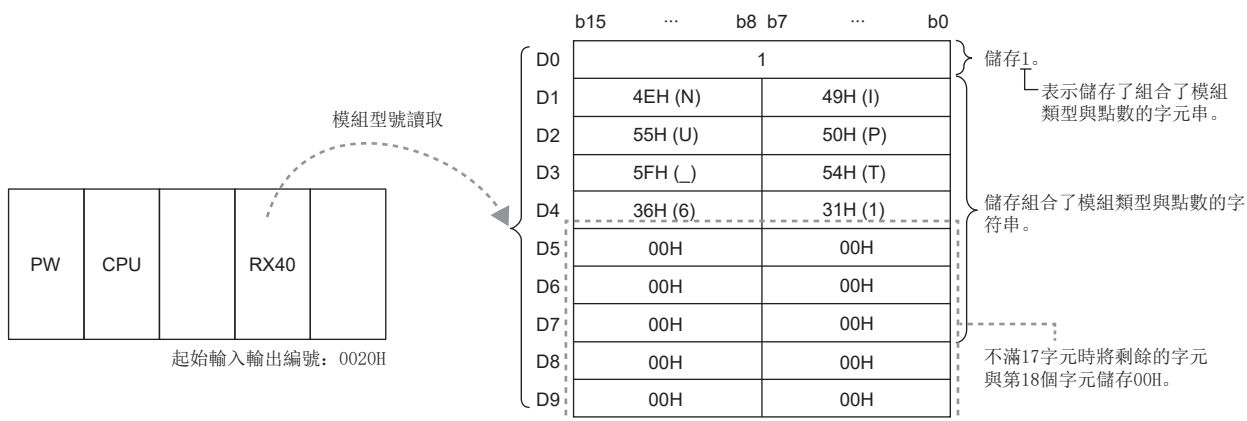
- 配置多CPU系統時，即使指定其它機號CPU模組管理的模組，也可讀取模組型號。
- 在下述程式示例中，M0變為0N時，將輸入輸出編號為0020H中安裝的模組的模組型號儲存到D0以後。



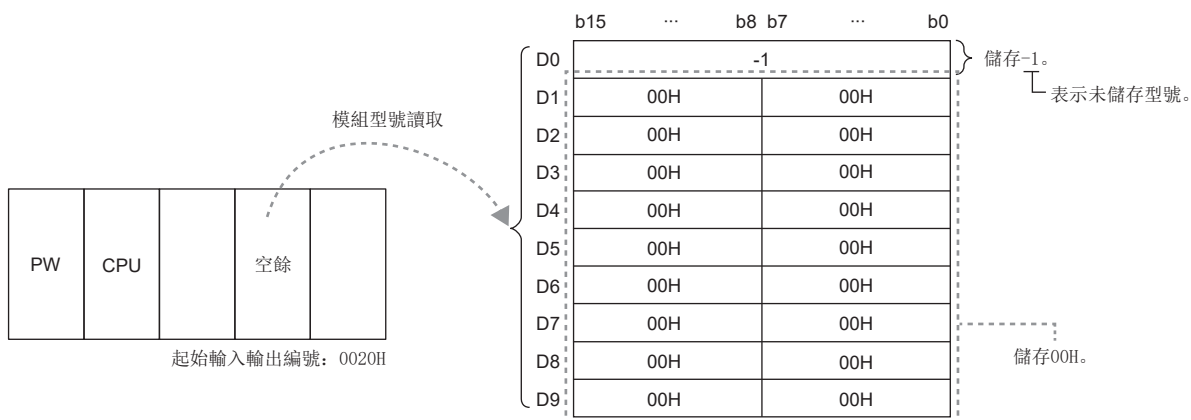
- 保持有型號的模組 (例: RJ71GP-SX)



- 未保持型號的模組 (例: RX40)



- 空餘插槽的情況下

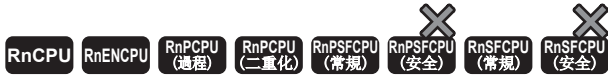


出錯

出錯代碼 (SD0)	內容
2800H	(H)的指定超出0~FFH、3E0~3E3H的範圍時。
2810H	由於讀取對象模組故障等原因無法通訊時。

模組特有資訊讀取

UNIINFRD (P)



讀取指定點數的模組資訊。

梯形圖	ST
	ENO:=UNIINFRD (EN, H, n, d) ; ENO:=UNIINFRDP (EN, H, n, d) ;

FBD/LD



■執行條件

指令	執行條件
UNIINFRD	
UNIINFRDP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(H)	讀取模組資訊的模組的起始輸入輸出編號(以16進制數4位表示時的高3位)	0H~FFH	無符號BIN16位元	ANY16
(d)	儲存模組資訊的起始元件	—	字元	ANY16*1
(n)	讀取資料點數	0~256	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

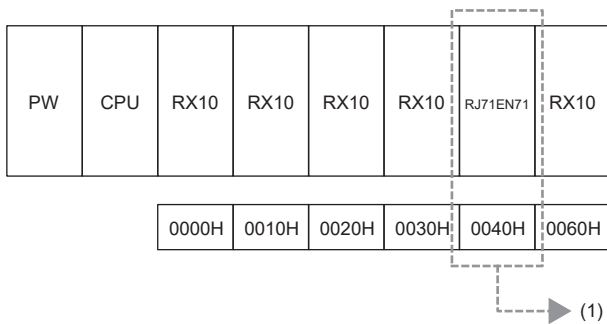
*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(H)	○	—	○	—	—	—	—	○	○	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	—	○	—	—	—	—	○	○	—	—	—	

功能

- 從(H)中指定的模組中，將(n)中指定的點數的模組資訊儲存到(d)中指定的元件及其以後。即使在I/O分配中更改模組類型及點數，也讀取實際安裝的模組狀態。
- 對於(H)，將讀取模組資訊的模組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。
- 未指定讀取模組的起始輸入輸出編號的情況下，將儲存僅模組安裝狀態為ON、模組安裝狀態以外為OFF的模組資訊。



(1): 讀取模組的起始輸入輸出編號指定K4或H4

- 模組資訊的詳細內容如下所示。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(d)																
(d)+1																

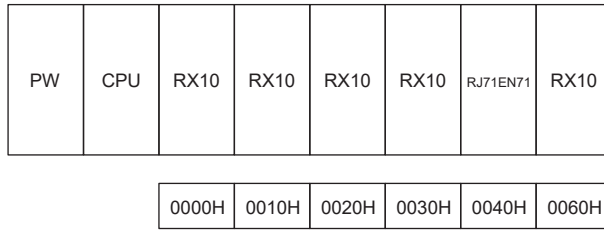
元件	位元	項目名	內容
(d)	b0	輸入輸出點數	0000: 16點
	b1		0001: 32點
	b2		0010: 48點
	b3		0011: 64點 0100: 128點 0101: 256點 0110: 512點 0111: 1024點
	b4	模組類型	0000: 輸入模組或未設定
	b5		0010: 輸出模組
	b6		0100: 輸入輸出混合模組(不同編號)
	b7		0110: 輸入輸出混合模組(同一編號) 1000: 智能功能模組*3
	b8	系列類別	010: MELSEC-Q系列的模組
	b9		011: MELSEC iQ-R系列的模組
	b10		111: 不明
	b11	組編號	000: 1號機管理
	b12		001: 2號機管理
	b13		010: 3號機管理 011: 4號機管理
	b14	插槽的起始判定	0: 不是插槽的起始 1: 是插槽的起始
b15	模組安裝狀態	0: 模組未安裝 1: 模組已安裝*2	
(d)+1	b0	出錯判定	00: 正常
	b1		01: 輕度異常 10: 中度異常 11: 重度異常
	b2	模組準備完成狀態	0: 準備未完成 1: 準備完成
	b3	空餘	0: 固定
	b4	空餘	0: 固定
	b5	空餘	0: 固定
	b6	模組之間同步狀態	00: 非同步對象
	b7		01: 同步準備中 10: 同步中 11: 同步出錯
	b8	空餘	0: 固定
	b9	空餘	0: 固定
	b10	外部供應電源狀態	0: 正常運行中或無外部供應電源 1: 發生電源斷開
	b11	保險絲熔斷發生有無	0: 正常運行中 1: 發生保險絲熔斷
	b12	空餘	0: 固定
	b13	在線模組更換狀態	0: 下述以外 1: 以下之一的情況下 • 在線模組更換中(模組選擇中除外) • 模組重新啟動中 • 嘗試由二重化系統的待機系統讀取擴展基板上的模組的資訊
	b14	空餘	0: 固定
b15	模組存取可否	0: 不能存取*1 1: 可以存取	

*1 適用於模組的拆裝，二重化擴展基板配置時的由待機系統至擴展基板上的模組的存取等

*2 佔用32點及其以上的模組的情況下，(d)+0、(d)+1中將儲存資訊，以後僅儲存模組安裝狀態ON的資訊。

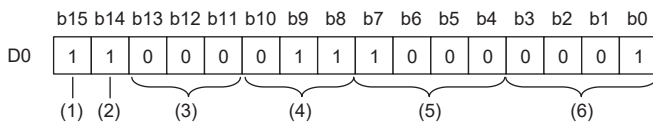
*3 對RnENCPU(網路部分)的輸入輸出編號進行了指定的情況下，將變為“1000: 智能功能模組”。

- 在下述程式示例中，M0為ON時，將輸入輸出編號為0040H以後的模組資訊儲存到D0以後。

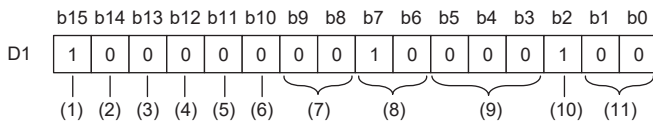


[讀取結果]

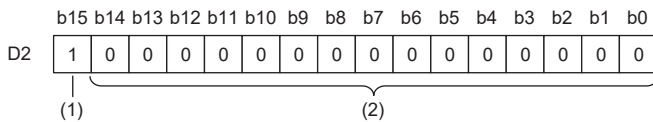
- 讀取RJ71EN71的模組資訊。



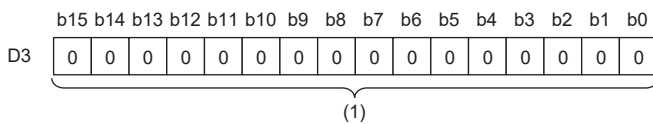
- (1) 模組已安裝
- (2) 是插槽的起始
- (3) 1號機管理
- (4) MELSEC iQ-R系列的模組
- (5) 智能功能模組
- (6) 32點模組



- (1) 可以存取模組
- (2) 固定為0
- (3) 不處於在線模組更換中，不是由待機系統執行
- (4) 固定為0
- (5) 無保險絲熔斷出錯
- (6) 固定為0
- (7) 固定為0
- (8) 模組之間同步中
- (9) 固定為0
- (10) 模組準備完成
- (11) 正常(無模組出錯)

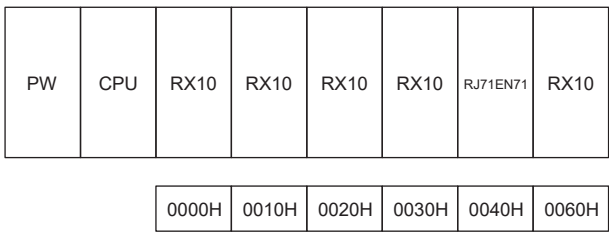


- (1) 作為32點模組的後半16點連接了模組。
- (2) D0、D1中可儲存資訊，因此全部變為0。

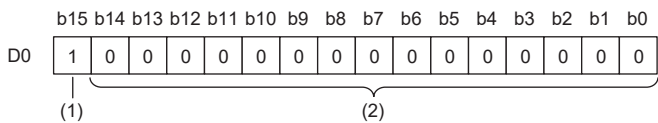


- (1) D0、D1中可儲存資訊，因此全部變為0。

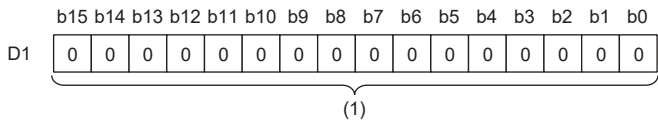
- 對於輸入輸出點數為32點及其以上的模組，在(H)中指定了起始輸入輸出編號以外的輸入輸出編號的情況下，僅將模組資訊的安裝狀態儲存為ON，模組安裝狀態以外則儲存OFF的模組資訊。在下述程式示例中，M0為ON時，將輸入輸出編號為0050H以後的模組資訊儲存到D0以後。



[讀取結果]
 • 由於在(H)中指定了RJ71EN71的後半16點的輸入輸出編號，因此僅模組安裝狀態ON。

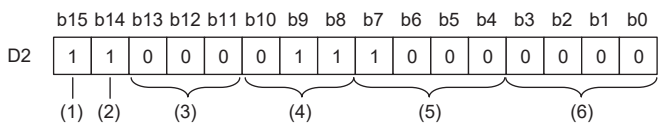


- (1) 作為32點模組的後半16點連接了模組。
- (2) 將32點模組的後半16點透過n1進行了指定，因此全部變為0。

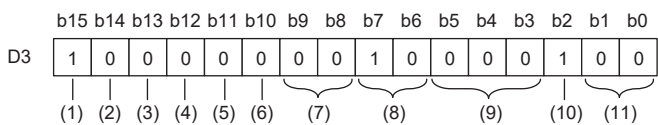


- (1) 將32點模組的後半16點透過n1進行了指定，因此全部變為0。

• 讀取RX10的模組資訊。



- (1) 模組已安裝
- (2) 是插槽的起始
- (3) 1號機管理
- (4) MELSEC iQ-R系列的模組
- (5) 輸入模組
- (6) 16點模組



- (1) 可以存取模組
- (2) 固定為0
- (3) 不處於在線模組更換中，不是由待機系統執行
- (4) 固定為0
- (5) 無保險絲熔斷出錯
- (6) 固定為0
- (7) 固定為0
- (8) 模組之間同步中
- (9) 固定為0
- (10) 模組準備完成
- (11) 正常(無模組出錯)

出錯

出錯代碼(SD0)	內容
2800H	(H) 為0~FFH以外時。
3405H	(n) 為0~256以外時。
	(H) 與 (n) 的合計為257及其以上時。

21 參數設定操作

21.1 路由資訊指令

路由資訊的讀取

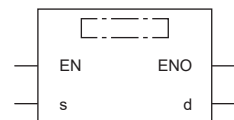
S(P). RTREAD



從參數中設定的路由資訊中讀取指定的傳送目標網路No. 的資料。

梯形圖	ST
	ENO:=S_RTREAD (EN, s, d) ; ENO:=SP_RTREAD (EN, s, d) ;

FBD/LD



■執行條件

指令	執行條件
S. RTREAD	
SP. RTREAD	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	傳送目標網路No.	1~239	帶符號BIN16位元	ANY16
(d)	儲存讀取的資料的起始元件	—	字元	ANY16_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	—	○	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—

功能

- 從參數中設定的路由資訊中讀取 (s) 中指定的傳送目標網路No. 的資料，並儲存到 (d) 及其以後。
- (s) 中指定的傳送目標網路No. 的資料未在參數中設定的情況下，(d) 及其以後中將儲存0。
- (d) 及其以後儲存的資料的內容如下所示。

起始元件	項目	範圍
(d)	中繼站目標網路No.	1~239
(d)+1	中繼站目標站號	參閱下表
(d)+2	虛擬	—

- (d)+1的指定範圍如下所示。

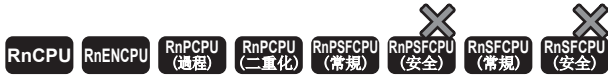
網路類別	指定範圍
MELSECNET/H	1~64
CC-Link IE控制網路	1~120
CC-Link IE現場網路(主站)	固定為125
CC-Link IE現場網路(本地站)	1~120
CC-Link IE TSN(主站)	固定為125
CC-Link IE TSN(本地站)	1~120

出錯

出錯代碼(SD0)	內容
3405H	(s) 為1~239以外時。

路由資訊的登錄

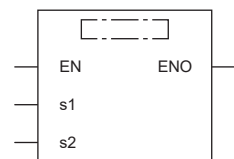
S(P). RTWRITE



將路由資訊登錄到指定的傳送目標網路No. 的區域中。

梯形圖	ST
	ENO:=S_RTWRITE (EN, s1, s2) ; ENO:=SP_RTWRITE (EN, s1, s2) ;

FBD/LD



■執行條件

指令	執行條件
S. RTWRITE	
SP. RTWRITE	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s1)	傳送目標網路No.	1~239	帶符號BIN16位元	ANY16
(s2)	儲存寫入資料的起始元件	—	字元	ANY16_ARRAY (要素數: 3)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	○	—	○	—	—	—	○	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—	—

功能

- 將 (s2) 及其以後的路由資料登錄到參數的 (s1) 中指定的傳送目標網路No. 的區域中。
- (s2) 及其以後儲存的資料的內容如下所示。

起始元件	項目	範圍
(s2)	中繼站目標網路No.	1~239
(s2)+1	中繼站目標站號	參閱下表
(s2)+2	虛擬	—

- (s2)+1的指定範圍如下所示。

網路類別	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)	固定為125
CC-Link IE現場網路(本地站)	1~120
CC-Link IE TSN(主站)	固定為125
CC-Link IE TSN(本地站)	1~120

- (s1) 中指定的傳送目標網路No. 的資料未在參數中設定的情況下，更改為 (s2) 及其以後的資料。
- (s2) 及其以後的資料 ((s2) ~ (s2)+2) 全部為0的情況下，將 (s1) 中指定的傳送目標網路No. 的資料從參數中刪除。

出錯

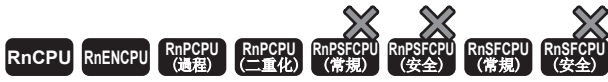
出錯代碼 (SD0)	內容
3405H	(s1) 為1~239以外時。
	(s2) 及其以後的資料超出以下範圍時。 <ul style="list-style-type: none">(s2): 1~239(s2)+1: 1~120、125
	網路模組的參數中登錄的路由資訊的登錄數與RTWRITE指令中登錄的路由資訊的登錄數的合計超過了最大登錄數238時。
	指定刪除未登錄到參數中的傳送目標網路No. 時。
	(s2)、(s2)+1中之一指定0的情況下。

22 CPU模組資料記錄功能

22.1 記錄用指令

觸發記錄設定

LOGTRG



• ROOCPU中不能使用。

在觸發記錄中，對指定的記錄設定No. 發生了觸發條件。

梯形圖	ST
	<pre>ENO:=LOGTRG (EN, s);</pre>

FBD/LD

■執行條件

指令	執行條件
LOGTRG	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	記錄設定No.	1~10	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	○	—	—	○	—	—	—

功能

- 在(s)中指定的記錄設定No. 的觸發記錄中發生觸發。
- (s)中可設定1~10。
- 執行LOGTRG指令時，(s)中指定的記錄設定No. 的特殊繼電器(記錄觸發)為ON，執行了觸發後記錄數中設定的記錄數的觸發記錄後，進行資料鎖存並停止觸發記錄。
- 觸發條件中選擇了“執行觸發指令時”時將生效。
- 下述情況下，即使執行LOGTRG指令也將變為無處理。
 - 觸發條件中指定了選擇為“執行觸發指令時”以外的記錄設定No. 的情況下。
 - 指定了未設定的記錄設定No. 的情況下。
 - 指定了正在執行連續記錄的記錄設定No. 的情況下。
 - 執行LOGTRG指令後，在未執行LOGTRGR指令的狀況下，再次執行了LOGTRG指令的情況下。

注意事項

中斷程式中使用LOGTRG時，觸發條件有可能不發生。

出錯

出錯代碼(SD0)	內容
3405H	(s)為1~10以外時。

觸發記錄重設

LOGTRGR

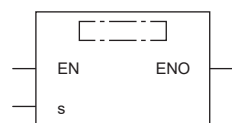


• R00CPU中不能使用。

對指定的記錄設定No. 的觸發條件進行重設。

梯形圖	ST
	ENO:=LOGTRGR(EN, s);

FBD/LD



■執行條件

指令	執行條件
LOGTRGR	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	記錄設定No.	1~10	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\□(H)G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	—	○	—	○	—	—	○	—	—	—

功能

- 是對(s)中指定的記錄設定No. 的LOGTRG指令進行重設的指令。執行LOGTRGR指令時，指定的觸發記錄設定No. 的LOGTRG指令將變為無效。
- 執行LOGTRGR指令時，(s)中指定的記錄設定No. 的特殊繼電器(記錄完成、記錄觸發、記錄觸發後)將變為OFF。
- 將緩衝內的資料儲存到SD記憶卡中時即使執行指令，指令的處理必須在所有資料的儲存完成之後始能處理。

注意事項

中斷程式中使用LOGTRGR時，觸發條件有可能不發生。

出錯

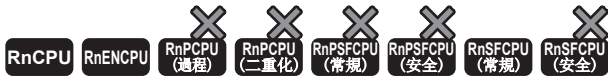
出錯代碼(SD0)	內容
3405H	(s)為1~10以外時。

23 記錄功能

23.1 資料採集指令

資料採集觸發設定

DATATRG



- R00CPU、R01CPU、R02CPU中不能使用。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“50”及其以後的版本使用。工程工具可以在版本為“1.065T”及其以後的版本使用。此外，在CPU模組的製造資訊中也有限制。關於對應的製造資訊，請參閱MELSEC iQ-R CPU模組用戶手冊(應用篇)。
- RnSFCPU(常規)可以在韌體版本為“23”及其以後的版本使用。工程工具可以在版本為“1.070Y”及其以後的版本使用。

在記錄功能中對於指定的設定No. 進行資料採集。

梯形圖	ST
	ENO:=DATEWRP (EN, s) ;

FBD/LD

■執行條件

指令	執行條件
DATATRG	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	設定No.	1~4	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(s)	—	—	○	—	○	—	—	○	—	—	—	—

功能

- 在記錄功能中，進行從(s)中指定的設定No. 的資料採集。
- (s)中可設定1~4。
- 選擇記錄設定的採集間隔設定中的“觸發指令”時將啟用。
- 下述情況下，即使執行DATATRIG指令也將變為無處理。
 - 記錄功能指定沒有設定的設定No. 時
 - 指定了記錄設定中採集間隔設定的“觸發指令”以外的設定No. 時
 - 對於同一設定No.，在指令執行中執行了DATATRIG指令時
 - 指定的設定No. 的記錄開始觸發在OFF狀態下執行時
 - 指定的設定No. 的記錄功能在準備中或者在儲存中的狀態下執行時

出錯

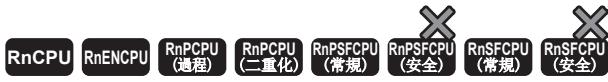
出錯代碼 (SDO)	內容
3405H	(s) 為1~4以外時。

24 內建乙太網路功能用指令

24.1 打開/關閉處理指令

連接的建立

SP.SOCOPEN



進行指定連接的打開處理。

梯形圖	ST
	ENO:=SP_SOCOPEN(EN, U, s1, s2, d);

FBD/LD

■執行條件

指令	執行條件
SP.SOCOPEN	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	連接編號	1~16	帶符號BIN16位元	ANY16
(s2)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 10)
(d)	透過指令完成讓1個掃描變為ON的元件 異常完成時(d)+1也將變為ON。	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H)G□	Z	LT、 LST、LC	LZ		K、H	E	\$		
(U)	—	—	○	—	—	—	—	○	—	—	○	○	
(s1)	—	—	○	—	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	○	—	○	—	—	—	—	○	—	—	—	—	

■控制資料

操作數：(s2)														
元件	項目	內容	設定範圍	設定方										
+0	執行/完成類型	指定連接的打開處理時，是使用透過工程工具進行的參數設定值，或是使用控制資料的(s2)+2～(s2)+9的設定值。 <ul style="list-style-type: none"> • 0000H：以工程工具的“打開設定”中設定的內容進行打開處理。 • 8000H：以控制資料的(s2)+2～(s2)+9中指定的內容進行打開處理。 	0000H 8000H	用戶										
+1	完成狀態	指令完成時儲存完成狀態。 <ul style="list-style-type: none"> • 0000H：正常完成 • 0000H以外：異常完成(出錯代碼) 	—	系統										
+2	使用用途設定區域	指定連接的使用用途。 <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">b15 b14</td> <td style="text-align: center;">...</td> <td style="text-align: center;">b10 b9 b8</td> <td style="text-align: center;">...</td> <td style="text-align: center;">b0</td> </tr> <tr> <td style="text-align: center;">(s2)+2</td> <td style="text-align: center;">(3)</td> <td style="text-align: center;">0</td> <td style="text-align: center;">(4)(2)(1)</td> <td style="text-align: center;">0</td> </tr> </table> </div> <p>(1) 通訊方式(協定)(位元8) <ul style="list-style-type: none"> • 0: TCP/IP • 1: UDP/IP (2) 套接字通訊功能的順序有無(位元9) <ul style="list-style-type: none"> • 1: 無順序(固定) (3) 打開方式(位元15、14) <ul style="list-style-type: none"> • 00: Active打開或UDP/IP • 10: Unpassive打開 • 11: Fullpassive打開 (4) 通訊協定設定(位元10) <ul style="list-style-type: none"> • 0: 不使用通訊協定支援功能(使用套接字通訊功能) • 1: 使用通訊協定支援功能 </p>	b15 b14	...	b10 b9 b8	...	b0	(s2)+2	(3)	0	(4)(2)(1)	0	—	用戶
b15 b14	...	b10 b9 b8	...	b0										
(s2)+2	(3)	0	(4)(2)(1)	0										
+3	本站埠編號	指定本站的埠編號。	0001H～1387H、 1392H～FFFEH (建議為0400H及其以後)	用戶										
+4 +5	對象設備IP地址*1	指定對象設備的IP地址。	00000001H～ FFFFFFFFH (FFFFFFFFH：廣播輪詢通訊)	用戶										
+6	對象設備埠編號	指定對象設備的埠編號。	0001H～FFFFH (FFFFH：廣播輪詢通訊)	用戶										
+7～+9	—	禁止使用	—	系統										

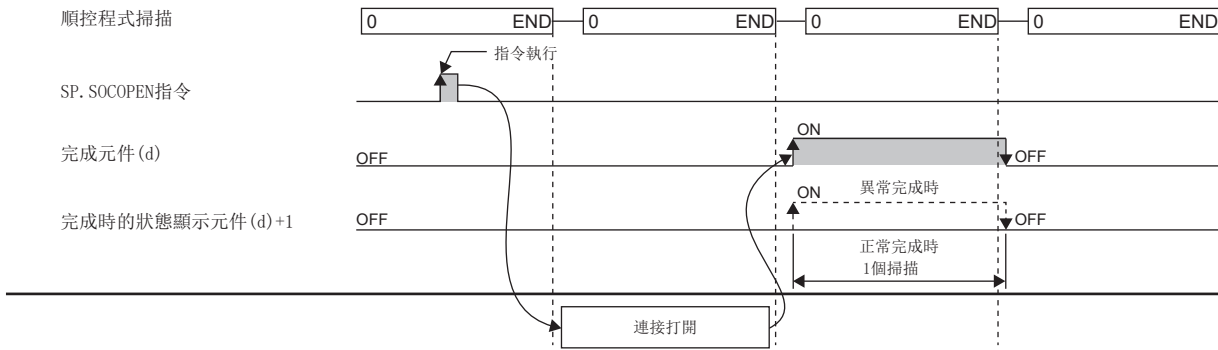
*1 Unpassive打開時對象設備IP地址、對象設備埠編號將被忽略。

要點

因為埠編號0001H～03FFH一般是保留的埠編號(WELL KNOWN PORT NUMBERS)、F000H～FFFEH是其它功能中動態使用的編號，因此建議使用本站埠編號0400H～1387H、1392H～EFFFH。

功能

- 進行 (s1) 中指定的連接的打開處理。打開處理中使用的設定值是在 (s2)+0 中選擇。
 - 對於 SP. SOCOPEN 指令執行及正常/異常完成，可以透過設定資料中指定的完成元件 (d)、完成時的狀態顯示元件 (d)+1 來確認。
 - 完成元件 (d)
- 在 SP. SOCOPEN 指令完成的掃描的 END 處理中 ON、在下一個 END 處理中 OFF。
- 完成時的狀態顯示元件 (d)+1
- 根據 SP. SOCOPEN 指令完成時的狀態而 ON/OFF。
- 正常完成時：保持為 OFF 不變。
- 異常完成時：在 SP. SOCOPEN 指令完成的掃描的 END 處理中 ON、在下一個 END 處理中 OFF。
- SP. SOCOPEN 指令的執行時機如下所示。



- 可以打開參數中未設定的連接(協定欄處於空欄狀態的連接)使用。在此情況下應將 (s2)+0 設定為 8000H，在控制資料的 (s2)+1 ~ (s2)+9 中指定打開的內容。

出錯

出錯代碼 (SD0)	內容
3405H	(s1) 中指定的連接編號超出 1~16 的範圍時。

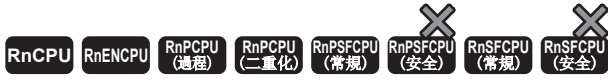
異常完成時，完成時的狀態顯示元件 (d)+1 將變為 ON、完成狀態 (s2)+1 中將儲存出錯代碼。

關於完成狀態 (s2)+1 中儲存的出錯代碼，請參閱以下手冊。

📖 MELSEC iQ-R 乙太網路用戶手冊(應用篇)

連接的斷開

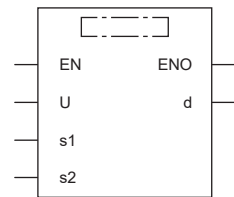
SP. SOCCLOSE



對指定的連接進行關閉處理。

梯形圖	ST
	ENO:=SP_SOCCLOSE(EN, U, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
SP. SOCCLOSE	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	連接編號	1~16	帶符號BIN16位元	ANY16
(s2)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 2)
(d)	透過指令完成讓1個掃描變為ON的元件 異常完成時(d)+1也將變為ON。	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

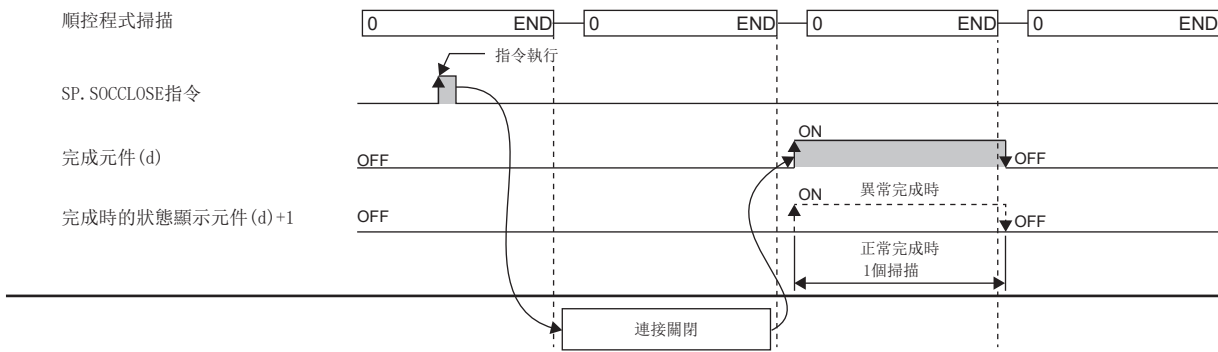
操作數	位元		字元		雙字元		間接指定	常數				其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H)G□	Z	LT、 LST、LC		LZ	K	H	E	
(U)	—	—	○	—	—	—	○	—	—	○	○	
(s1)	—	—	○	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	
(d)	○	—	○	—	—	—	○	—	—	—	—	

■控制資料

操作數: (s2)				
元件	項目	內容	設定範圍	設定方
+0	系統區域	—	—	—
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H: 正常完成 • 0000H以外: 異常完成(出錯代碼)	—	系統

功能

- 對(s1)中指定的連接進行關閉處理(連接的斷開)。打開處理中使用的設定值是在(s2)+0中選擇。
 - 對於SP.SOCCLOSE指令執行及正常/異常完成,可以透過設定資料中指定的完成元件(d)、完成時的狀態顯示元件(d)+1來確認。
 - 完成元件(d)
- 在SP.SOCCLOSE指令完成的掃描的END處理中ON、在下一個END處理中OFF。
- 完成時的狀態顯示元件(d)+1
- 根據SP.SOCCLOSE指令完成時的狀態而ON/OFF。
- 正常完成時:保持為OFF不變。
- 異常完成時:在SP.SOCCLOSE指令完成的掃描的END處理中ON、在下一個END處理中OFF。
- SP.SOCCLOSE指令的執行時機如下所示。



出錯

出錯代碼(SD0)	內容
3405H	(s1)中指定的連接編號超出1~16的範圍時。

異常完成時,完成時的狀態顯示元件(d)+1將變為ON、完成狀態(s2)+1中將儲存出錯代碼。

關於完成狀態(s2)+1中儲存的出錯代碼,請參閱以下手冊。

📖 MELSEC iQ-R 乙太網路用戶手冊(應用篇)

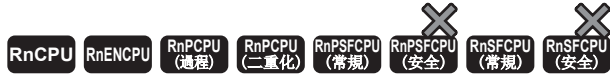
要點

在TCP Passive方式中,將SP.SOCCLOSE指令指定了打開等待狀態的連接的情況下,SP.SOCOPEN指令、SP.SOCCLOSE指令均將正常完成,連接將關閉。

24.2 套接字通訊用指令

接收資料的END處理時讀取

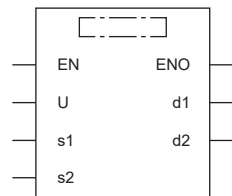
SP.SOCRCV



在執行指令後的END處理中，將指定的連接的接收資料從套接字通訊接收資料區域中讀取。

梯形圖	ST
	ENO:=SP_SOCRCV (EN, U, s1, s2, d1, d2) ;

FBD/LD



■執行條件

指令	執行條件
SP.SOCRCV	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	連接編號	1~16	帶符號BIN16位元	ANY16
(s2)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 2)
(d1)	儲存接收資料的起始元件	—	字元	ANY16*1
(d2)	透過指令完成讓1個掃描變為ON的元件 異常完成時(d2)+1也將ON。	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H) G□	Z	LT、 LST、LC	LZ		K、H	E	\$		
(U)	—	—	○	—	—	—	—	○	—	—	○	○	
(s1)	—	—	○	—	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d2)	○	—	○	—	—	—	—	○	—	—	—	—	

■控制資料

操作數：(s2)				
元件	項目	內容	設定範圍	設定方
+0	系統區域	—	—	—
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H：正常完成 • 0000H以外：異常完成(出錯代碼)	—	系統

操作數：(d1)				
元件	項目	內容	設定範圍	設定方
+0	接收資料長度	儲存從套接字通訊接收資料區域中讀取的資料的資料長度。 (字節數)	0~10238	系統
+1~+□	接收資料	從套接字通訊接收資料區域中讀取的資料從小編號地址開始依次被儲存。 ^{*1}	—	系統

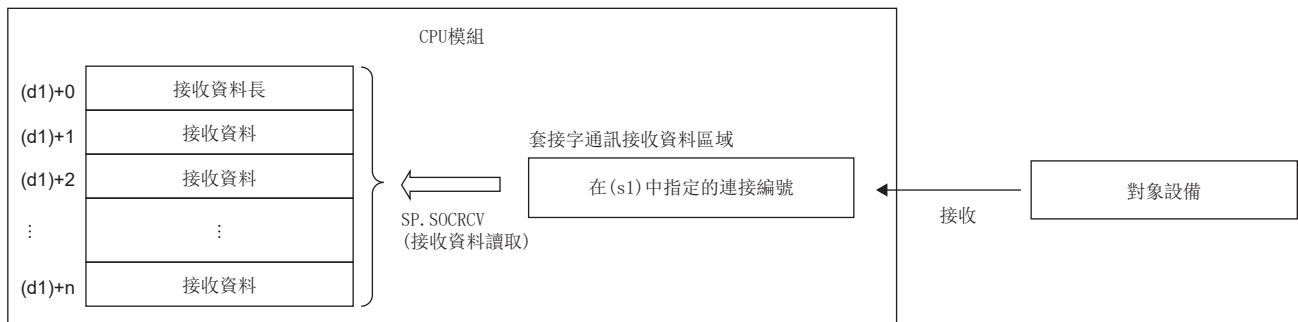
*1 接收的資料以字節單位從低位字節側開始依次被儲存。接收了奇數字節的資料時，最終資料儲存區域的低位字節中最後的接收資料將被儲存。

要點

- 執行SP.SOCRCV指令時，從接收資料的套接字通訊接收資料區域中的讀取是透過END處理進行。因此，執行SP.SOCRCV指令時掃描時間將延長。
- 接收了奇數字節資料的情況下，儲存了最後接收資料的元件的高位字節中將放入無效的資料。

功能

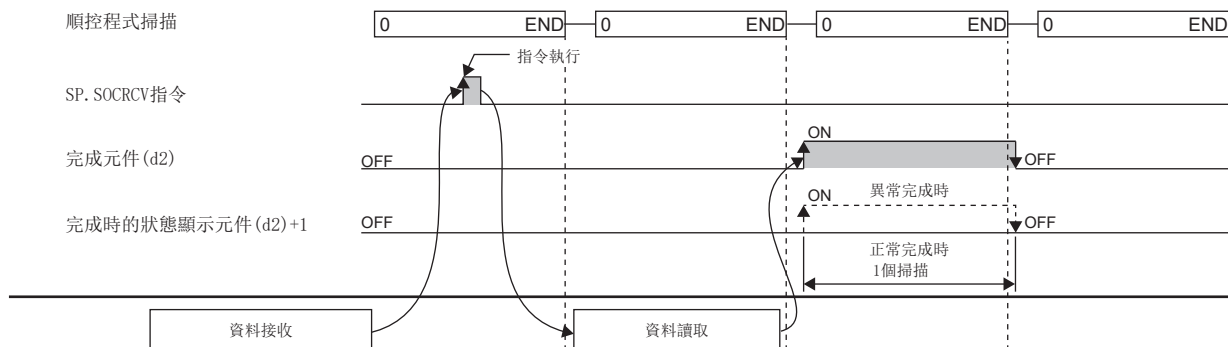
- 執行SP.SOCRCV指令後的END處理中，將(s1)中指定連接的接收資料從套接字通訊接收資料區域(儲存從各連接對象設備接收的資料的區域)中讀取。



- 對於SP.SOCRCV指令執行及正常/異常完成，可透過設定資料中指定的完成元件(d2)、完成時的狀態顯示元件(d2)+1進行確認。

- 完成元件(d2)
- 在SP.SOCRCV指令完成的掃描的END處理中ON、在下一個END處理中OFF。
- 完成時的狀態顯示元件(d2)+1
- 根據SP.SOCRCV指令完成時的狀態而ON/OFF。
 正常完成時：保持為OFF不變。
 異常完成時：在SP.SOCRCV指令完成的掃描的END處理中ON、在下一個END處理中OFF。

- SP.SOCRCV指令的執行時機如下所示。



注意事項

對同一連接進行接收資料的讀取的情況下，應勿同時使用S. SOCRCVS指令。

出錯

出錯代碼 (SD0)	內容
2820H	接收的資料超過了接收資料儲存目標元件的元件/標籤記憶體各設定區域時。
3405H	(s1)中指定的連接編號超出1~16的範圍時。

異常完成時，完成時的狀態顯示元件 (d2)+1將變為0N、完成狀態 (s2)+1中將儲存出錯代碼。

關於完成狀態 (s2)+1中儲存的出錯代碼，請參閱以下手冊。

📖 MELSEC iQ-R 乙太網路用戶手冊 (應用篇)

要點

- 為了防止接收過大的資料，透過SP. SOCRCV指令設定接收資料容量，可以抑制接收的資料量。
- 透過常閉觸點使SP. SOCRCV指令的完成元件連接到執行指示，即使連續接收資料的情況下也可連續進行讀取。

執行接收資料的指令時讀取

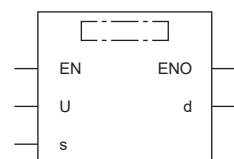
S. SOCRCVS



將指定連接的接收資料從套接字通訊接收資料區域中讀取。

梯形圖	ST
	ENO:=S_SOCRCVS (EN, U, s, d);

FBD/LD



■執行條件

指令	執行條件
S. SOCRCVS	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s)	連接編號	1~16	無符號BIN16位元	ANY16
(d)	儲存接收資料的起始元件	—	字元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其它 (U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(U)	—	—	○	—	—	—	—	○	—	—	○	○
(s)	—	—	○	—	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

■控制資料

操作數: (d)				
元件	項目	內容	設定範圍	設定方
+0	接收資料長度	儲存從套接字通訊接收資料區域中讀取的資料的資料長度。(字節數)	0~10238	系統
+1~+□	接收資料	從套接字通訊接收資料區域中讀取的資料從小編號地址開始依次被儲存。*1	—	系統

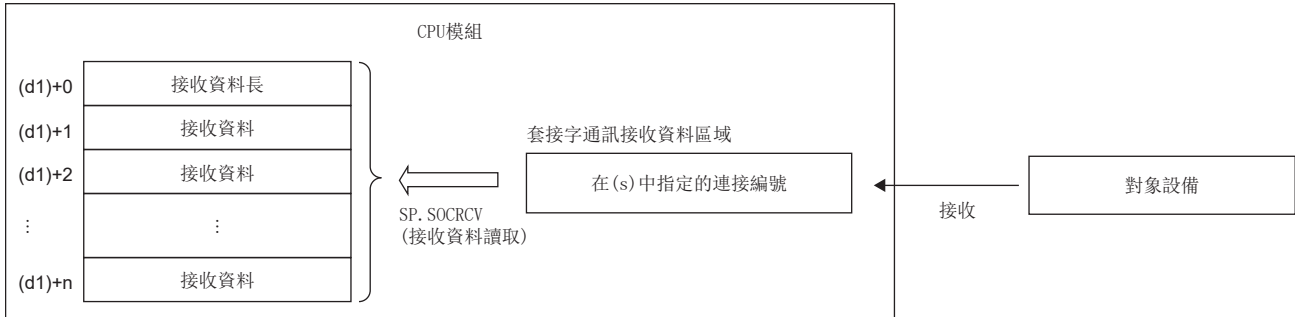
*1 接收的資料以字節單位從低位字節側開始依次被儲存。接收了奇數字節的資料時，最終資料儲存區域的低位字節中最後的接收資料將被儲存。

要點

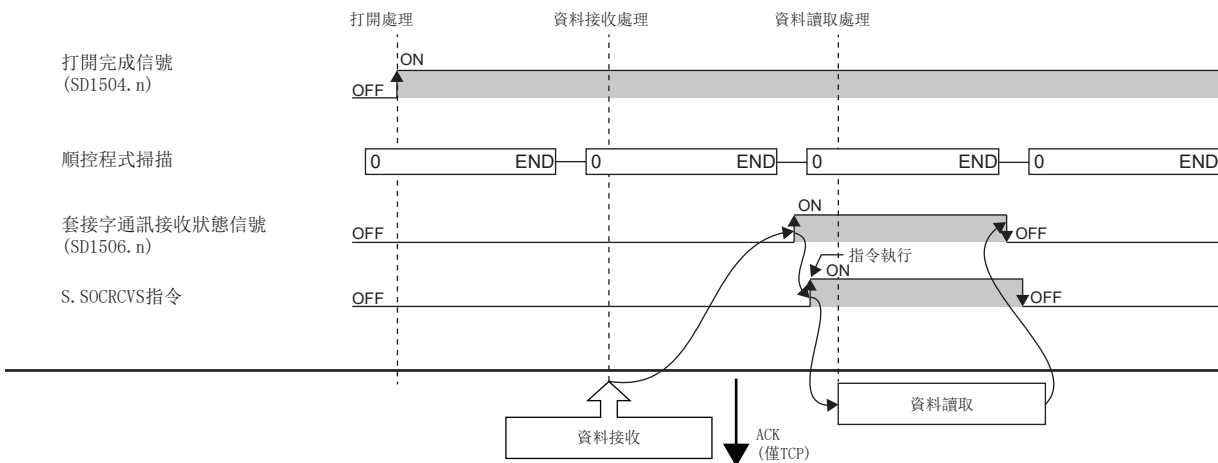
- 接收資料容量的預設認值為2046字節。接收不小於2047字節的資料的情況下，應透過SP. SOCRMODE指令更改接收資料容量。
- 接收了奇數字節資料的情況下，儲存了最後接收資料的元件的高位字節中將放入無效的資料。

功能

- 將(s)中指定連接的接收資料從套接字通訊接收資料區域(儲存從各連接的對象設備接收的資料的區域)中讀取。



- 使用了S. SOCRCVS指令的接收處理的時機如下所示。



注意事項

對同一連接進行接收資料讀取的情況下，應勿同時使用SP. SOCRCV指令。

出錯

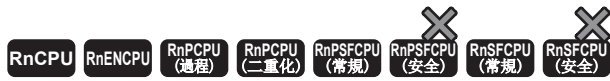
出錯代碼 (SD0)	內容
2820H	接收的資料超過了接收資料儲存目標元件的元件/標籤記憶體各設定區域時。
3405H	(s)中指定的連接編號超出1~16的範圍時。

要點

為了防止接收過大的資料，透過SP. SOCRMODE指令設定接收資料容量，可以抑制接收的資料量。

資料發送

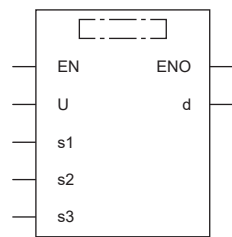
SP. SOCSND



將資料發送到指定連接的對象設備。

梯形圖	ST
	ENO:=SP_SOCSND (EN, U, s1, s2, s3, d);

FBD/LD



執行條件

指令	執行條件
SP. SOCSND	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	連接編號	1~16	帶符號BIN16位元	ANY16
(s2)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 2)
(s3)	儲存發送資料的起始元件	—	字元	ANY16*1
(d)	透過指令完成讓1個掃描變為ON的元件 異常完成時(d)+1也將變為ON。	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

操作數	位元		字元				雙字元		間接指定	常數			其它(U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ	K、H		E	\$		
(U)	—	—	○	—	—	—	—	○	—	—	○	○	
(s1)	—	—	○	—	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(s3)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	○	—	○	—	—	—	—	○	—	—	—	—	

■控制資料

操作數：(s2)				
元件	項目	內容	設定範圍	設定方
+0	系統區域	—	—	—
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H：正常完成 • 0000H以外：異常完成(出錯代碼)	—	系統

操作數：(s3)				
元件	項目	內容	設定範圍	設定方
+0	發送資料長	指定發送資料長。(字節數)	1~10238	用戶
+1~+□	發送資料	指定發送資料。*1	—	用戶

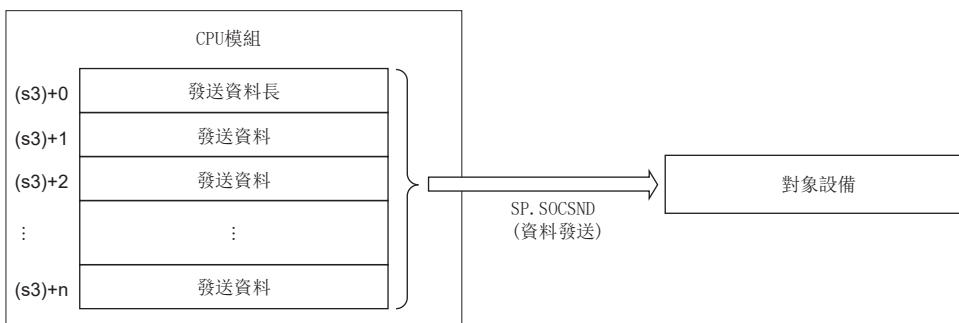
*1 發送資料以字節單位按照從低位字節側開始的順序被發送。接收了奇數字節的資料時，最終資料的儲存區域的低位字節中將儲存最後的發送資料。

要點

TCP的情況下，發送資料長不應超過對象設備的最大視窗尺寸(TCP的接收緩衝)。不能發送超出對象設備的最大視窗尺寸的資料。

功能

- 將(s3)中設定的資料發送到(s1)中指定連接的對象設備。



- 對於SP. SOCSND指令執行及正常/異常完成，可以透過設定資料中指定的完成元件(d)、完成時的狀態顯示元件(d)+1進行確認。

- 完成元件(d)

在SP. SOCSND指令完成的掃描的END處理中ON、在下一個END處理中OFF。

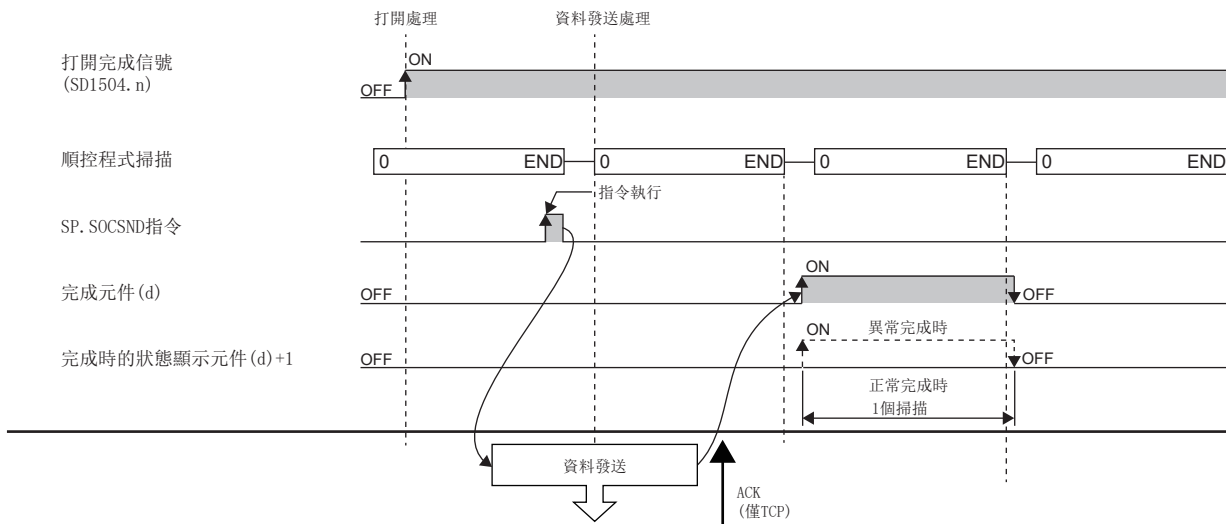
- 完成時的狀態顯示元件(d)+1

根據SP. SOCSND指令完成時的狀態而ON/OFF。

正常完成時：保持為OFF不變。

異常完成時：在SP. SOCSND指令完成的掃描的END處理中ON、在下一個END處理中OFF。

- 使用了SP. SOCSND指令的接收處理的時機如下所示。



完成元件ON後，發送也可能繼續進行。需要確認發送是否完成的情況下，應在接收側確認完成。

出錯

出錯代碼 (SD0)	內容
3405H	(s1) 中指定的連接編號超出1~16的範圍時。

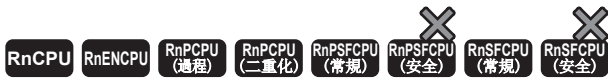
異常完成時，完成時的狀態顯示元件 (d)+1將變為ON、完成狀態 (s2)+1中將儲存出錯代碼。

關於完成狀態 (s2)+1中儲存的出錯代碼，請參閱以下手冊。

📖 MELSEC iQ-R 乙太網路用戶手冊 (應用篇)

連接資訊的讀取

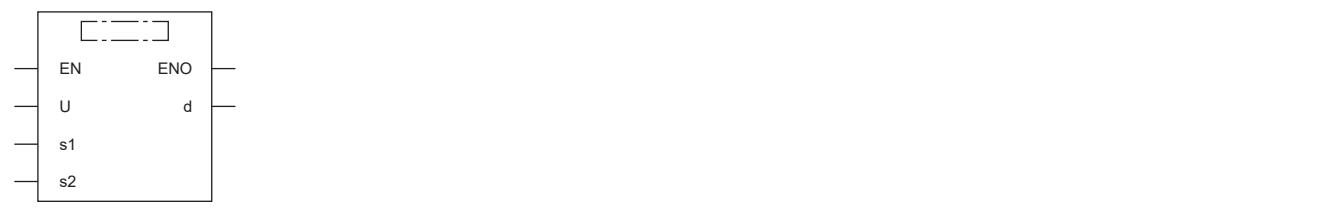
SP. SOCCINF



讀取 (s1) 中指定連接的連接資訊，儲存到 (d) 中指定的元件及其以後。

梯形圖	ST
<p>The diagram shows a single coil instruction box with inputs (U), (s1), (s2) and output (d).</p>	ENO:=SP_SOCCINF (EN, U, s1, s2, d);

FBD/LD



■執行條件

指令	執行條件
SP. SOCCINF	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	連接編號	1~16	無符號BIN16位元	ANY16
(s2)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 2)
(d)	儲存連接資訊的起始元件	—	字元	ANY16_ARRAY (要素數: 5)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其它 (U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(U)	—	—	○	—	—	—	—	○	—	—	○	○	
(s1)	—	—	○	—	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	

■控制資料

操作數：(s2)				
元件	項目	內容	設定範圍	設定方
+0	系統區域	—	—	—
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H：正常完成 • 0000H以外：異常完成(出錯代碼)	—	系統

操作數：(d)																				
元件	項目	內容	設定範圍	設定方																
+0 +1	對象設備IP地址	儲存對象設備的IP地址。	00000001H~ FFFFFFFH 00000000H：無通訊對象 (FFFFFFFH：廣播輪詢通訊)	系統																
+2	對象設備埠編號	儲存對象設備的埠編號。	0001H~ FFFFH (FFFFH：廣播輪詢通訊)	系統																
+3	本站埠編號	儲存本站埠編號。	0001H~1387H 1392H~FFFEH	系統																
+4	連接使用用途	儲存連接的使用用途。 <table border="1" style="margin-left: 20px;"> <tr> <td style="text-align: right;">b15</td> <td style="text-align: right;">b14</td> <td style="text-align: center;">...</td> <td style="text-align: right;">b10</td> <td style="text-align: right;">b9</td> <td style="text-align: right;">b8</td> <td style="text-align: center;">...</td> <td style="text-align: right;">b0</td> </tr> <tr> <td style="text-align: center;">(d)+4</td> <td style="text-align: center;">(3)</td> <td style="text-align: center;">0</td> <td style="text-align: center;">(4)</td> <td style="text-align: center;">(2)</td> <td style="text-align: center;">(1)</td> <td style="text-align: center;">0</td> <td></td> </tr> </table> (1) 通訊方式(協定)(位元8) • 0: TCP/IP • 1: UDP/IP (2) 套接字通訊功能的順序有無(位元9) • 1: 無順序(固定) (3) 打開方式(位元15、14) • 00: Active打開或UDP/IP • 10: Unpassive打開 • 11: Fullpassive打開 (4) 通訊協定設定(位元10)*1 • 0: 不使用通訊協定支援功能。(使用套接字通訊功能。) • 1: 使用通訊協定支援功能。	b15	b14	...	b10	b9	b8	...	b0	(d)+4	(3)	0	(4)	(2)	(1)	0		—	系統
b15	b14	...	b10	b9	b8	...	b0													
(d)+4	(3)	0	(4)	(2)	(1)	0														

*1 只有下述機型可以使用。此外，根據機型不同，所對應的韌體版本也不同。
 ·R00CPU、R01CPU、R02CPU：沒有版本的限制。
 ·上述以外的Rn(EN)CPU：“29”及其以後
 ·RnPCPU：“15”及其以後

功能

讀取(s1)中指定連接的連接資訊。

出錯

出錯代碼(SD0)	內容
3405H	(s1)中指定的連接編號超出1~16的範圍時。

異常完成時，完成狀態(s2)+1中將儲存出錯代碼。
 關於完成狀態(s2)+1中儲存的出錯代碼，請參閱以下手冊。
 MELSEC iQ-R 乙太網路用戶手冊(應用篇)

連接的通訊目標更改 (UDP/IP)

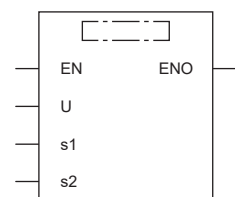
SP. SOCCSET



對指定連接的通訊對象IP地址、通訊對象埠編號進行更改。

梯形圖	ST
	ENO:=SP_SOCCSET (EN, U, s1, s2) ;

FBD/LD



■執行條件

指令	執行條件
SP. SOCCSET	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	連接編號	1~16	無符號BIN16位元	ANY16
(s2)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 5)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H) G□	Z	LT、 LST、LC	LZ		K	H	E	\$	
(U)	—	—	○	—	—	—	—	○	—	—	○	○	
(s1)	—	—	○	—	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	

■控制資料

操作數：(s2)				
元件	項目	內容	設定範圍	設定方
+0	系統區域	—	—	—
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H：正常完成 • 0000H以外：異常完成(出錯代碼)	—	系統
+2 +3	對象設備IP地址	儲存對象設備的IP地址。	00000001H~ FFFFFFFFH (FFFFFFFFH：廣播輪詢通訊)	用戶
+4	對象設備埠編號	儲存對象設備的埠編號。	0001H~FFFFH (FFFFH：廣播輪詢通訊)	用戶

功能

UDP/IP通訊時，對(s1)中指定連接的通訊對象IP地址、通訊對象埠編號進行更改。

要點

- 使用SP.SOCCSET指令時，即使未關閉連接也可更改通訊對象。
- SP.SOCCSET指令的執行時機接收資料區域中有資料時，在執行SP.SOCCSET指令後且執行一次SP.SOCRCV/S.SOCRCVS專用指令後將生效。此外，接收資料區域中沒有資料時，執行SP.SOCCSET指令後將生效。

注意事項

SP.SOCSND指令執行中應勿透過SP.SOCCSET指令更改通訊對象。

出錯

出錯代碼(SD0)	內容
3405H	(s1)中指定的連接編號超出1~16的範圍時。

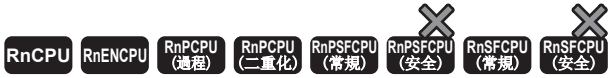
異常完成時，完成狀態(s2)+1中將儲存出錯代碼。

關於完成狀態(s2)+1中儲存的出錯代碼，請參閱以下手冊。

📖 MELSEC iQ-R 乙太網路用戶手冊(應用篇)

連接的接收模式更改

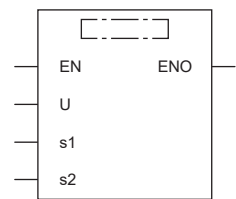
SP. SOCRMODE



對指定的連接更改TCP接收模式(對於UDP通訊的連接無效)及接收資料容量。

梯形圖	ST
	<pre>ENO:=SP_SOCRMODE(EN, U, s1, s2);</pre>

FBD/LD



■執行條件

指令	執行條件
SP. SOCRMODE	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	連接編號	1~16	帶符號BIN16位元	ANY16
(s2)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 4)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其它(U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(U)	—	—	○	—	—	—	—	○	—	—	○	○	—
(s1)	—	—	○	—	—	—	—	○	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—

■控制資料

操作數：(s2)				
元件	項目	內容	設定範圍	設定方
+0	系統區域	—	—	—
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H：正常完成 • 0000H以外：異常完成(出錯代碼)	—	系統
+2	TCP接收模式*1	儲存TCP接收模式。 • 0：TCP通常接收模式 • 1：TCP固定長接收模式	0、1	用戶
+3	接收資料容量	儲存套接字通訊接收資料容量。(字節數)	1~10238	用戶

*1 對UDP通訊的连接無效。

功能

- 對(s1)中指定的连接更改UDP通訊的连接以外的TCP接收模式及接收資料容量。
- 對於TCP的连接將變為(s2)+2中指定的模式。

■TCP通常接收模式

接收資料時，將資料儲存到套接字通訊接收資料區域中，將SD1506(套接字通訊接收狀態信號)設為ON。

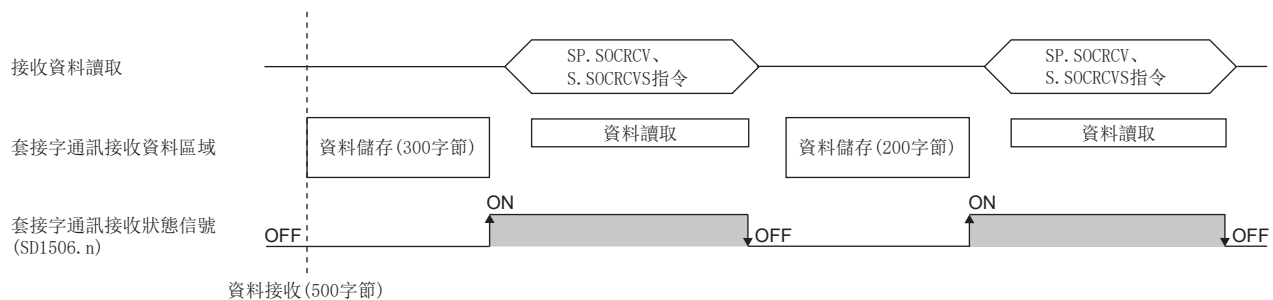
接收的資料超過了指定的接收資料容量的情況下，超出部分的接收資料將變為下一個接收資料。

此後，透過SP. SOCRCV/S. SOCRCVS指令從套接字通訊接收資料區域中讀取資料之前接收了資料的情況下，將被儲存到OS內接收資料中。

此外，透過SP. SOCRCV/S. SOCRCVS指令從套接字通訊接收資料區域中獲取資料時，OS內接收資料中如果有儲存的資料將被儲存到套接字通訊接收資料區域中，將SD1506(套接字通訊接收狀態信號)設為ON。

例

將接收資料容量設定為300字節，但接收了500字節的情況下



■TCP固定長接收模式

接收資料時，將資料儲存到套接字通訊接收資料區域中。未達到指定的接收資料容量的情況下，SD1506(套接字通訊接收狀態信號)不變為ON。

在接收資料達到接收資料容量之前重複接收，在達到接收資料容量的時刻SD1506(套接字通訊接收狀態信號)將變為ON。

接收的資料超過了指定的接收資料容量的情況下，超過部分的接收資料將變為下一個接收資料。

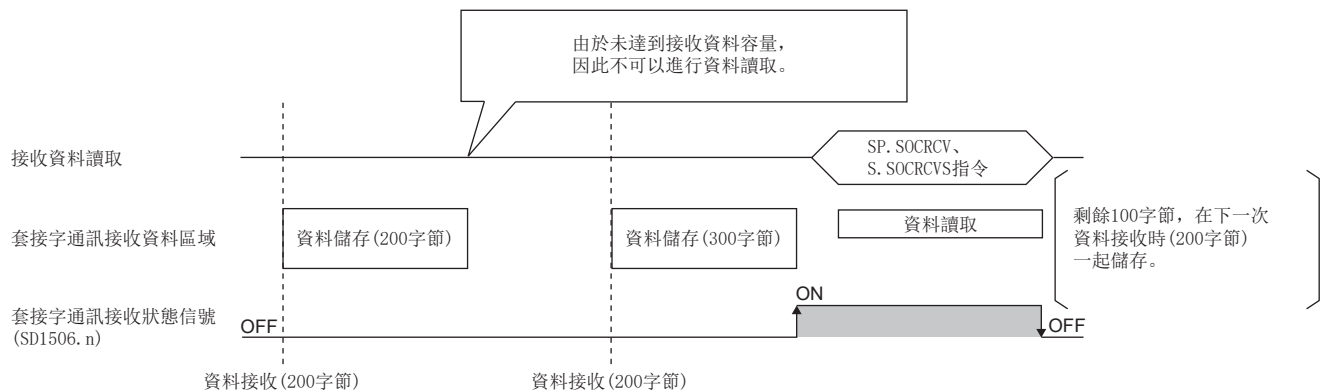
此後，在透過SP.SOCRCV/S.SOCRCVS指令從套接字通訊接收資料區域中讀取資料之前接收了資料的情況下，將被儲存到OS內接收資料中。

透過SP.SOCRCV/S.SOCRCVS指令從套接字通訊接收資料區域中獲取資料時，OS內接收資料中有儲存的資料時，將被儲存到套接字通訊接收資料區域中，但未達到指定的接收資料容量的情況下，SD1506(套接字通訊接收狀態信號)不變為ON。

此後，在接收資料達到接收資料容量之前將重複接收，在達到接收資料容量的時刻SD1506(套接字通訊接收狀態信號)將變為ON。

例

將接收資料容量設定為300字節，連續接收了200字節的情況下



要點

• 元件的有效利用

對於SP.SOCRCV/S.SOCRCVS指令中使用的接收資料的儲存用元件，預設必須預留1024字，透過將接收資料容量指定為1024字及其以下可以有效利用元件。

• 接收資料分離的防止

根據線路種類等，對象設備的資料可能會在到達前被分離，針對此情況，透過在TCP固定長接收模式中指定接收資料容量可以防止資料分離。

• 接收資料合併的防止

由於順控程式接收處理的延遲等，根據對象設備分開發送的資料有可能被合併。

透過在TCP固定長接收模式中指定接收資料容量，可以正確分開接收。

- 在(s2)+3中，指定1次的SP.SOCRCV/S.SOCRCVS指令中讀取的接收資料容量。UDP的情況下，接收的資料超出指定的接收資料容量的情況下，超出部分的接收資料將變為下一個接收資料。

出錯

出錯代碼(SD0)	內容
3405H	(s1)中指定的連接編號超出1~16的範圍時。

異常完成時，完成狀態(s2)+1中將儲存出錯代碼。

關於完成狀態(s2)+1中儲存的出錯代碼，請參閱以下手冊。

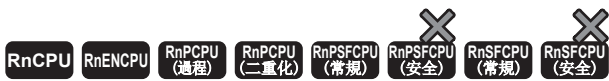
📖 MELSEC iQ-R 乙太網路用戶手冊(應用篇)

要點

在TCP固定長接收模式中，接收狀態信號不變為ON時，可以將當前時刻接收的資料透過SP.SOCRDATA指令讀取，可以確認來自於對象設備的發送資料是否不足。

套接字通訊接收資料讀取

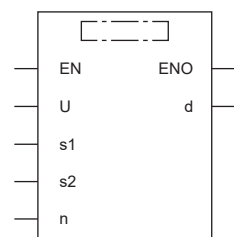
S(P). SOCRDATA



從(s1)中指定連接的套接字通訊接收資料區域中讀取(n)中指定字數的資料，儲存到(d)中指定的元件及其以後。

梯形圖	ST
	ENO:=S_SOCRDATA (EN, U, s1, s2, n, d) ; ENO:=SP_SOCRDATA (EN, U, s1, s2, n, d) ;

FBD/LD



執行條件

指令	執行條件
S. SOCRDATA	
SP. SOCRDATA	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	連接編號	1~16	帶符號BIN16位元	ANY16
(s2)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 2)
(d)	儲存讀取的資料的起始元件	—	字元	ANY16*1
(n)	讀取資料數	1~5120	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其它(U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K、H	E	
(U)	—	—	○	—	—	—	○	—	—	○	○
(s1)	—	—	○	—	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	—	○	—	—	—	○	○	—	—	—

■控制資料

操作數：(s2)				
元件	項目	內容	設定範圍	設定方
+0	系統區域	—	—	—
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H：正常完成 • 0000H以外：異常完成(出錯代碼)	—	系統

功能

從(s1)中指定連接的套接字通訊接收資料區域中，讀取(n)中指定字數的資料，儲存在(d)中指定的元件及其以後。讀取資料(n)為0時將無處理。

要點

- 將讀取資料數設定為1字元時，可以讀取接收資料長度。由此，可以更改執行SP. SOCRCV/S. SOCRCVS指令時儲存接收資料的元件。
- 透過S(P). SOCRDATA指令對本次接收的資料進行檢查、SP. SOCRM0DE指令指定下一次接收的資料容量後，可以透過SP. SOCRCV/S. SOCRCVS指令讀取本次的資料。由此，可以以本次接收的資料為基礎指定下一次接收的資料容量。

出錯

出錯代碼(SD0)	內容
3405H	(s1)中指定的連接編號超出1~16的範圍時。
	(n)中指定的元件的值超過了5120時。

異常完成時，完成狀態(s2)+1中將儲存出錯代碼。

關於完成狀態(s2)+1中儲存的出錯代碼，請參閱以下手冊。

📖 MELSEC iQ-R 乙太網路用戶手冊(應用篇)

24.3 通訊協定支援功能指令

登錄協定執行

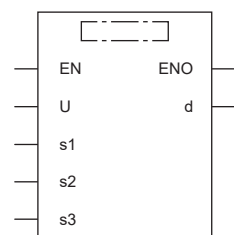
SP.ECPRTCL



執行通訊協定支援功能中設定的協定。

梯形圖	ST
	ENO:=SP_ECPRTCL (EN, U, s1, s2, s3, d);

FBD/LD



■執行條件

指令	執行條件
SP.ECPRTCL	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	連接編號	1~16	無符號BIN16位元	ANY16
(s2)	連續執行的協定數	1~8	無符號BIN16位元	ANY16
(s3)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 2)
(d)	透過指令完成讓1個掃描變為ON的元件 異常完成時 (d)+1也將變為ON。	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其它 (U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$		
(U)	—	—	○	—	—	—	—	○	—	—	○	○	
(s1)	○	—	○*1	—	—	—	—	○*1	○	—	—	—	
(s2)	○	—	○*1	—	—	—	—	○*1	○	—	—	—	
(s3)	○	—	○*1	—	—	—	—	○*1	—	—	—	—	
(d)	○*1	—	○	—	—	—	—	○*1	—	—	—	—	

*1 不能使用局部元件及各程式中設定的檔案寄存器。

■控制資料

操作數：(s3)				
元件	項目	內容	設定範圍	設定方
+0	執行數結果	儲存透過SP.ECPRTEL指令執行的協定數。發生出錯的協定也包含在執行數中。設定資料、控制資料的設定內容有錯誤的情況下將儲存“0”。	0、1~8	系統
+1	完成狀態	指令完成時儲存完成狀態。 執行多個協定的情況下，將儲存最後執行的協定的執行結果。 • 0：正常完成 • 0以外：異常完成(出錯代碼)	—	系統
+2	執行協定編號指定1	指定第1個執行的協定的協定編號。	1~128	用戶
+3	執行協定編號指定2	指定第2個執行的協定的協定編號。	0、1~128	用戶
+4	執行協定編號指定3	指定第3個執行的協定的協定編號。	0、1~128	用戶
+5	執行協定編號指定4	指定第4個執行的協定的協定編號。	0、1~128	用戶
+6	執行協定編號指定5	指定第5個執行的協定的協定編號。	0、1~128	用戶
+7	執行協定編號指定6	指定第6個執行的協定的協定編號。	0、1~128	用戶
+8	執行協定編號指定7	指定第7個執行的協定的協定編號。	0、1~128	用戶
+9	執行協定編號指定8	指定第8個執行的協定的協定編號。	0、1~128	用戶
+10	校驗一致 接收資料包編號1	第1個執行的協定的通訊類型中包含有接收的情況下，儲存校驗一致的接收資料包編號。通訊類型為“僅發送”的情況下，將儲存“0”。執行第1個協定時發生了出錯的情況下將儲存“0”。	0、1~16	系統
+11	校驗一致 接收資料包編號2	第2個執行的協定的通訊類型中包含有接收的情況下，儲存校驗一致的接收資料包編號。通訊類型為“僅發送”的情況下，將儲存“0”。執行第2個協定時發生了出錯的情況下將儲存“0”。執行的協定數不足2時將儲存“0”。	0、1~16	系統
+12	校驗一致 接收資料包編號3	第3個執行的協定的通訊類型中包含有接收的情況下，儲存校驗一致的接收資料包編號。通訊類型為“僅發送”的情況下，將儲存“0”。執行第3個協定時發生了出錯的情況下將儲存“0”。執行的協定數不足3時將儲存“0”。	0、1~16	系統
+13	校驗一致 接收資料包編號4	第4個執行的協定的通訊類型中包含有接收的情況下，儲存校驗一致的接收資料包編號。通訊類型為“僅發送”的情況下，將儲存“0”。執行第4個協定時發生了出錯的情況下將儲存“0”。執行的協定數不足4時將儲存“0”。	0、1~16	系統
+14	校驗一致 接收資料包編號5	第5個執行的協定的通訊類型中包含有接收的情況下，儲存校驗一致的接收資料包編號。通訊類型為“僅發送”的情況下，將儲存“0”。執行第5個協定時發生了出錯的情況下將儲存“0”。執行的協定數不足5時將儲存“0”。	0、1~16	系統
+15	校驗一致 接收資料包編號6	第6個執行的協定的通訊類型中包含有接收的情況下，儲存校驗一致的接收資料包編號。通訊類型為“僅發送”的情況下，將儲存“0”。執行第6個協定時發生了出錯的情況下將儲存“0”。執行的協定數不足6時將儲存“0”。	0、1~16	系統
+16	校驗一致 接收資料包編號7	第7個執行的協定的通訊類型中包含有接收的情況下，儲存校驗一致的接收資料包編號。通訊類型為“僅發送”的情況下，將儲存“0”。執行第7個協定時發生了出錯的情況下將儲存“0”。執行的協定數不足7時將儲存“0”。	0、1~16	系統
+17	校驗一致 接收資料包編號8	第8個執行的協定的通訊類型中包含有接收的情況下，儲存校驗一致的接收資料包編號。通訊類型為“僅發送”的情況下，將儲存“0”。執行第8個協定時發生了出錯的情況下將儲存“0”。執行的協定數不足8時將儲存“0”。	0、1~16	系統

功能

- 執行工程工具中登錄的協定。使用(s1)中指定的連接後，執行的協定取決於(s3)中指定的元件及其以後的控制資料。
- 1次的指令執行中，連續執行(s2)中指定的數(最大8)的協定。
- 執行了協定的數將被儲存到執行數結果(s3)+0中。
- 對於協定的執行狀態，可透過通訊協定支援功能執行狀態確認用區域(Un\G350~Un\G669)進行確認。(MELSEC iQ-R 乙太網路用戶手冊(應用篇))
- 對於SP.ECPRTCL指令執行及正常/異常完成，可透過設定資料中指定的完成元件(d)、完成時的狀態顯示元件(d)+1進行確認。

- 完成元件(d)

在SP.ECPRTCL指令完成的掃描的END處理中ON、在下一個END處理中OFF。

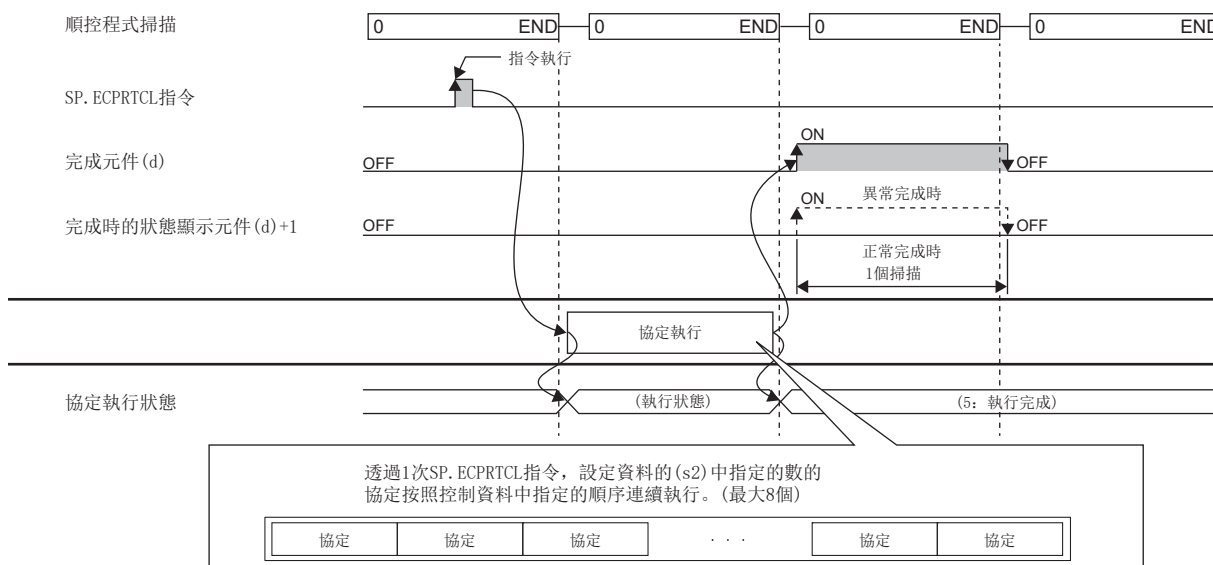
- 完成時的狀態顯示元件(d)+1

根據SP.ECPRTCL指令完成時的狀態而ON/OFF。

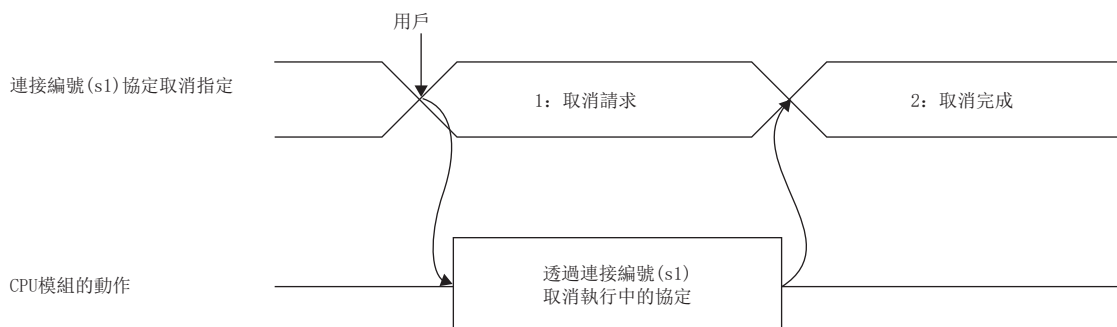
正常完成時：保持為OFF不變。

異常完成時：在SP.ECPRTCL指令完成的掃描的END處理中ON、在下一個END處理中OFF。異常完成時，(s3)+1的完成狀態中將儲存出錯代碼。

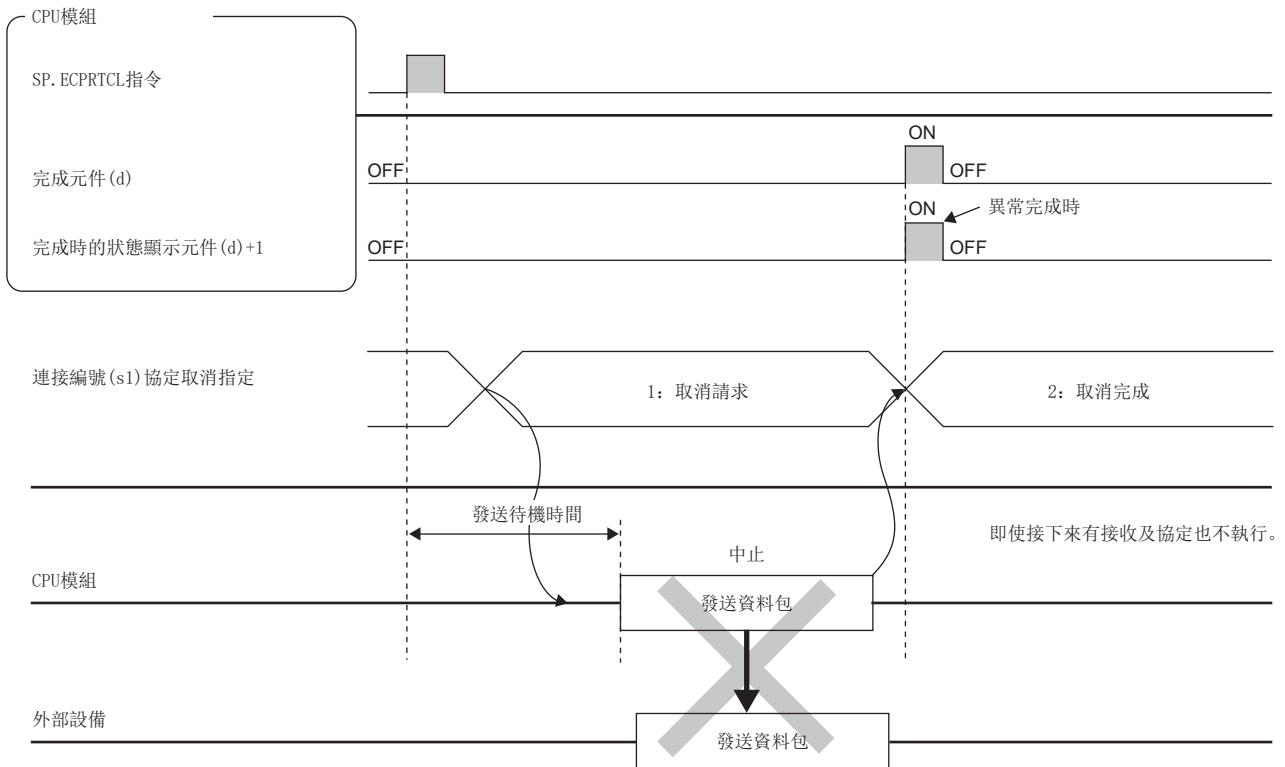
- SP.ECPRTCL指令的執行時機如下所示。



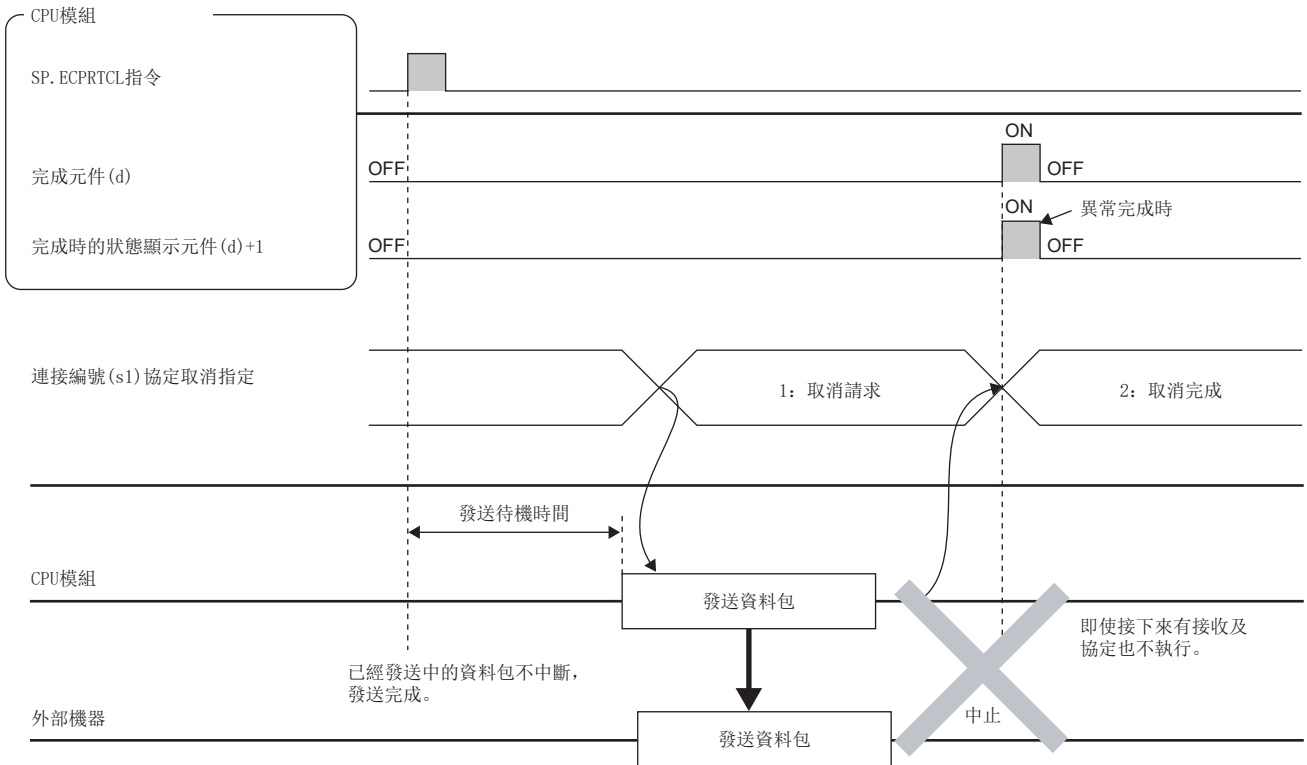
- 透過設定協定取消請求，可以取消協定的執行。協定取消指定時，使用通訊協定支援功能執行狀態確認用區域(Un\G350~Un\G669)。(MELSEC iQ-R 乙太網路用戶手冊(應用篇))



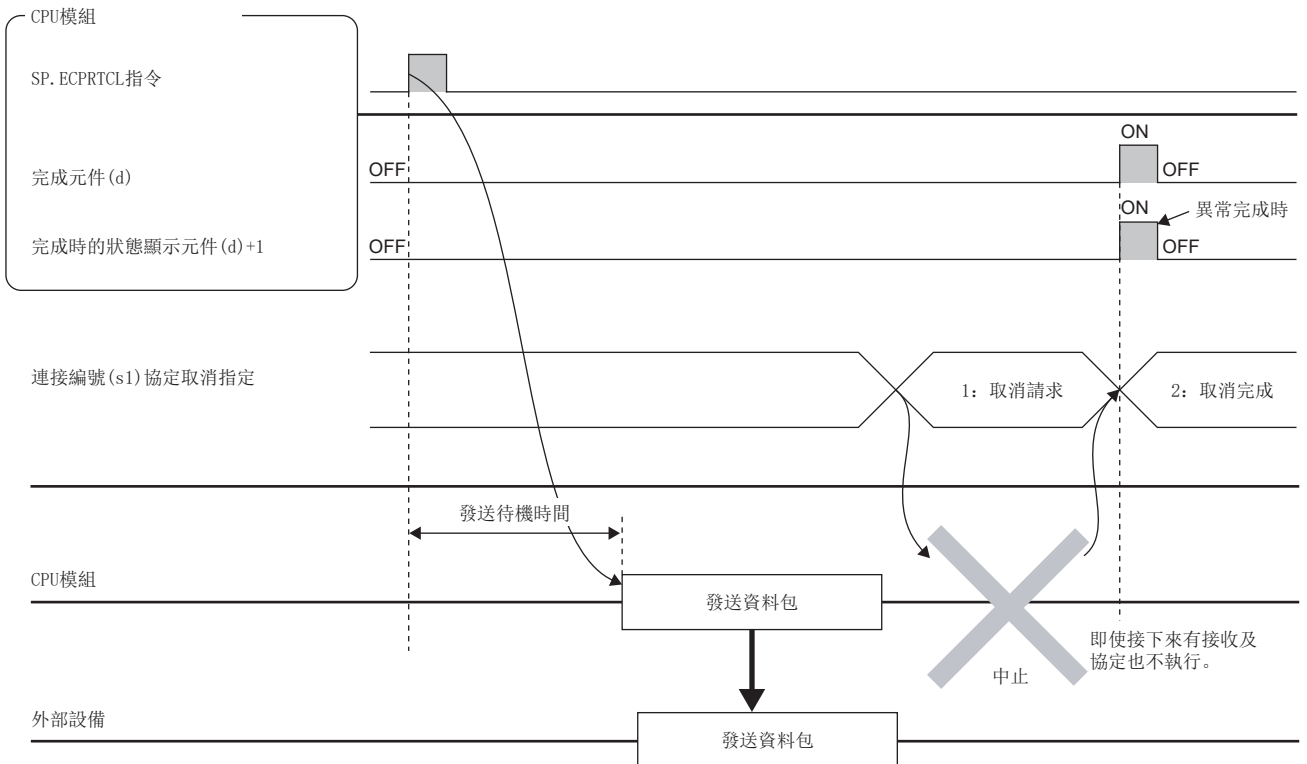
- 各時機中協定取消的動作如下所示。
 - 發送之前有取消請求的情況下
- 協定執行狀態為“1: 發送等待”情況下的動作如下所示。



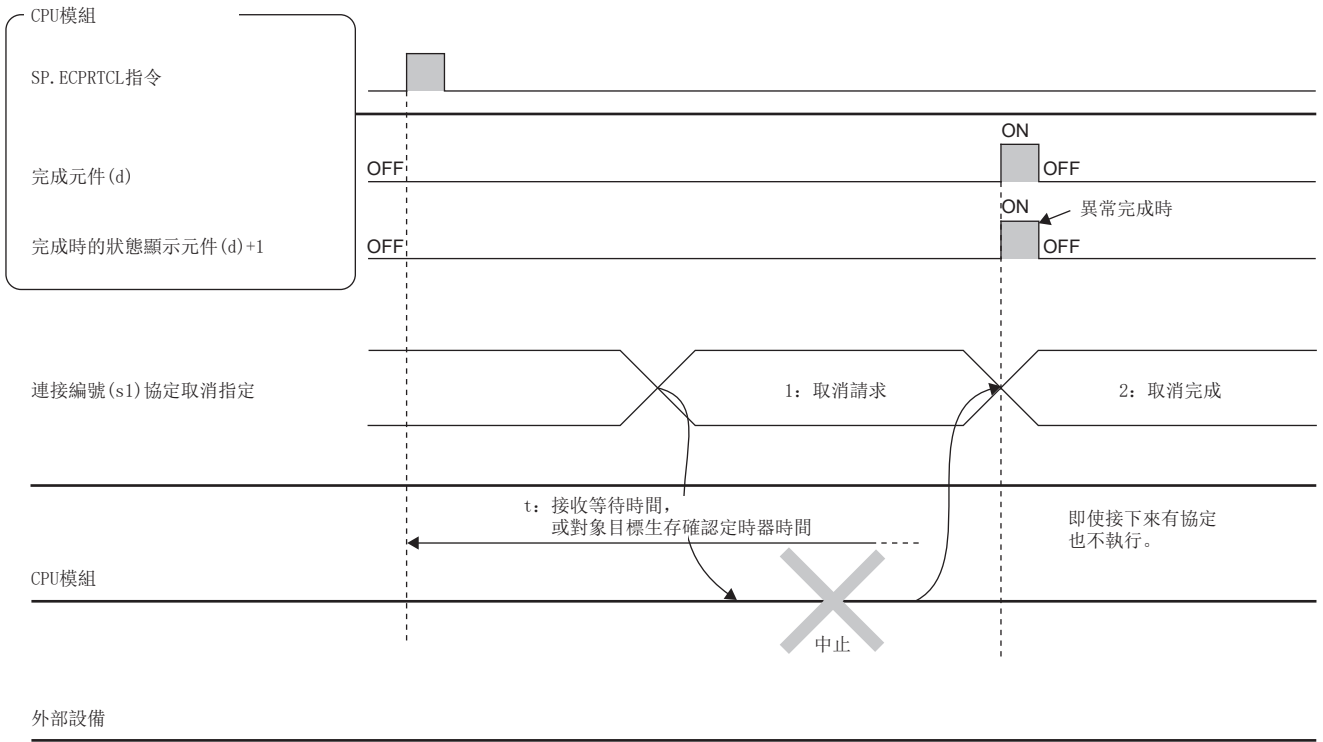
- 發送完成之前有取消請求的情況下
協定執行狀態為“2: 發送中”，未完成發送情況下的動作如下所示。



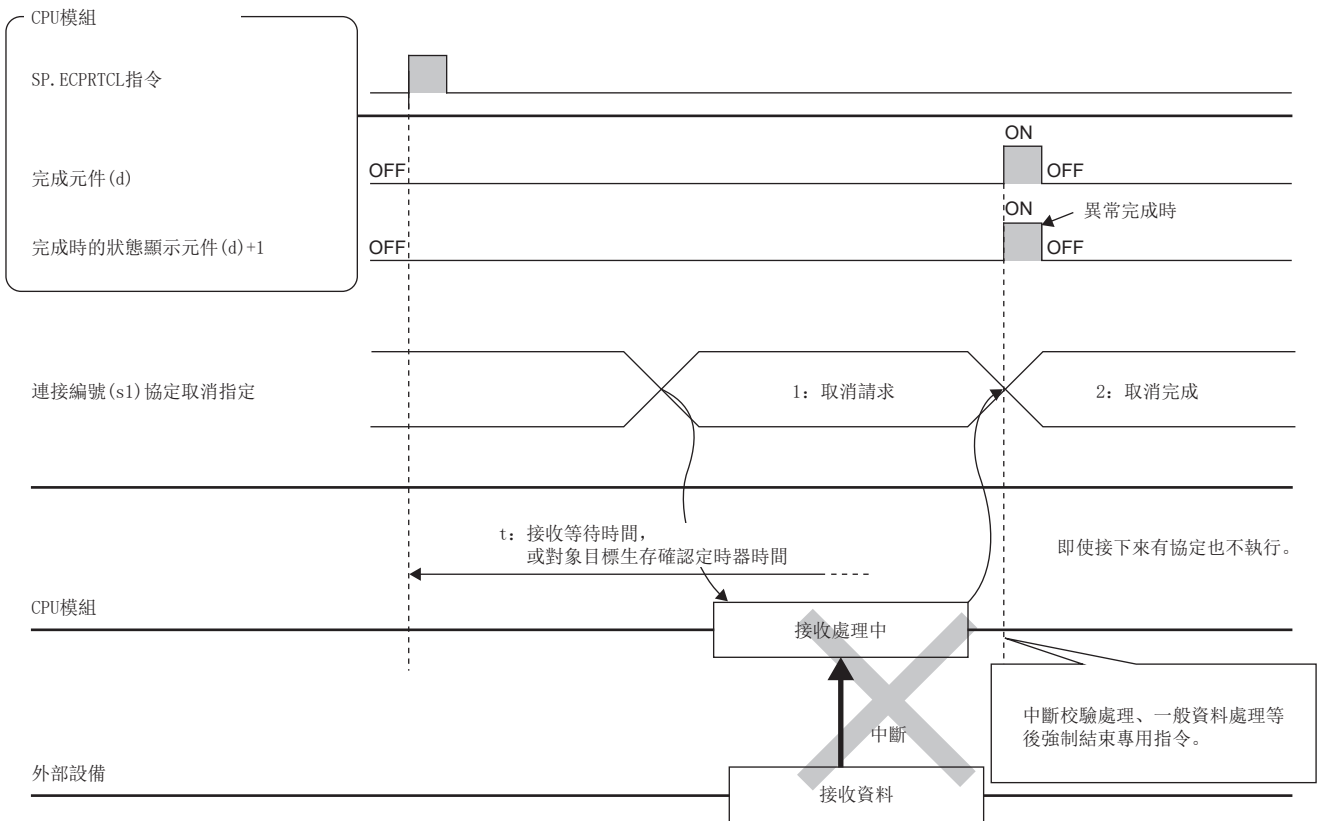
- 發送完成時有取消請求的情況下
協定執行狀態為“2: 發送中”，發送完成情況下的動作如下所示。



• 接收等待中有取消請求的情況下
 協定執行狀態為“3: 接收資料等待”情況下的動作如下所示。



• 接收處理中有取消請求的情況下
 協定執行狀態為“4: 接收處理中”情況下的動作如下所示。



出錯

出錯代碼 (SD0)	內容
3405H	(s1) 超出1~16的範圍時。

異常完成時，完成時的狀態顯示元件 (d)+1 將變為 0N、完成狀態 (s3)+1 中將儲存出錯代碼。

關於完成狀態 (s3)+1 中儲存的出錯代碼，請參閱以下手冊。

📖 MELSEC iQ-R 乙太網路用戶手冊 (應用篇)

注意事項

- 在執行多個協定的情況下，第 m 個協定中發生出錯時，第 m+1 個及其以後的協定將不執行，指令異常完成。
- 執行包含有未轉換轉變數的協定的情況下，因為 1 個資料包中使用的變數的合計資料長超過了 1920 字節的情況下，有可能獲取跨越了數個掃描的 CPU 元件值，因此指令啟動時開始至執行完成為止應勿對未轉換變數中指定的 CPU 元件值進行更改。
- 關於協定取消
 - SP.ECPRTCL 指令異常完成、(s3)+1 的完成狀態中將儲存協定取消請求出錯 (C404H)。
 - 在沒有執行中的協定的狀態下，設定了取消請求的情況下，CPU 模組將以無處理進行取消完成。
 - 不使用通訊協定的情況下，即使設定了取消請求也將被忽略。
 - 連續執行多個協定的情況下，第 n 個協定執行中如果進行了取消請求，CPU 模組將強制結束第 n 個協定，此外，不執行第 n+1 個及其以後的協定。執行數結果 ((s3)+0) 中將儲存執行中的協定編號 n，校驗一致接收資料包編號 1~(n-1) 中將儲存校驗一致的接收資料包編號，完成狀態 ((s3)+1) 中將儲存協定取消請求出錯 (C404H)。
 - CPU 模組透過定期處理確認有無取消請求。因此，從取消請求指示起至取消處理被進行為止有可能需要耗費時間。
- 在 SP.ECPRTCL 指令單體中，因為不進行連接的打開/關閉，因此需要透過 SP.SOCOPEN/SP.SOCCLDSE 指令進行連接的打開/關閉處理。

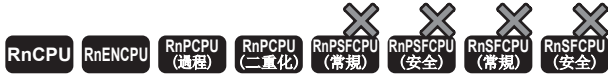
📖 1145 頁 SP.SOCOPEN, 1148 頁 SP.SOCCLDSE

- 對同一連接執行同一指令的情況下，在先執行中的指令完成之前，後一個指令將被忽略而不執行。
- 將接收等待時間設定為“0: 無限等待”的情況下，在協定設定中指定的資料被接收之前 SP.ECPRTCL 指令不完成。

24.4 SLMP幀發送指令

SLMP幀發送

SP.SLMPSND



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- RnPCPU可以在韌體版本為“13”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

對支援SLMP的設備發送SLMP的報文。

梯形圖	ST
	ENO:=SP_SLMPSND(EN, U, s1, s2, d1, d2)

FBD/LD

■執行條件

指令	執行條件
SP.SLMPSND	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	儲存控制資料的起始元件	☞ 1175頁 控制資料	字元	ANY16* ¹
(s2)	儲存請求幀的起始元件	☞ 1177頁 請求幀	字元	ANY16* ¹
(d1)	儲存響應幀的起始元件	☞ 1177頁 響應幀	字元	ANY16_ARRAY* ²
(d2)	透過指令完成讓1個掃描變為ON的元件 異常完成時(d2)+1也將ON。	—	位元	ANYBIT_ARRAY* ² (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。
*2 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組。

■可使用的元件

操作數	位元		字元				雙字元		間接指定	常數			其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H) G□	Z	LT、 LST、LC	LZ	K、H		E	\$		
(U)	—	—	○	—	—	—	—	○	—	—	○	○	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d2)	○	—	○*1	—	—	—	—	○	—	—	—	—	

*1 不能使用T、ST、C、FD。

■控制資料

操作數：(s1)									
元件	項目	內容	設定範圍	設定方					
+0	執行・異常時完成類型	b15 b7 b0 <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 10%;">1</td> <td style="width: 40%;">0</td> <td style="width: 10%;">(2)</td> <td style="width: 40%;">0</td> <td style="width: 10%;">(1)</td> </tr> </table> (1) 執行類型(位元0) • 0: 無到達確認(透過從本站發送請求報文而變為完成狀態。)*1 • 1: 有到達確認(透過從對象設備接收響應報文而變為完成狀態。) (2) 異常時完成類型(位元7) 指定異常完成時的資料的設定狀態。 • 0: 在(s1)+13及其以後不設定異常完成時的資料。(清除(s1)+13及其以後。) • 1: 在(s1)+13及其以後設定異常完成時的資料。	1	0	(2)	0	(1)	—	用戶
1	0	(2)	0	(1)					
+1	完成狀態	指令完成時的狀態將被儲存。 0: 正常 0以外: 異常(出錯代碼)	—	系統					
+2	本站使用通道	指定本站使用的通道。在請求報文中附加/未附加序列號*2由於是由通道決定，所以應根據用途指定通道。 • 1: 未附加序列號的通道 • 2~9: 附加序列號的通道	1~9	用戶					
+3	對象設備IP地址(第3、4八位字節)	指定對象設備的IP地址(第3、4八位字節)。 b15 b8 b7 b0 <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 50%;">3</td> <td style="width: 50%;">4</td> </tr> </table> 3、4: 顯示IP地址的八位字節。	3	4	根據(s1)+3、(s1)+4, 0000001H~FFFFFFEH (1~4294967294)	用戶			
3	4								
+4	對象設備IP地址(第1、2八位字節)	指定對象設備的IP地址(第1、2八位字節)。 b15 b8 b7 b0 <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 50%;">1</td> <td style="width: 50%;">2</td> </tr> </table> 1、2: 顯示IP地址的八位字節。	1	2		用戶			
1	2								
+5	對象設備埠編號	指定對象設備的埠編號。	1~65534 (1~FFFEH)	用戶					
+6	請求目標網路編號	0000H固定	0000H	用戶					
+7	請求目標站號	00FFH固定	00FFH	用戶					
+8	請求目標模組I/O編號	指定請求目標的模組I/O編號。 • 03FFH: 本站/管理CPU • 03E0H: 多CPU1號機 • 03E1H: 多CPU2號機 • 03E2H: 多CPU3號機 • 03E3H: 多CPU4號機 • 03D0H: 控制系統CPU • 03D1H: 待機系統CPU • 03D2H: A系統CPU • 03D3H: B系統CPU	03FFH、03E0H~03E3H、03D0H~03D3H	用戶					
+9	請求目標多支路站號	0000H固定	0000H	用戶					

操作數: (s1)						
元件	項目	內容	設定範圍	設定方		
+10	重新發送次數	(s1)+0中指定的執行類型為“1: 有到達確認”時變為有效。 ■執行指令時 指定在(s1)+11中指定的監視時間內未完成的情況下, 重新發送的次數。 • 0~15(次) ■指令完成時 儲存重新發送的次數(結果)。 • 0~15(次)	0~15	用戶/系統		
+11	到達監視時間	指定處理完成為止的監視時間。在監視時間內未完成的情況下, 將進行(s1)+10中指定的次數為止的重新發送。 • 0: 10秒 • 1~32767: 1~32767秒	0~32767	用戶		
+12	時鐘設定標誌	儲存(s)+13及其以後的資料有效/無效狀態。此外, 在正常完成的情況下, (s1)+13及其以後的資料將被清除。 • 0: 無效 • 1: 有效	—	系統		
+13	時鐘資料	高位8位元: 月(01H~12H) 低位8位元: 年(00H~99H) 公曆低2位	—	系統		
+14		高位8位元: 時(00H~23H) 低位8位元: 日(01H~31H)				
+15		高位8位元: 秒(00H~59H) 低位8位元: 分(00H~59H)				
+16		高位8位元: 年(00H~99H) 公曆高2位 低位8位元: 星期(00H(日)~06H(六))				
+17	異常檢測設備IP地址(第3、4八位字節)	儲存檢測出異常的站的IP地址(第3、4八位字節)。 b15 b8 b7 b0 <table border="1" style="margin-left: 40px;"> <tr> <td style="width: 40px; text-align: center;">3</td> <td style="width: 40px; text-align: center;">4</td> </tr> </table> 3、4: 顯示IP地址的八位字節。	3	4	—	系統
3	4					
+18	異常檢測設備IP地址(第1、2八位字節)	儲存檢測出異常的站的IP地址(第1、2八位字節)。 b15 b8 b7 b0 <table border="1" style="margin-left: 40px;"> <tr> <td style="width: 40px; text-align: center;">1</td> <td style="width: 40px; text-align: center;">2</td> </tr> </table> 1、2: 顯示IP地址的八位字節。	1	2	—	系統
1	2					

*1 在(s1)+0的無到達確認中已設定的情況下, 接收資料未被設定。應在下述的情況下設定(s1)+0的無到達確認。

- 使用未返回響應報文的指令的情況下
- 未參閱響應報文的情況下

*2 對同一個支援SLMP的設備發送多個的請求報文的情況下附加。附加的序列號在系統中自動決定。另外, 過程CPU(二重化模式)中, 在A系統/B系統中各自保持序列號。
關於序列號, 請參閱下述手冊。

📖 SLMP參考手冊

■請求幀

操作數：(s2)				
元件	項目	內容	設定範圍	設定方
+0	請求資料長	指定從監視定時器到請求資料的資料長。(字節單位)	1~2000	用戶
+1	監視定時器	接收請求報文的對象設備，設定從請求存取目標的處理到響應返回為止的等待時間的定時器。(單位：250ms) <ul style="list-style-type: none"> • 0：無限等待 • 1~65535：1~65535×250ms 	0~65535	用戶
+2~+□	請求資料	儲存SLMP的報文的請求資料。	—	用戶

■響應幀

操作數：(d1)				
元件	項目	內容	設定範圍	設定方
+0	響應資料長	儲存從結束代碼到響應資料為止的資料長。(字節單位)	2~2000	系統
+1	結束代碼	儲存指令處理結果。 正常結束時儲存為0。異常結束時儲存對象設備中設定的出錯代碼。	—	系統
+2~+□	響應資料*1	設定對請求資料的執行結果。 (根據指令的不同，也有不返回響應資料的情況。)	—	系統

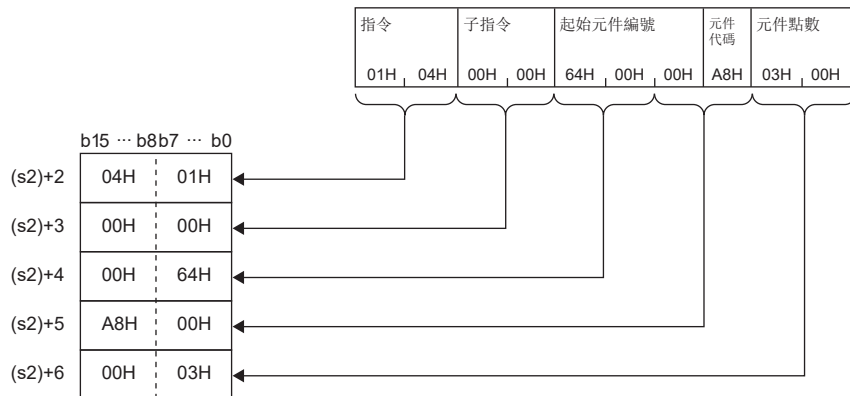
*1 響應資料以字節單位從低位字節側開始依次被儲存。接收了奇數字節的響應資料時，最後的響應資料將儲存於最終資料儲存區域的低位字節中。

- 發送“Read(指令: 0401H)”(字元單位讀出)時的請求資料及正常完成時的響應資料如下所示。

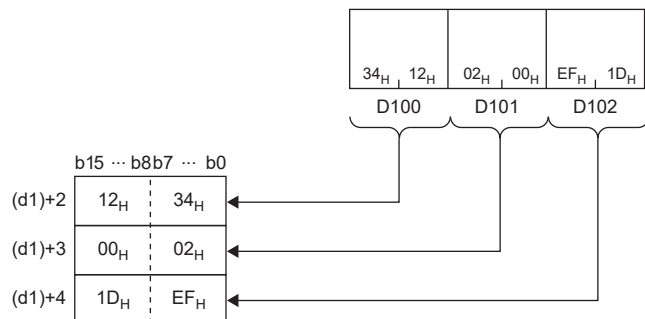
例

讀取D100~D102的值的的情況下

[請求資料]

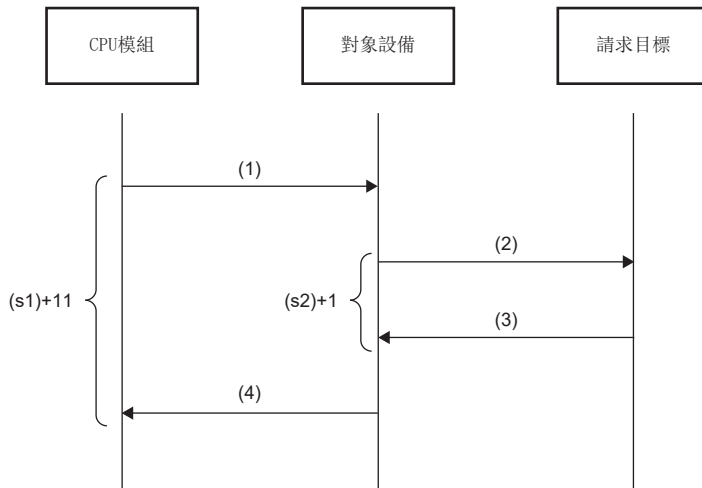


[響應資料]



注意事項

- 同時執行多個的SP. SLMPSND指令的情況下，應避免SP. SLMPSND指令的通道重複。對於設定了同一通道的SP. SLMPSND指令，不能同時使用。在同一順控掃描內滿足同一通道的SP. SLMPSND指令的執行條件的情況下，僅執行最初執行的SP. SLMPSND指令，不執行在那之後的SP. SLMPSND指令。此外，即使執行了與執行中的SP. SLMPSND指令相同的通道的設定的SP. SLMPSND指令的情況下，之後也不執行的SP. SLMPSND指令的處理。在CPU模組沒執行SP. SLMPSND指令的處理的情況下，SM699(專用指令未執行標誌)將變為ON。
- 關於控制資料的到達監視時間((s1)+11)及請求幀的監視定時器((s2)+1)，應設定為到達監視時間≥監視定時器。



- (1) 請求報文
 (2) 從對象設備向請求目標發出的處理請求
 (3) 從請求目標向對象設備發出的處理響應
 (4) 響應報文

要點

即使在對象設備返回異常響應的情況下，SP. SLMPSND指令也將正常完成。SP. SLMPSND指令正常完成的情況下，透過響應幀的結束代碼判斷是正常響應還是異常響應。異常響應的情況下，應根據使用的支援SLMP的設備的手冊進行確認及處理。

出錯

出錯代碼 (SD0)	內容
3405H	(s1)+2中指定的本站使用通道為1~9以外時。
	(s2)+0中指定的請求資料長為0、或超過2000時。

異常完成時，完成時的狀態顯示元件(d2)+1將變為ON、完成狀態(s1)+1中將儲存出錯代碼。關於完成狀態(s1)+1中儲存的出錯代碼，請參閱以下手冊。

📖 MELSEC iQ-R 乙太網路用戶手冊(應用篇)

24.5 檔案傳送功能用指令

發送FTP客戶端檔案

SP.FTPPUT

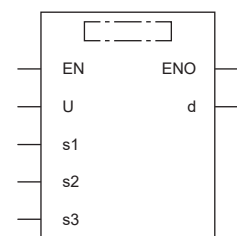


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“22”及其以後的版本使用。工程工具可以在版本為“1.025B”及其以後的版本使用。
- RnPCPU可以在韌體版本為“13”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

將 (s2) 中指定的CPU模組的檔案發送至 (s3) 中指定的FTP伺服器的資料夾中。

梯形圖	ST
	<pre>ENO:=SP_FTPPUT (EN, U, s1, s2, s3, d);</pre>

FBD/LD



■執行條件

指令	執行條件
SP.FTPPUT	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 4)
(s2)	CPU模組中儲存的檔案名(傳送源)*1	—	Unicode字元串*2	ANYSTRING_DOUBLE
(s3)	FTP伺服器的資料夾路徑(傳送目標)*1	—	Unicode字元串*2	ANYSTRING_DOUBLE
(d)	透過指令完成讓1個掃描變為ON的元件 異常完成時 (d)+1也將變為ON。	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 Unicode字元串或儲存了Unicode字元串的起始元件。

*2 資料類型為Unicode字元串時僅可使用半形英文數字、符號、假名字元、全形字元(移位JIS代碼)。不支援的字元作為“_”處理。

■可使用的元件

操作數	位元		字元				雙字元		間接指定	常數			其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H) G□	Z	LT、 LST、LC	LZ	K、H		E	\$		
(U)	—	—	○	—	—	—	—	○	—	—	○	○	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	○	—	
(s3)	—	—	○	—	—	—	—	○	—	—	○	—	
(d)	○	—	○*1	—	—	—	—	○	—	—	—	—	

*1 不能使用T、ST、C、FD。

■控制資料

操作數：(s1)				
元件	項目	內容	設定範圍	設定方
+0	使用用途設定區域	b15 ... b3 b2 b1 b0 <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> 0 (2) (1) 0 </div> (1) 傳送完成檔案刪除設定 (位元2) 傳送後指定傳送完成的檔案的處理。 • 0: 不刪除 • 1: 刪除 (2) 檔案傳送時的暫時檔案建立設定 (位元3) 指定建立/不建立檔案傳送時的暫時檔案 (FTPCLI_I.TMP)。 透過在“0: 建立”中設定, 在檔案傳送中(檔案改寫中)即使發生電纜斷開或電源斷開等, 傳送目標檔案也不會變為不正確的狀態。 • 0: 建立 • 1: 不建立	如左所示	用戶
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H: 正常完成 • 0000H以外: 異常完成(出錯代碼)	—	系統
+2	處理檔案總數	儲存透過SP.FTTPUT指令處理的檔案的總數。	—	系統
+3	處理完成檔案數	儲存處理完成的檔案數。	—	系統

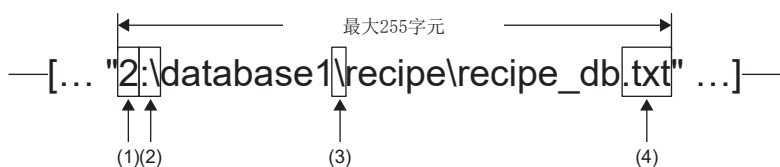
功能

- 將(s2)中指定的CPU模組的檔案發送至(s3)中指定的FTP伺服器的資料夾中。執行SP.FTTPUT指令時, 連接“模組參數”的“FTP客戶端設定”中指定的FTP伺服器, 檔案發送後斷開與FTP伺服器的連接。關於“FTP客戶端設定”的詳細內容, 請參閱下述手冊。

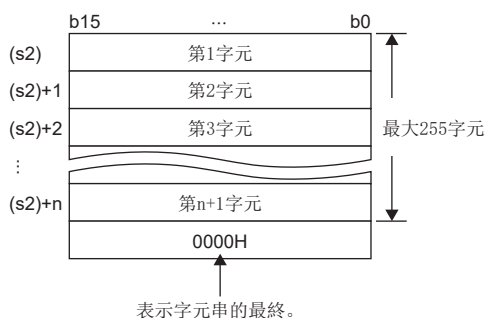
📖 MELSEC iQ-R 乙太網路用戶手冊(應用篇)

- 透過SP.FTTPUT指令處理的檔案的總數儲存於(s1)+2的“處理檔案總數”中、處理完成的檔案數儲存於(s1)+3的“處理完成檔案數”中。

- 在(s2)中以Unicode字元串形式，指定CPU模組的傳送源的驅動器No. 2~4與儲存了檔案的資料夾路徑、檔案名(包含副檔名)。檔案路徑長度的最大值為255字元。除去檔案名的路徑長度在不含檔案的分割字元的情況下，最多246字元。驅動器、資料夾路徑、檔案名的分割字元中使用半形的“\”或“/”。



- (1) 可以指定的驅動器No. 為2~4。只有R00CPU為3、4。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
- (2) 驅動器No. 的分割字元使用“:”或“/”。
- (3) 資料夾路徑、檔案的分割字元使用半形的“\”或“/”。
- (4) 指定的檔案名中包含副檔名。



- (s2)中指定的檔案名或副檔名中，可以使用通配符指定(*、?)。

符號	內容
*	以從指定為“*”位置開始的任意的字元串(包含沒有的情況)的全部的檔案為對象。
?	以從指定為“?”位置開始的任意的字元(不包含沒有的情況)的全部的檔案為對象。(可以使用多個“?”。)

點號不可以使用通配符指定。

下述指定方法會出錯。

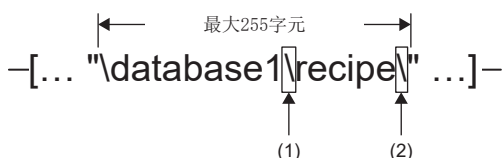
- 檔案名(到點號為止)或副檔名中至少使用2個“*”。(例: “*abc*.txt”)
- 檔案名(到點號為止)或副檔名中混用“*”與“?”。(例: “*ab?.txt”)

透過通配符指定的可以傳送的檔案數是由指定的各檔案的檔案名容量的合計所決定。可以傳送的檔案數及個檔案名容量滿足下述公式的情況下可以傳送。不滿足下述公式的情況下執行檔案傳送功能指令會發生異常完成。

$$(F_i + NM) + 1 < 65536 \text{ [字節]}$$

N: 通配符指定中一致的檔案的總數
 F_i: 通配符指定中一致的檔案名的合計容量
 M: 指定資訊容量(固定值: 6字節)

- 在SP.FTPPUT指令執行期間，傳送的某1個檔案發生了出錯時，在此同時中止傳送、剩餘檔案的傳送也不會被執行。
- 不能傳送的檔案即使符合通配符指定的條件也無法作為傳送的對象。
- 包含檔案名、副檔名的檔案路徑超過255字元的情況下，即使符合通配符指定的條件也無法作為傳送的對象。
- 在(s3)中，以Unicode字元串的形式指定FTP伺服器的傳送目標的資料夾路徑。透過FTP伺服器的主目錄的相對路徑指定資料夾路徑。資料夾路徑的分割字元使用半形的“\”或“/”。*1資料夾路徑長度的最大值為255字元。但是，資料夾路徑(包含末尾的分割字元)與(s2)的檔案名部分的合計，應指定為不超過傳送目標FTP伺服器支援的最大路徑長度。雖然末尾的分割字元可以省略，但省略的情況下，末尾視為設定了“\”。此外，不存在資料夾路徑的情況下，指令執行時在系統中自動建立、進行檔案傳送。



- (1) 資料夾的分割字元使用半形的“\”或“/”。*1
- (2) 末尾的分割字元可以省略。

*1 根據FTP伺服器，有無法將“\”作為分隔符號使用的情況。

- 在(s3)的NULL或元件中僅指定了“0000H”時，直接存取FTP伺服器的主目錄。詳細內容應符合FTP伺服器的規格。
- 傳送目標中存在同名的檔案的情況，應對同名的檔案進行覆蓋。
- 可以發送的最大檔案容量為1檔案4G字節。

- 對於SP.FTPPUT指令的執行及正常/異常完成，可以透過設定資料中指定的完成元件(d)、完成時的狀態顯示元件(d)+1進行確認。

- 完成元件(d)

SP.FTPPUT指令透過已完成的掃描的END處理變為ON、透過下一個END處理變為OFF。

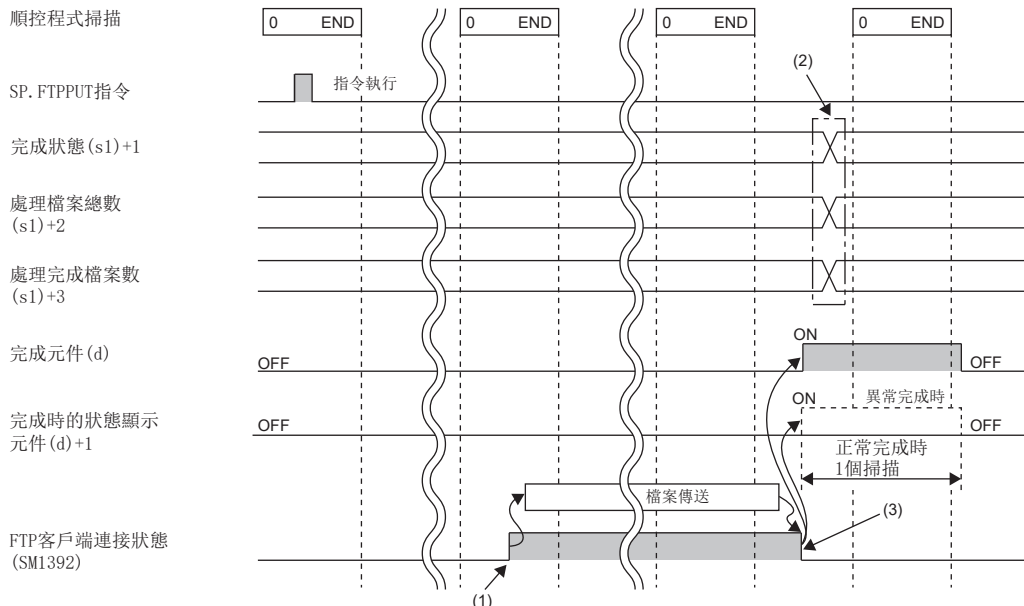
- 完成時的狀態顯示元件(d)+1

根據SP.FTPPUT指令已完成時的狀態而變為ON/OFF。

正常完成時：保持為OFF不變。

異常完成時：SP.FTPPUT指令透過已完成的掃描的END處理變為ON、透過下一個END處理變為OFF。異常完成時，(s1)+1的完成狀態中將儲存出錯代碼。

- SP.FTPPUT指令的執行時機如下所示。



(1) FTP伺服器連接後的END處理中ON。

(2) 指令完成時儲存值。

(3) 全部檔案的傳送完成時變為OFF。

- FTP伺服器連接中SM1392 (FTP客戶端連接狀態) 變為ON、斷開後SM1392變為OFF。
- SP.FTPPUT指令執行中，SM753 (檔案存取中) 變為ON。SM753為ON時，不能執行SP.FTPPUT指令。執行的情況下將變為無處理。
- 在SP.FTPPUT指令或SP.FTPGET指令執行中執行SP.FTPPUT指令的情況下，在先執行中的指令完成之前，後一個指令將被忽略而不執行。SP.FTPPUT指令被忽略的情況下，SM699 (專用指令未執行標誌) 變為ON。
- 傳送目標沒有空域容量的情況或(s1)+0的位元3 (檔案傳送時的暫時檔案建立設定) 為0 (建立) 時，向傳送目標傳送的檔案與暫時檔案 (與傳送檔案相同容量) 中沒有空域容量的情況下，將會異常完成。
- 檔案傳送執行中即使將CPU模組的狀態RUN→STOP，檔案還是會繼續傳送直到完成。
- 過程CPU (二重化模式) 的情況下，透過A系統/B系統各自的IP地址進行通訊。

注意事項

- 檔案傳送中發生電纜斷開或CPU模組的電源斷開・重設的情況下，應根據需要刪除FTP伺服器中剩餘的不需要的檔案(暫時檔案及不完全的檔案)後再次執行。
- 關於刪除傳送完成後的傳送源檔案，在(s1)+0的位元2(傳送完成檔案刪除設定)設定為1(刪除)的情況下，應注意下述內容。

項目	內容
傳送CPU模組的檔案的情況	即使有了CPU模組的動作中需要的檔案，檔案也會從CPU模組刪除。不保證刪除的動作。
將傳送檔案進行通配符指定後傳送的情況	預期外的必要檔案有可能被刪除。

- 透過檔案密碼功能限制檔案的存取時，SP.FTPPUT指令將會異常完成。但是，將(s1)+0的位元2(傳送完成檔案刪除設定)設定為0(不刪除)，檔案密碼的設定為“禁止寫入”的情況下可以執行SP.FTPPUT指令。
- 將(s1)+0的位元3(檔案傳送時的暫時檔案建立設定)設定為0(進行建立)的情況下，為了建立傳送目標中12字元的暫時檔案(FTPCLI_I.TMP)，根據傳送目標檔案路徑的路徑長度，不應超過傳送目標FTP伺服器的支援的最大路徑長度。
- 應勿使用不被支援的字元。(s2)、(s3)中指定的檔案名或檔案名中包含不支援的字元時，該字元將會轉換為“_”。通配符指定時，傳送源CPU模組中存在的檔案名中包含不支援的字元時，該字元將會轉換為“_”。因此，在檔案傳送中，與轉換為“_”之後的檔案名或被指定的資料夾名的情況下作同等的處理。
- 過程CPU(二重化模式)的情況下，在檔案傳送中的系統切換後，有可能出現透過FTP伺服器與舊控制系統的連接不被切斷，在新控制系統中無法與FTP伺服器連接的情況。在此情況下，應再次執行SP.FTPPUT指令，直到可以連接FTP伺服器為止。

出錯

出錯代碼(SD0)	內容
2820H	(s2)、(s3)中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在0000H時。
3405H	(s2)中指定的元件中儲存的字元串超過255字元時。
	除去(s2)中指定的檔案名的路徑(不包含檔案的分割字元)超過246字元時。
	(s2)中指定的檔案名(除去驅動器、資料夾路徑)與(s3)中指定的字元串的合計超過255字元時。
	(s2)中指定的驅動器No.在可以傳送範圍外時。
3426H	(s2)中未指定檔案名時。
	(s2)中指定的檔案名為無法傳送檔案時。
	(s2)的驅動器No.的分割字元為“:\”或“:/”之外的字元時。
	(s2)中指定的檔案名(到點號為止)或副檔名中至少指定了2個“*”時。
	(s2)中指定的檔案名(到點號為止)或副檔名中混用了“*”與“?”時。
(s3)中包含通配符指定字元(“*”、“?”)時。	
3430H	在未設定FTP客戶端設定的參數的狀態下執行SP.FTPPUT指令時。

獲取FTP客戶端檔案

SP. FTPGET



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(除了R00CPU、R01CPU、R02CPU)、RnENCPU可以在韌體版本為“22”及其以後的版本使用。工程工具可以在版本為“1.025B”及其以後的版本使用。
- RnPCPU可以在韌體版本為“13”及其以後的版本使用。工程工具可以在版本為“1.040S”及其以後的版本使用。

在 (s3) 中指定的CPU模組的資料夾路徑中獲取 (s2) 中指定的FTP伺服器的檔案。

梯形圖	ST
	ENO:=SP_FTPGET (EN, U, s1, s2, s3, d);

FBD/LD

■執行條件

指令	執行條件
SP. FTPGET	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U)	虛擬	—	字元串	ANYSTRING_SINGLE
(s1)	儲存控制資料的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 4)
(s2)	FTP伺服器中儲存的檔案名(傳送源)*1	—	Unicode字元串*2	ANYSTRING_DOUBLE
(s3)	CPU模組的資料夾路徑(傳送目標)*1	—	Unicode字元串*2	ANYSTRING_DOUBLE
(d)	透過指令完成讓1個掃描變為ON的元件 異常完成時(d)+1也將變為ON。	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 Unicode字元串或儲存了Unicode字元串的起始元件。

*2 資料類型為Unicode字元串時僅可使用半形英文數字、符號、假名字元、全形字元(移位JIS代碼)。不支援的字元作為“_”處理。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H) G□	Z	LT、 LST、LC	LZ		K、H	E	\$	
(U)	—	—	○	—	—	—	—	○	—	—	○	○
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(s3)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	○	—	○*1	—	—	—	—	○	—	—	—	—

*1 不能使用T、ST、C、FD。

■控制資料

操作數：(s1)																
元件	項目	內容	設定範圍	設定方												
+0	使用用途設定區域	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">b15</td> <td style="width: 40%;">...</td> <td style="width: 10%;">b3</td> <td style="width: 10%;">b2</td> <td style="width: 10%;">b1</td> <td style="width: 10%;">b0</td> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td style="text-align: center;">(2)</td> <td style="text-align: center;">(1)</td> <td></td> <td style="text-align: center;">0</td> </tr> </table> <p>(1) 傳送完成檔案刪除設定 (位元2) 傳送後指定傳送完成的檔案的處理。 • 0: 不刪除 • 1: 刪除</p> <p>(2) 檔案傳送時的暫時檔案建立設定 (位元3) 指定建立/不建立檔案傳送時的暫時檔案 (FTPCLI_I.TMP)。 透過在“0: 建立”中設定，在檔案傳送中 (檔案改寫中) 即使發生電纜斷開或電源斷開等，傳送目標檔案也不會變為不正確的狀態。 • 0: 建立 • 1: 不建立</p>	b15	...	b3	b2	b1	b0	0		(2)	(1)		0	如左所示	用戶
b15	...	b3	b2	b1	b0											
0		(2)	(1)		0											
+1	完成狀態	指令完成時儲存完成狀態。 • 0000H: 正常完成 • 0000H以外: 異常完成 (出錯代碼)	—	系統												
+2	處理檔案總數	儲存透過SP.FTPGET指令處理的檔案的總數。	—	系統												
+3	處理完成檔案數	儲存處理完成的檔案數。	—	系統												

功能

- 在 (s3) 中指定的 CPU 模組的資料夾路徑中獲取 (s2) 中指定的 FTP 伺服器的檔案。執行 SP.FTPGET 指令時，連接“模組參數”的“FTP 客戶端設定”中指定的 FTP 伺服器，並在獲取檔案後斷開與 FTP 伺服器的連接。關於“FTP 客戶端設定”的詳細內容，請參閱下述手冊。

📖 MELSEC iQ-R 乙太網路用戶手冊 (應用篇)

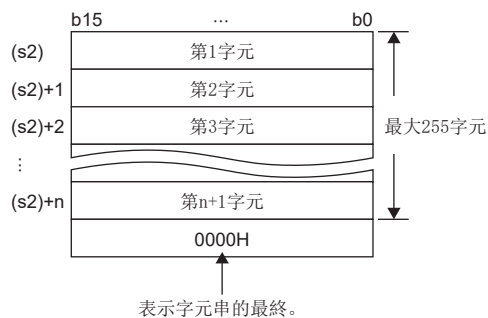
- 透過 SP.FTPGET 指令處理的檔案的總數儲存於 (s1)+2 的“處理檔案總數”中、處理完成的檔案數儲存於 (s1)+3 的“處理完成檔案數”中。
- 在 (s2) 中以 Unicode 字元串形式，指定 FTP 伺服器的傳送源的檔案儲存的資料夾路徑、檔案名 (包含副檔名)。檔案路徑長度的最大值為 255 字元。透過 FTP 伺服器的主目錄的相對路徑指定檔案路徑。資料夾路徑、檔案名的分割字元中使用半形的“\”或“/”。*1



(1) 資料夾路徑、檔案的分割字元使用半形的“\”或“/”。*1

(2) 指定的檔案名中包含副檔名。

*1 根據 FTP 伺服器，有無法將“\”作為分隔符號使用的情況。



- (s2)中指定的檔案名或副檔名中，可以使用通配符指定(*、?)。

符號	內容
*	以從指定為“*”位置開始的任意的字元串(包含沒有的情況)的全部的檔案為對象。
?	以從指定為“?”位置開始的任意的字元(不包含沒有的情況)的全部的檔案為對象。(可以使用多個“?”。)

點號不可以使用通配符指定。

下述指定方法會出錯。

- 檔案名(到點號為止)或副檔名中至少使用2個“*”。(例: “*abc*.txt”)
- 檔案名(到點號為止)或副檔名中混用“*”與“?”。(例: “*ab?.txt”)

透過通配符指定的可以傳送的檔案數是由指定的各檔案的檔案名容量的合計所決定。可以傳送的檔案數及個檔案名容量滿足下述公式的情況下可以傳送。不滿足下述公式的情況下執行檔案傳送功能指令會發生異常完成。

$$(F_i + NM) + 1 < 65536 \text{ [字節]}$$

N: 通配符指定中一致的檔案的總數
F_i: 通配符指定中一致的檔案名的合計容量
M: 指定資訊容量(固定值: 6字節)

- 執行SP. FTPGET指令中，途中的1檔案的獲取發生出錯的情況下會在發生出錯的同時中止，剩餘的檔案的傳送將無法執行。
- 不能傳送的檔案即使符合通配符指定的條件也無法作為傳送的對象。
- 包含檔案名、副檔名的檔案路徑超過255字元的情況下，即使符合通配符指定的條件也無法作為傳送的對象。
- (s2)中只指定了檔案名的情況下，直接存取FTP伺服器的主目錄。在僅指定檔案名的情況下，應從分割字元開始指定。詳細內容應符合FTP伺服器的規格。
- 在(s3)中，以Unicode字元串的形式指定CPU模組的傳送目標的資料夾路徑。資料夾路徑的分割字元使用半形的“\”或“/”。資料夾路徑的最大長度為247字元(末尾的分割字元被省略的情況下最大為246字元)。但是，資料夾路徑(包含末尾的分割字元)與(s2)的檔案名部分的合計，應指定為不超過CPU模組中支援的最大路徑長度(255字元)。雖然末尾的分割字元可以省略，但省略的情況下，末尾視為設定了“\”。此外，不存在資料夾路徑的情況下，指令執行時在系統中自動建立、進行檔案傳送。



- (1) 可以指定的驅動器No. 為2、4。只有R00CPU為4。(MELSEC iQ-R CPU模組用戶手冊(應用篇))
- (2) 驅動器No. 的分割字元使用“:\”或“:/”。
- (3) 資料夾的分割字元使用半形的“\”或“/”。
- (4) 末尾的分割字元可以省略。

- 傳送目標中存在同名的檔案的情況，應對同名的檔案進行覆蓋。
- 可以獲取的最大檔案容量為1檔案4G字節。
- 對於SP. FTPGET指令的執行及正常/異常完成，可以透過設定資料中指定的完成元件(d)、完成時的狀態顯示元件(d)+1進行確認。

• 完成元件(d)

SP. FTPGET指令透過已完成的掃描的END處理變為ON、透過下一個END處理變為OFF。

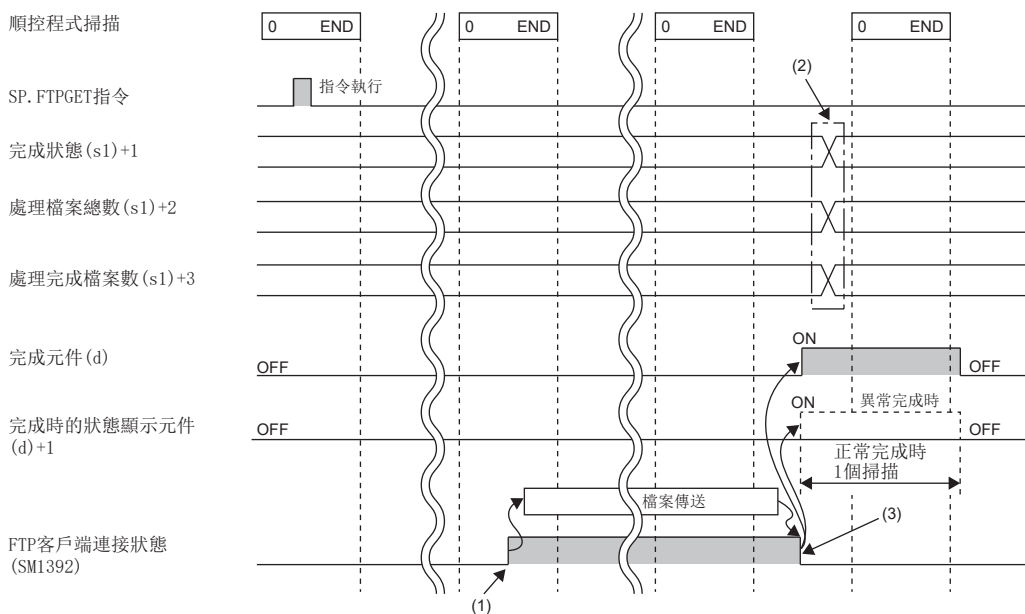
- 完成時的狀態顯示元件(d)+1

根據SP. FTPGET指令已完成時的狀態而變為ON/OFF。

正常完成時：保持為OFF不變。

異常完成時：SP. FTPGET指令透過已完成的掃描的END處理變為ON、透過下一個END處理變為OFF。異常完成時，(s1)+1的完成狀態中將儲存出錯代碼。

- SP.FTPGET指令的執行時機如下所示。



(1) FTP伺服器連接後的END處理中ON。

(2) 指令完成時儲存值。

(3) 全部檔案的傳送完成時變為OFF。

- FTP伺服器連接中SM1392 (FTP客戶端連接狀態) 變為ON、斷開後SM1392變為OFF。
- SP.FTPGET指令執行中，SM753 (檔案存取中) 將變為ON。SM753為ON中，不能執行SP.FTPGET指令。執行的情況下將變為無處理。
- 在SP.FTPPUT指令或SP.FTPGET指令執行中執行SP.FTPGET指令的情況下，在先執行中的指令完成之前，後一個指令將被忽略而不執行。忽略SP.FTPGET指令時，SM699 (專用指令未執行標誌) 變為ON。
- 傳送目標沒有空域容量的情況或 (s1)+0的位元3 (檔案傳送時的暫時檔案建立設定) 為0 (建立) 時，向傳送目標傳送的檔案與暫時檔案 (與傳送檔案相同容量) 中沒有空域容量的情況下，將會異常完成。
- 檔案傳送執行中即使將CPU模組的狀態RUN→STOP，檔案還是會繼續傳送直到完成。
- 過程CPU (二重化模式) 的情況下，透過A系統/B系統各自的IP地址進行通訊。

注意事項

- 檔案傳送中發生電纜斷開或CPU模組的電源斷開・重設的情況下，應再次執行。發生電纜斷開或CPU模組的電源斷開・重設的情況下，不需要的檔案透過下述被刪除。

不需要的檔案	刪除動作
暫時檔案 (FTPCLI_I.TMP)	FTP客戶端中剩餘的暫時檔案在與下一次的同一資料夾相對的檔案傳送開始時自動被刪除。
不完全的傳送對象檔案	FTP客戶端中剩餘的不完全的傳送對象檔案，在與下一次的同一檔案相對的檔案傳送時被覆蓋。

- 對於傳送源的檔案，傳送目標中存在透過檔案密碼功能中存取限制的同名的檔案時，SP.FTPGET指令將異常完成。
- 將 (s1)+0的位元3 (檔案傳送時的暫時檔案建立設定) 設定為0 (建立) 的情況下，應將CPU模組的資料夾路徑設定為最大長度243字元 (省略末尾的分割字元時最多242字元)。此外，由於建立傳送目標中12字元的暫時檔案 (FTPCLI_I.TMP)，根據傳送目標資料夾路徑的路徑長度不應超過CPU模組中支援的最大路徑長度 (255字元)。
- 應勿使用不被支援的字元。(s2)、(s3) 中指定的檔案名或檔案名中包含不支援的字元時，該字元將會轉換為“_”。因此，在檔案傳送中，與轉換為“_”之後的檔案名或被指定的資料夾名的情況下作同等的處理。此外，通配符指定時，傳送源FTP伺服器中存在的檔案名中包含不支援的字元的情況下，無法正確的獲取檔案。(關於如何處理不支援的字元的部分，應根據FTP伺服器的規格進行處理)
- 過程CPU (二重化模式) 的情況下，在檔案傳送中的系統切換後，有可能出現透過FTP伺服器與舊控制系統的連接不被切斷，在新控制系統中無法與FTP伺服器連接的情況。在此情況下，應再次執行SP.FTPGET指令，直到可以連接FTP伺服器為止。

出錯

出錯代碼 (SD0)	內容
2820H	(s2)、(s3) 中指定的元件編號及其以後，元件/標籤記憶體的各設定區域之間不存在0000H時。
3405H	(s2) 中指定的元件中儲存的字元串超過255字元時。
	除去 (s2) 中指定的檔案名的路徑 (不包含檔案的分割字元) 超過246字元時。
	(s3) 中指定的字元串與 (s2) 中指定的字元串的檔案名部分合計超過255字元時。
	(s3) 中指定的驅動器No. 在可以傳送範圍外時。
3426H	(s2) 中未指定檔案名時。
	(s2) 中指定的檔案名為無法傳送檔案時。
	(s3) 的驅動器No. 的分割字元使用 “:\” 或 “:/” 之外的字元時。
	(s2) 中指定的檔案名 (到點號為止) 或副檔名中至少指定了2個 “*” 時。
	(s2) 中指定的檔案名 (到點號為止) 或副檔名中混用了 “*” 與 “?” 時。
	(s3) 中包含通配符指定字元 (“*”、“?”) 時。
3430H	在未設定FTP客戶端設定的參數的狀態下執行SP.FTPGET指令時。

25 PID運算指令

PID運算指令及PID控制指令使用區分

為進行PID控制的指令有以下種類。

種類	使用	參閱目標(概要)	參閱目標(指令詳細)
PID運算指令	實施使用自動調諧控制的情況下	1192頁 概要	1203頁 PID運算指令
PID控制指令	實施與MELSEC-Q系列和MELSEC-L系列相同的PID控制的情況下	1207頁 概要	1214頁 PID控制指令(不完全微分), 1226頁 PID控制指令(完全微分)

PID運算指令和PID控制指令的比較如下所示。

項目	PID運算指令	PID控制指令
PID運算方式	不完全微分	不完全微分/完全微分
採樣週期/採樣時間	1~32767ms	10~60000ms
控制環路數	1環路/指令	最大32環路
自動調諧	有(限位週期法/步響應法)	無

25.1 概要

使用PID運算指令、說明PID控制的概要。

PID運算指令

PID是運算指令因接近目標值(SV)，從測定值(PV)組合P動作(比例動作)、I動作(積分動作)、D動作(微分動作)運算輸出值(MV)的指令。

■報警輸出功能

關於輸入(測定值)變化量和輸出(值)變化量可以ON報警輸出。

■輸出值的上下限設定

設定輸出的上下限值可以抑制PID控制積分項的增大。

■自動調諧功能

比例增益(K_p)、積分時間(T_I)、微分時間(T_D)可以自動設定。可以選擇限位週期法或步響應法。

■PID運算指令的運算方式

在速度型・測定值微分型中進行運算。

PID運算指令的基本運算公式[參考]

本指令透過速度型・測定值微分型的運算公式進行PID運算。

根據 (s3) +1 (動作設定 (ACT)) 的位0的內容執行正動作或逆動作的運算公式。

運算需要的各值透過 (s3) 及其以後指定的控制用資料的內容運算。

• 運算公式

正動作/逆動作 ((s3)+1位0)	運算公式
正動作 (OFF)	$\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} EV_n + D_n \}$ $EV_n = PV_{nf} - SV$ $D_n = \frac{T_D}{T_S + K_D \cdot T_D} (-2PV_{nf-1} + PV_{nf} + PV_{nf-2}) + \frac{K_D \cdot T_D}{T_S + K_D \cdot T_D} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$
逆動作 (ON)	$\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} EV_n + D_n \}$ $EV_n = SV - PV_{nf}$ $D_n = \frac{T_D}{T_S + K_D \cdot T_D} (2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{K_D \cdot T_D}{T_S + K_D \cdot T_D} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$

運算公式中的符號含義如下所示。

符號	含義
EV_n	本次樣本時的偏差
EV_{n-1}	1週期前的偏差
SV	目標值
PV_{nf}	本次樣本時的測定值 (濾波器後)
PV_{nf-1}	1週期前的測定值 (濾波器後)
PV_{nf-2}	2週期前的測定值 (濾波器後)
ΔMV	輸出變化量
MV_n	本次的操作量
D_n	本次的微分項
D_{n-1}	1週期前的微分項
K_p	比例增益
T_S	採樣週期
T_I	積分常數
T_D	微分常數
K_D	微分增益

本次樣本時的測定值 (濾波器後) (PV_{nf}) 是將讀取測定值透過以下方式運算後的值。未設定輸入資料的濾波器係數的情況下，將變為與輸入資料的測定值 (PV) 相同的值。

$$PV_{nf} = PV_n + L (PV_{nf-1} - PV_n)$$

PV_{nf} : 本次樣本時的測定值、L: 濾波器係數、 PV_{nf-1} : 1週期前的測定值 (濾波器後)

控制用資料

說明關於PID運算指令的控制用資料的內容。

採樣時間：(s3)

設定範圍：1~32767[ms]

設定進行PID運算的週期(ms)。

- 自動調諧(限位週期法)時

按(可程式控制器的運算週期) < (採樣時間) 設定。

- 自動調諧(步響應法)時

設定為1000ms及其以上。

■關於最大誤差

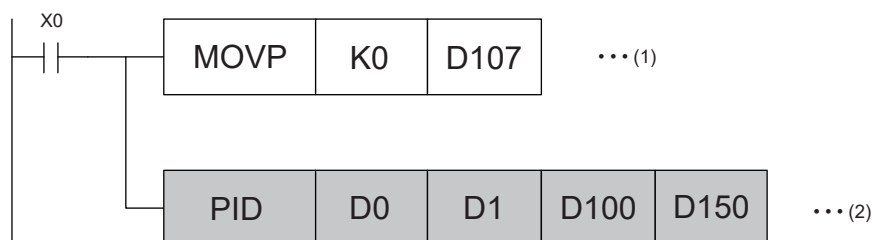
採樣時間(T_s)的最大誤差為 $-(1\text{運算週期}+1\text{ms}) \sim +(1\text{運算週期})$ 。

- 採樣時間(T_s)為小的值的情況下

有時最大誤差的變動會成為問題。應在執行設定恆定掃描或在定時器中斷常規內進行程式。

- 比可程式控制器的1運算週期短的情況下

雖然發生PID運算出錯(11A6H)，但將採樣時間(T_s)=作為運算週期執行PID運算。這種情況下，應在定時器中斷內使用PID運算指令、在PID運算指令的執行之前清除(s3)+7之後使用。



(1) 重設(s3)+7(在首次中斷常規執行時，透過脈衝化指令清除內部處理用寄存器。)

(2) 執行PID運算。

動作設定：(s3)+1

■動作方向(正動作/逆動作)：(s3)+1 位元0

設定範圍：OFF=正動作/ON=逆動作

選擇正動作或逆動作。

- 自動調諧(限位週期法)的情況下

自動調諧需要設定想要的正動作或逆動作的PID控制方向。

- 自動調諧(步響應法)的情況下

自動調諧即使執行正動作或逆動作中之一，完成時自動進行設定。

[正動作((s3)+1 位元0=OFF)]

[逆動作((s3)+1 位元0=ON)]

■報警設定(輸入變化量、輸出變化量)：(s3)+1 位元1、2

設定範圍：OFF=無報警/ON=報警有效

可以進行輸入變化量、輸出變化量的檢查。檢查結果可以透過(s3)+24確認。(☞ 1199頁 報警輸出的標誌動作：(s3)+24)

- 輸入變化量((s3)+1 位1)

使用輸入變化量報警的情況下，將下述的位元設為ON，需要設定想要檢查的值。

設定項目			內容	設定範圍
動作設定 (ACT)	(s3)+1	位元1	輸入變化量報警	ON: 使用 OFF: 不使用
輸入變化量報警設定值	(s3)+20		輸入變化量(增加側)報警設定值	0~32767
	(s3)+21		輸入變化量(減少側)報警設定值	

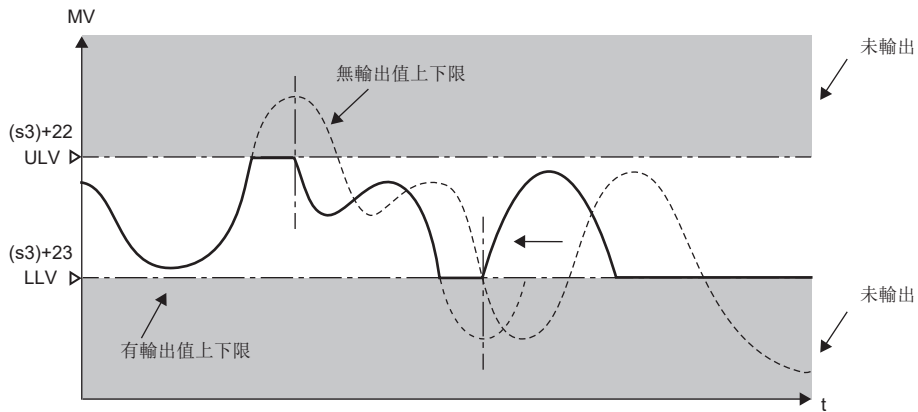
- 輸出變化量((s3)+1 位元2)

使用輸出變化量報警的情況下，將下述的位元設為ON，需要設定想要檢查的值。

設定項目			內容	設定範圍
動作設定 (ACT)	(s3)+1	位元2	輸出變化量報警	ON: 使用 OFF: 不使用
		位元5	輸出值上下限設定	必須設定為OFF
輸出變化量報警設定值	(s3)+22		輸出變化量(增加側)報警設定值	0~32767
	(s3)+23		輸出變化量(減少側)報警設定值	

■輸出值上下限設定：(s3)+1 位元5

根據輸出值上下限設定的輸出值如下所示。



MV: 輸出值
 ULV: 輸出上限值
 LLV: 輸出下限值
 t: 時間

輸出值上下限設定有抑制PID控制的積分項增大的效果。使用本功能的情況下，必須將(s3)+1的位元2設定為OFF。

設定項目			內容	設定範圍
動作設定 (ACT)	(s3)+1	位元2	輸出變化量報警	必須設定為OFF
		位元5	輸出值上下限設定	ON: 使用 OFF: 不使用

輸入濾波器：(s3)+2

設定範圍：0~99[%]

PID控制：比例動作、積分動作、微分動作

輸入濾波器(α)是透過測定值(PV)的噪聲減低變動的軟體濾波器。由於設定了與控制對象的特性和其噪聲等級相適應的濾波器的輸入濾波器(α)，因此可以抑制噪聲的影響。

- 若小，作為濾波器的效果降低。
- 若過大，輸入的響應變差。

因輸入濾波器(α)對目標值(SV)起作用，因此影響比例動作、積分動作、微分動作。



比例增益：(s3)+3

設定範圍：1~32767[%]

PID控制：比例動作

輸出值(MV)增加在比例動作偏差(目標值(SV)和測定值(PV)的差)成正比。此百分比稱為比例增益(K_p)，透過以下關係式表示。

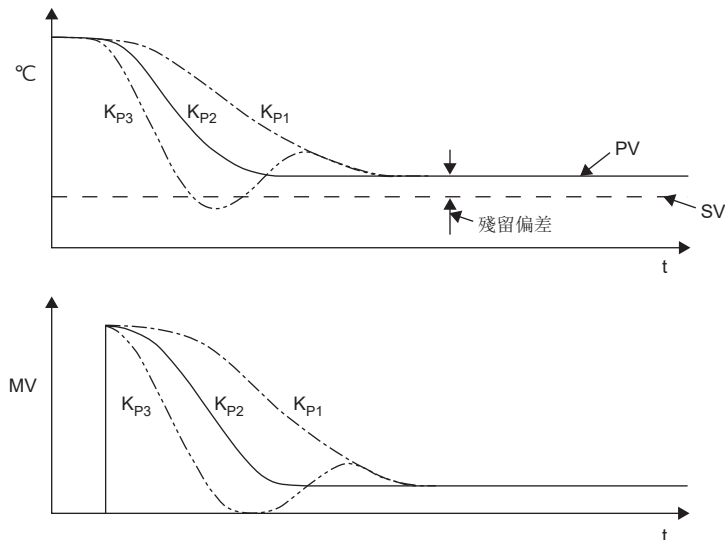
輸出值(MV)=比例增益(K_p)×偏差(EV)

此外，比例增益(K_p)的倒數稱為比例帶。

隨著比例增益(K_p)變大，測定值(PV)越接近目標值(SV)動作會變越大。

例

在冷卻(正動作)的情況下的比例動作(P動作)



比例增益(K_p): $K_{P3} > K_{P2} > K_{P1}$

°C: 溫度

SV: 目標值

PV: 測定值

MV: 輸出

t: 時間

積分時間：(s3)+4

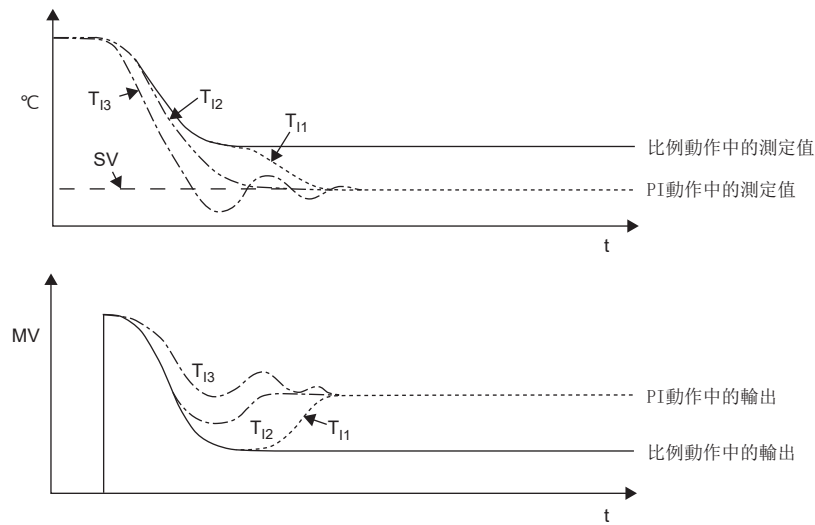
設定範圍：0~32767[×100ms] (0的情況下，作為 ∞ 處理。(無積分))

PID控制：積分動作

在積分動作發生偏差之後，積分動作的輸出到比例動作的輸出為止的時間稱為積分時間，用 T_I 來表示。
若 T_I 變小，積分動作變強。

例

在冷卻(正動作)的情況下的PI動作



積分時間(T_I): $0 < T_{I3} < T_{I2} < T_{I1}$

°C: 溫度

SV: 目標值

MV: 輸出

t: 時間

微分增益: $(s^3)+5$

設定範圍: 0~200[%]

PID控制：微分動作

根據微分動作於輸出施加濾波器。微分增益(K_D)僅影響微分動作。

- 若微分增益(K_D)變小，對於由干擾等對測定值(PV)的變化將瞬間響應。
- 若微分增益(K_D)變大，對於由干擾等對測定值(PV)的變化將花費長時間響應。

要點

應在最初設定微分增益(K_D)為0、在輸入濾波器(α)調整。輸出的變化對干擾響應過好時，應增大微分增益(K_D)的值。

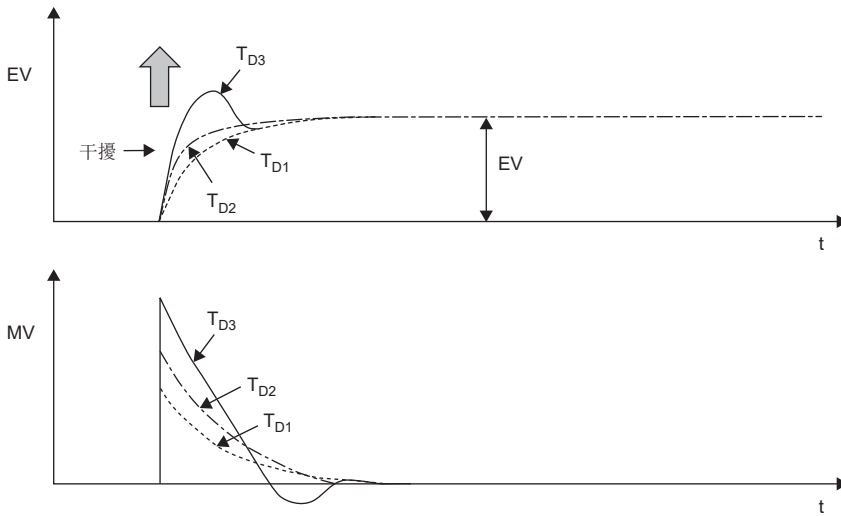
微分時間 (T_D): (s3)+6

設定範圍: 0~32767 [$\times 10\text{ms}$] (0的情況下, 為無微分。)

PID控制: 微分動作

針對測定值 (PV) 的干擾等造成的變動, 為了將因敏感而造成的變動抑制在最小限而使用。

- 若微分時間 (T_D) 變大, 防止由干擾等造成控制對象的大變動之動作給予加強。



微分時間 (T_D): $T_{D3} > T_{D2} > T_{D1}$

EV: 偏差

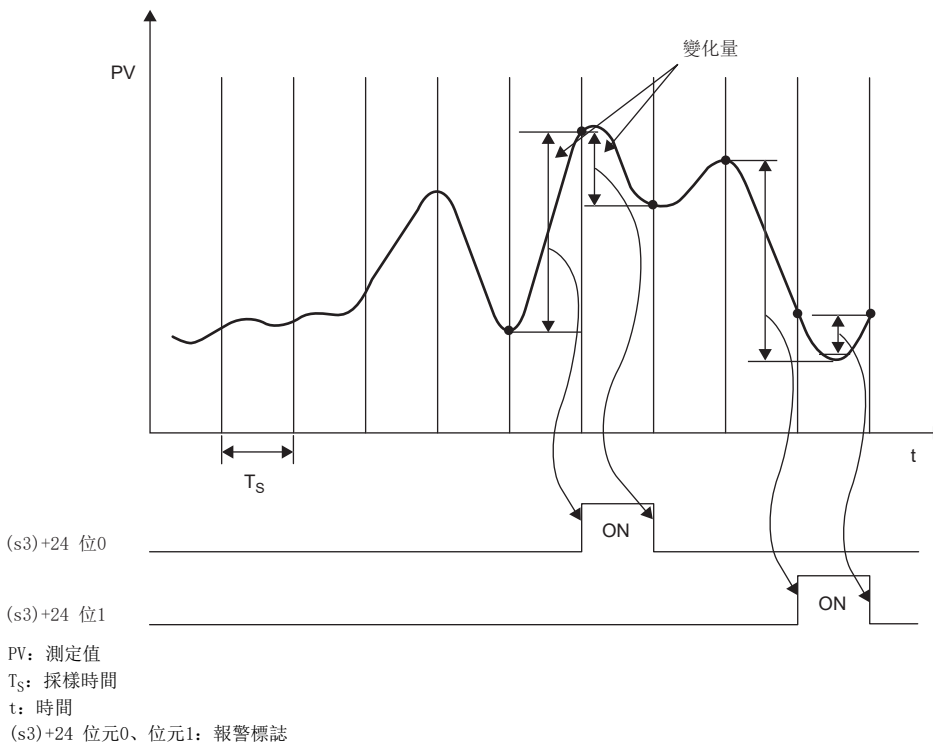
MV: 輸出

t: 時間

報警輸出的標誌動作: (s3)+24

超過設定的輸入輸出變化量時, 作為報警標誌 (s3)+24 的各位元在PID運算指令執行之後ON。

- 輸入變化量報警 ((s3)+1 位元1為ON) 的情況下



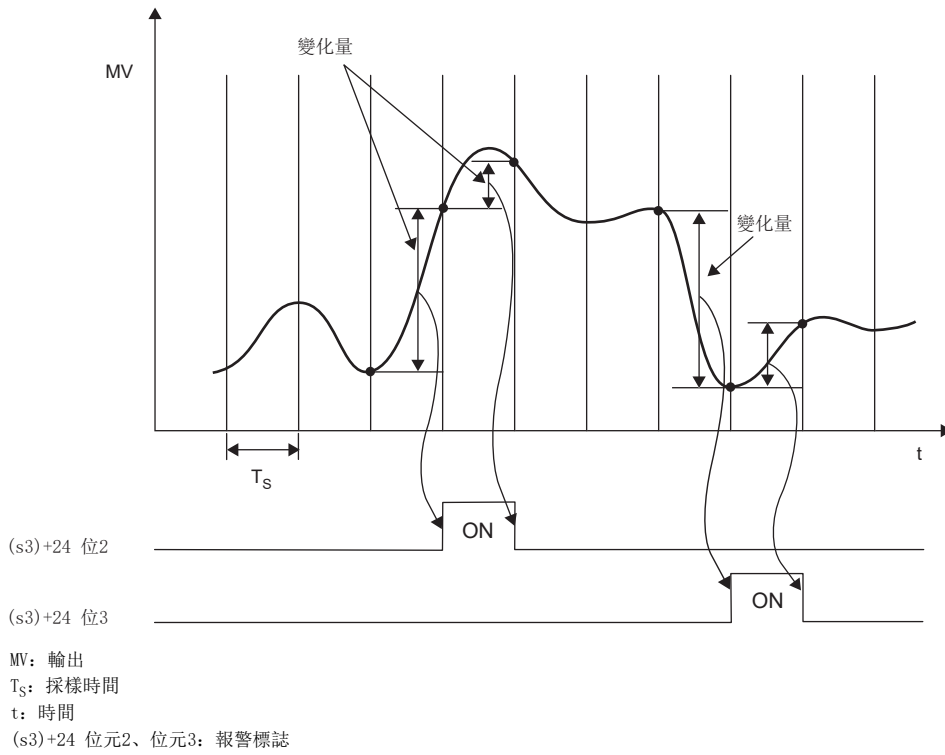
PV: 測定值

T_s : 採樣時間

t: 時間

(s3)+24 位元0、位元1: 報警標誌

• 輸出變化量報警((s3)+1的位元2為ON)的情況下



自動調諧

為了在PID控制得到良好的控制結果，因此要求出與控制對象相吻合的各常數(控制用資料)的最適合值。自動調諧功能為具有為了最佳PID控制而自動設定重要的常數的功能，常數包含比例增益、積分時間、微分時間。

PID運算指令的自動調諧功能有限位週期法和步響應法2種方法。說明各自的順序。

要點

自動調諧應從系統穩定狀態開始。若從不穩定的狀態開始，會有不能正確自動調諧的情況。

限位週期法

限位週期法的概要請參閱下述章節。

☞ 1550頁 限位週期法的概要

■透過自動調諧(限位週期法)設定的控制用資料

- 比例增益(K_p): (s3)+3
- 積分時間(T_I): (s3)+4
- 微分時間(T_D): (s3)+6

■自動調諧步驟

1. 正動作/逆動作的設定

在(s3)+1(動作設定(ACT))的位元0中，設定正動作或逆動作。

2. 限位週期法的選擇

將(s3)+1(動作設定(ACT))的位元6設為ON、選擇限位週期法。(OFF時透過步響應法動作。)

3. 自動調諧執行標誌的ON

將(s3)+1(動作設定(ACT))的位4設為ON。

4. 輸入濾波器的設定

設定(s3)+2(輸入濾波器(α))。

5. 採樣時間的設定

設定(s3)(採樣時間(T_s))。

6. 輸出值上限(ULV)的設定

在(s3)+26(輸出值上限(ULV))設定輸出值(MV)的最大輸出值。

7. 輸出值下限(LLV)的設定

在(s3)+27(輸出值下限(LLV))設定輸出值(MV)的最小輸出值。

8. PV值門檻(滯後)寬度(SHPV)的設定

設定(s3)+25(PV值門檻(滯後)寬度(SHPV))。

9. 目標值(SV)的設定

在PID運算指令的(s1)設定目標值(SV)。

10. 自動調諧的開始

若PID運算指令的啟動觸點為ON時，將透過測定值(PV)自動調諧開始。

在自動調諧完成時刻，(s3)+1(動作設定(ACT))的自動調諧標誌(位4、位6)變為OFF。

步響應法

步響應法的概要，請參閱下述章節。

☞ 1551頁 步響應法的概要

■透過自動調諧(步響應法)設定的控制用資料

- 動作方向(正動作/逆動作)：(s3)+1 位元0
- 比例增益(K_p)：(s3)+3
- 積分時間(T_I)：(s3)+4
- 微分時間(T_D)：(s3)+6

■自動調諧步驟

1. 自動調諧用輸出值的設定

在輸出值(MV)傳送自動調諧用輸出值。

自動調諧用輸出值，應設定為輸出設備可輸出的最大值 $\times 0.5\sim 1$ 的值。

2. 自動調諧未設定資料的設定

根據系統設定自動調諧未設定的下述項目。

設定項目	備注	
(s1)	目標值(SV)	設定和測定值(PV)的差為150及其以上。*1
(s3)	採樣時間(T_S)	設定為1000ms及其以上。*2
(s3)+2	輸入濾波器(α)	—
(s3)+5	微分增益(K_D)	設定輸入濾波器的情況下，微分增益通常設定為“0”。
其他		根據需要設定。

*1 關於目標值(SV)的設定值和測定值(PV)的差

若自動調諧開始時的測定值和目標值的差小於150，不能正確自動調諧。因此，不足150的情況下，應設定自動調諧用的目標值。自動調諧完成後，應重新設定目標值。

*2 關於採樣時間(T_S)的設定時間

自動調諧時的採樣時間(T_S)務必設為1000ms及其以上。
此外，應將此採樣週期設為比輸出變化週期更長的時間。

3. 自動調諧的開始

若(s3)+1(動作設定(ACT))的位元4為ON，則自動調諧開始。

若從自動調諧開始時的測定值到目標值的變化量為1/3以上，則自動調諧完成、(s3)+1(動作設定(ACT))的位元4自動變為OFF。

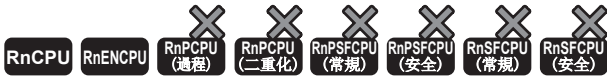
自動調諧時的注意事項

- 測定值(PV)不變化的情況下的程式對策

由於類比輸入斷線等原因造成測定值(PV)不正常變化的情況下，自動調諧將不結束。應透過導入監視從輸入值和調諧開始經過時間的順控，來發現、迴避。

25.2 PID運算指令

PID



• R00CPU、R01CPU、R02CPU沒有版本的限制。

• RnCPU、RnENCPU (除了R00CPU、R01CPU、R02CPU) 可以在韌體版本為“17”及其以後的版本使用。(工程工具可以在版本為“1.020W”及其以後的版本使用。)

以(s1)、(s2)、(s3)中設定的值進行PID運算、將每採樣時間運算結果儲存於(d)。

梯形圖	ST
	$ENO := PID(EN, s1, s2, s3, d);$

FBD/LD

■執行條件

指令	執行條件
PID	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存目標值(SV)的元件	-32768~32767	帶符號BIN16位元	ANY16
(s2)	儲存測定值(PV)的元件	-32768~32767	帶符號BIN16位元	ANY16
(s3)	儲存控制用資料的起始元件	—	帶符號BIN16位元	ANY16
(d)	儲存輸出值(MV)的元件	-32768~32767	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	○	—	—	—	○	—	—	—	—	
(s3)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	○	—	—	—	○	—	—	—	—	

功能

設定目標值(s1)、測定值(s2)、控制用資料(s3)～(s3)+6，若執行程式，將各採樣時間(s3)的運算結果(MV)儲存至輸出值(d)中。詳細內容如下所示。

☞ 1192頁 概要

■各引數的設定項目

設定項目	內容	佔用點數
(s1)	目標值(SV) 設定目標值(SV)。 ■自動調諧：限位週期法的情況下 自動調諧用的目標值和進行PID控制時的目標值不同的情況下應設定加法運算偏置值的值，在自動調諧標誌變為OFF時應設定實際的目標值。	1點
(s2)	測定值(PV) 設定PID運算的輸入值。	1點
(s3)	控制用資料*1 ■自動調諧：限位週期法的情況下 從(s3)中指定的起始元件開始佔用29點的元件。 ■自動調諧：步響應法的情況下 ① 動作設定(ACT)的設定的位元1、位元2、位元5全部為“0”以外時，從(s3)中指定的起始元件開始佔用25點的元件。 ② 動作設定(ACT)的設定的位元1、位元2、位元5全部為“0”時，從(s3)中指定的起始元件開始佔用20點的元件。	29點 ① 25點 ② 20點
(d)	輸出值(MV) ■PID控制(普通處理)的情況下 在指令執行前用戶側設定初始輸出值。指令執行後運算結果儲存。 ■自動調諧：限位週期法的情況下 在自動調諧中自動輸出輸出上限值(ULV)或輸出下限值(LLV)，在自動調諧結束後規定的輸出值(MV)將被設定。 ■自動調諧：步響應法的情況下 在指令執行前用戶側設定步輸出值。自動調諧中，輸出值(MV)在PID指令側將不變更。	1點

*1 不使用自動調諧的情況下，將與使用步響應法的情況下佔用相同的點數。

■控制用資料的設定項目

操作數：(s3)				
元件	項目	內容	備注	
+0	採樣時間(T _S)	1~32767[ms]	無法執行比運算週期短的值。	
+1	動作設定(ACT)	位元0	0: 正動作 1: 逆動作	動作方向指定
		位元1	0: 無輸入變化量報警 1: 輸入變化量報警有效	—
		位元2	0: 無輸出變化量報警 1: 輸出變化量報警有效	應不要將位元2與位元5同時ON。
		位元3	不能使用	—
		位元4	0: 自動調諧不動作 1: 自動調諧執行	—
		位元5	0: 無輸出值上下限設定 1: 輸出值上下限設定有效	應不要將位元2與位元5同時ON。
		位元6	0: 步響應法 1: 限位週期法	自動調諧模式選擇
	位元7~位元15	不能使用	—	
+2	輸入濾波器常數(α)	0~99[%]	指定0的情況下，為無輸入濾波器。	
+3	比例增益(K _P)	1~32767[%]	—	
+4	積分時間(T _I)	1~32767[×100ms]	指定0的情況下，作為∞處理。(無積分)	
+5	微分增益(K _D)	0~200[%]	指定0的情況下，為無微分增益。	
+6	微分時間(T _D)	1~32767[×10ms]	指定0的情況下，為無微分。	
+7~+19	為了在PID運算的內部處理佔有，禁止資料變更			
+20*1	輸入變化量(增加側)報警設定值	0~32767	動作方向(ACT)：(s3)+1 位元1=1時有效	
+21*1	輸入變化量(減少側)報警設定值	0~32767	動作方向(ACT)：(s3)+1 位元1=1時有效	
+22*1	輸出變化量(增加側)報警設定值	0~32767	動作方向(ACT)：(s3)+1 位元2=1、位元5=0時有效	
	輸出上限設定值	-32768~32767	動作方向(ACT)：(s3)+1 位元2=0、位元5=1時有效	
+23*1	輸出變化量(減少側)報警設定值	0~32767	動作方向(ACT)：(s3)+1 位元2=1、位元5=0時有效	
	輸出下限設定值	-32768~32767	動作方向(ACT)：(s3)+1 位元2=0、位元5=1時有效	
+24*1	報警輸出	位元0	0: 輸入變化量(增加側)未溢出 1: 輸入變化量(增加側)溢出	動作方向(ACT)：(s3)+1 位元1=1或位元2=1時有效
		位元1	0: 輸入變化量(減少側)未溢出 1: 輸入變化量(減少側)溢出	—
		位元2	0: 輸出變化量(增加側)未溢出 1: 輸出變化量(增加側)溢出	—
		位元3	0: 輸出變化量(減少側)未溢出 1: 輸出變化量(減少側)溢出	—
+25*2	PV值門檻(滯後)寬度(SHPV)	根據測定值(PV)的擺動設定	動作設定(ACT)：(s3)+1	
+26*2	輸出值上限(ULV)	輸出值(MV)的最大輸出值(ULV)設定	位元6=1(選擇限位週期法)時佔用	
+27*2	輸出值下限(LLV)	輸出值(MV)的最小輸出值(LLV)設定		
+28*2	從調諧週期結束到PID控制開始的視窗設定參數(K _W)	-5~32717[%]		

*1 在(s3)+1的動作設定(ACT)的位元1=1、位元2=1或位元5=1時佔用。

*2 (s3)+1的動作設定(ACT)的位元6=1時(限位週期法的選擇時)被佔用。

出錯

出錯代碼 (SDO)	內容
11A0H	在採樣時間 (T_S) 指定範圍外 ($T_S \leq 0$) 值時。
11A1H	在輸入濾波器常數 (α) 指定範圍外 ($\alpha < 0$ 或 $100 \leq \alpha$) 的值時。
11A2H	在比例增益 (K_P) 指定範圍外 ($K_P < 0$) 的值時。
11A3H	在積分時間 (T_I) 指定範圍外 ($T_I < 0$) 的值時。
11A4H	在微分增益 (K_D) 指定範圍外 ($K_D < 0$ 或 $201 \leq K_D$) 的值時。
11A5H	在微分時間 (T_D) 指定範圍外 ($T_D < 0$) 的值時。
11A6H	在採樣時間 (T_S) 中指定了不超過運算週期的值時。
11A7H	測定值變化量 (ΔPV) 上溢時。
11A8H	偏差 (EV) 上溢時。
11A9H	積分計算值上溢時。
11AAH	微分增益 (K_D) 的值上溢時。
11ABH	微分計算值上溢時。
11ACH	PID運算結果上溢時。
11ADH	在輸出上限設定值指定比輸出下限設定值小的值時。
11AEH	在輸入變化量報警設定值或輸出變化量報警設定值指定比0小時。
11AFH	<p>■步響應法</p> <p>自動調諧結果不良時。</p> <ul style="list-style-type: none"> 自動調諧(步響應法)開始時的偏差超過150。 自動調諧(步響應法)結束時的偏差在開始時的偏差的1/3及其以上結束。
11B0H	<p>■步響應法</p> <p>自動調諧動作方向不一致時。</p> <ul style="list-style-type: none"> 從自動調諧(步響應法)開始時的目標值和測定值的關係推測的動作方向，與輸出值的動作方向不一致。
11B1H	<p>■步響應法</p> <p>自動調諧動作不良時。</p> <ul style="list-style-type: none"> 輸入值(PV)不正常變化、自動調諧(步響應法)不能正確動作。
11B2H	<p>■限位週期法</p> <p>在自動調諧(限位週期法)用的輸出值上限(ULV)中指定了不超過輸出值下限(LLV)的值時。</p>
11B3H	<p>■限位週期法</p> <p>在自動調諧(限位週期法)用的PV值門檻(滯後)寬度(SHPV)指定範圍外(SHPV < 0)的值。</p>
11B4H	<p>■限位週期法</p> <p>透過自動調諧(限位週期法)使用的系統區域被改寫時。</p>
11B5H	<p>■限位週期法</p> <p>因超過自動調諧(限位週期法)的測定時間，τ及τ_{on}的時間無法正常獲取時。$(\tau_{on} > \tau, \tau_{on} < 0, \tau < 0)$</p>
11B6H	<p>■限位週期法</p> <p>透過自動調諧(限位週期法)計算出比例增益(K_P)上溢時。</p>
11B7H	<p>■限位週期法</p> <p>透過自動調諧(限位週期法)計算出的積分時間(T_I)為0~32767以外的結果時。</p>
11B8H	<p>■限位週期法</p> <p>透過自動調諧(限位週期法)計算出的微分時間(T_D)為0~32767以外的結果時。</p>

26 PID控制指令

進行PID控制的指令的種類如下所示。

- PID運算指令
- PID控制指令

關於各個的正確的使用方法與比較，請參閱下述內容。

☞ 1192頁 PID運算指令及PID控制指令使用區分

26.1 概要

以下介紹PID控制指令中的運算方式、PID控制步驟、其它功能有關內容。

要點

PID控制指令中，存在不完全微分與完全微分的各個指令。

不完全微分透過微分項的輸入中置入一次延遲濾波器的PID控制，在下述情況下使用時有效。

- 易受到高頻噪聲影響的控制
- 完全微分方式中有步狀變化時、無法使操作端動作獲得有效能量的情況下

完全微分是原樣不變地使用微分項輸入的PID控制。

運算方式

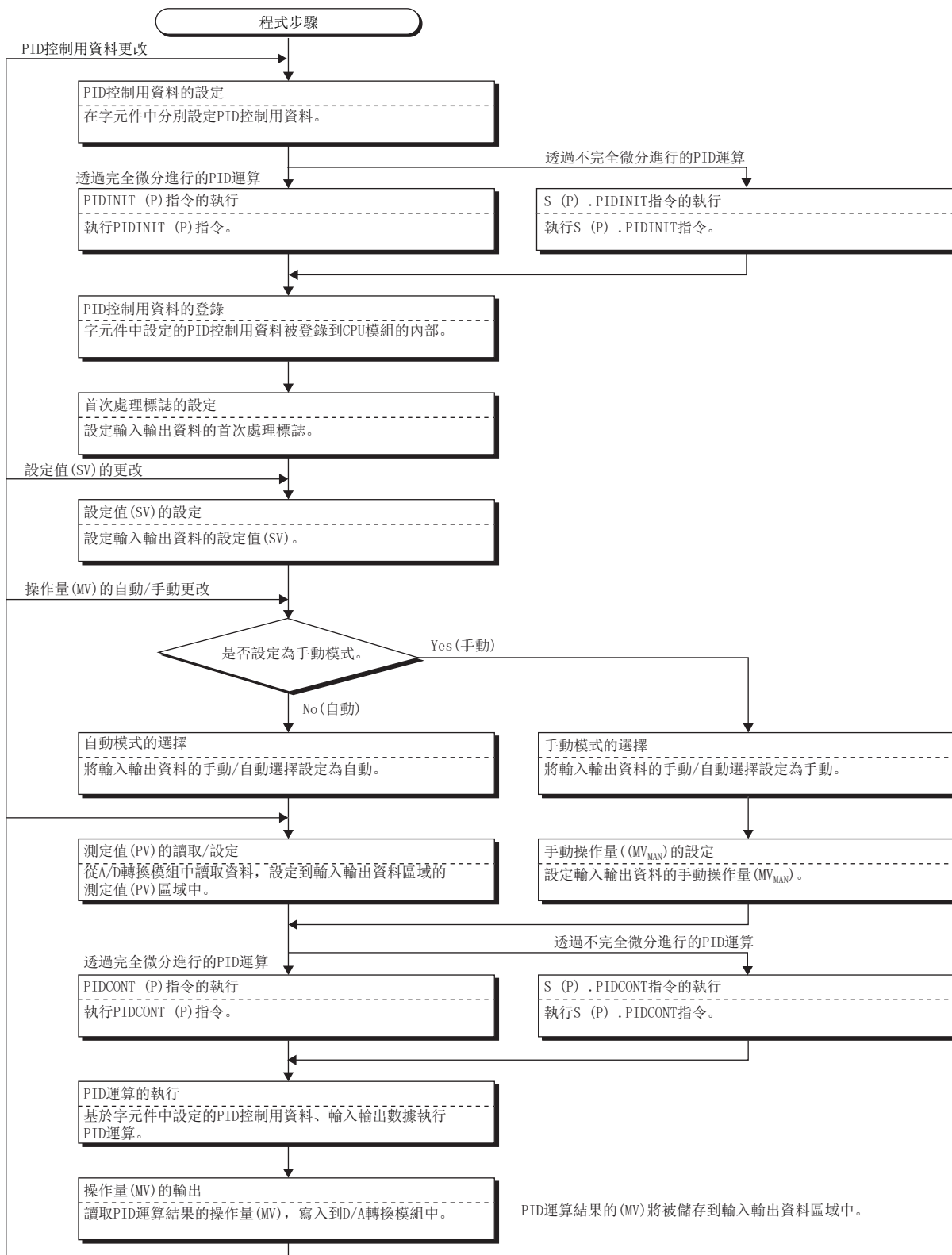
PID控制指令的PID控制中運算方式為速度型・測定值微分型。

在速度型・測定值微分型中進行以下控制。

運算方式	內容
速度型運算	是透過PID運算對操作量(MV)的變化進行計算的方式。實際的操作量為各採樣週期中計算的操作量變化的累計值。
測定值微分型	是透過PID運算在微分項中使用測定值(PV)進行運算的方式。因為微分項中不使用偏差，因此可以減少因設定值更改而引起偏差變化時微分動作導致的輸出驟變。

PID控制步驟

使用了用於進行PID控制的元件的程式步驟如下所示。



要點

- 對於PID控制用資料，可以在程式的各掃描中登錄或更改。但是，進行了登錄或更改的情況下，必須執行PIDINIT指令。如果未執行PIDINIT指令，在執行PIDCONT指令時登錄或更改的資料將不被反映。
- 透過PIDPRMW指令在每個環路中更改PID控制用資料時，無需執行PIDINIT指令。

PID控制用資料

是在PID運算中用於設定基準值的資料。透過PIDCONT指令開始PID運算之前，透過PIDINIT指令登錄到CPU模組內部。此外，可以設定為字元件的任意編號。但是，需要設定使用環路的所有資料連續的元件編號。

關於PID控制資料的分配，請參閱以下內容。

- 不完全微分：1216頁 S(P).PIDINIT
- 完全微分：1228頁 PIDINIT(P)

在PID控制用資料中，有在全部環路通用設定的資料及在各環路進行設定的資料這2種。

資料的類型	項目	內容	設定範圍	設定資料超出指定範圍時的處理	
通用設定資料	使用環路數	是執行PID運算的環路數的設定。	1~32	變為出錯狀態、全部環路不執行PID運算。	
	1個掃描的執行環路數	是在有多個達到採樣週期的環路時，1次的PID運算中能執行多少個環路的設定。	1~32		
各環路的設定資料	運算公式選擇	是進行PID運算式的正動作還是逆動作的選擇。	0: 正動作 1: 逆動作	變為出錯狀態、不執行相應環路的PID運算。	
	採樣週期(T_S)	是進行PID運算的週期的設定。	1~6000(單位: 10ms)		
	比例常數(K_P)	是PID運算的比。	1~10000(單位: 0.01)		
	積分常數(T_I)	是表示積分動作(I動作)效果大小的常數。增大積分常數時，操作量的變化將變緩。	1~32767(單位: 100ms)		
	微分常數(T_D)	是表示微分動作(D動作)效果大小的常數。增大微分常數時，控制對象的輕微變化將引起操作量較大的變化。	0~30000(單位: 10ms)		
	濾波器係數(α)	是對測定值(來自於A/D轉換模組的輸入)施加多大程度濾波器的設定。越趨近於0則濾波器效果越小。	1~100		
	操作量下限值(MVLL)	是自動模式時，透過PID運算計算的操作量的下限值的設定。操作量低於操作量下限值(MVLL)時，操作量下限值(MVLL)將變為操作量。	-50~2050* ¹ -32768~32767* ²		在“有PID極限制”中，操作量下限值(MVLL)或操作量上限值(MVHL)的值超出指定範圍時，將被轉換為下述值。 • 小於-50時，變為-50。 • 超過2050時，變為2050。
	操作量上限值(MVHL)	是自動模式時，透過PID運算計算的操作量的上限值的設定。操作量超過操作量上限值(MVHL)時，操作量上限值(MVHL)將變為操作量。	-50~2050* ¹ -32768~32767* ²		
	操作量變化率極限值(Δ MVL)	是上次與本次的操作量中變化量的限制值的設定。操作量的變化量超過限制值時，報警用元件的b1將變為1。但是，操作量的變化量無限制。即使操作量的變化量超過限制值的情況下，也仍然原樣不變地作為操作量的變化量使用，計算出操作量。	0~2000* ¹ 0~32767* ²		在“有PID極限制”中，操作量變化率極限值(Δ MVL)或測定值變化率極限值(Δ PVL)的值超出指定範圍時，將被轉換為下述的值。 • 小於0時，變為0。 • 超過2000時，變為2000。
	測定值變化率極限值(Δ PVL)	是上次與本次的測定值中變化量的限制值的設定。測定值的變化量超過限制值時，報警用元件的b0將變為1。但是，測定值的變化量無限制。即使測定值的變化量超過限制值的情況下，也仍然原樣不變地作為變化量使用，進行PID運算。	0~2000* ¹ 0~32767* ²		
微分增益(K_D)* ³	是對微分動作附加時間寬度(動作延遲)的設定。值越大則時間寬度越小，動作將趨近於完全微分。	0~32767(單位: 0.01)	變為出錯狀態、不執行相應環路的PID運算。		

*1 是在“有PID極限制”設定時。

*2 是在“無PID極限制”設定時。

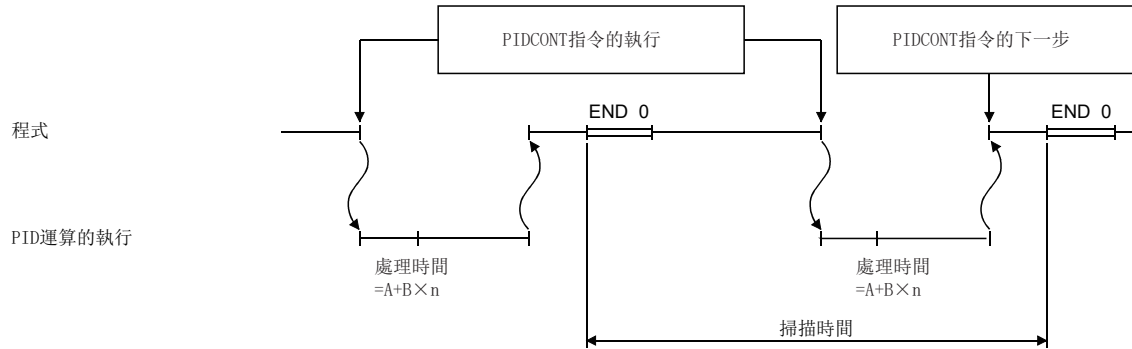
*3 僅為不完全微分時。

■使用環路數及1個掃描的執行環路數

使用環路數是使其執行PID運算的環路數。執行PIDCONT指令時進行設定的環路數的採樣週期的計測、進行達到採樣週期的環路的PID運算。

執行了PIDCONT指令情況下的處理時間與執行PID運算的環路數成正比。

1個掃描的執行環路數的含義是，在有多個達到採樣週期的環路時，1個掃描中則進行多少個環路的PID運算的設定。如果預先指定1個掃描的執行環路數，即使執行PIDCONT指令時達到採樣週期的環路數較多的情況下也只執行設定的1個掃描的執行環路數的PID運算，剩餘的將在下一個掃描中執行。



A: 採樣週期的計測等的固定時間

B: 用於進行環路的PID運算的時間

n: 環路數

要點

達到採樣週期的環路多於1個掃描的執行環路數情況下的優先順序如下所示。

- 環路No. 較小編號的優先順序越高。
- 有上一個掃描中未執行PID運算的環路及已執行過運算的環路的情況下，上一個掃描中未執行PID運算的環路將優先。

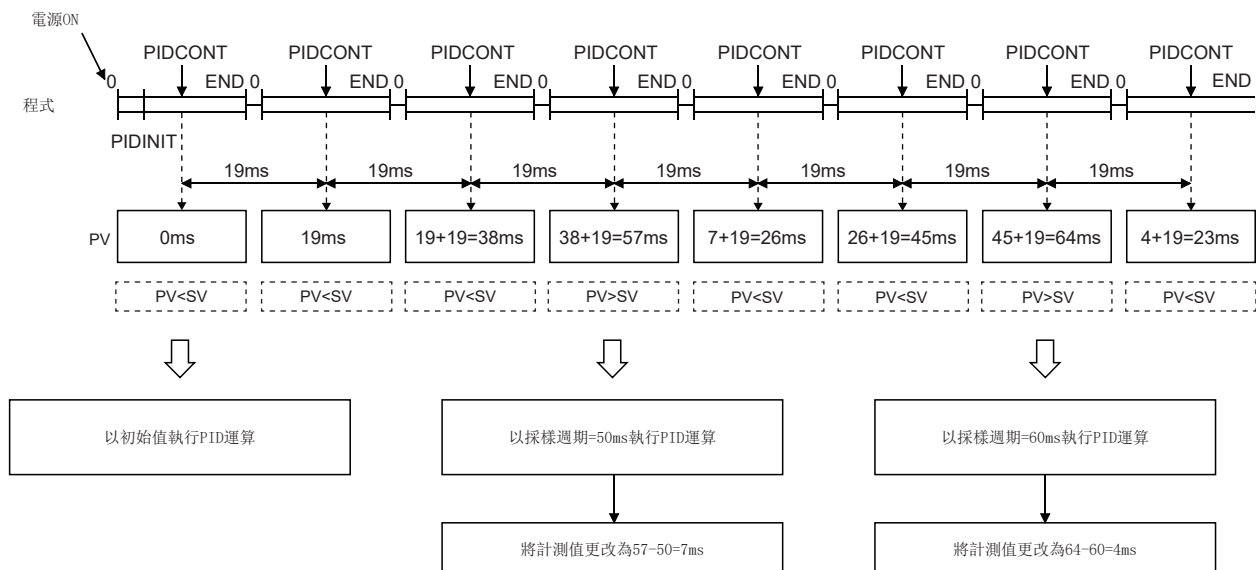
■採樣週期

採樣週期是進行PID運算的週期。每次執行PIDCONT指令時將1個掃描的計測時間加到上一個為止的計測時間中。相加後的值為設定的採樣週期及其以上時，進行相應環路的PID運算。

用於PID運算的採樣週期使用10ms單位的執行值。

例

採樣週期的設定值為50ms的情況下



PV: 計測值

SV: 設定值

因為採樣週期的計測是在執行PIDCONT指令時進行，因此採樣週期中不能設定小於程式掃描時間的值。設定了小於掃描時間的值的情況下，將以掃描時間值進行PID運算。

輸入輸出資料

輸入輸出資料是設定值(SV)、測定值(PV)等用於進行PID運算而設定的輸入資料及運算結果等的輸出資料。此外，可以指定字元元件的任意編號。但是，需要設定使用環路的所有資料連續的元件編號。

關於輸入輸出資料的分配，請參閱以下內容。

- 不完全微分：1219頁 S(P).PIDCONT
- 完全微分：1230頁 PIDCONT(P)

輸入輸出資料區域中，有分配至各環路中的以下項目及用於進行PID運算的系統所使用的工作區域。

項目	內容	設定範圍	備註
設定值	SV 是PID控制的目標值。	0~2000* ¹ -32768~32767* ²	在“有PID極限制”中，設定值(SV)的值超出指定範圍時，將被轉換為下述的值。 • 小於0時，變為0。 • 超過2000時，變為2000。
測定值	PV 是從控制對象反饋至A/D轉換模組中的資料。	-50~2050* ¹ -32768~32767* ²	在“有PID極限制”中，測定值(PV)的值超出指定範圍時，將被轉換為下述的值。 • 小於-50時，變為-50。 • 超過2050時，變為2050。
自動操作量	MV 是透過PID運算計算出的操作量。從D/A轉換模組輸出到控制對象。	-50~2050* ¹ -32768~32767* ²	—
濾波器後的測定值	PVf 是透過以下運算公式計算出的測定值。 $PV_{fn} = PV_n + \alpha(PV_{fn-1} - PV_n)$	-50~2050* ¹ -32768~32767* ²	—
手動操作量	MV _{MAN} 儲存手動時從D/A轉換模組輸出的資料。	-50~2050* ¹ -32768~32767* ²	在“有PID極限制”中，手動操作量(MV _{MAN})的值超出指定範圍時，將被轉換為下述的值。 • 小於-50時，變為-50。 • 超過2050時，變為2050。
手動/自動選擇	MAN/AUTO 選擇至D/A轉換模組的輸出資料是手動操作量還是自動操作量。手動時，自動操作量不變化。	0: 自動操作量 1: 手動操作量	設定為0、1以外時將出錯，不執行相應環路的PID運算。
報警	ALARM 用於判別自動操作量(MV)、測定值(PV)的變化率是否超出極限值範圍。進行了1次設定時，在重設之前將被保持。 <div style="text-align: center;"> <p>(2) (1)</p> </div> (1) 測定值(PV)超出極限範圍時，b0將變為1。 (2) 自動操作量(MV)超出極限範圍時，b1將變為1。	—	—

*1 是在“有PID極限制”設定時。
*2 是在“無PID極限制”設定時。

其它功能

在透過PID控制指令進行的PID控制中，自動進行如下所示的無衝擊切換、操作量上下限限制器控制。

無衝擊切換

無衝擊是模式切換(手動↔自動)時連續控制操作量(MV)的功能。進行模式切換時，按以下方式在自動模式的操作量區域與手動模式的操作量區域之間進行資料傳送。

- 從手動模式切換為自動模式時：將手動模式的操作量傳送至自動模式的操作量區域中。
- 從自動模式切換為手動模式時：將自動模式的操作量傳送至手動模式的操作量區域中。

模式切換透過輸入輸出資料區域進行。(☞ 1211頁 輸入輸出資料)

要點

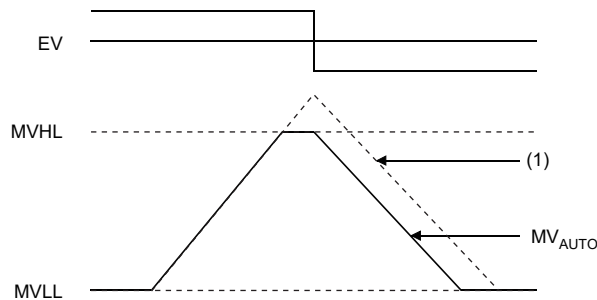
PID控制的自動模式、手動模式如下所示。

- 自動模式：是根據透過PID控制指令進行PID運算而計算出的操作量，對控制對象進行控制的模式。
- 手動模式：是在不透過PID控制指令進行PID運算的狀況下根據計算出的操作量對控制對象進行控制的模式。設定為手動模式的環路將在各採樣週期將測定值(PV)儲存到設定值區域中。

操作量上下限限制器控制

操作量上下限限制器控制是對透過PID運算計算出的操作量的上限或下限進行限制的功能。該功能僅在自動模式時才有效，手動模式時不執行。

透過設定操作量上限值(MVHL)、操作量下限值(MVLL)，可將PID運算計算出的操作量限制在下限值至上限值的範圍內。在操作量上下限限制器功能中，其動作如下所示。



(1)：未執行限制器控制時的 MV_{AUTO}

對於操作量上限值(MVHL)、操作量下限值(MVLL)，可在各環路中設定-50~2050或任意範圍。預設值如下所示。

- 操作量上限值：2000
- 操作量下限值：0

對於操作量上限值(MVHL)、操作量下限值(MVLL)，如果設定為(上限值)<(下限值)將出錯。

手動模式時的測定值至設定值的傳送功能

手動模式時也執行PIDCONT指令。手動模式時，根據SM792(PID無衝擊處理(完全微分用))、SM794(PID無衝擊處理(不完全微分用))的ON/OFF，可以選擇執行PIDCONT指令時是否將從A/D轉換模組獲取的測定值傳送至設定值。預設為OFF。此外，透過PIDCONT指令設定值及測定值將被儲存到輸入輸出資料區域中指定的元件或標籤中。

SM792/SM794	動作內容
OFF	<ul style="list-style-type: none"> 執行PIDCONT指令時，將測定值傳送至設定值。 手動模式→自動模式的切換時，可繼續手動模式時的操作量輸出。 切換為自動模式後如果更改設定值，可以進行從輸出的操作量至設定值的控制。
ON	<ul style="list-style-type: none"> 執行PIDCONT指令時，不將測定值傳送至設定值。 手動模式→自動模式的切換時，可以進行從手動模式時的操作量輸出至設定值的控制。 至自動模式的切換前，應儲存設定值。

要點

根據SM792(PID無衝擊處理(完全微分用))、SM794(PID無衝擊處理(不完全微分用))的ON/OFF，手動模式→自動模式切換時，有以下不同。

- OFF時，因為將測定值傳送至設定值，因此模式切換時測定值與設定值無差異，不發生操作量的急劇變化。另一方面，由於切換後的設定值與自動模式中的目標值不同，因此應透過程式使設定值逐漸變為目標值。
- ON時，因為不將測定值傳送至設定值，因此模式切換時測定值與設定值有差異。切換時該差異較大的情況下，因為有可能發生操作量的急劇變化，因此應在測定值充分趨近於設定值時進行切換的系統中使用。無需透過程式使設定值逐漸變化，可以立即透過自動模式實施。

PID控制用資料、輸入輸出資料的設定範圍更改功能

可以將PID控制用資料、輸入輸出資料的設定範圍透過設定更改為任意範圍。

使設定生效的情況下，應將SD792~SD793(PID極限制設定(完全微分用))、SD794~SD795(PID極限制設定(不完全微分用))的相應環路的位元設定為ON。

不完全微分	完全微分	動作內容																																
SD794	SD792	<table border="1"> <tr> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td> </tr> </table> <p>b0: 環路1 b1: 環路2 ⋮ b14: 環路15 b15: 環路16</p>	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																			
1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0																			
SD795	SD793	<table border="1"> <tr> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td> </tr> </table> <p>b0: 環路17 b1: 環路18 ⋮ b14: 環路31 b15: 環路32</p>	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																			
1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0																			

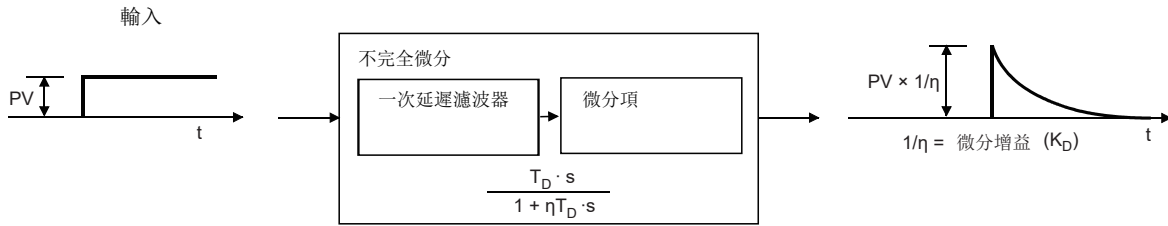
0: 有PID極限制(預設)

1: 無PID極限制

26.2 PID控制指令(不完全微分)

不完全微分是在微分項的輸入中置入一次延遲濾波器的PID控制。不完全微分在以下情況下有效。

- 易受到高頻噪聲影響的控制的情況下
- 完全微分方式中，有步狀變化時無法賦予使操作端動作的有效能量的情況下

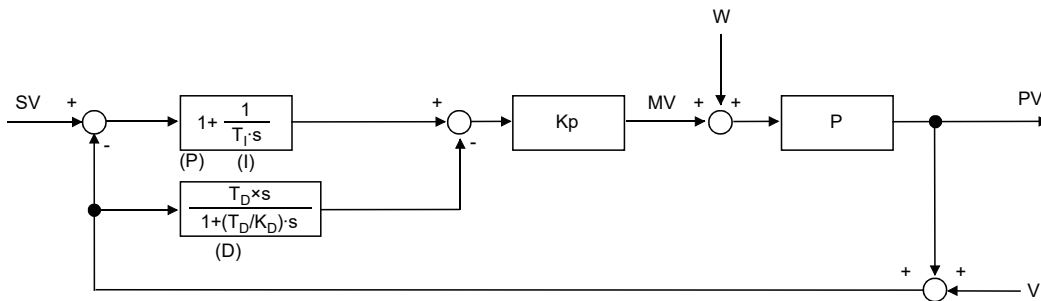


透過不完全微分進行的PID控制指令的性能規格如下所示。

項目		“有PID極限制制”時	“無PID極限制制”時
PID控制環路數		—	最大32環路
採樣週期		T_S	0.01~60.00s
PID運算方式		—	測定值微分型不完全微分(正動作/逆動作)
PID常數設定範圍	比例常數	K_P	0.01~100.00
	積分常數	T_I	0.1~3000.0s
	微分常數	T_D	0.00~300.00s
	微分增益	K_D	0.00~300.00
設定值設定範圍		SV	0~2000
測定值設定範圍		PV	-50~2050
操作量輸出範圍		MV	-32768~32767

PID運算的塊圖及運算公式如下所示。

- 不完全微分的PID運算塊



K_P : 增益
 W : 干擾
 P : 控制對象
 V : 檢測雜音

• 運算公式

正動作/逆動作	運算公式
正動作	$EV_n = PV_{fn} - SV$ $\Delta MV = K_p \left\{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} \cdot EV_n + D_n \right\}$ $D_n = \frac{T_D}{T_S + \frac{T_D}{K_D}} (PV_{fn} - 2PV_{fn-1} + PV_{fn-2}) + \frac{\frac{T_D}{K_D}}{T_S + \frac{T_D}{K_D}} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$
逆動作	$EV_n = SV - PV_{fn}$ $\Delta MV = K_p \left\{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} \cdot EV_n + D_n \right\}$ $D_n = \frac{T_D}{T_S + \frac{T_D}{K_D}} (-PV_{fn} + 2PV_{fn-1} - PV_{fn-2}) + \frac{\frac{T_D}{K_D}}{T_S + \frac{T_D}{K_D}} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$

運算公式中的符號含義如下所示。

符號	含義
EV_n	本次樣本時的偏差
EV_{n-1}	1週期前的偏差
SV	設定值
PV_{fn}	本次樣本時的測定值(濾波器後)
PV_{fn-1}	1週期前的測定值(濾波器後)
PV_{fn-2}	2週期前的測定值(濾波器後)
ΔMV	輸出變化量
MV_n	本次的操作量
D_n	本次的微分項
D_{n-1}	1週期前的微分項
T_S	採樣週期
K_p	比例常數
T_I	積分常數
T_D	微分常數
K_D	微分增益

本次樣本時的測定值(濾波器後) (PV_{fn}) 是將輸入資料的測定值透過下式運算後的值。未設定輸入資料的濾波器係數的情況下，將變為與輸入資料的測定值(PV)相同的值。

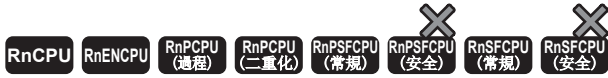
$$PV_{fn} = PV_n + \alpha (PV_{fn-1} - PV_n)$$

PV_n : 本次樣本時的測定值, α : 濾波器係數, PV_{fn-1} : 1週期前的測定值(濾波器後)

此外, PV_{fn} 將被儲存到輸入輸出資料區域中。(參閱 1211 頁 輸入輸出資料)

PID控制用資料的設定

S(P).PIDINIT



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

將指定的元件編號及其以後設定的使用環路數的PID控制資料批量登錄到CPU內部。

梯形圖	ST
	<pre>ENO:=S_PIDINIT(EN, s); ENO:=SP_PIDINIT(EN, s);</pre>

FBD/LD

■執行條件

指令	執行條件
S.PIDINIT	
SP.PIDINIT	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	設定了PID控制用資料的起始元件	—	字元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將(s)中指定的元件編號及其以後設定的使用環路數的PID控制用資料批量登錄到CPU模組內部，設為可PID控制狀態。
(☞ 1209頁 PID控制用資料)
- PID控制用資料的分配如下所示。

項目	PID控制用資料	
全環路通用	(s)+0	使用環路數
	(s)+1	1個掃描的執行環路數
No. 1環路用(14字)	(s)+2	運算公式選擇
	(s)+3	採樣週期(T _S)
	(s)+4	比例常數(K _P)
	(s)+5	積分常數(T _I)
	(s)+6	微分常數(T _D)
	(s)+7	濾波器係數(α)
	(s)+8	操作量下限值(MVLL)
	(s)+9	操作量上限值(MVHL)
	(s)+10	操作量變化率極限值(ΔMVL)
	(s)+11	測定值變化率極限值(ΔPVL)
	(s)+12	固定0(指定的0以外的情況下出錯)
	(s)+13	微分增益(K _D)
	(s)+14	固定0(指定的0以外的情況下出錯)
	(s)+15	固定0(指定的0以外的情況下出錯)
⋮	⋮	⋮
No. n環路用(14字)*1	(s)+(m+0)	運算公式選擇
	(s)+(m+1)	採樣週期(T _S)
	(s)+(m+2)	比例常數(K _P)
	(s)+(m+3)	積分常數(T _I)
	(s)+(m+4)	微分常數(T _D)
	(s)+(m+5)	濾波器係數(α)
	(s)+(m+6)	操作量下限值(MVLL)
	(s)+(m+7)	操作量上限值(MVHL)
	(s)+(m+8)	操作量變化率極限值(ΔMVL)
	(s)+(m+9)	測定值變化率極限值(ΔPVL)
	(s)+(m+10)	固定0(指定的0以外的情況下出錯)
	(s)+(m+11)	微分增益(K _D)
	(s)+(m+12)	固定0(指定的0以外的情況下出錯)
	(s)+(m+13)	固定0(指定的0以外的情況下出錯)

*1 $m=(n-1) \times 14+2$

(1) 固定為“0”。指定了“0”以外的情況下，將變為出錯狀態。

- PID控制用資料的設定中使用的元件點數透過下式計算。

元件點數=2+14×n (n: 使用環路數)

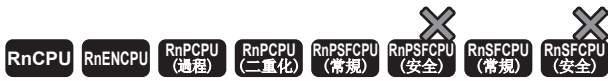
- 各資料應以BIN值進行設定。
- 使用環路的元件點數超過指定元件的最終元件編號時將出錯且不進行處理。
- 1個掃描中在多個位置執行了S(P).PIDINIT指令的情況下，在最靠近S(P).PIDCONT指令處執行的S(P).PIDINIT指令的設定值將有效。
- S(P).PIDINIT指令的執行必須在執行S(P).PIDCONT指令之前。未執行S(P).PIDINIT指令的情況下，不能進行PID控制。

出錯

出錯代碼 (SD0)	內容
3405H	(s) 中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">• PID控制用資料中設定的值超出了允許設定範圍時。• (使用環路數) < (1個掃描的執行環路數) 時。• (操作量上限值) < (操作量下限值) 時。• PID控制用資料的固定為“0”的區域不是“0”時。

PID運算

S(P).PIDCONT



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

透過執行指令的ON，進行採樣週期的計測及PID運算。

梯形圖	ST
	ENO:=S_PIDCONT(EN, s); ENO:=SP_PIDCONT(EN, s);

FBD/LD

執行條件

指令	執行條件
S.PIDCONT	
SP.PIDCONT	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	輸入輸出資料區域中分配的起始元件	—	字元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 執行S(P).PIDCONT指令時，進行採樣週期的計測及PID運算。
- 對於S(P).PIDCONT指令，以(s)中指定的元件編號及其以後設定的輸入輸出資料區域的設定值(SV)、測定值(PV)為基礎進行PID運算，將運算結果儲存到輸入輸出資料區域的自動操作量(MV)區域中。
- 對於PID運算，是在經過採樣週期的設定時間後的首個S(P).PIDCONT指令執行時進行。
- PID控制中，應將控制指令設為ON、執行每個掃描執行S(P).PIDCONT指令。未執行每個掃描的情況下，將無法以正常的採樣週期進行PID運算。此外，1個掃描中不能多次執行S(P).PIDCONT指令。1個掃描中執行多次S(P).PIDCONT指令的情況下，將無法以正常的採樣週期進行PID運算。
- S(P).PIDCONT指令不能記述到中斷程式中使用。將S(P).PIDCONT指令記述到中斷程式中的情況下，將無法以正常的採樣週期進行PID運算。
- 在(s)中，指定輸入輸出資料區域中指定的元件編號的起始。(☞ 1211頁 輸入輸出資料)
- 將檔案寄存器指定為輸入輸出資料區域的情況下，應勿對檔案寄存器施加記憶體保護。雖施加了記憶體保護時不會變為出錯，但無法進行正常的PID運算。
- 輸入輸出資料的分配如下所示。

項目		輸入輸出資料	
寫入		(s)+0	初次處理標誌
讀取/寫入禁止		(s)+1 ⋮ (s)+9	PID控制用工作區域 (用戶不可使用)
No. 1環路用輸入輸出資料區域 (23字)	寫入	(s)+10	設定值(SV)
		(s)+11	測定值(PV)
	讀取	(s)+12	自動操作量(MV)
		(s)+13	濾波器後的測定值(PVf)
	寫入	(s)+14	手動操作量(MV _{MAN})
		(s)+15	手動/自動選擇(MAN/AUTO)
	讀取/寫入	(s)+16	報警(ALARM)
讀取/寫入禁止	(s)+17 ⋮ (s)+32	No. 1環路用工作區域 (用戶不可使用)	
⋮	⋮	⋮	⋮
No. n環路用輸入輸出資料區域 (23字)*1	寫入	(s)+(m+0)	設定值(SV)
		(s)+(m+1)	測定值(PV)
	讀取	(s)+(m+2)	自動操作量(MV)
		(s)+(m+3)	濾波器後的測定值(PVf)
	寫入	(s)+(m+4)	手動操作量(MV _{MAN})
		(s)+(m+5)	手動/自動選擇(MAN/AUTO)
	讀取/寫入	(s)+(m+6)	報警(ALARM)
讀取/寫入禁止	(s)+(m+7) ⋮ (s)+(m+22)	No. n環路用工作區域 (用戶不可使用)	

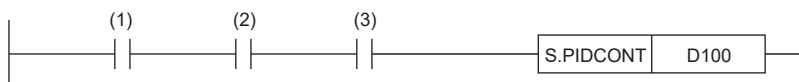
*1 $m=(n-1) \times 23+10$

- 輸入輸出資料的設定中使用的元件點數由下式計算。

元件點數=10+23×n (n: 使用環路數)

- 各資料應以BIN值進行設定。
- 首次處理標誌是PID運算開始時的處理方法的設定。
- 首次運算處理時，將視為到達設定的採樣週期進行運算。
- 首次處理標誌為0的情況下，透過1個掃描將使用環路數的PID運算處理進行批量處理。此外，0以外的情況下，將使用環路數的PID運算處理分割為數個掃描進行處理後，從首次處理完成的環路開始依次開始循環。每個掃描的處理環路數將變為設定的1個掃描的執行環路數。

- 對於輸入輸出資料區域中處於“寫入”的資料，應由用戶透過程式進行寫入。對於處於“讀取”的資料，由用戶透過程式讀取使用。因為對處於“讀取/寫入禁止”及“讀取”的資料進行寫入時將無法正常進行運算，因此應勿進行。但是，從初始狀態開始控制的情況下，需要透過程式進行資料清除。
- 使用環路的元件點數超過指定元件的最終元件編號時將出錯且不進行處理。
- 即使透過手動模式進行手動操作量(MV_{MAN})的輸出的情況下，也應每個掃描執行S(P).PIDCONT指令。如果不執行S(P).PIDCONT指令，將無法執行無衝擊功能。
- 對於S(P).PIDCONT指令，應透過各模組的READY信號採取互鎖，以便只有在用於進行測定值(PV)獲取的A/D轉換模組以及用於進行操作量(MV)輸出的D/A轉換模組正常時才執行指令。



- (1): 控制指令
 (2): A/D轉換模組的READY信號
 (3): D/A轉換模組的READY信號

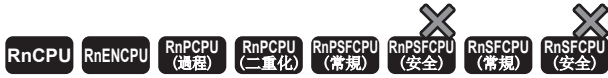
如果在各模組異常時執行，將無法正常進行測定值(PV)的獲取或正常進行操作量(MV)的輸出，因此也無法正常進行PID運算。

出錯

出錯代碼(SD0)	內容
3405H	(s)中指定的輸入輸出資料區域中設定的資料值超出允許設定範圍時。
3422H	在執行S(P).PIDCONT指令之前未執行S(P).PIDINIT指令時。

指定環路No. 的運算停止

S(P).PIDSTOP



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

停止指定環路No. 的PID運算。

梯形圖	ST
	<pre>ENO:=S_PIDSTOP(EN, s); ENO:=SP_PIDSTOP(EN, s);</pre>

FBD/LD

執行條件

指令	執行條件
S.PIDSTOP	
SP.PIDSTOP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	停止的環路No.	1~32	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(s)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 停止(s)中指定的環路No. 的PID運算。對於透過S(P).PIDSTOP指令停止的環路，即使執行S(P).PIDINIT指令也不重啟PID運算。
- 停止中運算資料將被保持。

出錯

出錯代碼(SD0)	內容
3405H	(s)中指定了超出允許指定範圍的資料時。 • 指定的環路No. 不存在 • 指定的值超出了1~32的範圍
3422H	在執行S(P).PIDSTOP指令之前未執行S(P).PIDINIT指令、S(P).PIDCONT指令時。

指定環路No. 的運算開始

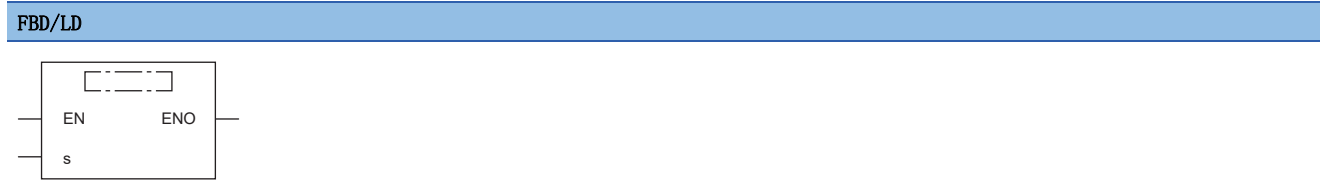
S(P).PIDRUN



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

開始指定環路No. 的運算。

梯形圖	ST
	<pre>ENO:=S_PIDRUN(EN, s); ENO:=SP_PIDRUN(EN, s);</pre>



執行條件

指令	執行條件
S.PIDRUN	
SP.PIDRUN	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	停止的環路No.	1~32	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	○	—	—	—	

功能

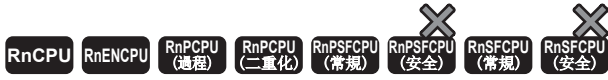
- 開始(s)中指定的環路No. 的PID運算。是用於對透過S(P).PIDSTOP指令停止PID運算的環路No. 再次執行運算的指令。
- 對正在執行PID運算的環路No. 執行了S(P).PIDRUN指令的情況下，將變為無處理。

出錯

出錯代碼(SD0)	內容
3405H	(s)中指定的環路No. 不存在時。
	(s)超出了1~32的範圍時。
3422H	在執行S(P).PIDRUN指令之前未執行S(P).PIDINIT指令、S(P).PIDCONT指令時。

指定環路No. 的參數更改

S(P).PIDPRMW



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(LJMESEC iQ-R CPU模組用戶手冊(應用篇))

將指定環路No. 的運算參數更改為指定的元件編號及其以後儲存的PID控制用資料。

梯形圖	ST
	<pre>ENO:=S_PIDPRMW(EN, s1, s2); ENO:=SP_PIDPRMW(EN, s1, s2);</pre>

FBD/LD

■執行條件

指令	執行條件
S.PIDPRMW	
SP.PIDPRMW	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	更改的環路No.	1~32	無符號BIN16位元	ANY16
(s2)	儲存了更改的PID控制用資料的起始元件	—	字元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○	—	○	—	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	

功能

- 將(s1)中指定的環路No. 的運算參數更改為(s2)中指定的元件編號及其以後儲存的PID控制用資料。
- (s2)中指定的元件編號及其以後的PID控制用資料的配置如下所示。

PID控制用資料	
(s2)+0	運算公式選擇
(s2)+1	採樣週期(T_s)
(s2)+2	比例常數(K_p)
(s2)+3	積分常數(T_I)
(s2)+4	微分常數(T_D)
(s2)+5	濾波器係數(α)
(s2)+6	操作量下限值(MVLL)
(s2)+7	操作量上限值(MVHL)
(s2)+8	操作量變化率極限值(ΔMVL)
(s2)+9	測定值變化率極限值(ΔPVL)
(s2)+10	0
(s2)+11	微分增益(K_D)
(s2)+12	0
(s2)+13	0

出錯

出錯代碼(SD0)	內容
3405H	<p>(s1)中設定了超出允許指定範圍的資料時。</p> <ul style="list-style-type: none"> • 指定的環路No. 不存在 • 指定的值超出了1~32的範圍 <p>(s2)中設定了超出允許指定範圍的資料時。</p> <ul style="list-style-type: none"> • PID控制用資料超出了允許設定範圍 • PID控制用資料的(s2)+10、(s2)+12、(s2)+13不為0
3422H	在執行S(P).PIDPRMW指令之前未執行S(P).PIDINIT指令時。

26.3 PID控制指令(完全微分)

完全微分是原樣不變地使用微分項輸入的PID控制。

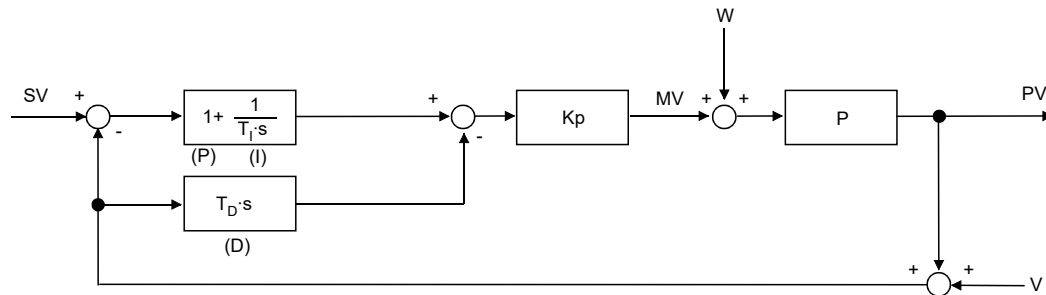


透過完全微分進行的PID控制指令的性能規格如下所示。

項目		“有PID極限制”時	“無PID極限制”時
PID控制環路數	—	最大32環路	
採樣週期	T_S	0.01~60.00s	
PID運算方式	—	測定值微分型完全微分(正動作/逆動作)	
PID常數設定範圍	比例常數	K_P	0.01~100.00
	積分常數	T_I	0.1~3000.0s
	微分常數	T_D	0.00~300.00s
設定值設定範圍	SV	0~2000	-32768~32767
測定值設定範圍	PV	-50~2050	-32768~32767
操作量輸出範圍	MV		

PID運算的塊圖及運算公式如下所示。

- 完全微分的PID運算塊



K_P : 增益
 W : 干擾
 P : 控制對象
 V : 檢測雜音

• 運算公式

正動作/逆動作	運算公式
正動作	$EV_n = PV_{fn} - SV$ $\Delta MV = K_p \left\{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} \cdot EV_n + D_n \right\}$ $D_n = \frac{T_D}{T_S} (PV_{fn} - 2PV_{fn-1} + PV_{fn-2})$ $MV_n = \Sigma \Delta MV$
逆動作	$EV_n = SV - PV_{fn}$ $\Delta MV = K_p \left\{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} \cdot EV_n + D_n \right\}$ $D_n = \frac{T_D}{T_S} (-PV_{fn} + 2PV_{fn-1} - PV_{fn-2})$ $MV_n = \Sigma \Delta MV$

運算公式中的符號含義如下所示。

符號	含義
EV_n	本次樣本時的偏差
EV_{n-1}	1週期前的偏差
SV	設定值
PV_{fn}	本次樣本時的測定值(濾波器後)
PV_{fn-1}	1週期前的測定值(濾波器後)
PV_{fn-2}	2週期前的測定值(濾波器後)
ΔMV	輸出變化量
MV_n	本次的操作量
D_n	本次的微分項
T_S	採樣週期
K_p	比例常數
T_I	積分常數
T_D	微分常數

本次樣本時的測定值(濾波器後)(PV_{fn})是將輸入資料的測定值透過下式運算後的值。未設定輸入資料的濾波器係數的情況下，將變為與輸入資料的測定值(PV)相同的值。

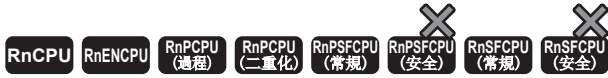
$$PV_{fn} = PV_n + \alpha (PV_{fn-1} - PV_n)$$

PV_n : 本次樣本時的測定值, α : 濾波器係數, PV_{fn-1} : 1週期前的測定值(濾波器後)

此外, PV_{fn} 將被儲存到輸入輸出資料區域中。(參見1211頁 輸入輸出資料)

PID控制用資料的設定

PIDINIT(P)



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

將指定的元件編號及其以後設定的使用環路數的PID控制資料批量登錄到CPU內部。

梯形圖	ST
	ENO:=PIDINIT(EN, s); ENO:=PIDINITP(EN, s);

FBD/LD

■執行條件

指令	執行條件
PIDINIT	
PIDINITP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	設定了PID控制用資料的起始元件	—	字元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將 (s) 中指定的元件編號及其以後設定的使用環路數的PID控制用資料批量登錄到CPU模組內部，設為可PID控制狀態。
(參見 1209 頁 PID 控制用資料)
- PID 控制用資料的分配如下所示。

項目	PID 控制用資料	
全環路通用	(s)+0	使用環路數
	(s)+1	1個掃描的執行環路數
No. 1環路用 (10字元)	(s)+2	運算公式選擇
	(s)+3	採樣週期 (T _S)
	(s)+4	比例常數 (K _P)
	(s)+5	積分常數 (T _I)
	(s)+6	微分常數 (T _D)
	(s)+7	濾波器係數 (α)
	(s)+8	操作量下限值 (MVLL)
	(s)+9	操作量上限值 (MVHL)
	(s)+10	操作量變化率極限值 (ΔMVL)
	(s)+11	測定值變化率極限值 (ΔPVL)
⋮	⋮	⋮
No. n環路用 (10字元)*1	(s)+(m+0)	運算公式選擇
	(s)+(m+1)	採樣週期 (T _S)
	(s)+(m+2)	比例常數 (K _P)
	(s)+(m+3)	積分常數 (T _I)
	(s)+(m+4)	微分常數 (T _D)
	(s)+(m+5)	濾波器係數 (α)
	(s)+(m+6)	操作量下限值 (MVLL)
	(s)+(m+7)	操作量上限值 (MVHL)
	(s)+(m+8)	操作量變化率極限值 (ΔMVL)
	(s)+(m+9)	測定值變化率極限值 (ΔPVL)

*1 $m = (n-1) \times 10 + 2$

- PID 控制用資料的設定中使用的元件點數透過下式計算。

元件點數 = $2 + 10 \times n$ (n: 使用環路數)

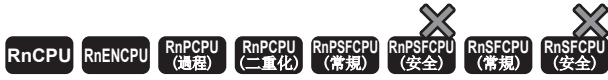
- 各資料應以 BIN 值進行設定。
- 使用環路的元件點數超過指定元件的最終元件編號時將出錯且不進行處理。
- 1個掃描中在多個位置執行了 PIDINIT (P) 指令的情況下，在最靠近 PIDCONT (P) 指令處執行的 PIDINIT (P) 指令的設定值將有效。
- PIDINIT (P) 指令的執行必須在執行 PIDCONT (P) 指令之前。未執行 PIDINIT (P) 指令的情況下，不能進行 PID 控制。

出錯

出錯代碼 (SD0)	內容
3405H	(s) 中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none"> PID 控制用資料中設定的值超出了允許設定範圍 (使用環路數) < (1個掃描的執行環路數) (操作量上限值) < (操作量下限值)

PID運算

PIDCONT (P)



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

透過執行指令的ON，進行採樣週期的計測及PID運算。

梯形圖	ST
	ENO:=PIDCONT (EN, s) ; ENO:=PIDCONT (EN, s) ;

FBD/LD

執行條件

指令	執行條件
PIDCONT	
PIDCONT (P)	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	輸入輸出資料區域中分配的起始元件	—	字元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(s)	—	—	○	—	—	—	○	—	—	—	—	—

功能

- 執行PIDCONT (P) 指令時，進行採樣週期的計測及PID運算。
- 對於PIDCONT (P) 指令，以 (s) 中指定的元件編號及其以後設定的輸入輸出資料區域的設定值 (SV)、測定值 (PV) 為基礎進行PID運算，將運算結果儲存到輸入輸出資料區域的自動操作量 (MV) 區域中。
- 對於PID運算，是在經過採樣週期的設定時間後的首個PIDCONT (P) 指令執行時進行。
- 在PID控制中，應將控制指令設為ON、讓每個掃描執行PIDCONT (P) 指令。未執行每個掃描的情況下，將無法以正常的採樣週期進行PID運算。此外，1個掃描中不能多次執行PIDCONT (P) 指令。1個掃描中執行多次PIDCONT (P) 指令的情況下，將無法以正常的採樣週期進行PID運算。
- PIDCONT (P) 指令不能記述到中斷程式中使用。將PIDCONT (P) 指令記述到中斷程式中的情況下，將無法以正常的採樣週期進行PID運算。
- 在 (s) 中，指定輸入輸出資料區域中指定的元件編號的起始。(☞ 1211頁 輸入輸出資料)
- 將檔案寄存器指定為輸入輸出資料區域的情況下，應勿對檔案寄存器施加記憶體保護。雖施加了記憶體保護時不會變為出錯，但無法進行正常的PID運算。
- 輸入輸出資料的分配如下所示。

項目		輸入輸出資料	
寫入		(s)+0	初次處理標誌
讀取/寫入禁止		(s)+1 ⋮ (s)+9	PID控制用工作區域 (用戶不可使用)
No. 1環路用輸入輸出資料區域 (18字)	寫入	(s)+10	設定值 (SV)
		(s)+11	測定值 (PV)
	讀取	(s)+12	自動操作量 (MV)
		(s)+13	濾波器後的測定值 (PVf)
	寫入	(s)+14	手動操作量 (MV _{MAN})
		(s)+15	手動/自動選擇 (MAN/AUTO)
	讀取/寫入	(s)+16	報警 (ALARM)
讀取/寫入禁止	(s)+17 ⋮ (s)+27	No. 1環路用工作區域 (用戶不可使用)	
⋮	⋮	⋮	⋮
No. n環路用輸入輸出資料區域 (18字)*1	寫入	(s)+(m+0)	設定值 (SV)
		(s)+(m+1)	測定值 (PV)
	讀取	(s)+(m+2)	自動操作量 (MV)
		(s)+(m+3)	濾波器後的測定值 (PVf)
	寫入	(s)+(m+4)	手動操作量 (MV _{MAN})
		(s)+(m+5)	手動/自動選擇 (MAN/AUTO)
	讀取/寫入	(s)+(m+6)	報警 (ALARM)
讀取/寫入禁止	(s)+(m+7) ⋮ (s)+(m+17)	No. n環路用工作區域 (用戶不可使用)	

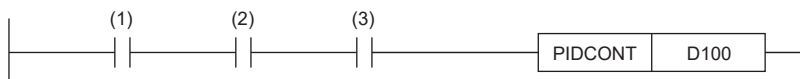
*1 $m=(n-1) \times 18+10$

- 輸入輸出資料的設定中使用的元件點數由下式計算。

元件點數=10+18×n (n: 使用環路數)

- 各資料應以BIN值進行設定。
- 首次處理標誌是PID運算開始時的處理方法的設定。
- 首次運算處理時，將視為到達設定的採樣週期進行運算。
- 首次處理標誌為0的情況下，透過1個掃描將使用環路數的PID運算處理進行批量處理。此外，0以外的情況下，將使用環路數的PID運算處理分割為數個掃描進行處理後，從首次處理完成的環路開始依次開始循環。每個掃描的處理環路數將變為設定的1個掃描的執行環路數。

- 對於輸入輸出資料區域中處於“寫入”的資料，應由用戶透過程式進行寫入。對於處於“讀取”的資料，由用戶透過程式讀取使用。因為對處於“讀取/寫入禁止”及“讀取”的資料進行寫入時將無法正常進行運算，因此應勿進行。但是，從初始狀態開始控制的情況下，需要透過程式進行資料清除。
- 即使透過手動模式進行手動操作量(MV_{MAN})的輸出的情況下，也應每個掃描執行PIDCONT(P)指令。如果不執行PIDCONT(P)指令，將無法執行無衝擊功能。
- 對於PIDCONT(P)指令，應透過各模組的READY信號採取互鎖，以便只有有用於進行測定值(PV)獲取的A/D轉換模組以及用於進行操作量(MV)輸出的D/A轉換模組正常時才執行指令。如果在各模組異常時執行，將無法正常進行測定值(PV)的獲取或正常進行操作量(MV)的輸出，因此也無法正常進行PID運算。



(1): 控制指令

(2): A/D轉換模組的READY信號

(3): D/A轉換模組的READY信號

出錯

出錯代碼(SD0)	內容
3405H	(s)中指定的輸入輸出資料區域中設定的資料值超出允許設定範圍時。
3422H	執行PIDCONT(P)指令之前，未執行PIDINIT(P)指令時。

指定環路No. 的運算停止

PIDSTOP (P)



- 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

停止 (s) 中指定的環路No. 的PID運算。

梯形圖	ST
	ENO:=PIDSTOP (EN, s) ; ENO:=PIDSTOPP (EN, s) ;

FBD/LD

執行條件

指令	執行條件
PIDSTOP	
PIDSTOPP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	停止的環路No.	1~32	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	—	○	—	—	—	—	○	○	—	—	—

功能

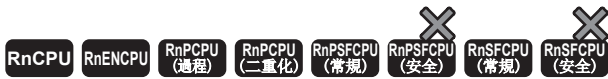
- 停止 (s) 中指定的環路No. 的PID運算。對於透過PIDSTOP (P) 指令停止的環路，即使執行PIDINIT (P) 指令，也不重啟PID運算。
- 停止中運算資料將被保持。

出錯

出錯代碼 (SD0)	內容
3405H	(s) 中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none"> 指定的環路No. 不存在 指定的值超出了1~32的範圍
3422H	在執行PIDSTOP (P) 指令之前未執行PIDINIT (P) 指令、PIDCONT (P) 指令時。

指定環路No. 的運算開始

PIDRUN (P)



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

開始指定環路No. 的運算。

梯形圖	ST
	ENO:=PIDRUN (EN, s) ; ENO:=PIDRUNP (EN, s) ;

FBD/LD

執行條件

指令	執行條件
PIDRUN	
PIDRUNP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	停止的環路No.	1~32	無符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	—	○	—	—	—	—	○	○	—	—	—

功能

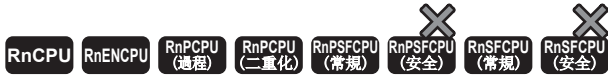
- 開始(s)中指定的環路No. 的PID運算。是用於對透過PIDSTOP (P) 指令停止了PID運算的環路No. 再次執行運算的指令。
- 對正在執行PID運算的環路No. 執行了PIDRUN (P) 指令的情況下，將變為無處理。

出錯

出錯代碼(SD0)	內容
3405H	(s) 中設定了超出允許指定範圍的資料時。 • 指定的環路No. 不存在 • 指定的值超出了1~32的範圍
3422H	在執行PIDRUN (P) 指令之前未執行PIDINIT (P) 指令、PIDCONT (P) 指令時。

指定環路No. 的參數更改

PIDPRMW (P)



- 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統中的動作將有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))

將指定的環路No. 的運算參數更改為指定的元件編號及其以後儲存的PID控制用資料。

梯形圖	ST
	ENO:=PIDPRMW (EN, s1, s2) ; ENO:=PIDPRMWP (EN, s1, s2) ;

FBD/LD

執行條件

指令	執行條件
PIDPRMW	
PIDPRMWP	

設置數據

內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	更改的環路No.	1~32	無符號BIN16位元	ANY16
(s2)	儲存了更改的PID控制用資料的起始元件	—	字元	ANY16*1
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○	—	○	—	—	—	—	○	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 將(s1)中指定的環路No. 的運算參數更改為(s2)中指定的元件編號及其以後儲存的PID控制用資料。
- (s2)中指定的元件編號及其以後的PID控制用資料的配置如下所示。

PID控制用資料	
(s2)+0	運算公式選擇
(s2)+1	採樣週期(T_s)
(s2)+2	比例常數(K_p)
(s2)+3	積分常數(T_I)
(s2)+4	微分常數(T_D)
(s2)+5	濾波器係數(α)
(s2)+6	操作量下限值(MVLL)
(s2)+7	操作量上限值(MVHL)
(s2)+8	操作量變化率極限值(ΔMVL)
(s2)+9	測定值變化率極限值(ΔPVL)

出錯

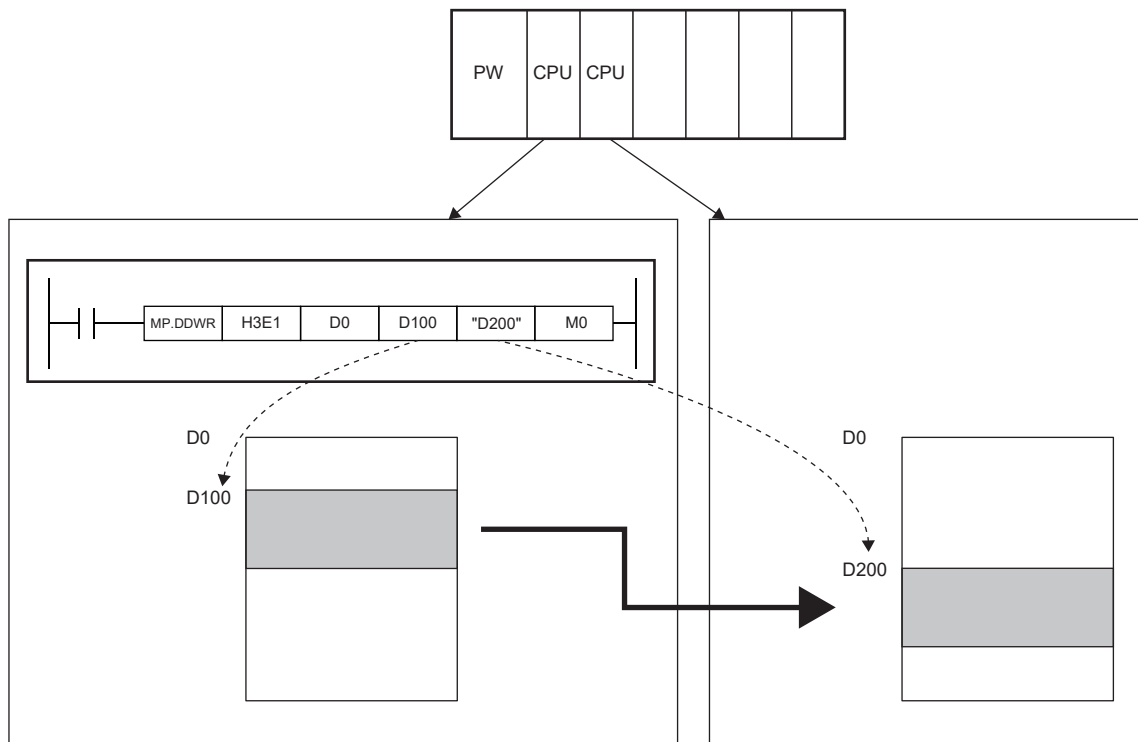
出錯代碼(SD0)	內容
3405H	(s1)中設定了超出允許指定範圍的資料時。 指定的環路No. 不存在 指定的值超出了1~32的範圍
	(s2)中指定的PID控制用資料超出允許設定範圍時。
3422H	在執行PIDPRMW(P)指令之前未執行PIDINIT(P)指令時。

27 多CPU之間專用指令

27.1 其它機號CPU模組存取指令

概要

其它機號CPU模組存取指令是從本機CPU模組向其它機號CPU模組進行元件資料的寫入/讀取的指令。
根據其它機號CPU模組存取指令，進行從1號機至2號機的寫入時的動作如下所示。



其它機號CPU模組存取指令的內容及用途如下所示。

指令名	內容	用途
D(P).DDR	將其它機號CPU模組的元件資料讀取到本機CPU模組的元件中。	在CPU模組之間以使用了CPU之間恆定週期通訊功能的時機進行資料發送接收的情況下使用。
D(P).DDWR	將本機CPU模組的元件資料寫入到其它機號CPU模組的元件中。	在以各自的CPU模組的時機進行資料發送接收的情況下使用。
M(P).DDR	將其它機號CPU模組的元件資料讀取到本機CPU模組的元件中。	在以各自的CPU模組的時機進行資料發送接收的情況下使用。
M(P).DDWR	將本機CPU模組的元件資料寫入到其它機號CPU模組的元件中。	

參數記憶體

使用D(P).DDR指令、D(P).DDWR指令的情況下，需要在系統參數內進行恆定週期通訊功能設定。

可寫入/讀取的元件

根據其它機號CPU模組存取指令，可對其它機號CPU模組進行寫入/讀取的元件如下所示。

分類	種類	元件名*1	對象目標元件 設定可否	設定時的條件
用戶元件 系統元件	位元元件	X、Y、M、L、B、F、 SB、SM	可以	應滿足以下條件。 • 必須是16位元(4位)的位指定。 • 開始位元元件必須是16(10H)的倍數。
	字元元件	T、ST、C、D、W、SW、 SD	可以	無
檔案寄存器	字元元件	R、ZR	可以	無
間接指定		—	否	無

*1 可以進行使用了字元串指定的變址寄存器的變址修飾(Z、ZZ表示)。以透過本機變址寄存器的值進行了變址修飾的值對其它機號進行存取。

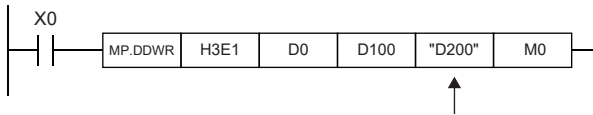
例如，“K4M0Z0”中Z0=16的情況下，則M0+16=M16，以K4M16對其它機號進行存取。

此外，“ZROZZ0”中Z0、Z1=100000的情況下，則ZR0+100000=ZR100000，以ZR100000對其它機號進行存取。

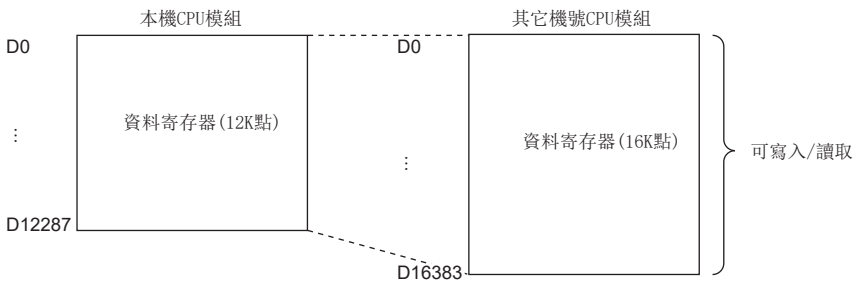
元件的指定方法及可寫入/讀取的範圍

其它機號CPU模組的元件指定是透過字元串進行指定。

直接指定其它機號CPU模組的寫入元件“D200”。



在字元串指定中，可以對其它機號的元件的所有範圍進行寫入/讀取。例如，本機的資料寄存器為12K點，其它機號的資料寄存器為16K點的情況下，可以從其它機號的資料寄存器的起始開始進行16K點的寫入/讀取。



要點

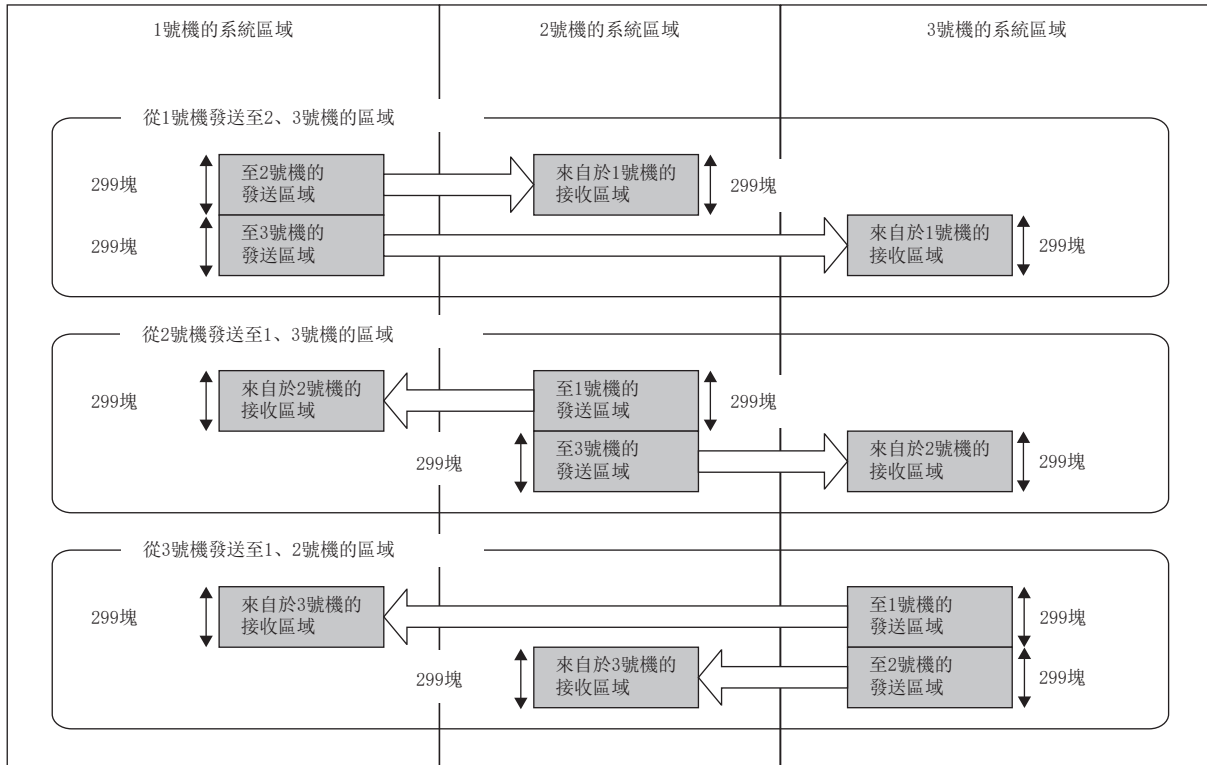
- 即使在元件No. 的高位中附加“0”，也將被作為與未附加情況下相同的元件進行處理。例如，“D1”及“D0001”均被作為D1處理。
- 對於以大寫字母及小寫字母指定的元件，將被作為同一元件處理。例如，“D1”與“d1”均被作為D1處理。
- 透過字元串指定了其它機號CPU模組中不存在的元件的情況下，將變為異常完成，應加以注意。

可使用的塊數

對於其它機號CPU模組存取指令，將系統區域以1個容量為16字構成的“塊”為最小單位使用。
其它機號CPU模組存取指令中可使用的塊數如下所示。

CPU模組個數	最大塊數
2個配置	599塊
3個配置	299塊
4個配置	199塊

由3個CPU模組構成的多CPU系統中的塊數如下所示。



可讀取/寫入的最大資料點數

指令中可讀取/寫入的最大資料點數取決於多CPU系統配置的CPU模組個數。

CPU模組個數	可讀取的最大資料點數	可寫入的最大資料點數
2個	8192點	8192點
3個	4096點	4096點
4個	2048點	2048點

指令中使用的塊數

指令中使用的塊數取決於寫入/讀取資料點數。指令中使用的塊數如下所示。

讀取/寫入	塊數	示例
讀取時	指令中使用的塊數=(21+讀取資料點數)÷16	<ul style="list-style-type: none"> 讀取資料點數為100點的情況下 指令中使用的塊數=(21+100)÷16=7[塊]
寫入時	指令中使用的塊數=(19+寫入資料點數)÷16	<ul style="list-style-type: none"> 寫入資料點數為100點的情況下 指令中使用的塊數=(19+100)÷16=7[塊]

其它機號CPU模組存取指令的同時執行

對於其它機號CPU模組存取指令，如果在下式範圍內則可以同時執行。

可同時執行其它機號CPU模組存取指令的條件

[各號機中可使用的塊數] ≥ [同時執行的指令使用的合計塊數]

由於執行其它機號CPU模組存取指令，其它機號CPU模組存取指令使用的塊數超過系統區域的總塊數的情況下，在該掃描中將不執行本指令（變為無處理），在下一個掃描中將再次執行本指令。

但是，執行了本指令時，系統區域的空餘塊數少於SD796～SD799（多CPU之間專用指令最大塊數設定）的設定值的情況下，本指令將變為異常完成。

系統區域的空餘塊數少於其它機號CPU模組存取指令使用的塊數或SD796～SD799的設定值的情況下，其它機號CPU模組存取指令的執行可否如下所示。

SD設定值*3與空餘塊數*2的大小關係	指令使用塊數*1與空餘塊數的大小關係	
	指令使用塊數 ≤ 空餘塊數	指令使用塊數 > 空餘塊數
SD設定值 ≤ 空餘塊數	執行	不執行（進行無處理）
SD設定值 > 空餘塊數	異常完成	

*1 其它機號CPU模組存取指令使用的塊數

*2 系統區域的空餘塊數

*3 SD796～SD799的設定值

使用其它機號CPU模組存取指令時的互鎖

作為其它機號CPU模組存取指令的互鎖用，有特殊繼電器的SM796～SM799（多CPU之間專用指令最大使用塊數設定）。

同時執行多個其它機號CPU模組存取指令的情況下，其它機號CPU模組存取指令的互鎖應使用SM796～SM799。

要點

使用SM796～SM799的情況下，應在SD796～SD799中設定各機號中使用的指令的最大塊數。例如，對3號機執行的其它機號CPU模組存取指令的塊數的最大值為5的情況下，應在SD798中設定5。

變為SD796～SD799中設定的塊數以下時，相應特殊繼電器（SM796～SM799）將變為ON。

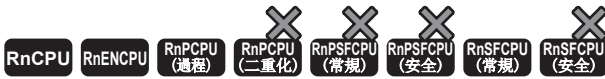
注意事項

- 應在寫入/讀取對象CPU模組處於已啟動的狀態下，執行D(P).DDWR指令、M(P).DDWR指令、D(P).DDRDR指令、M(P).DDRDR指令。如果在寫入/讀取對象CPU模組未啟動的狀態下執行，將變為無處理。
- 執行D(P).DDWR指令、M(P).DDWR指令、D(P).DDRDR指令、M(P).DDRDR指令之後，至完成元件變為ON為止期間，如果對設定資料中指定的元件範圍等進行更改，系統中儲存的資料（完成狀態、完成元件）將無法時常儲存。
- SB/SW、SM/SD中包含有系統資訊區域。透過D(P).DDWR指令、M(P).DDWR指令進行寫入的情況下，應避免覆蓋系統資訊區域。
- 執行指令的使用塊數多於SD796～SD799的設定值的情況下，即使透過SM796～SM799採取互鎖指令也可能無法執行（異常完成）。
- 對於SD796～SD799，應在以相應機號為對象的指令執行之前進行設定。（建議對CPU模組RUN後的第1個掃描進行設定。）

從其它機號CPU模組的元件讀取

D(P). DDRD、M(P). DDRD

- D(P). DDRD



- RnENCPU可在韌體版本為“25”及其以後的版本使用。工程工具可以在版本為“1.032J”及其以後的版本使用。
- RnSFCPU(常規)可以在韌體版本為“17”及其以後的版本使用。工程工具可以在版本為“1.055H”及其以後的版本使用。

- M(P). DDRD

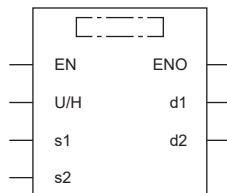


RnENCPU可在韌體版本為“25”及其以後的版本使用。工程工具可以在版本為“1.032J”及其以後的版本使用。

多CPU配置系統時，從其它機號CPU模組中讀取元件資料。

梯形圖	ST
	ENO:=D_DDRD(EN, U/H, s1, s2, d1, d2); ENO:=DP_DDRD(EN, U/H, s1, s2, d1, d2); ENO:=M_DDRD(EN, U/H, s1, s2, d1, d2); ENO:=MP_DDRD(EN, U/H, s1, s2, d1, d2);

FBD/LD



■執行條件

指令	執行條件
D. DDRD M. DDRD	
DP. DDRD MP. DDRD	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U/H)	其它機號CPU模組的起始輸入輸出編號 (以16進制數4位表示時的高3位)	3E0H~3E3H	無符號BIN16位元	ANY16
(s1)	儲存了控制資料的本機CPU模組的起始元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 2)
(s2)	儲存讀取資料的其它機號CPU模組的起始元件	—	字元串	ANYSTRING_SINGLE
(d1)	儲存讀取的資料的本機CPU模組的起始元件	—	字元	ANY16* ¹
(d2)	完成元件	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其它 (U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(U/H)	○	—	○	—	—	—	—	○	○	—	—	○
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	—	—	—	—	—	—	—	—	○	—
(d1)	○	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	—	○	—	—	—	—	○	—	—	—	—

■控制資料

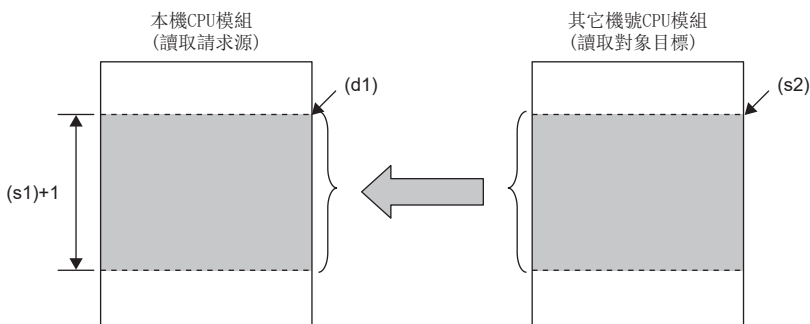
操作數：(s1)				
元件	項目	內容	設定範圍	設定方
+0	完成狀態	儲存完成時的狀態。 • 0000H：正常完成 • 0000H以外：異常完成(出錯代碼)	—	系統
+1	讀取資料點數	將讀取資料點數以字元單位進行設定。	1~8192*1	用戶

*1 變為配置2個CPU模組的多CPU系統時的最大設定範圍。

由於根據系統配置可讀取的資料點數將變化，因此有可能會變為8192點以下。(☞ 1237頁 其它機號CPU模組存取指令)

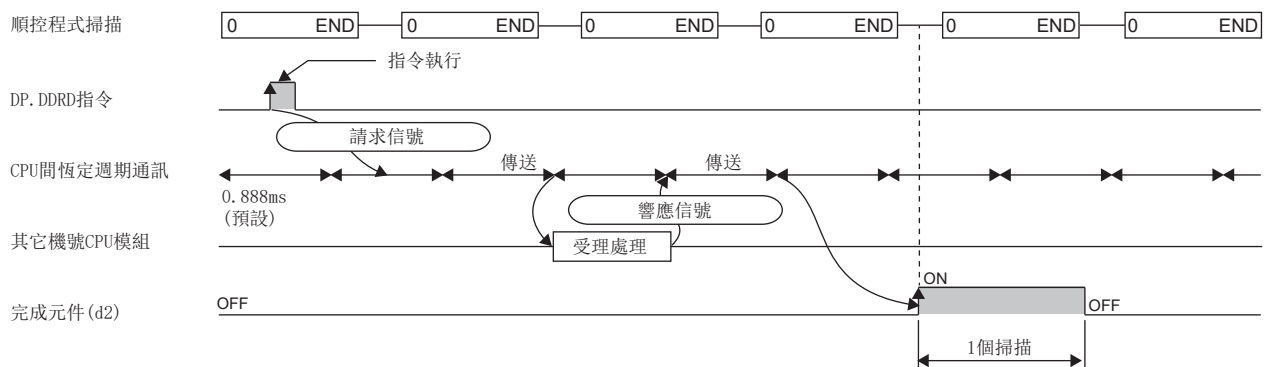
功能

- 多CPU配置系統時，將本機CPU模組中指定的元件(d1)以後的資料，以(s1)+1中指定的讀取資料點數，儲存到其它機號CPU模組(U/H)的指定的元件(d1)以後。

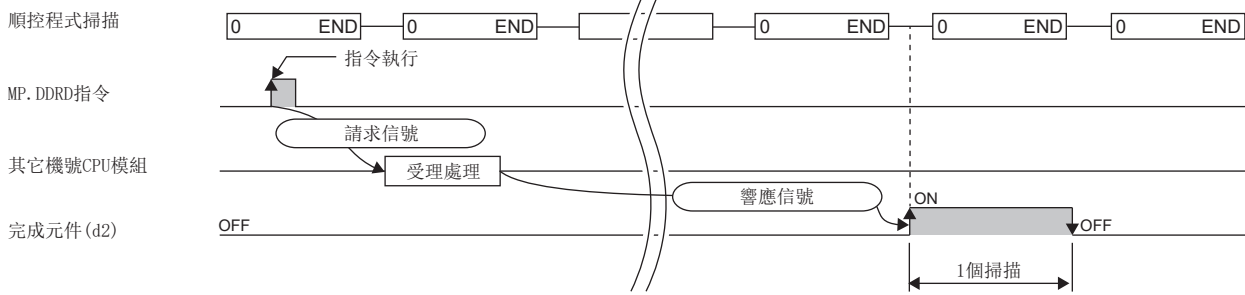


- D(P). DDRD指令、M(P). DDR指令的動作概要如下所示。

- D(P). DDRD指令的動作概要



• M(P). DDRD指令的動作概要



• 對於D(P). DDRD指令、M(P). DDRD指令的執行及正常/異常完成，可透過設定資料中指定的完成元件 (d2)、完成時的狀態顯示元件 (d2)+1進行確認。

• 完成元件 (d2)

D(P). DDRD指令、M(P). DDRD指令在完成的掃描的END處理中ON、在下一個END處理中OFF。

• 完成時的狀態顯示元件 (d2)+1

D(P). DDRD指令根據M(P). DDRD指令完成時的狀態而ON/OFF。

正常完成時：保持為OFF不變。

異常完成時：D(P). DDRD指令、M(P). DDRD指令在完成的掃描的END處理中ON、在下一個END處理中OFF。

異常完成時，完成狀態 (s1)+0中將儲存出錯代碼。

• 指令中使用的塊數取決於讀取資料點數。關於指令中使用的塊數，請參閱以下內容。

☞ 1237頁 其它機號CPU模組存取指令

• 關於讀取對象CPU模組的對象目標元件中可指定的元件，請參閱以下內容。

☞ 1237頁 其它機號CPU模組存取指令

• 系統區域中沒有空餘塊的情況下，即使執行指令也將變為異常完成。透過指令中將使用的塊數在SD796~SD799的範圍內進行設定，將SM796~SM799作為互鎖使用可以防止變為異常完成狀態。

出錯

出錯代碼 (SD0)	內容
2800H	指定的其它機號CPU模組的起始輸入輸出編號(以16進制數4位表示時的高3位)超出3E0H~3E3H的範圍時。
2801H	指定的其它機號CPU模組有錯誤。 • 指定了被設定為保留的機號時。 • 指定了未安裝的機號時。
2802H	其它機號CPU模組不支援D(P). DDRD指令、M(P). DDRD指令。
2803H	指定的其它機號CPU模組中指定了本機CPU模組時。
2810H	指定的其它機號CPU模組中指定了不能執行指令的CPU模組時。
3404H	用於指定元件的字元串發生了出錯時。
3405H	讀取資料點數 (s1)+1超出0~8192的範圍。*2
3440H	在設定為不使用CPU之間恆定週期通訊的狀況下執行了D(P). DDRD指令。
3441H	指定的資料點數超出了各號機中可以使用的系統區域容量。

*2 變為配置2個CPU模組的多CPU系統時的最大設定範圍。

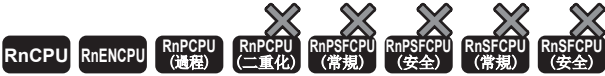
由於根據系統配置可讀取的資料點數將變化，因此有可能會變為8192點以下。(☞ 1237頁 其它機號CPU模組存取指令)

出錯代碼 ((s1)+0)	內容
0010H	至對象目標CPU模組的指令請求為允許值及其以上。(系統區域中沒有空餘塊。)
1001H	(s2)中指定的其它機號CPU模組的元件是在其它機號CPU模組中不能使用的元件。或者超出了元件範圍。
1081H	D(P). DDRD指令、M(P). DDRD指令中設定的讀取資料點數為0。

至其它機號CPU模組的元件寫入

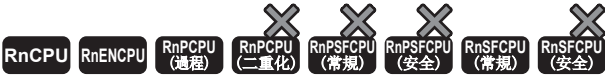
D(P). DDWR、M(P). DDWR

- D(P). DDWR



RnENCPU可在韌體版本為“25”及其以後的版本使用。工程工具可以在版本為“1.032J”及其以後的版本使用。

- M(P). DDWR

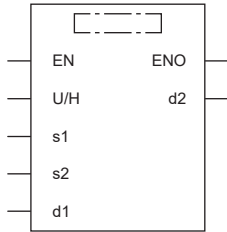


RnENCPU可在韌體版本為“25”及其以後的版本使用。工程工具可以在版本為“1.032J”及其以後的版本使用。

多CPU配置系統時，將元件資料寫入其它機號CPU模組。

梯形圖	ST
	ENO:=D_DDWR (EN, U/H, s1, s2, d1, d2) ; ENO:=DP_DDWR (EN, U/H, s1, s2, d1, d2) ; ENO:=M_DDWR (EN, U/H, s1, s2, d1, d2) ; ENO:=MP_DDWR (EN, U/H, s1, s2, d1, d2) ;

FBD/LD



■執行條件

指令	執行條件
D. DDWR M. DDWR	
DP. DDWR MP. DDWR	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(U/H)	其它機號CPU模組的起始輸入輸出編號 (以16進制數4位表示時的高3位)	3E0H~3E3H	無符號BIN16位元	ANY16
(s1)	儲存了控制資料的本機CPU模組的起始 元件	參閱控制資料	字元	ANY16_ARRAY (要素數: 2)
(s2)	儲存了寫入資料的本機CPU模組的起始 元件	—	字元	ANY16* ¹
(d1)	儲存寫入資料的其它機號CPU模組的起 始元件	—	字元串	ANYSTRING_SINGLE
(d2)	完成元件	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

*1 在透過標籤進行設定的情況下，應在確保動作所需區域的前提下定義數組，指定該數組型標籤的要素。

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其它 (U)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(U/H)	○	—	○	—	—	—	○	○	—	—	○	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	○	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	—	—	—	—	—	—	—	○	—	
(d2)	○	—	○	—	—	—	○	—	—	—	—	

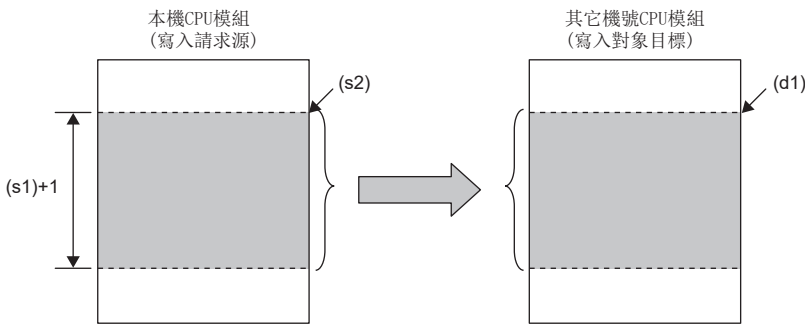
■控制資料

操作數：(s1)				
元件	項目	內容	設定範圍	設定方
+0	完成狀態	儲存完成時的狀態。 • 0000H：正常完成 • 0000H以外：異常完成(出錯代碼)	—	系統
+1	寫入資料點數	將寫入資料點數以字單位進行設定。	1~8192*1	用戶

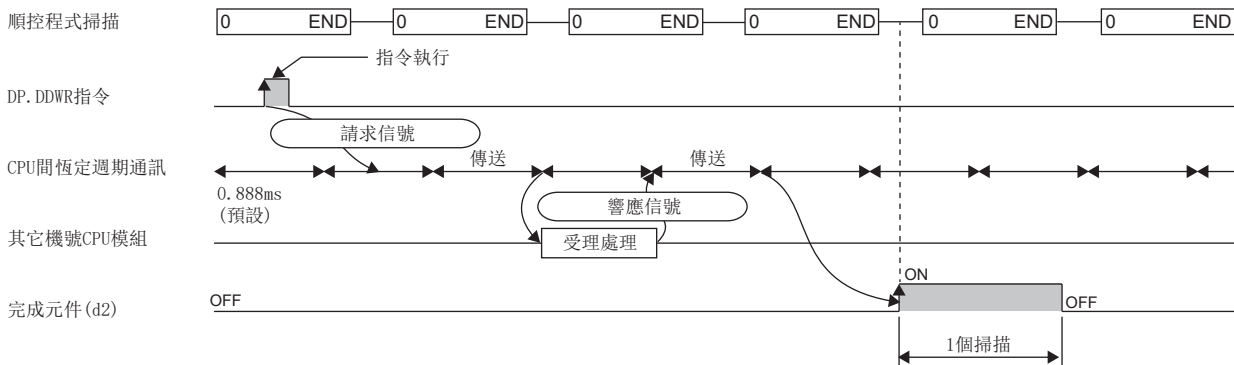
*1 變為配置2個CPU模組的多CPU系統時的最大設定範圍。
由於根據系統配置可寫入的資料點數將變化，因此有可能會變為8192點以下。(☞ 1237頁 其它機號CPU模組存取指令)

功能

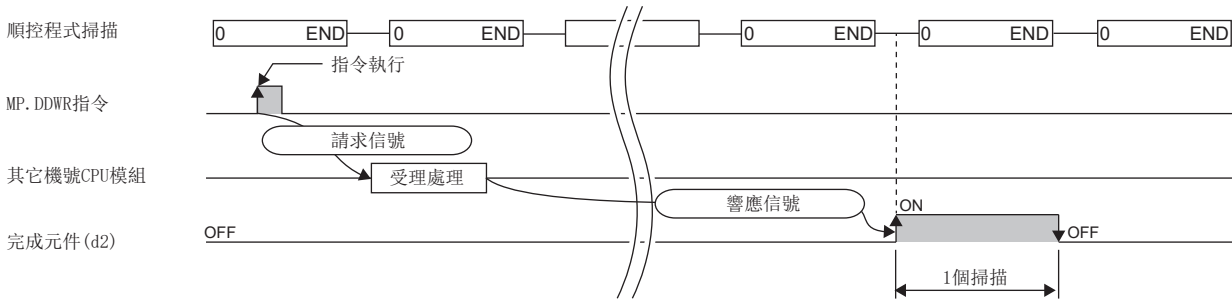
- 多CPU配置系統時，將本機CPU模組中指定的元件(s2)以後的資料，以(s1)+1中指定的寫入資料點數、儲存到其它機號CPU模組(U/H)的指定的元件(d1)以後。



- D(P). DDWR指令、M(P). DDWR指令的動作概要如下所示。
- D(P). DDWR指令的動作概要



• M(P). DDWR指令的動作概要



• 對於D(P). DDWR指令、M(P). DDWR指令的執行及正常/異常完成，可透過設定資料中指定的完成元件 (d2)、完成時的狀態顯示元件 (d2)+1進行確認。

• 完成元件 (d2)

D(P). DDWR指令、M(P). DDWR指令在完成的掃描的END處理中ON、在下一個END處理中OFF。

• 完成時的狀態顯示元件 (d2)+1

D(P). DDWR指令根據M(P). DDWR指令完成時的狀態而ON/OFF。

正常完成時：保持為OFF不變。

異常完成時：D(P). DDWR指令、M(P). DDWR指令在完成的掃描的END處理中ON、在下一個END處理中OFF。

異常完成時，完成狀態 (s1)+0中將儲存出錯代碼。

• 指令中使用的塊數取決於讀取資料點數。關於指令中使用的塊數，請參閱以下內容。

☞ 1237頁 其它機號CPU模組存取指令

• 關於寫入對象目標CPU模組的對象目標元件中可指定的元件，請參閱以下內容。

☞ 1237頁 其它機號CPU模組存取指令

• 系統區域中沒有空餘塊的情況下，即使執行指令也將變為異常完成。透過指令中將使用的塊數在SD796~SD799的範圍內進行設定，將SM796~SM799作為互鎖使用可以防止變為異常完成狀態。

出錯

出錯代碼 (SD0)	內容
2800H	指定的其它機號CPU模組的起始輸入輸出編號 (以16進制數4位表示時的高3位) 超出3E0H~3E3H的範圍時。
2801H	指定的其它機號CPU模組有錯誤。 • 指定了被設定為保留的機號時。 • 指定了未安裝的機號時。
2802H	其它機號CPU模組不支援D(P). DDWR指令、M(P). DDWR指令。
2803H	指定的其它機號CPU模組中指定了本機CPU模組時。
2810H	指定的其它機號CPU模組中指定了不能執行指令的CPU模組時。
3404H	用於指定元件的字元串發生了出錯時。
3405H	寫入資料點數 (s1)+1超出了0~8192的範圍。 ^{*2}
3440H	在設定為不使用CPU之間恆定週期通訊的狀況下執行了D(P). DDWR指令。
3441H	指定的資料點數超出了各號機中可以使用的系統區域容量。

*2 變為配置2個CPU模組的多CPU系統時的最大設定範圍。

由於根據系統配置可寫入的資料點數將變化，因此有可能會變為8192點以下。(☞ 1237頁 其它機號CPU模組存取指令)

出錯代碼 ((s1)+0)	內容
0010H	至對象目標CPU模組的指令請求為允許值及其以上。(系統區域中沒有空餘塊。)
1001H	(d1)中指定的其它機號CPU模組的元件是其它機號CPU模組中不能使用的元件。或者超出了元件範圍。
1080H	D(P). DDWR指令、M(P). DDWR指令中設定的寫入資料點數為0。
1090H	透過D(P). DDWR指令、M(P). DDWR指令設定的寫入目標的元件被設定為禁止外部的元件寫入。

28 SFC程式用指令

要點

本章中對SFC程式的進行控制的指令進行說明。關於SFC程式的詳細內容，請參閱下述手冊。
 MELSEC iQ-R 程式手冊(程式設計篇)

28.1 SFC控制指令

步活性檢查

LD、LDI、AND、ANI、OR、ORI [S□/BL□\S□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU(過程模式)可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU(二重化模式)在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。
- LD: 常開觸點指令/LDI: 常閉觸點指令

將指定步的激活/非激活當作運算結果。

- AND: 常開觸點串聯連接指令/ANI: 常閉觸點串聯連接指令

將指定步的激活/非激活、與至當時為止的運算結果之AND運算當作運算結果。

- OR: 1個常開觸點的並聯連接指令/ORI: 1個常閉觸點的並聯連接指令

將指定步的激活/非激活、與至當時為止的運算結果之OR運算當作運算結果。

梯形圖	ST
	不對應。
FBD/LD	
不對應。	

■執行條件

指令	執行條件
LD LDI AND ANI OR ORI	常時執行

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	作為觸點使用的元件	—	位元	ANY_BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他 (BL□\ S□)	
	X、Y、M、L、 SM、F、B、S、 SB、FX、FY	J□\ □	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\ G□、J□\ □、 U3E□\ (H)G□	Z	LT、 LST、LC		LZ	K、H	E		\$
(s)	○*1	—	—	—	—	—	—	—	—	—	—	○

*1 僅可使用S。

功能

- 檢查指定塊的指定步是否為激活狀態。
- 根據已指定步的激活/非激活狀態，觸點的ON/OFF如下所示。

步的狀態	常開觸點指令的觸點	常閉觸點指令的觸點
激活狀態	ON	OFF
非激活狀態	OFF	ON

- 步的指定方法如下所示。

程式	指定方法	
SFC程式	指定本塊內的步時	透過S□指定。
	指定其它塊的步時	透過BL□\ S□指定。
順控程式	透過BL□\ S□指定。 無指定塊No.時，應透過BRSET指令指定塊No.。在BRSET指令無塊指定時，對象塊變為塊0。	

- 應在下述範圍內指定塊No.、步No.。

CPU模組	塊No.	步No.
R00CPU、R01CPU、R02CPU	0~127	0~127
上述以外的CPU模組	0~319	0~511

- 指定了範圍外的塊No. 或者步No. 時，常開觸點指令、常閉觸點指令均變為OFF。
- SFC程式存在的狀態中(SM320(SFC程式的有無)為ON)，應將SM321(SFC程式的啟動/停止)僅在ON時執行。
- SFC程式為不存在狀態(SM320為OFF)或SM321為OFF時，指定的塊No. 及步No. 皆在範圍內的情況下，常開觸點指令為OFF，常閉觸點指令為ON。

注意事項

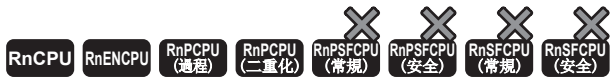
- 順控程式上的塊未指定的步指定S□不能透過工程工具上觸點的ON/OFF監視。但是由於進行運算，若內部為ON則線圈輸出將被設定為ON。

出錯

沒有運算出錯。

塊激活檢查

LD、LDI、AND、ANI、OR、ORI [BL□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU(過程模式)可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU(二重化模式)在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

• LD: 常開觸點指令/LDI: 常閉觸點指令

將指定塊的激活/非激活、與至當時為止的運算結果之AND運算當作運算結果。

• AND: 常開觸點串聯連接指令/ANI: 常閉觸點串聯連接指令

將指定塊的激活/非激活與至當時為止的激活/非激活的AND運算作為運算結果。

• OR: 1個常開觸點的並聯連接指令/ORI: 1個常閉觸點的並聯連接指令

將指定塊的激活/非激活、與至當時為止的運算結果之OR運算當作運算結果。

梯形圖	ST
	不對應。

FBD/LD
不對應。

■執行條件

指令	執行條件
LD	常時執行
LDI	
AND	
ANI	
OR	
ORI	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	作為觸點使用的元件	—	位元	ANY_BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他 (BL□)
	X、Y、M、L、 SM、F、B、S、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H)G□	Z	LT、 LST、LC	LZ		K、H	E	\$		
(s)	—	—	—	—	—	—	—	—	—	—	—	—	○

功能

- 檢查指定塊是否為激活狀態。
- 根據已指定的塊的激活/非激活狀態，觸點的ON/OFF如下所示。

指定了塊的狀態	常閉觸點指令的觸點	常開觸點指令的觸點
激活狀態	ON	OFF
非激活狀態	OFF	ON

- 應在下述範圍內指定塊No.。

CPU模組	塊No.
R00CPU、R01CPU、R02CPU	0~127
上述以外的CPU模組	0~319

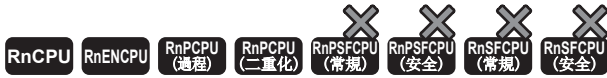
- 指定了範圍外的塊No.時，常閉觸點指令、常開觸點指令的觸點皆變為OFF。
- SFC程式存在的狀態中(SM320(SFC程式的有無)為ON)，應將SM321(SFC程式的啟動/停止)僅在ON時執行。
- SFC程式為不存在狀態(SM320為OFF)或SM321為OFF時，指定的塊No.在範圍內的情況下，常閉觸點指令為OFF，常開觸點指令為ON。

出錯

沒有運算出錯。

激活步批量讀取

MOV(P) [K4S□/BL□\K4S□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU(過程模式)可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU(二重化模式)在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

指定塊的步激活狀態，作為位資訊在BIN16位元資料單位中讀取至指定元件。

梯形圖	ST
	ENO:=MOV(EN, s, d); ENO:=MOV(P)(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
MOV	
MOV(P)	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存傳送源的資料的元件編號	—	帶符號BIN16位元	ANY16
(d)	傳送目標的元件編號	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他 (BL□\ S□)	
	X、Y、M、L、 SM、F、B、S、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H) G□	Z	LT、 LST、LC		LZ	K	H		E
(s)	○*1	—	—	—	—	—	—	—	—	—	—	○
(d)	○	○	○	○	○	—	—	○	—	—	—	—

*1 僅可使用S。

DMOV (P) [K8S□/BL□\K8S□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU (過程模式) 可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU (二重化模式) 在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

指定塊的步激活狀態，作為位資訊在BIN32位元資料單位中讀取至指定元件。

梯形圖	ST
	ENO:=DMOV (EN, s, d) ; ENO:=DMOV (EN, s, d) ;

FBD/LD

■執行條件

指令	執行條件
DMOV	
DMOV P	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	儲存傳送源的資料的元件編號	—	帶符號BIN32位元	ANY32
(d)	傳送目標的元件編號	—	帶符號BIN32位元	ANY32
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他 (BL□\S□)
	X、Y、M、L、SM、F、B、S、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○*1	—	—	—	—	—	—	—	—	—	—	—	○
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

*1 僅可使用S。

出錯

出錯代碼 (SD0)	內容
2820H	進行塊指定時，指定塊No. 超過可指定的塊No. 的範圍時。
	進行塊指定時，指定步No. 超過可指定的步No. 的範圍時。

要點

透過位指定來指定步。

- 透過SFC程式指定本塊內的步時，透過K8S□指定。
- 透過SFC程式指定其他塊內的步時，透過BL□\K8S□指定。
- 透過順控程式指定步時，透過BL□\K8S□指定。

BMOV (P) [K4S□/BL□\K4S□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU (過程模式) 可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU (二重化模式) 在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

從指定步中，將指定塊的步激活狀態，以指定字元批量讀取。

梯形圖	ST
	ENO:=BMOV (EN, s, n, d) ; ENO:=BMOV (EN, s, n, d) ;

FBD/LD

■執行條件

指令	執行條件
BMOV	
BMOVP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(s)	傳送資料被儲存的元件的起始編號	—	帶符號BIN16位元	ANY16
(d)	傳送目標的元件的起始編號	—	帶符號BIN16位元	ANY16
(n)	傳送數	0~65535	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

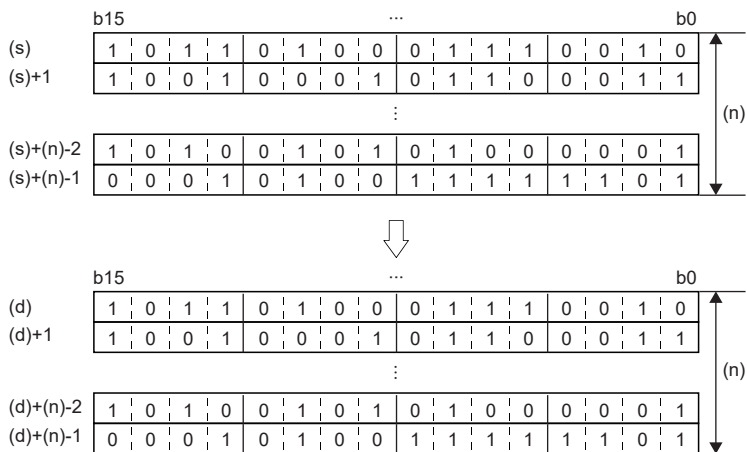
■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他 (BL□\S□)	
	X、Y、M、L、SM、F、B、S、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(s)	○*1	—	—	—	—	—	—	—	—	—	—	○
(d)	○	○	○	○	—	○	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

*1 僅可使用S。

功能

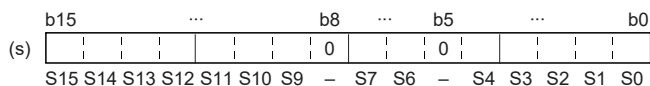
- 指定塊的活性狀態，透過指定字元分從指定步中批量讀取。
- 未指定塊時，讀取如下塊的步激活狀態。
 - 在順控程式中執行指令時：塊0
 - 在SFC程式中(動作輸出內)執行指令時：執行指令的塊(本塊)
- 將讀取的結果儲存到(d)中指定的元件中。



- 在讀取資料中，缺少編號的步No. 中對應的位元變為0。

例

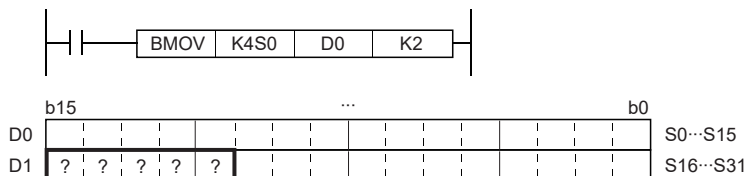
指定塊的S5與S8不存在時如下所示。(空欄中，步的激活狀態被儲存。)



- 在無塊指定時，讀取資料範圍超過塊中的最大步No. 時，不固定資料被儲存。

例

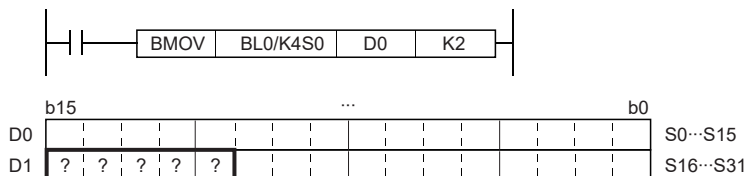
塊的最終步No. 在S26中，將從S0開始的2字元傳送到D0至D1中的情況如下所示。(空欄中，步的激活狀態被儲存。)



- 塊中存在指定時，讀取資料範圍超過塊內存在的步數、或指定了不存在步的起始時，不固定資料被儲存。

例

塊的最終步No. 為S26時，將從S0開始的2字傳送到D0至D1中的情況如下所示。(空欄中，步的激活狀態被儲存。)



- 在SFC程式不存在狀態中執行的情況及存在塊指定的情況下，指定了不存在讀取資料的起始塊時，讀取位元資料變為“0”。

出錯

出錯代碼 (SD0)	內容
2820H	進行塊指定時，指定塊No. 超過可指定的塊No. 的範圍時。
	進行塊指定時，指定步No. 超過可指定的步No. 的範圍時。

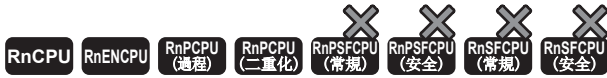
要點

透過位指定來指定步。

- 透過SFC程式指定本塊內的步時，透過K4S□指定。
- 透過SFC程式指定其它塊的步時，透過BL□\K4S□指定。
- 透過順控程式指定步時，透過BL□\K4S□指定。

塊啟動

SET [BL□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU (過程模式) 可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU (二重化模式) 在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

單獨激活指定塊，並從初始步執行。

梯形圖	ST
	ENO:=SET (EN, d);

FBD/LD

■執行條件

指令	執行條件
SET	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	被激活指定塊No. (設定 (ON) 位元元件編號)	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他 (BL□)	
	X、Y、M、L、SM、F、B、S、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(d)	—	—	—	—	—	—	—	—	—	—	—	○*1

*1 BL可變址修飾。

功能

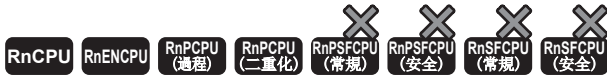
- 單獨激活指定塊，並從初始步執行。初始步為多個步時，激活全部的初始步。
- 設定SFC用資訊元件的塊啟動結束位時，該位元元件變為OFF→ON。
- 指令執行時，指定塊已是激活狀態時，作為無處理指令 (相當於NOP指令) 繼續處理。
- 指令執行時，指定塊為SFC塊運行中寫入中的狀態時，將指令作為無處理 (相當於NOP指令) 繼續處理。
- 過程CPU (二重化模式) 的情況下，即使以備份模式的待機系統執行也將變為無處理。

出錯

出錯代碼 (SD0)	內容
31A1H	不存在指定塊時。 不存在SFC程式 (不存在掃描執行類型的SFC程式)、或將SM321 (SFC程式的啟動/停止) 設定為OFF且執行指令時。
31A2H	指定塊No. 超過可指定的塊No. 的範圍時。

塊結束

RST [BL□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU (過程模式)可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU (二重化模式)在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

單獨非激活指定塊。

梯形圖	ST
	ENO:=RST(EN, d);

FBD/LD

■執行條件

指令	執行條件
RST	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	被結束的指定塊No. (重設(OFF)位元元件編號)	—	位元	ANY_ELEMENTARY
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數			其他 (BL□)
	X、Y、M、L、SM、F、B、S、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	
(d)	—	—	—	—	—	—	—	—	—	—	—	○*1

*1 BL可變址修飾。

功能

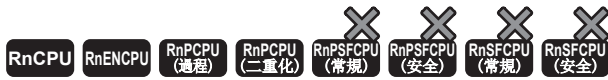
- 單獨的指定塊設為非激活。
- 激活中的存在步的情況下，全部設為非激活後線圈輸出也為OFF。
- 設定SFC用資訊元件的塊啟動結束位時，該位元元件變為ON→OFF。
- 指令執行時，指定塊已是非激活狀態時，作為無處理指令(相當於NOP指令)繼續處理。
- 過程CPU(二重化模式)的情況下，即使以備份模式的待機系統執行也將變為無處理。

出錯

出錯代碼 (SD0)	內容
31A1H	不存在指定塊時。
	不存在SFC程式 (不存在掃描執行類型的SFC程式)、或將SM321 (SFC程式的啟動/停止) 設定為OFF且執行指令時。
31A2H	指定塊No. 超過可指定的塊No. 的範圍時。

塊停止

PAUSE [BL□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU (過程模式) 可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU (二重化模式) 在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

指定塊為暫時停止狀態。

梯形圖	ST
	ENO:=PAUSE (EN, d);

FBD/LD

■執行條件

指令	執行條件
PAUSE	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	暫時停止狀態中指定塊No.	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他 (BL□)	
	X、Y、M、L、SM、F、B、S、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC		LZ	K	H		E
(d)	—	—	—	—	—	—	—	—	—	—	—	○*1

*1 BL可變址修飾。

功能

- 指定塊為暫時停止狀態。
- 設定SFC用資訊元件的塊停止重啟位時，該位元元件也變為ON。
- 因為SFC用資訊元件的停止時模式位元的設定透過即時停止或移轉後停止，因此指令執行後各步的停止時機有所不同。

停止時模式位元的設定	停止時機
即時停止	指令執行後，所有步停止。
切換後停止	指令執行後步的動作仍持續。切換條件成立後，透過切換目標的步停止。

- 暫時停止時，根據OUT指令線圈輸出狀態與SM325(塊停止時的輸出模式)的狀態相比有所不同。

SM325的狀態	根據OUT指令線圈輸出狀態
ON時	線圈輸出在停止中仍保持停止前的狀態。
OFF時	線圈輸出，停止時為OFF。

- 過程CPU(二重化模式)的情況下，即使以備份模式的待機系統執行也將變為無處理。

注意事項


線圈保持步，在保持中SM325在OFF時停止的情況下，因變為非激活，故保持的狀態不可重啟。SM325停止在ON時，可以重啟。

出錯

出錯代碼(SD0)	內容
31A1H	不存在指定塊時。 不存在SFC程式(不存在掃描執行類型的SFC程式)、或將SM321(SFC程式的啟動/停止)設定為OFF且執行指令時。
31A2H	指定塊No. 超過可指定的塊No. 的範圍時。

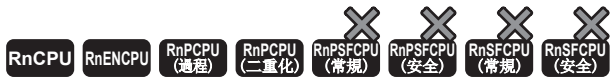
要點

塊停止/重啟時的動作由SM325與SFC用資訊元件的停止時模式位元來設定，根據步的保持/非保持的組合決定。詳細內容，請參閱下述手冊。

 MELSEC iQ-R 程式手冊(程式設計篇)

塊重啟

RSTART [BL□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU (過程模式) 可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU (二重化模式) 在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

解除指定塊的暫時停止、透過停止步執行重啟。

梯形圖	ST
	<pre>ENO:=RSTART (EN, d);</pre>

FBD/LD

■執行條件

指令	執行條件
RSTART	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	解除暫時停止指定塊No.	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數			其他 (BL□)
	X、Y、M、L、SM、F、B、S、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K、H	E	\$	
(d)	—	—	—	—	—	—	—	—	—	—	—	○*1

*1 BL可變址修飾。

功能

- 指定塊透過停止的步執行被設為重啟。在動作保持狀態停止中變為動作保持步(有過度檢查/無過度檢查)，透過動作保持狀態執行重啟。
- SFC用資訊元件的塊停止重啟位元被設定時，該位元元件仍為OFF。
- 根據SM325 (塊停止時的輸出模式)的狀態，塊重啟後的PLS指令，僅在上升沿時執行的指令的動作有所不同。

SM325的狀態	PLS指令，僅在上升沿時執行指令的動作
ON時(線圈輸出保持)	不再次執行。
OFF時(線圈輸出OFF)	再次執行。

- 過程CPU(二重化模式)的情況下，即使以備份模式的待機系統執行也將變為無處理。

注意事項

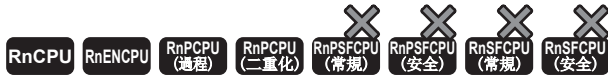
線圈保持步，在保持中SM325在OFF時停止的情況下，因變為非激活，故保持的狀態不可重啟。SM325停止在ON時，可以重啟。

出錯

出錯代碼(SD0)	內容
31A1H	不存在指定塊時。
	不存在SFC程式(不存在掃描執行類型的SFC程式)、或將SM321(SFC程式的啟動/停止)設定為OFF且執行指令時。
31A2H	指定塊No. 超過可指定的塊No. 的範圍時。

步啟動

SET [S□/BL□\S□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU (過程模式) 可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU (二重化模式) 在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

指定步被激活。

梯形圖	ST
	ENO:=SET (EN, d);

FBD/LD

■執行條件

指令	執行條件
SET	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	被啟動指定步No. (設定 (ON) 位元元件編號)	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他 (BL□\S□)	
	X、Y、M、L、SM、F、B、S、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H) G□	Z	LT、LST、LC		LZ	K	H		E
(d)	○*1	—	—	—	—	—	—	—	—	—	—	○*1

*1 僅可使用S、BL□\S□。也可進行變址修飾。

功能

- 指定塊的指定步被激活。根據指定塊的激活/非激活，指定塊的動作如下所示。

指定塊的狀態	動作
非激活的情況下	指定塊在指令時執行時變為激活，透過指定步開始處理。設定SFC用資訊元件的塊啟動結束位時，該位元件變為OFF→ON。
已變為激活的情況下	已激活狀態的步繼續進行原有處理、進行新的指定步激活。

- 在已激活中的步執行指令時，作為無處理指令(相當於NOP指令)繼續處理。但是，在保持步中指定的步為保持中狀態時，解除保持狀態後變為一般的激活狀態，與動作輸出、移轉條件同時執行。
- 指令執行時，指定步的塊為SFC塊運行中寫入中的狀態時，將指令作為無處理(相當於NOP指令)繼續處理。
- 初始步存在多個步時，透過指定特定步使其激活來開始初始步的選擇。
- 無塊指定時，如下所示。
 - 在順控程式中執行指令時：塊0
 - 在SFC程式中(動作輸出內)執行指令時：執行指令的塊(本塊)
- 過程CPU(二重化模式)的情況下，即使以備份模式的待機系統執行也將變為無處理。

注意事項

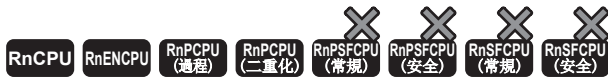
- 透過步啟動指令，指定並聯分支中的步時應啟動並聯步。若存在未激活分支梯形圖，則不移轉並聯合併。此外，對於並聯分支中的步，若執行步結束指令，並聯合併的條件同樣不成立。
- SFC程式在動作輸出內中，不可指定步No. 中的本步No.。指定了指定步No. 中本步No. 時，會出錯。

出錯

出錯代碼(SDO)	內容
31A1H	不存在指定塊時。(SET/RST BL□\S□)
31A2H	進行塊指定時，指定塊No. 超過可指定的塊No. 的範圍時。
31B1H	不存在SFC程式(不存在掃描執行類型的SFC程式)、或將SM321(SFC程式的啟動/停止)設定為OFF且執行指令時。 指定步在指定塊內不存在時。(SET/RST BL□\S□/S□)
31B2H	指定步No. 超過可指定的步No. 的範圍時。
31B5H	在動作輸出內中，指定了指定步中的本步時。

步結束

RST [S□/BL□\S□]



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU (R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU (過程模式) 可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU (二重化模式) 在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

非激活指定步。

梯形圖	ST
	ENO:=RST(EN, d);

FBD/LD

■執行條件

指令	執行條件
RST	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型 (標籤)
(d)	被結束指定塊No. (重設 (OFF) 位元元件編號)	—	位元	ANY_ELEMENTARY
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元		間接指定	常數			其他 (BL□\ S□)	
	X、Y、M、L、 SM、F、B、S、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H) G□	Z	LT、 LST、LC		LZ	K、H	E		\$
(d)	○*1	—	—	—	—	—	—	—	—	—	—	○*1

*1 僅可使用S、BL□\S□。也可進行變址修飾。

功能

- 指定塊的指定步為非激活。
- 根據指令執行，指定塊的激活步數變為0時，塊變為非激活。設定SFC用資訊元件的塊啟動結束位時，該位元件從ON變為OFF。
- 在已變為非激活的步中執行時，作為無處理指令(相當於NOP指令)。
- 無塊指定時，如下所示。
 - 在順控程式中執行指令時：塊0
 - 在SFC程式中(動作輸出內)執行指令時：執行指令的塊(本塊)
- 過程CPU(二重化模式)的情況下，即使以備份模式的待機系統執行也將變為無處理。

注意事項

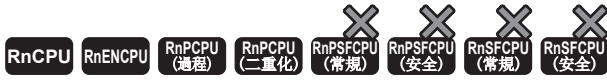
- 透過步啟動指令，指定並聯分支中的步時應啟動並聯步。若存在未激活分支梯形圖，則不移轉並聯合併。此外，對於並聯分支中的步，若執行步結束指令，並聯合併的條件同樣不成立。
- SFC程式在動作輸出內中，不可指定步No. 中的本步No.。指定了指定步No. 中本步No. 時，會出錯。

出錯

出錯代碼(SD0)	內容
31A1H	不存在指定塊時。(SET/RST BL□\S□)
31A2H	進行塊指定時，指定塊No. 超過可指定的塊No. 的範圍時。
31B1H	不存在SFC程式(不存在掃描執行類型的SFC程式)、或將SM321(SFC程式的啟動/停止)設定為OFF且執行指令時。 指定步在指定塊內不存在時。(SET/RST BL□\S□/□)
31B2H	指定步No. 超過可指定的步No. 的範圍時。
31B5H	在動作輸出內中，指定了指定步中的本步時。

切換對象塊

BRSET

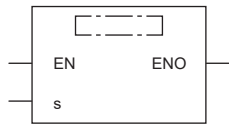


- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU(過程模式)可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU(二重化模式)在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

指定SFC控制指令的對象塊。

梯形圖	ST
	ENO:=BRSET(EN, s);

FBD/LD



■執行條件

指令	執行條件
BRSET	

設置數據

■內容、範圍、資料類型

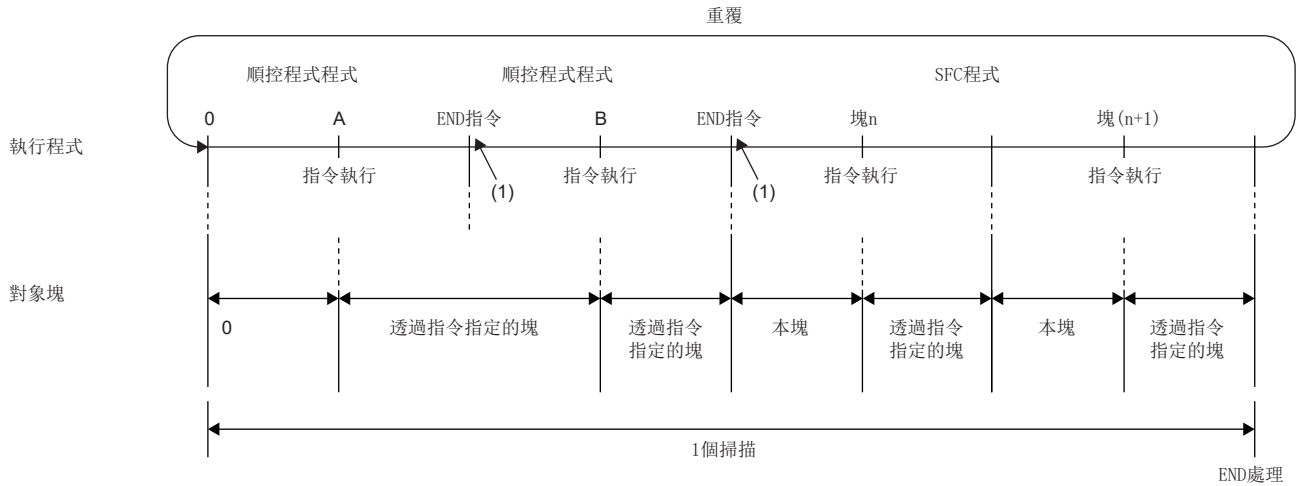
操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	指定指令的對象塊	—	帶符號BIN16位元	ANY16
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元			雙字元		間接指定	常數				其他
	X、Y、M、L、SM、F、B、S、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 僅在步(S□)指定SFC控制指令的對象塊，在(s)中指定的元件中切換設定的編號。
- 執行BRSET指令時，根據執行的程式，對象塊切換變為有效的情況如下所示。
- 透過順控程式執行BRSET指令時，對象塊切換變為有效的情況，從指令執行開始到SFC程式執行為止。下述掃描時，到再次執行指令為止作為預設的對象塊變為塊0。
- 在SFC程式中執行了BRSET指令時，對象塊切換變為有效的情況，僅為執行中的步。即使對象步相同時，針對每S□指令的塊，有必要執行BRSET指令。此外，塊內也變為有效的情況僅從指令執行到指定步的處理結束。在指定步的處理結束、執行下述的程式時，到再次執行指令為止作為預設，對象塊將變為本塊。



(1) 不進行END處理。

- BL□\S□中指定的塊No. 與BRSET指令的執行/不執行無關，均將變為有效。
- 透過並聯分支等激活了多個的步時，因僅在執行了指令步中變為有效，因此在多個步中指定塊時，有必要執行步指令。

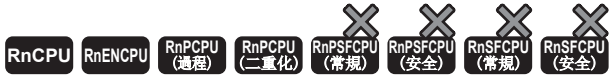
出錯

出錯代碼 (SD0)	內容
31A1H	指定的塊不存在、或SFC程式處於待機狀態時。 不存在SFC程式 (不存在掃描執行類型的SFC程式)、或將SM321 (SFC程式的啟動/停止) 設定為OFF且執行指令時。
31A2H	指定塊No. 超過可指定的塊No. 的範圍時。

28.2 SFC專用指令

移轉條件虛擬輸出

TRAN



- R00CPU、R01CPU、R02CPU沒有版本的限制。
- RnCPU(R00CPU、R01CPU、R02CPU除外)、RnENCPU可以在韌體版本為“12”及其以後的版本使用。工程工具可以在版本為“1.015R”及其以後的版本使用。
- 過程CPU(過程模式)可以在韌體版本為“03”及其以後的版本使用。工程工具可以在版本為“1.020W”及其以後的版本使用。
- 過程CPU(二重化模式)在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

移轉條件被成立時的虛擬輸出。

梯形圖	ST
	TRAN(s);

FBD/LD

要點

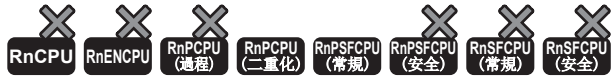
移轉條件的詳細內容，請參閱下述手冊。

MELSEC iQ-R 程式手冊(程式設計篇)

29 二重化系統用指令

29.1 系統切換

SP. CONTSW



執行SP. CONTSW指令的掃描的END處理時，切換控制系統和待機系統。

梯形圖	ST
	ENO:=SP_CONTSW(EN, s, d);

FBD/LD

■執行條件

指令	執行條件
SP. CONTSW	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	系統切換指令識別編號 指定用於特定發行了系統切換請求的處理的值。	-32768~32767	帶符號BIN16位元	ANY16
(d)	異常完成元件 無法執行系統切換時設定為ON、正常地完成了系統切換時設定為OFF。	—	位元	ANY_BOOL
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

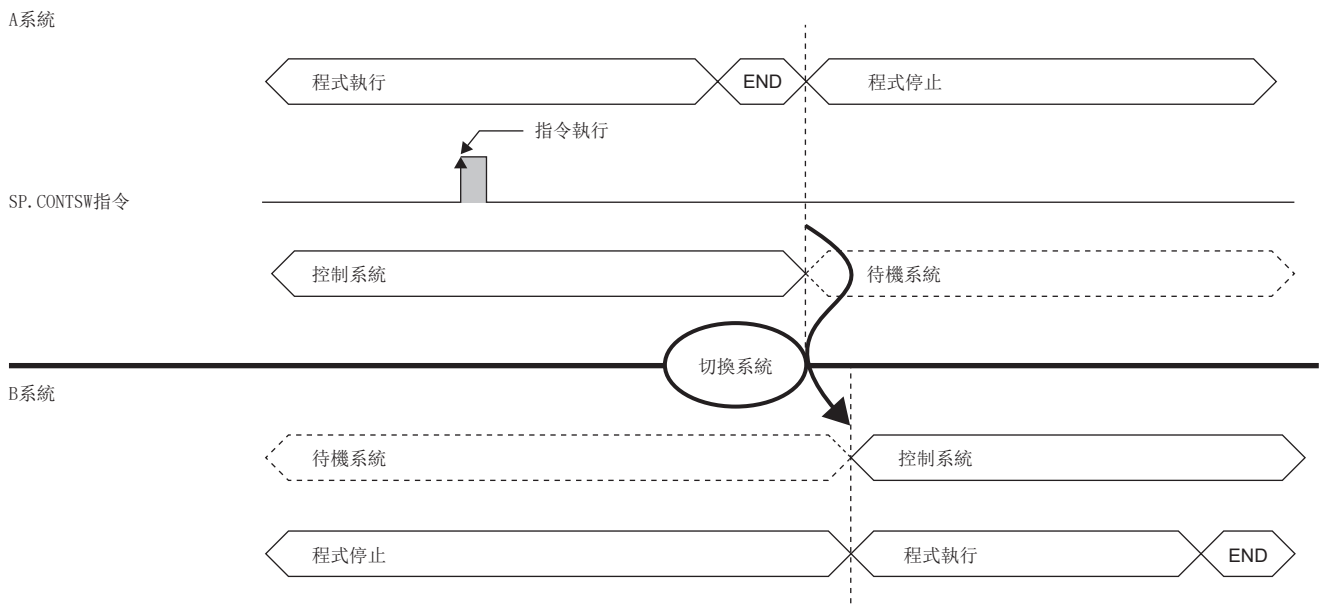
■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	—	○	—	—	—	—	○	○	—	—	—	
(d)	○	—	○	—	—	—	—	○	—	—	—	—	

功能

- 執行SP. CONTSW指令的掃描的END處理時，切換控制系統和待機系統。在控制系統中執行SP. CONTSW指令。在待機系統中執行的情況下將變為無處理。

系統切換時的動作如下所示。



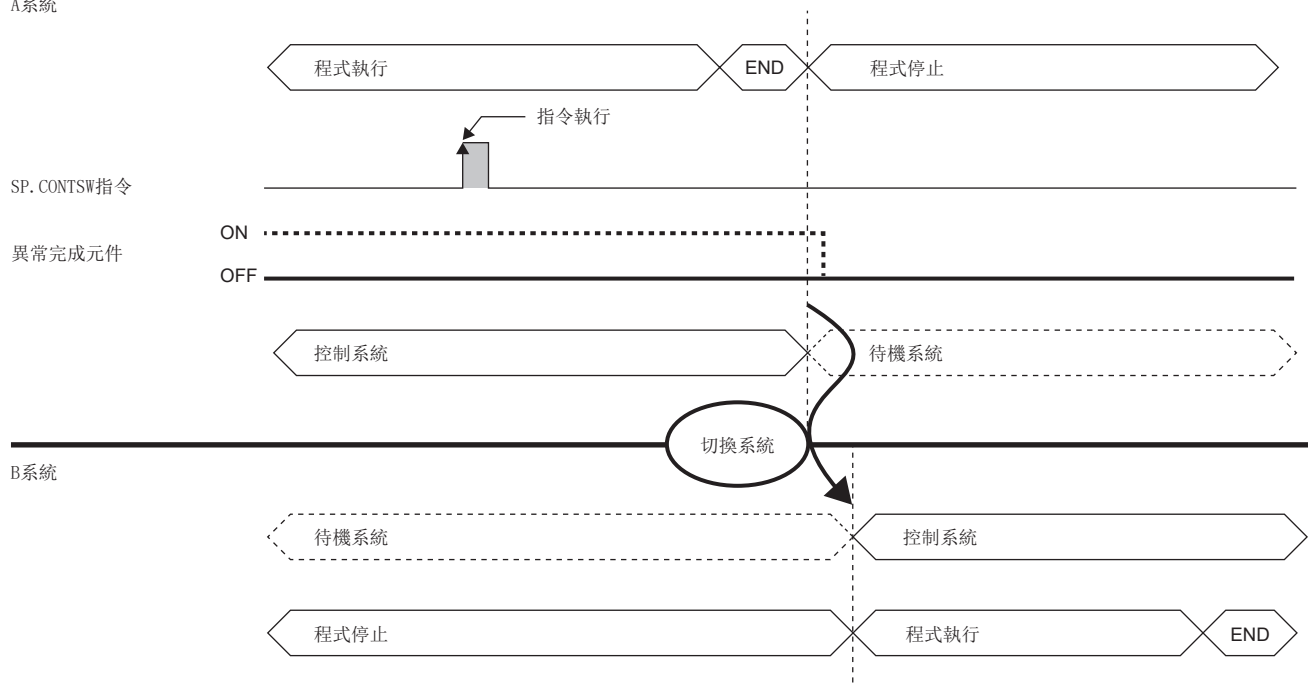
- 透過SP. CONTSW指令進行系統切換時，事先將控制系統的SM1646(用戶系統切換允許)設定為ON狀態。此外，透過DCONTSW指令設定為系統切換禁止狀態時，應在待機系統中執行ECONTSW指令、設定為系統切換允許狀態。
- 在系統切換正常完成時，(s)中指定的值被儲存至兩個系統的SD1650(系統切換指令識別編號)。程式內存在多個系統切換指令時，透過讀取SD1650，可以確認是透過哪個系統切換指令切換了系統。在同一掃描內執行了多個系統切換指令時，僅儲存最初執行的指令的引數。

- 在透過執行系統切換指令切換系統時，將(d)中指定的異常完成元件設定為OFF、無法切換時設定為ON。系統不能切換原因導致無法執行系統切換時，16(系統切換指令發出的系統切換請求)被儲存至控制系統的SD1643(系統切換原因)，系統不能切換原因No. 被儲存至SD1644(系統不能切換原因)。關於SD1644中儲存的系統不能切換原因No.，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

- 透過SP. CONTSW指令切換系統時

A系統



- 執行SP. CONTSW指令時SM1646為OFF時

A系統



B系統

注意事項

- 執行SP. CONTSW指令時，應將SM1646(用戶系統切換允許)設定為ON(允許)。設定為OFF(禁止)時，無法執行透過SP. CONTSW指令的系統切換。
- 在SM1646為ON(允許)的狀態下執行SP. CONTSW指令後，實際上即使在執行系統切換處理到END處理期間將SM1646設定為OFF狀態，系統切換處理也將被執行。

出錯

出錯代碼 (SD0)	內容
1BD0H	END處理時，系統不能切換原因導致無法進行系統切換時。
1BD1H	指令執行時，SM1646(用戶系統切換允許)為OFF(禁止)時。

29.2 系統切換禁止/允許

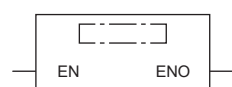
DCONTSW、ECONTSW



- DCONTSW：禁止系統切換(用戶切換)。
- ECONTSW：允許系統切換(用戶切換)。

梯形圖	ST
	ENO:=DCONTSW (EN) ; ENO:=ECONTSW (EN) ;

FBD/LD



■執行條件

指令	執行條件
DCONTSW	
ECONTSW	

功能

- 若在待機系統中執行DCONTSW指令，則控制系統變為系統切換禁止狀態。
- 若在待機系統中執行ECONTSW指令，則控制系統變為系統切換允許狀態。
- 初始狀態為系統切換允許狀態。
- DCONTSW指令/ECONTSW指令僅可在待機系統中執行。即使在控制系統中執行DCONTSW指令/ECONTSW指令，也將變為無處理。

要點

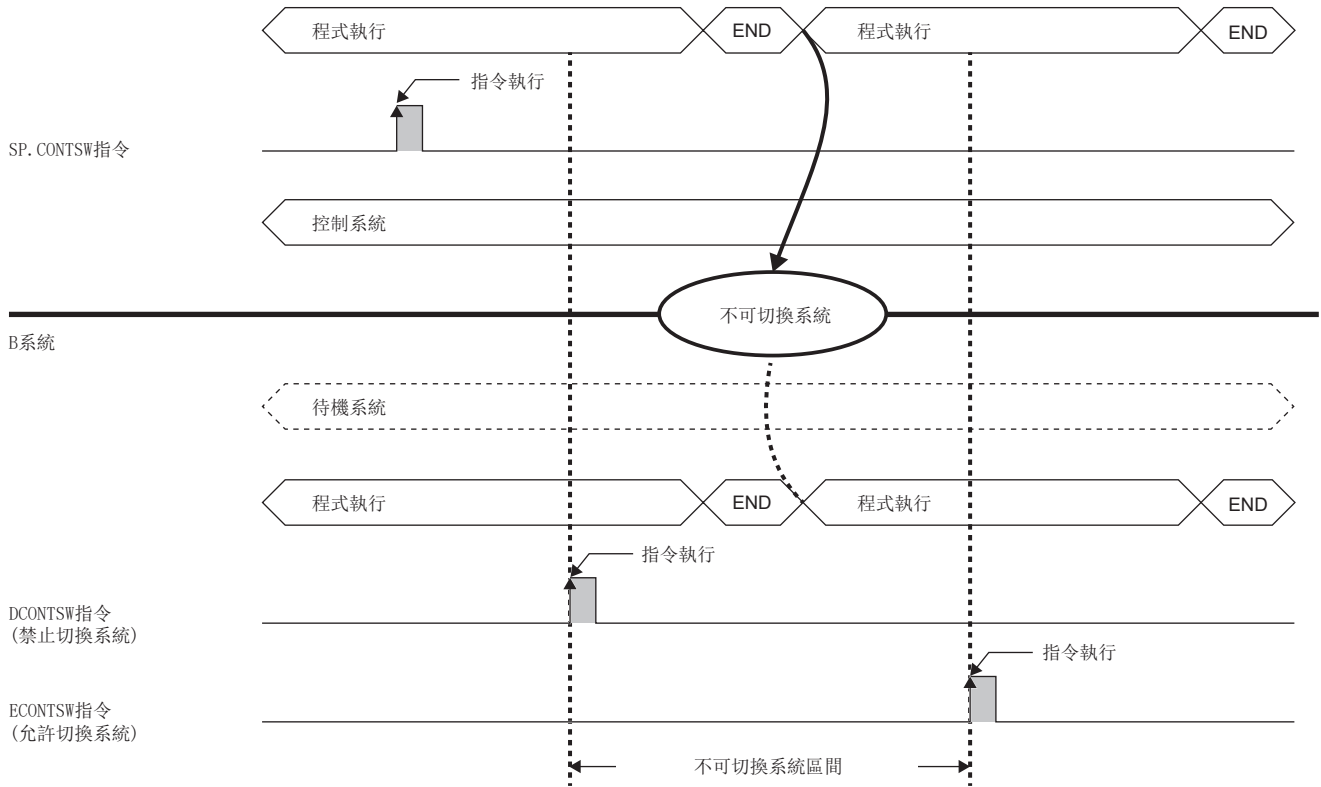
備份模式時，為了在待機系統中執行DCONTSW指令/ECONTSW指令，需要將指令記述至兩個系統執行程式、或兩個系統執行程式中調用的程式部件中。關於程式的兩個系統執行功能的詳細內容，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

例

透過DCONTSW指令禁止系統切換時，透過SP.CONTSW指令執行系統切換的情況

A系統



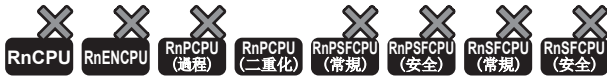
- 下述情況為系統切換允許狀態。想要返回系統切換禁止時，應再次執行DCONTSW指令。
- 待機系統的電源OFF→ON
- 待機系統CPU模組的重設
- 待機系統CPU模組的RUN→STOP
- 待機系統CPU模組的停止出錯發生
- 運行模式切換(備份模式→分離模式)
- 系統切換(系統切換)發生

出錯

沒有運算出錯。

29.3 由待機系統寫入資料至控制系統

CONTWR (P)



• 過程CPU(二重化模式)在韌體版本為“18”及其以後的版本可以使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

雙系統執行程式中，由待機系統寫入資料至控制系統。

梯形圖	ST
	<pre>ENO:=CONTWR(EN, s1, s2, d1, d2); ENO:=CONTWRP(EN, s1, s2, d1, d2);</pre>

FBD/LD

■執行條件

指令	執行條件
CONTWR	
CONTWRP	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s1)	儲存控制資料的起始元件	—	字元	ANY16_ARRAY (要素數: 2)
(s2)	儲存寫入資料的起始元件	—	字元	ANY16
(d1)	寫入目標(控制系統)的起始元件	—	字元	ANY16
(d2)	透過指令完成將1次掃描後變為ON的元件 異常完成時(d2)+1也將變為ON。	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

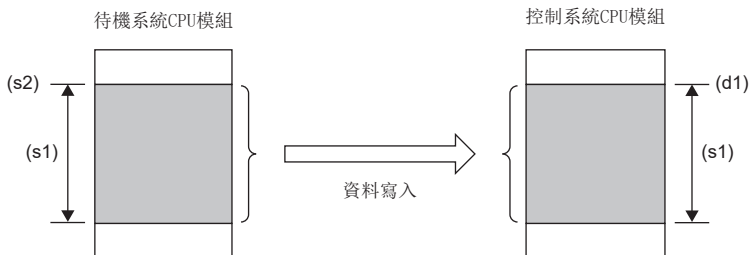
操作數	位元		字元		雙字元			間接指定	常數				其他
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□、U3E□\ (H)G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d2)	○	—	○	—	—	—	—	○	—	—	—	—	

■控制資料

操作數：(s1)				
元件	項目	內容	設定範圍	設定方
+0	寫入資料點數	將寫入資料點數以字元單位進行設定。	1~ 1024	用戶
+1	完成狀態	儲存指令完成時的狀態。 • 0：正常完成 • 0以外：異常完成(無法寫入資料的原因)	—	系統

功能

- 從(s2)中指定的待機系統的元件/標籤中, 將(s1)中指定的寫入資料點數的資料寫入至(d1)中指定的控制系統的元件/標籤。至控制系統的資料寫入完成時, (d2)中指定的待機系統的完成元件將變為ON。

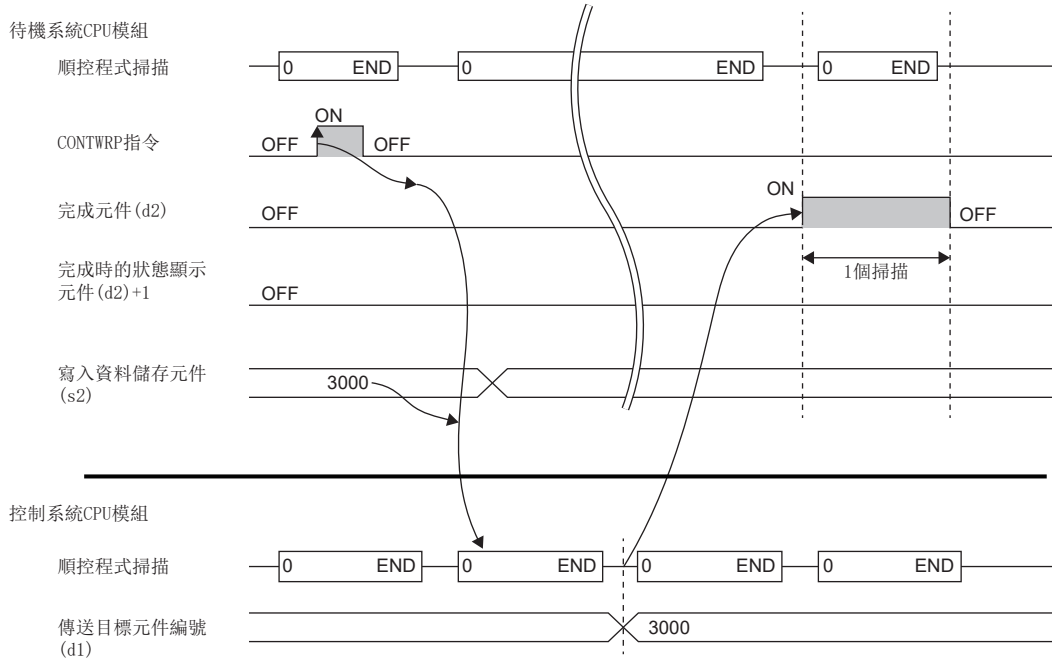


- CONTWR(P) 指令僅能在待機系統中執行。即使在控制系統中執行本指令, 也將變為無處理。
- 多次執行了CONTWR(P) 指令的情況下, 在第1次執行的指令完成前, 第2次及其以後執行的指令將設為無處理。
- CONTWR(P) 指令執行開始到完成前發生系統切換的情況下, 將不中止指令且繼續執行。系統切換後新控制系統的完成元件(d2)將設為ON。
- 對於CONTWR(P) 指令的正常/異常完成, 可透過設定資料中指定的完成元件(d2)、完成時的狀態顯示元件(d2)+1進行確認。異常完成時, 完成狀態(s1)+1中將儲存無法寫入資料的原因。

項目	內容
完成元件(d2)	在CONTWR(P) 指令完成的掃描的END處理中設為ON, 在下一個END處理中設為OFF。
完成時的狀態顯示元件(d2)+1	根據CONTWR(P) 指令完成後的狀態而變為ON/OFF。 正常完成時: 保持為OFF不變。 異常完成時: 在CONTWR(P) 指令完成的掃描的END處理中設為ON, 在下一個END處理中設為OFF。
完成狀態(s1)+1	下述的無法寫入資料的原因將被儲存。 0: 寫入正常完成 1: 無法追蹤通訊 2: 其他系統為過程模式 3: 控制系統不支援CONTWR(P) 指令 4: 元件/標籤的分割與控制系統不一致 5: 控制系統的元件/標籤超出元件/標籤範圍

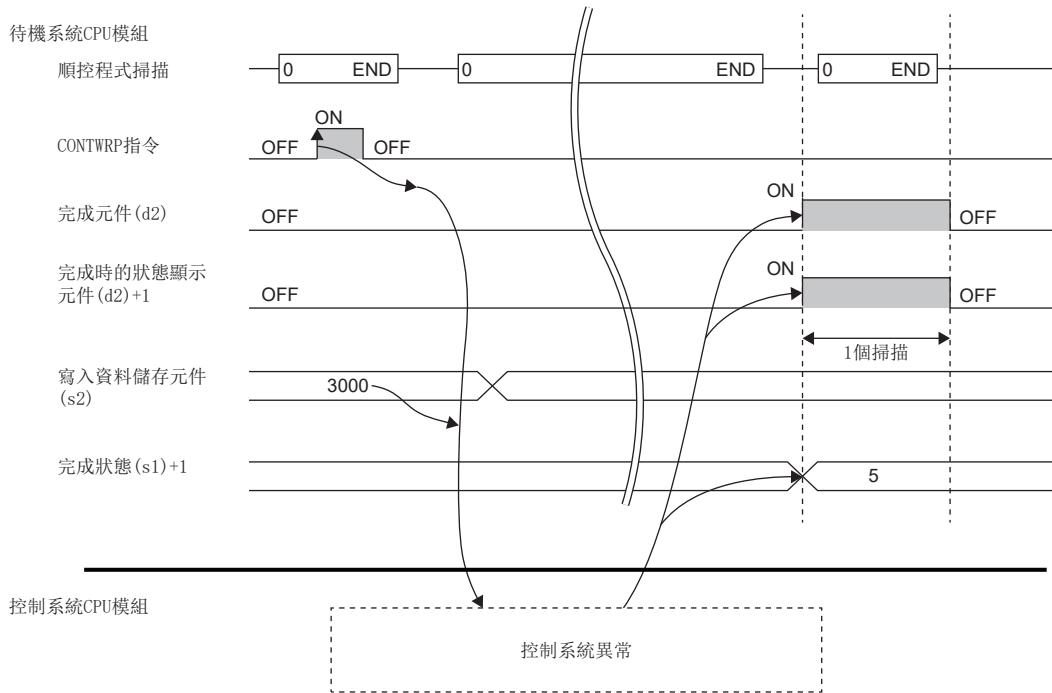
例

正常完成的情況下



例

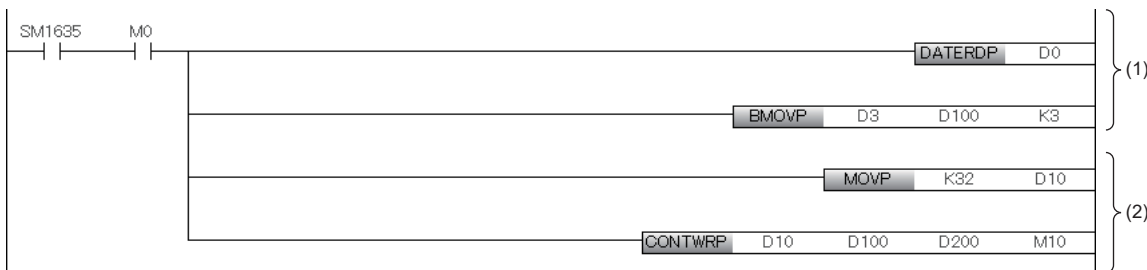
在控制系統的元件/標籤範圍外，異常完成的情況下 (完成狀態 (s1)+1 = 5)



- 由待機系統寫入至控制系統寫入的資料中，包含了指令執行時的時間資訊，因此可透過控制系統確認最後CONTWR(P)指令正常完成的時間。

例

起始中包含了時間資訊的資料，寫入至控制系統的情況下

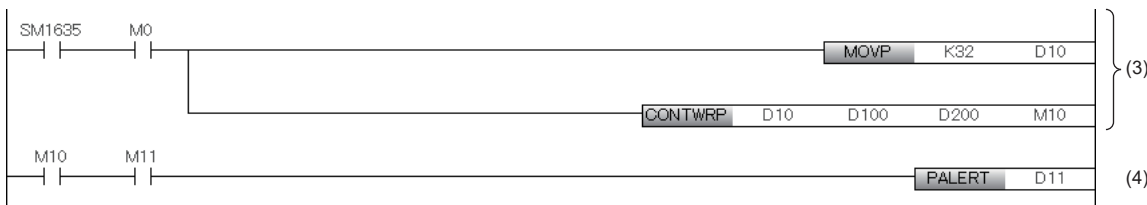


- 透過DATERDP指令將取得的時間資訊(時、分、秒)儲存至D100~D102(寫入至控制系統資料的起始)。
- 透過CONTWRP指令將待機系統的D100~D131的資料寫入至控制系統的D200~D231。

- CONTWR(P)指令異常完成時，可透過PALERT(P)指令使其發生繼續執行錯誤，通知異常完成。待機系統中執行的PALERT(P)指令的出錯代碼及詳細資訊2，可以透過控制系統的SD1681(最新自我診斷出錯代碼(其它系統))和SD1722~SD1752(詳細資訊2(其它系統))確認。(但是僅限於最新出錯)

例

異常完成時與無法寫入資料的原因合併通知的情況下



- 透過CONTWRP指令將待機系統的D100~D131的資料寫入至控制系統的D200~D231。
- CONTWRP指令異常完成時，將執行PALERT指令，D11(無法寫入資料的原因)儲存至出錯代碼1810H的詳細資訊2。

注意事項

- (s1)與(s2)，應使用非跟蹤傳送對象的元件/標籤，或是區域元件/區域標籤中。指定跟蹤傳送對象的元件/標籤的情況下，可能有控制系統的資料被待機系統覆蓋，發生預料之外的動作情形。
- 應在雙系統的檔案為相同狀態下執行CONTWR(P)指令。雙系統的檔案不在相同狀態下時執行指令，將可能發生指令異常完成的情況。
- 在(d1)指定下述的元件/標籤的情況下，資料的寫入目標將根據指令執行時的待機系統的狀態決定。

項目	內容
檔案暫存器(R10等)	執行指令時，對與待機系統正在使用的檔案暫存器的檔案名稱及塊No.相同控制系統的檔案暫存器，執行資料寫入。在控制系統進行資料寫入時，沒有相同名稱的檔案暫存器檔案、或是檔案大小不一致的情況下，指令將異常完成。
變址修飾(W100Z0等)	由指令執行時的待機系統的變址暫存器中儲存的值，決定控制系統寫入目標的元件。
間接指定(@D2000等)	由指令執行時的待機系統的元件中儲存的間接位址，決定控制系統寫入目標的元件。
標籤數組(wLabe11[D0]等)	由指令執行時的待機系統的數組的添加文字中儲存的值，決定控制系統寫入目標的標籤。

- 透過寫入至PLC、RUN中寫入、SLMP、FTP伺服器功能，在檔案寫入中執行CONTWR(P)指令或是在CONTWR(P)指令執行中進行檔案寫入的情況下，若雙系統的元件/標籤分配的不一致，可能有指令異常完成的情況。異常完成的情況下，檔案寫入後應再度執行指令。

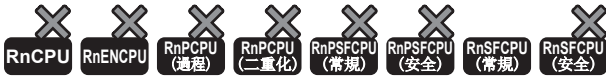
出錯

出錯代碼(SD0)	內容
3405H	寫入資料點數(s1)+0超出了1~1024的範圍。

30 安全系統用指令

30.1 安全資料一致性檢查資訊讀取指令

SP.SIDRD



• RnSFCPU(常規)可以在韌體版本為“15”及其以後的版本使用。工程工具可以在版本為“1.050C”及其以後的版本使用。

讀取指定檔案中的安全資料一致性檢查用識別元。

梯形圖	ST
	ENO:=SP_SIDRD (EN, s, d1, d2, d3);

FBD/LD

■執行條件

指令	執行條件
SP.SIDRD	

設置數據

■內容、範圍、資料類型

操作數	內容	範圍	資料類型	資料類型(標籤)
(s)	儲存檔案名稱的起始元件	—	Unicode字元串	ANYSTRING_DOUBLE
(d1)	儲存完成狀態的元件 0000: 正常完成 0000以外: 異常完成(出錯代碼)	—	字元	ANY16
(d2)	儲存讀取的資料識別元的起始元件	—	雙字元	ANY32
(d3)	處理完成時變為ON的位元元件 (但是, 異常完成時(d3)+1也將變為ON)	—	位元	ANYBIT_ARRAY (要素數: 2)
EN	執行條件	—	位元	BOOL
ENO	執行結果	—	位元	BOOL

■可使用的元件

操作數	位元		字元		雙字元			間接指定	常數				其它 (U)
	X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	T、ST、C、D、 W、SD、SW、FD、 R、ZR、RD	U□\G□、J□\□、 U3E□\ (H)G□	Z	LT、 LST、LC	LZ		K	H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d3)	○	—	○	—	—	—	—	○	—	—	—	—	

功能

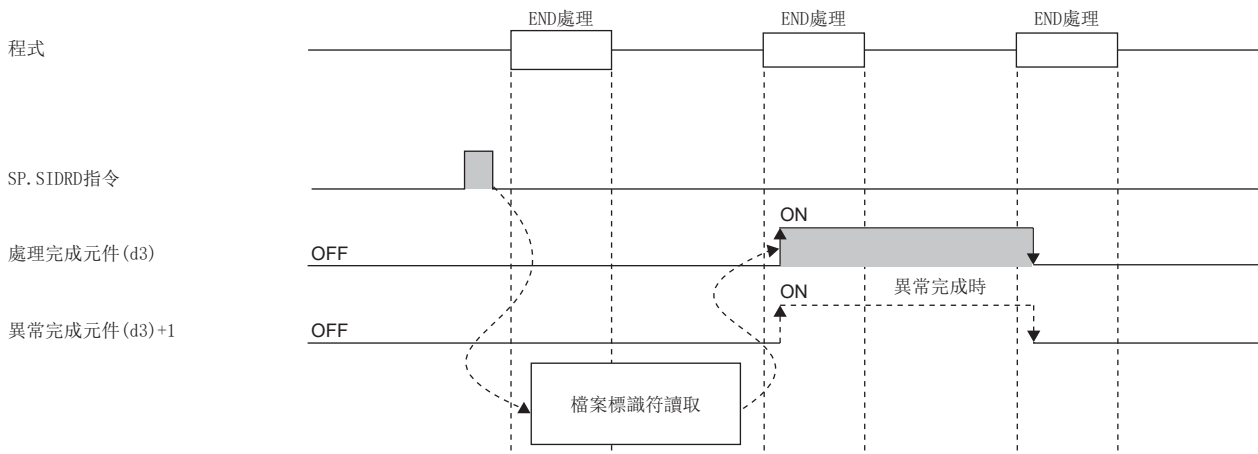
- 讀取在 (s) 指定檔案中的安全資料一致性檢查用識別元，並儲存至在 (d2) 指定的元件中。在 (s) 指定的檔案名的字元數，應包含副檔名不超過64字元。安全資料一致性檢查用識別元的詳細內容，請參閱下述手冊。

📖 MELSEC iQ-R CPU模組用戶手冊(應用篇)

- (s) 中可指定的檔案如下所示。

檔案類型	檔案名	副檔名
安全程式	任意	SPG
安全FB檔案	任意	SPB
安全CPU參數	CPU	SPR
安全模組參數	UNIT	

- 處理完成 (d3) 的位元元件，在檢測出 SP.SIDRD 指令的處理完成掃描的 END 指令執行時將被自動設為 ON，透過下一個掃描的 END 指令變為 OFF。
- 當 SP.SIDRD 指令異常完成時，異常完成 (d3)+1 的元件將與處理完成 (d3) 的元件在相同的時機設為 ON/OFF。
- 在執行指令時檢測到運算出錯的情況下，處理完成 (d3)、異常完成 (d3)+1 將不變為 ON。
- SP.SIDRD 指令執行中，SM753 (檔案存取中) 將變為 ON。在 SM753 已處於 ON 狀態的情況下，不能執行 SP.SIDRD 指令。(執行時將變為無處理。)
- 執行 SP.SIDRD 指令時的完成元件的動作如下所示。



注意事項

- 請勿透過中斷程式執行 SP.SIDRD 指令。若透過中斷程式執行，可能導致誤動作。
- 請勿同時執行 SP.SIDRD 指令與檔案存取的其他功能。
- 指令執行中即使將 CPU 模組的狀態更改為 RUN→STOP，本指令的處理仍會繼續進行。

出錯

出錯代碼 (SD0)	內容
3405H	無法讀取 (s) 中指定的檔案名稱字元串時。 <ul style="list-style-type: none"> 檔案名稱未指定任何字元。 檔案名稱指定了65字元及其以上。

在安全系統用指令中發生的出錯代碼

安全系統用指令的完成狀態 (d1) 中儲存的出錯代碼的一覽如下所示。

出錯代碼 (d1)	出錯內容	處理方法
8000H	指定的檔案不存在。	確認指定的檔案是否存在。
8001H	指定的檔案的副檔名不正確。	確認指定的檔案的副檔名是否存在。
8002H	檔案存取的其他功能正在執行中。	應在檔案存取的其他功能完成後，再次執行指令。
8004H	指定的檔案已損壞。	應再次將指定的檔案寫入至CPU模組中。

第6部分 模組專用指令

本部分由下述章節構成。

31 模組專用指令

31 模組專用指令

關於模組專用指令的詳細內容，請參閱下述手冊。

📖 MELSEC iQ-R 程式手冊(模組專用指令篇)

備忘錄

第7部分 通用FUN

本部分由下述章節構成。

32 類型轉換函數

33 單數值變數函數

34 算術運算函數

35 位移函數

36 位型布爾函數

37 選擇函數

38 比較函數

39 字元串函數

40 時間資料類型函數

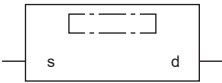
32 類型轉換函數

32.1 BOOL型→WORD型轉換

BOOL_TO_WORD(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將BOOL型資料轉換為WORD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p> 	<p>[無EN/ENO] d:=BOOL_TO_WORD(s);</p> <p>[帶EN/ENO] d:=BOOL_TO_WORD_E(EN, ENO, s);</p>

設置數據

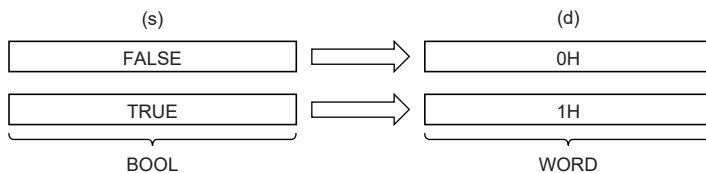
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	BOOL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	WORD

功能

■運算處理

- 將(s)中輸入的BOOL型的資料轉換為WORD型的資料後從(d)中輸出。
- 輸入值為FALSE的情況下，以WORD型的資料值輸出0H。
- 輸入值為TRUE的情況下，以WORD型的資料值輸出1H。



- 至(s)的輸入值為BOOL型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

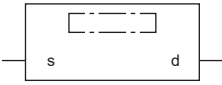
沒有運算出錯。

32.2 BOOL型→DWORD型轉換

BOOL_TO_DWORD(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCEPU (常規) RnSFCEPU (安全)

將BOOL型資料轉換為DWORD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p> 	<p>[無EN/ENO] d:=BOOL_TO_DWORD(s);</p> <p>[帶EN/ENO] d:=BOOL_TO_DWORD_E(EN, ENO, s);</p>

設置數據

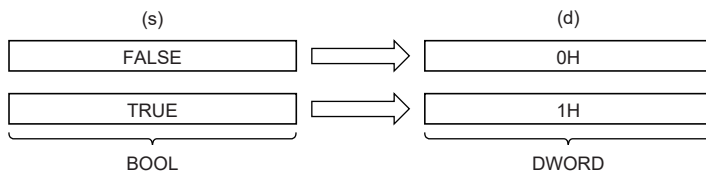
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	BOOL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DWORD

功能

■運算處理

- 將(s)中輸入的BOOL型的資料轉換為DWORD型的資料後從(d)中輸出。
- 輸入值為FALSE的情況下，以DWORD型的資料值輸出0H。
- 輸入值為TRUE的情況下，以DWORD型的資料值輸出1H。



- 至(s)的輸入值為BOOL型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

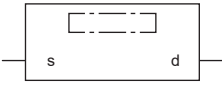
沒有運算出錯。

32.3 BOOL型→INT型轉換

BOOL_TO_INT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將BOOL型資料轉換為INT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p> 	<p>[無EN/ENO] d:=BOOL_TO_INT(s);</p> <p>[帶EN/ENO] d:=BOOL_TO_INT_E(EN, ENO, s);</p>

設置數據

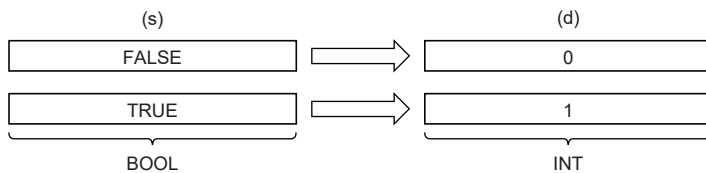
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	BOOL
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 將(s)中輸入的BOOL型的資料轉換為INT型的資料後從(d)中輸出。
- 輸入值為FALSE的情況下，以INT型的資料值輸出0。
- 輸入值為TRUE的情況下，以INT型的資料值輸出1。



- 至(s)的輸入值為BOOL型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32.4 BOOL型→DINT型轉換

BOOL_TO_DINT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將BOOL型資料轉換為DINT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=BOOL_TO_DINT(s);</p> <p>[帶EN/ENO] d:=BOOL_TO_DINT_E(EN, ENO, s);</p>

設置數據

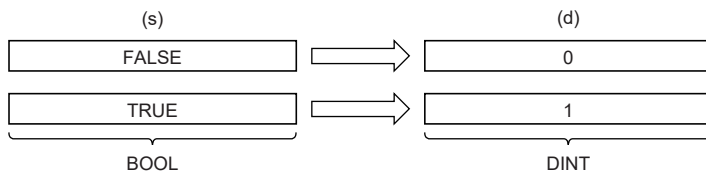
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	BOOL
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出	輸出變數	DINT

功能

■運算處理

- 將(s)中輸入的BOOL型的資料轉換為DINT型的資料後從(d)中輸出。
- 輸入值為FALSE的情況下，以DINT型的資料值輸出0。
- 輸入值為TRUE的情況下，以DINT型的資料值輸出1。



- 至(s)的輸入值為BOOL型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32.5 BOOL型→TIME型轉換

BOOL_TO_TIME(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將BOOL型資料轉換為TIME型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=BOOL_TO_TIME(s);</p> <p>[帶EN/ENO] d:=BOOL_TO_TIME_E(EN, ENO, s);</p>

設置數據

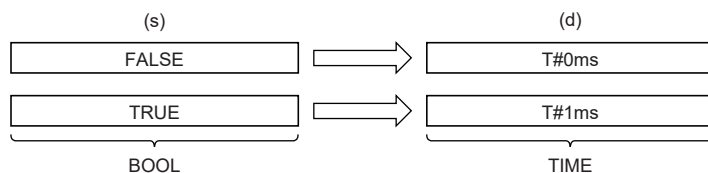
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(IN)	輸入	輸入變數	BOOL
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出	輸出變數	TIME

功能

■運算處理

- 將(s)中輸入的BOOL型的資料轉換為TIME型的資料後從(d)中輸出。
- 輸入值為FALSE的情況下，以TIME型的資料值輸出0。
- 輸入值為TRUE的情況下，以TIME型的資料值輸出1。



- 至(s)的輸入值為BOOL型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

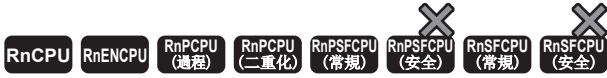
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32.6 BOOL型→STRING型轉換

BOOL_TO_STRING(_E)



將BOOL型資料轉換為STRING型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=BOOL_TO_STRING(s);</p> <p>[帶EN/ENO] d:=BOOL_TO_STRING_E(EN, ENO, s);</p>

設置數據

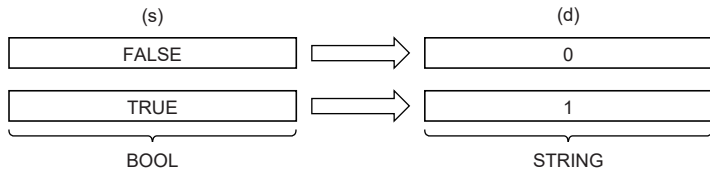
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	BOOL
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出	輸出變數	STRING

功能

■運算處理

- 將(s)中輸入的BOOL型的資料轉換為STRING型的資料後從(d)中輸出。
- 輸入值為FALSE的情況下，以STRING型的資料值輸出0。
- 輸入值為TRUE的情況下，以STRING型的資料值輸出1。



- 至(s)的輸入值為BOOL型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

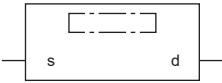
沒有運算出錯。

32.7 WORD型→BOOL型轉換

WORD_TO_BOOL(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將WORD型資料轉換為BOOL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p> 	<p>[無EN/ENO] d:=WORD_TO_BOOL(s);</p> <p>[帶EN/ENO] d:=WORD_TO_BOOL_E(EN, ENO, s);</p>

設置數據

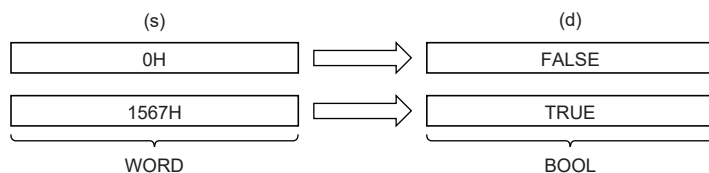
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	WORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	BOOL

功能

■運算處理

- 將(s)中輸入的WORD型的資料轉換為BOOL型的資料後從(d)中輸出。
- 輸入值為0H的情況下，輸出FALSE。
- 輸入值為0H以外的情況下，輸出TRUE。



- 至(s)的輸入值為WORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32.8 WORD型→DWORD型轉換

WORD_TO_DWORD(_E)



將WORD型資料轉換為DWORD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=WORD_TO_DWORD(s);</p> <p>[帶EN/ENO] d:=WORD_TO_DWORD_E(EN, ENO, s);</p>

設置數據

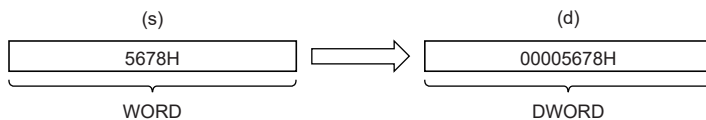
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(IN)	輸入	輸入變數	WORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DWORD

功能

■運算處理

- 將(s)中輸入的WORD型的資料轉換為DWORD型的資料後從(d)中輸出。
- 資料轉換後，高位16位元將變為0。



- 至(s)的輸入值為WORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32.9 WORD型→INT型轉換

WORD_TO_INT(_E)



將WORD型資料轉換為INT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=WORD_TO_INT(s);</p> <p>[帶EN/ENO] d:=WORD_TO_INT_E(EN, ENO, s);</p>

設置數據

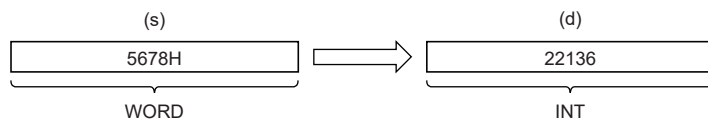
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	WORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 將(s)中輸入的WORD型的資料轉換為INT型的資料後從(d)中輸出。



- 至(s)的輸入值為WORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

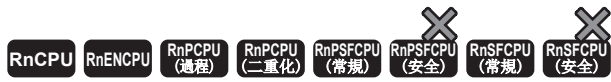
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 10 WORD型→DINT型轉換

WORD_TO_DINT(_E)



將WORD型資料轉換為DINT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=WORD_TO_DINT(s);</p> <p>[帶EN/ENO] d:=WORD_TO_DINT_E(EN, ENO, s);</p>

設置數據

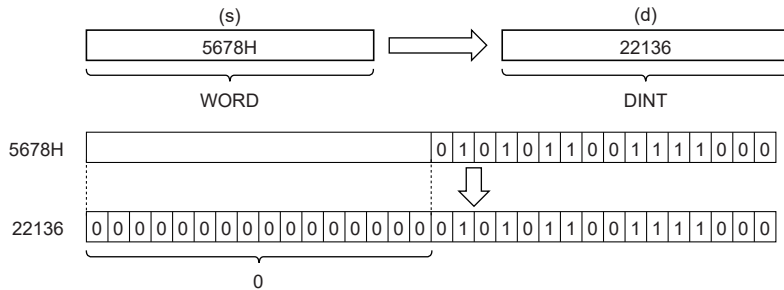
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	WORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DINT

功能

■運算處理

- 將(s)中輸入的WORD型的資料轉換為DINT型的資料後從(d)中輸出。
- 資料轉換後，高位16位元將變為0。



- 至(s)的輸入值為WORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

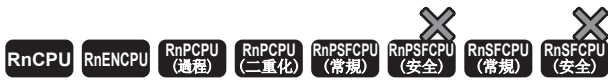
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 11 WORD型→TIME型轉換

WORD_TO_TIME(_E)



將WORD型資料轉換為TIME型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=WORD_TO_TIME(s);</p> <p>[帶EN/ENO] d:=WORD_TO_TIME_E(EN, ENO, s);</p>

設置數據

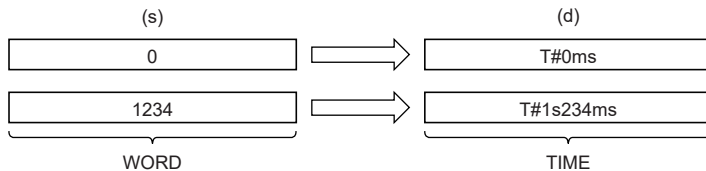
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	WORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	TIME

功能

■運算處理

- 將(s)中輸入的WORD型的資料轉換為TIME型的資料後從(d)中輸出。



- 至(s)的輸入值為WORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 12 WORD型→STRING型轉換

WORD_TO_STRING(_E)



將WORD型資料轉換為STRING型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=WORD_TO_STRING(s);</p> <p>[帶EN/ENO] d:=WORD_TO_STRING_E(EN, ENO, s);</p>

設置數據

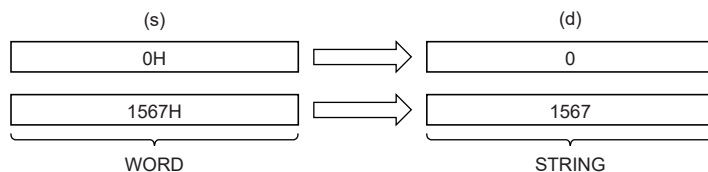
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	WORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	STRING (4)

功能

■運算處理

- 將(s)中輸入的WORD型的資料轉換為STRING型的資料後從(d)中輸出。



- 至(s)的輸入值為WORD型的資料值。
- SM701(輸出字數切換)為OFF的情況下, 字元串的最後將儲存00H。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下, 從(d)中輸出的資料將變為不定值。在此情況下, 應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 13 DWORD型→BOOL型轉換

DWORD_TO_BOOL(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將DWORD型資料轉換為BOOL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DWORD_TO_BOOL(s);</p> <p>[帶EN/ENO] d:=DWORD_TO_BOOL_E(EN, ENO, s);</p>

設置數據

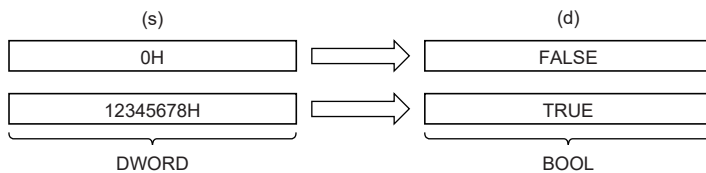
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DWORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	BOOL

功能

■運算處理

- 將(s)中輸入的DWORD型的資料轉換為BOOL型的資料後從(d)中輸出。
- 輸入值為0H的情況下，輸出FALSE。
- 輸入值為0H以外的情況下，輸出TRUE。



- 至(s)的輸入值為DWORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 14 DWORD型→WORD型轉換

DWORD_TO_WORD(_E)



將DWORD型資料轉換為WORD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DWORD_TO_WORD(s);</p> <p>[帶EN/ENO] d:=DWORD_TO_WORD_E(EN, ENO, s);</p>

設置數據

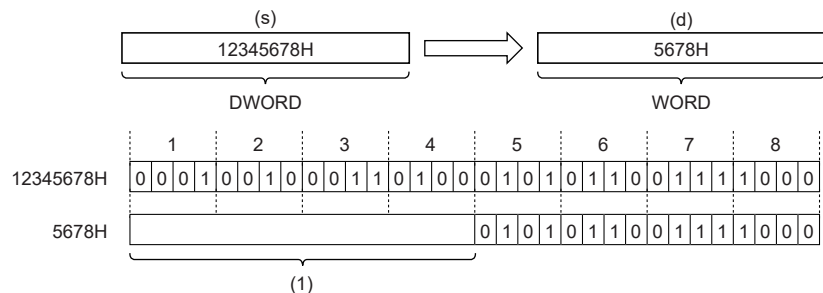
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DWORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	WORD

功能

■運算處理

- 將(s)中輸入的DWORD型的資料轉換為WORD型的資料後從(d)中輸出。
- 輸入值的高位16位元的資訊將被刪除。(如下所示(1))



- 至(s)的輸入值為DWORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

要點

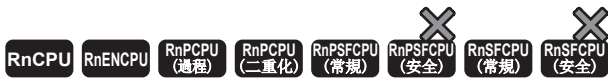
執行DWORD_TO_WORD(_E)時，從(s)輸入的DWORD型資料值的高位16位元的資訊將被刪除。

出錯

沒有運算出錯。

32. 15 DWORD型→INT型轉換

DWORD_TO_INT(_E)



將DWORD型資料轉換為INT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DWORD_TO_INT(s);</p> <p>[帶EN/ENO] d:=DWORD_TO_INT_E(EN, ENO, s);</p>

設置數據

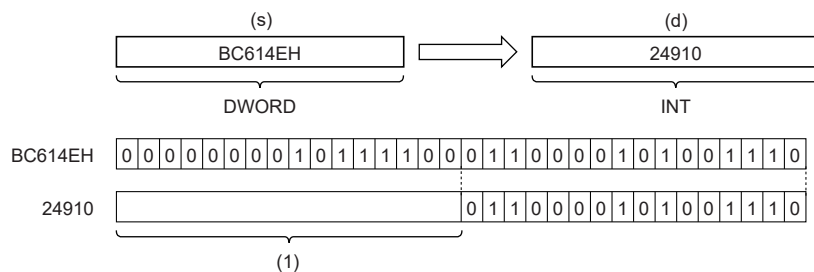
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DWORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 將(s)中輸入的DWORD型的資料轉換為INT型的資料後從(d)中輸出。
- 輸入值的高位16位元的資訊將被刪除。(如下所示(1))



- 至(s)的輸入值為DWORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

要點

執行DWORD_TO_INT(_E)時，從(s)輸入的DWORD型資料值的高位16位元的資訊將被刪除。

出錯

沒有運算出錯。

32. 16 DWORD型→DINT型轉換

DWORD_TO_DINT(_E)



將DWORD型資料轉換為DINT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DWORD_TO_DINT(s);</p> <p>[帶EN/ENO] d:=DWORD_TO_DINT_E(EN, ENO, s);</p>

設置數據

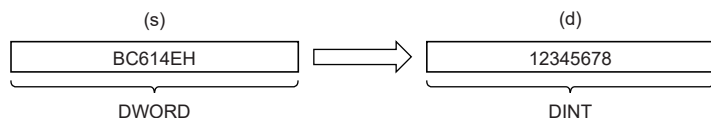
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DWORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DINT

功能

■運算處理

- 將(s)中輸入的DWORD型的資料轉換為DINT型的資料後從(d)中輸出。



- 至(s)的輸入值為DWORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 17 DWORD型→TIME型轉換

DWORD_TO_TIME(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFPCPU (常規) RnSFPCPU (安全)

將DWORD型資料轉換為TIME型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DWORD_TO_TIME(s);</p> <p>[帶EN/ENO] d:=DWORD_TO_TIME_E(EN, ENO, s);</p>

設置數據

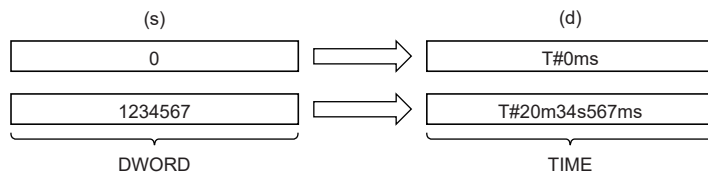
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(IN)	輸入	輸入變數	DWORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	TIME

功能

■運算處理

- 將(s)中輸入的DWORD型的資料轉換為TIME型的資料後從(d)中輸出。



- 至(s)的輸入值為DWORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 18 DWORD型→STRING型轉換

DWORD_TO_STRING(_E)



將DWORD型資料轉換為STRING型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DWORD_TO_STRING(s);</p> <p>[帶EN/ENO] d:=DWORD_TO_STRING_E(EN, ENO, s);</p>

設置數據

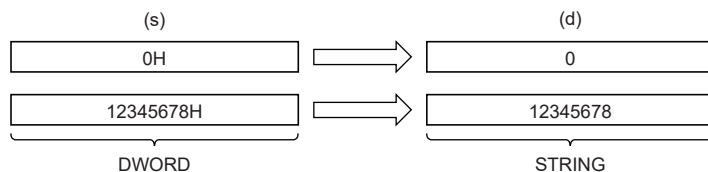
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DWORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	STRING (8)

功能

■運算處理

- 將(s)中輸入的DWORD型的資料轉換為STRING型的資料後從(d)中輸出。



- 至(s)的輸入值為DWORD型的資料值。
- SM701(輸出字數切換)為OFF的情況下, 字元串的最後將儲存00H。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下, 從(d)中輸出的資料將變為不定值。在此情況下, 應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 19 INT型→BOOL型轉換

INT_TO_BOOL(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFPCPU (常規) RnSFPCPU (安全)

將INT型資料轉換為BOOL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=INT_TO_BOOL(s);</p> <p>[帶EN/ENO] d:=INT_TO_BOOL_E(EN, ENO, s);</p>

設置數據

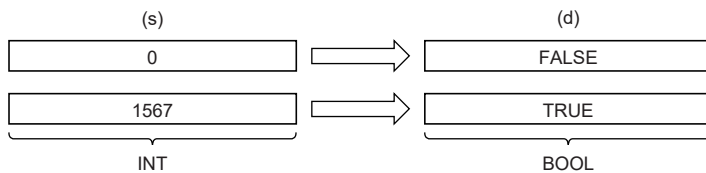
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	BOOL

功能

■運算處理

- 將(s)中輸入的INT型的資料轉換為BOOL型的資料後從(d)中輸出。
- 輸入值為0的情況下，輸出FALSE。
- 輸入值為0以外的情況下，輸出TRUE。



- 至(s)的輸入值為INT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 20 INT型→WORD型轉換

INT_TO_WORD(_E)



將INT型資料轉換為WORD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=INT_TO_WORD(s);</p> <p>[帶EN/ENO] d:=INT_TO_WORD_E(EN, ENO, s);</p>

設置數據

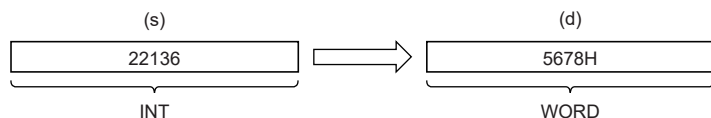
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	WORD

功能

■運算處理

- 將(s)中輸入的INT型的資料轉換為WORD型的資料後從(d)中輸出。



- 至(s)的輸入值為INT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

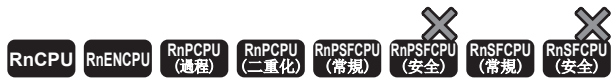
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 21 INT型→DWORD型轉換

INT_TO_DWORD(_E)



將INT型資料轉換為DWORD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=INT_TO_DWORD(s);</p> <p>[帶EN/ENO] d:=INT_TO_DWORD_E(EN, ENO, s);</p>

設置數據

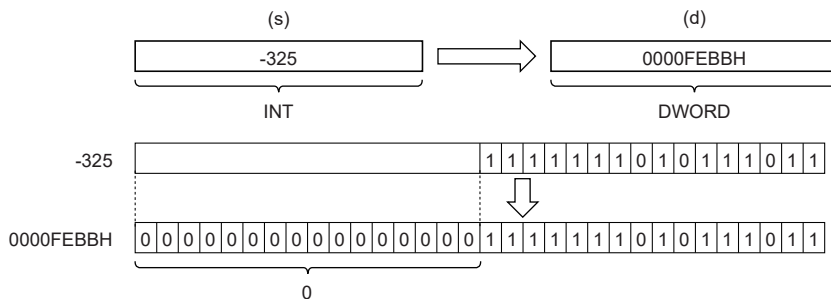
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DWORD

功能

■運算處理

- 將(s)中輸入的INT型的資料轉換為DWORD型的資料後從(d)中輸出。
- 資料轉換後，高位16位元將變為0。



- 至(s)的輸入值為INT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 22 INT型→DINT型轉換

INT_TO_DINT(_E)



將INT型資料轉換為DINT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] <code>d:=INT_TO_DINT(s);</code> [帶EN/ENO] <code>d:=INT_TO_DINT_E(EN, ENO, s);</code></p>

設置數據

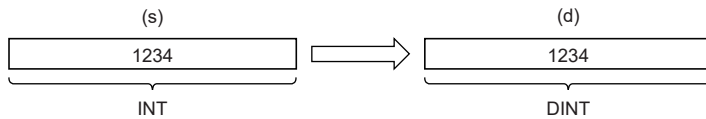
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DINT

功能

■運算處理

- 將(s)中輸入的INT型的資料轉換為DINT型的資料後從(d)中輸出。



- 至(s)的輸入值為INT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 23 INT型→BCD型轉換

INT_TO_BCD(_E)



將INT型資料轉換為BCD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=INT_TO_BCD(s);</p> <p>[帶EN/ENO] d:=INT_TO_BCD_E(EN, ENO, s);</p>

設置數據

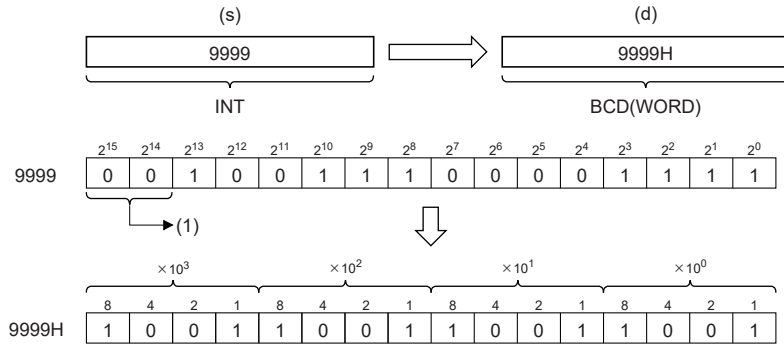
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	INT
ENO	輸出狀態 (TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	WORD

功能

■運算處理

- 將(s)中輸入的INT型資料轉換為BCD型(WORD型)資料後從(d)中輸出。



- (1): 必須設定為0。
- 至(s)的輸入值是INT類型的資料值，且在0~9999的範圍內。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼 (SD0)	內容
3401H	(s) 的資料超出0~9999的範圍時。

32.24 INT型→REAL型轉換

INT_TO_REAL(_E)



將INT型資料轉換為REAL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=INT_TO_REAL(s);</p> <p>[帶EN/ENO] d:=INT_TO_REAL_E(EN, ENO, s);</p>

設置數據

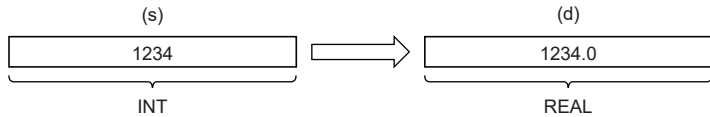
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	REAL

功能

■運算處理

- 將(s)中輸入的INT型的資料轉換為REAL型的資料後從(d)中輸出。



- 至(s)的輸入值為INT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

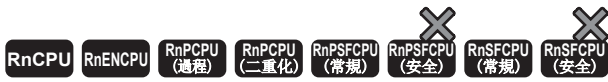
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 25 INT型→LREAL型轉換

INT_TO_LREAL(_E)



將INT型資料轉換為LREAL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=INT_TO_LREAL(s);</p> <p>[帶EN/ENO] d:=INT_TO_LREAL_E(EN, ENO, s);</p>

設置數據

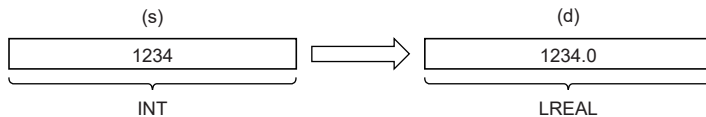
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	LREAL

功能

■運算處理

- 將(s)中輸入的INT型的資料轉換為LREAL型的資料後從(d)中輸出。



- 至(s)的輸入值為INT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

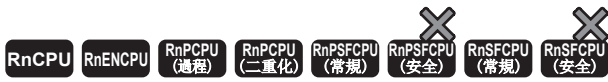
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 26 INT型→TIME型轉換

INT_TO_TIME(_E)



將INT型資料轉換為TIME型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=INT_TO_TIME(s);</p> <p>[帶EN/ENO] d:=INT_TO_TIME_E(EN, ENO, s);</p>

設置數據

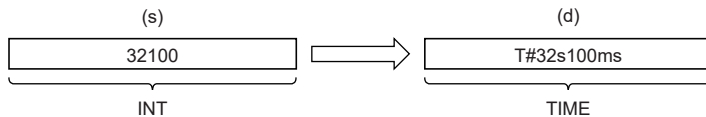
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	TIME

功能

■運算處理

- 將(s)中輸入的INT型的資料轉換為TIME型的資料後從(d)中輸出。



- 至(s)的輸入值為INT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

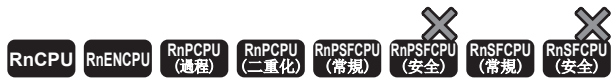
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 27 INT型→STRING型轉換

INT_TO_STRING(_E)



將INT型資料轉換為STRING型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=INT_TO_STRING(s);</p> <p>[帶EN/ENO] d:=INT_TO_STRING_E(EN, ENO, s);</p>

設置數據

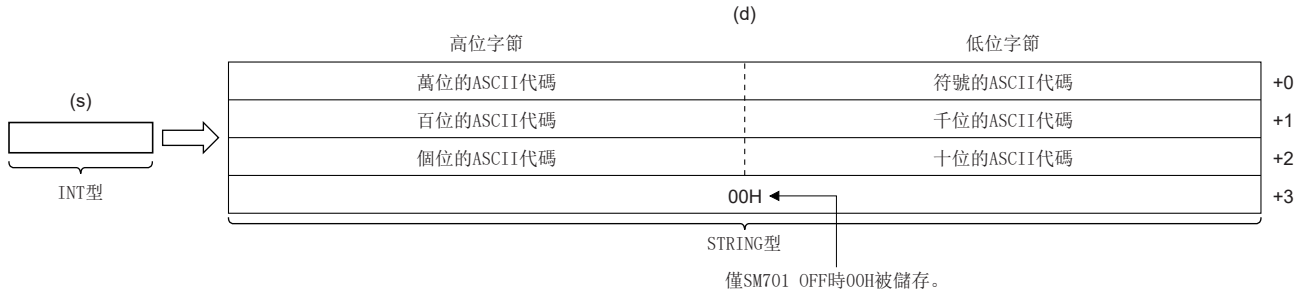
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	STRING (6)

功能

■運算處理

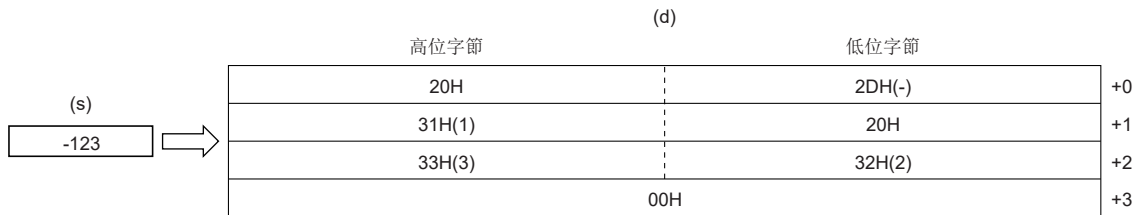
- 將(s)中輸入的INT型的資料轉換為STRING型的資料後從(d)中輸出。



- 至(s)的輸入值為INT型的資料值。
- 在符號資料中，輸入的值為正時儲存20H(空格)、為負時儲存2DH(-)。
- 有效位數較少的情況下，高位位中將儲存20H(空格)。

例

輸入了-123的情況下



- SM701 (輸出字數切換) 為OFF的情況下，字元串的最後(第4字)將儲存00H。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 28 DINT型→BOOL型轉換

DINT_TO_BOOL(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFPCPU (常規) RnSFPCPU (安全)

將DINT型資料轉換為BOOL型資料。

梯形圖、FBD/LD		ST
[無EN/ENO]		[無EN/ENO] d:=DINT_TO_BOOL(s);
[帶EN/ENO]		[帶EN/ENO] d:=DINT_TO_BOOL_E(EN, ENO, s);

設置數據

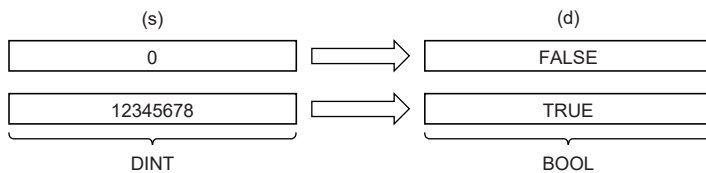
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DINT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	BOOL

功能

■運算處理

- 將(s)中輸入的DINT型的資料轉換為BOOL型的資料後從(d)中輸出。
- 輸入值為0的情況下，輸出FALSE。
- 輸入值為0以外的情況下，輸出TRUE。



- 至(s)的輸入值是DINT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

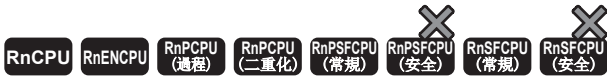
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 29 DINT型→WORD型轉換

DINT_TO_WORD(_E)



將DINT型資料轉換為WORD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DINT_TO_WORD(s);</p> <p>[帶EN/ENO] d:=DINT_TO_WORD_E(EN, ENO, s);</p>

設置數據

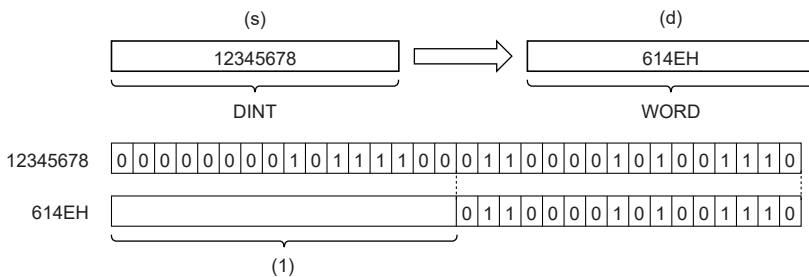
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DINT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	WORD

功能

■運算處理

- 將(s)中輸入的DINT型的資料轉換為WORD型的資料後從(d)中輸出。
- 輸入值的高位16位元的資訊將被刪除。(如下所示(1))



- 至(s)的輸入值是DINT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

要點

執行DINT_TO_WORD(_E)時，從(s)輸入的DINT型資料值的高位16位的資訊將被刪除。

出錯

沒有運算出錯。

32. 30 DINT型→DWORD型轉換

DINT_TO_DWORD(_E)



將DINT型資料轉換為DWORD型資料。

梯形圖、FBD/LD		ST
[無EN/ENO]		[無EN/ENO] d:=DINT_TO_DWORD(s);
[帶EN/ENO]		[帶EN/ENO] d:=DINT_TO_DWORD_E(EN, ENO, s);

設置數據

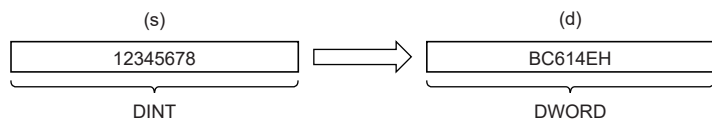
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DINT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DWORD

功能

■運算處理

- 將(s)中輸入的DINT型的資料轉換為DWORD型的資料後從(d)中輸出。



- 至(s)的輸入值是DINT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 31 DINT型→INT型轉換

DINT_TO_INT(_E)



將DINT型資料轉換為INT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DINT_TO_INT(s);</p> <p>[帶EN/ENO] d:=DINT_TO_INT_E(EN, ENO, s);</p>

設置數據

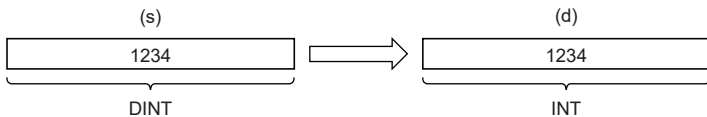
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DINT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 將(s)中輸入的DINT型的資料轉換為INT型的資料後從(d)中輸出。



- 至(s)的輸入值是DINT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3401H	(s)中設定的帶符號BIN32位元資料超出-32768~32767的範圍時。

32. 32 DINT型→BCD型轉換

DINT_TO_BCD(_E)



將DINT型資料轉換為BCD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DINT_TO_BCD(s);</p> <p>[帶EN/ENO] d:=DINT_TO_BCD_E(EN, ENO, s);</p>

設置數據

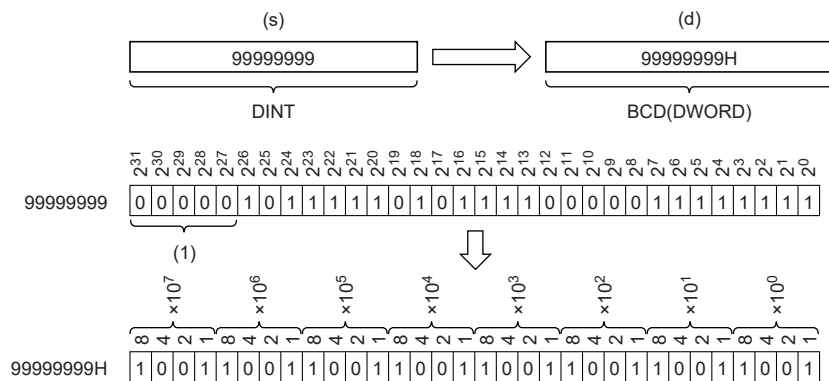
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DINT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_BIT

功能

■運算處理

- 將(s)中輸入的DINT型資料轉換為BCD型(DWORD型)資料後從(d)進行輸出。



(1): 必須設定為0。

- 至(s)的輸入值是DINT型的資料值。(d)為WORD的情況下，輸入值的範圍為0~9999。(d)為DWORD型的情況下，輸入值的範圍為0~99999999。
- (d)中可以指定WORD型、DWORD型。不能指定BOOL型。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

- (d)為WORD型的情況下

出錯代碼(SD0)	內容
3401H	(s)中設定的帶符號BIN32位元資料超出-32768~32767的範圍時。
	(s)的資料超出0~9999的範圍時。

- (d)為DWORD型的情況下

出錯代碼(SD0)	內容
3401H	(s)的資料超出0~99999999的範圍時。

32. 33 DINT型→REAL型轉換

DINT_TO_REAL(_E)



將DINT型資料轉換為REAL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DINT_TO_REAL(s);</p> <p>[帶EN/ENO] d:=DINT_TO_REAL_E(EN, ENO, s);</p>

設置數據

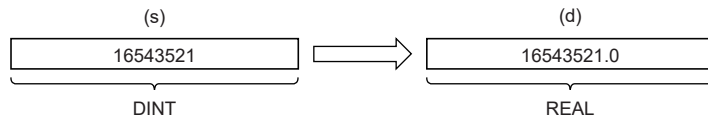
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DINT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	REAL

功能

■運算處理

- 將(s)中輸入的DINT型的資料轉換為REAL型的資料後從(d)中輸出。



- 至(s)的輸入值是DINT型的資料值。
- 因為REAL型資料是以32位的單精度進行處理，因此有效位數約為7位。
- 整數值超出-16777216~16777215的範圍的情況下，轉換後的值將產生誤差。(化整誤差)

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 34 DINT型→LREAL型轉換

DINT_TO_LREAL(_E)



將DINT型資料轉換為LREAL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DINT_TO_LREAL(s);</p> <p>[帶EN/ENO] d:=DINT_TO_LREAL_E(EN, ENO, s);</p>

設置數據

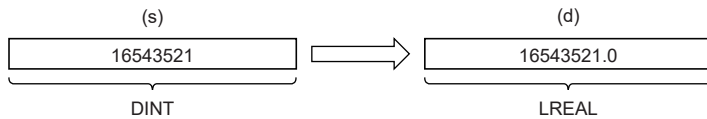
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DINT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	LREAL

功能

■運算處理

- 將(s)中輸入的DINT型的資料轉換為LREAL型的資料後從(d)中輸出。



- 至(s)的輸入值是DINT型的資料值。
- 因為LREAL型資料是以64位的雙精度進行處理，因此有效位數約為15位。
- 整數值超出-2147483648~2147483647的範圍的情況下，轉換後的值將產生誤差。(化整誤差)

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 35 DINT型→TIME型轉換

DINT_TO_TIME(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將DINT型資料轉換為TIME型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DINT_TO_TIME(s);</p> <p>[帶EN/ENO] d:=DINT_TO_TIME_E(EN, ENO, s);</p>

設置數據

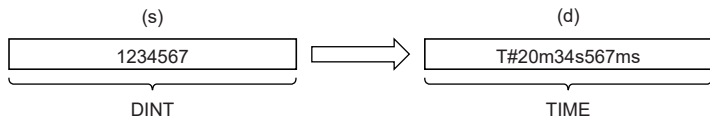
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DINT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	TIME

功能

■運算處理

- 將(s)中輸入的DINT型的資料轉換為TIME型的資料後從(d)中輸出。



- 至(s)的輸入值是DINT型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯


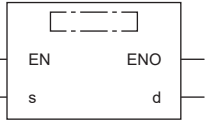
沒有運算出錯。

32. 36 DINT型→STRING型轉換

DINT_TO_STRING(_E)



將DINT型資料轉換為STRING型資料。

梯形圖、FBD/LD		ST
[無EN/ENO] 	[帶EN/ENO] 	[無EN/ENO] d:=DINT_TO_STRING(s); [帶EN/ENO] d:=DINT_TO_STRING_E(EN, ENO, s);

設置數據

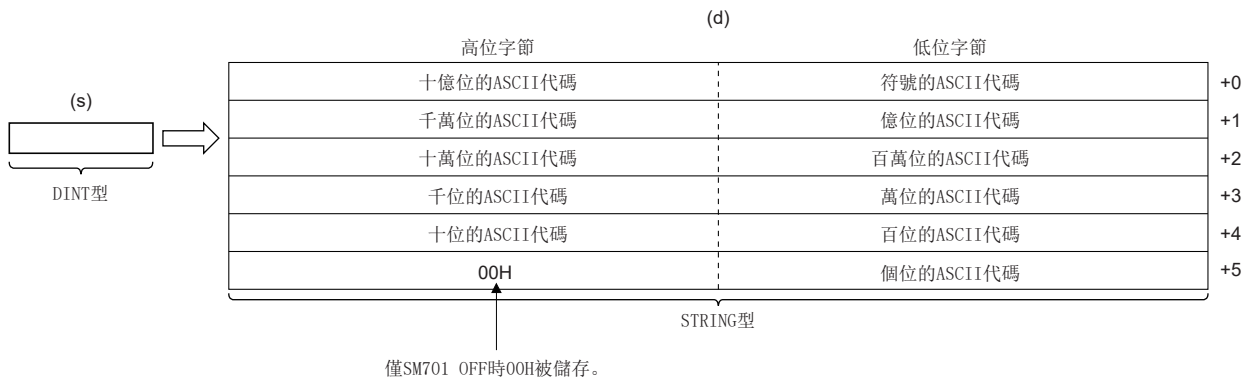
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	DINT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	STRING(11)

功能

■ 運算處理

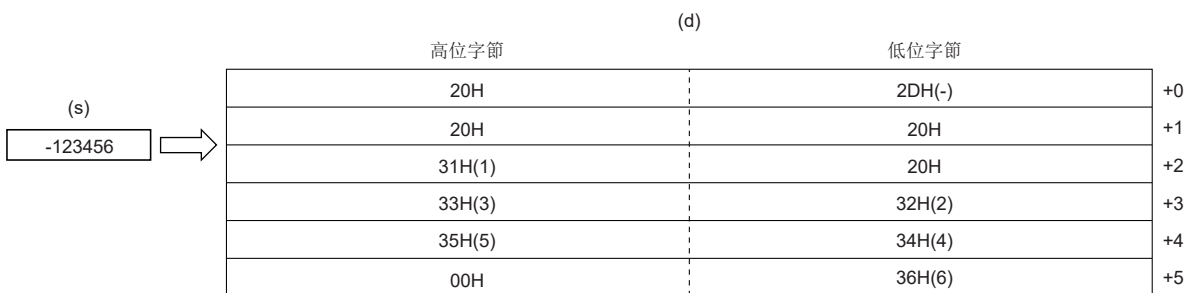
- 將 (s) 中輸入的 DINT 型的資料轉換為 STRING 型的資料後從 (d) 中輸出。



- 至 (s) 的輸入值是 DINT 型的資料值。
- 在符號資料中，輸入的值為正時儲存 20H (空格)、為負時儲存 2DH (-)。
- 有效位數較少的情况下，高位位中將儲存 20H (空格)。

例

輸入了 -123456 的情况下



- SM701 (輸出字數切換) 為 OFF 的情况下，字元串的最後 (第 6 字高位字節) 將儲存 00H。

■ 運算結果

1. 無 EN/ENO 函數

執行運算處理。從 (d) 輸出運算輸出值。

2. 帶 EN/ENO 函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

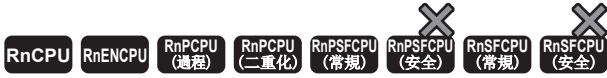
*1 在 ENO 輸出了 FALSE 的情况下，從 (d) 中輸出的資料將變為不定值。在此情况下，應進行不使用從 (d) 中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 37 BCD型→INT型轉換

BCD_TO_INT(_E)



將BCD型資料轉換為INT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=BCD_TO_INT(s);</p> <p>[帶EN/ENO] d:=BCD_TO_INT_E(EN, ENO, s);</p>

設置數據

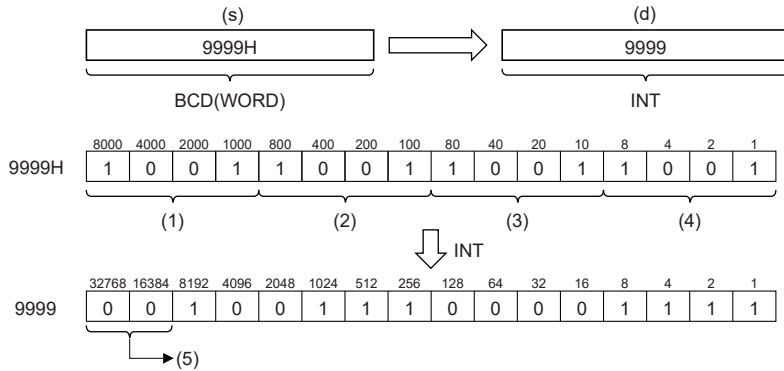
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	WORD
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 將(s)中輸入的BCD型(WORD型)資料轉換為INT型資料後從(d)進行輸出。



- (1): 千位
- (2): 百位
- (3): 十位
- (4): 個位
- (5): 必須為0。

- 至(s)的輸入值是WORD型的資料值，且在0H~9999H(各位為0~9的範圍)的範圍內。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3401H	(s)的各位中有0~9以外的值時。


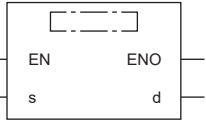
- 透過預先將SM754設為ON，可以避免上述出錯。設定了超出範圍的數值的情況下，與SM754的ON/OFF無關，指令將不被執行。

32. 38 BCD型→DINT型轉換

BCD_TO_DINT(_E)



將BCD型資料轉換為DINT型資料。

梯形圖、FBD/LD		ST
[無EN/ENO] 	[帶EN/ENO] 	[無EN/ENO] d:=BCD_TO_DINT(s); [帶EN/ENO] d:=BCD_TO_DINT_E(EN, ENO, s);

設置數據

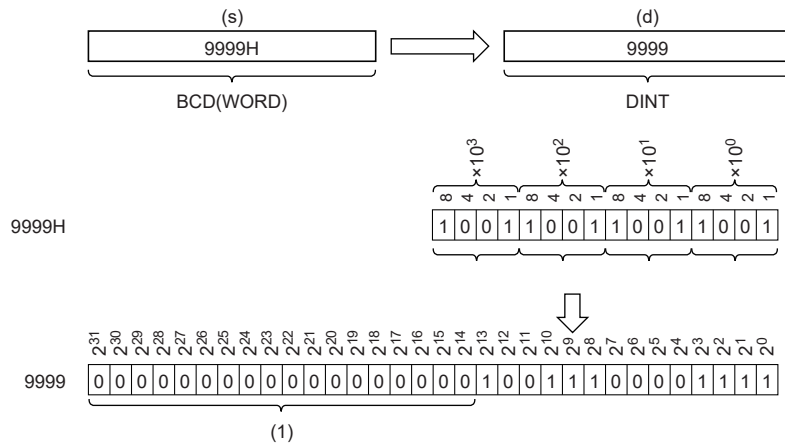
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(IN)	輸入	輸入變數	ANY_BIT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DINT

功能

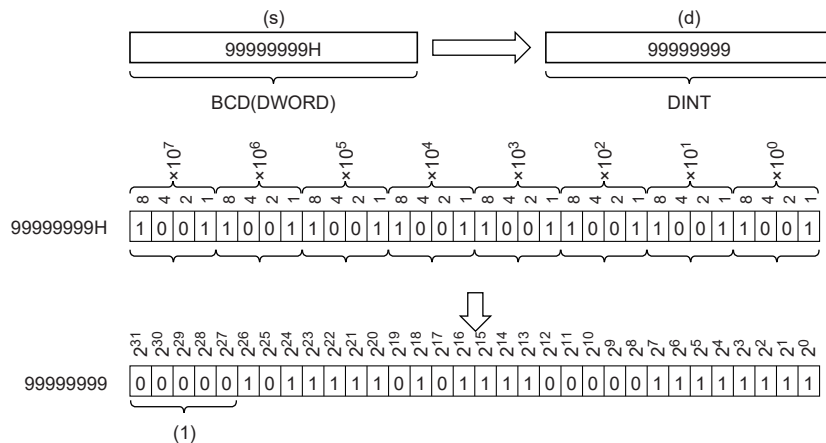
■ 運算處理

- 將(s)中輸入的BCD型(WORD型或DWORD型)資料轉換為DINT型資料後從(d)進行輸出。
- (s)中指定了WORD型的情況下



(1): 必須為0。

- (s)中指定了DWORD型的情況下



(1): 必須為0。

- 對於至(s)的輸入值，為WORD型資料值時在0H~9999H(各位為0~9的範圍)的範圍內，為DWORD型的資料值時在0H~99999999H(各位為0~9的範圍)的範圍內。
- (s)中可以指定WORD型、DWORD型。不能指定BOOL型。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼 (SD0)	內容
3401H	(s) 的各位中有0~9以外的值時。

- 透過預先將SM754設為ON，可以避免上述出錯。此外，在設定了超出範圍的數值的情況下，與SM754的ON/OFF無關，指令將不被執行。

32. 39 BCD型→STRING型轉換

BCD_TO_STRING(_E)



將BCD型資料轉換為STRING型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=BCD_TO_STRING(s);</p> <p>[帶EN/ENO] d:=BCD_TO_STRING_E(EN, ENO, s);</p>

設置數據

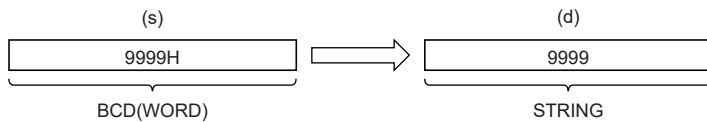
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_BIT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	STRING (8)

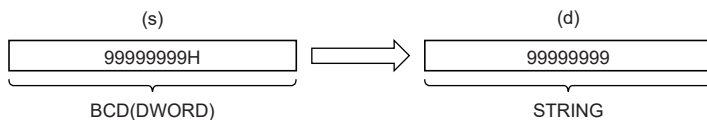
功能

■運算處理

- 將(s)中輸入的BCD型(WORD型或DWORD型)資料轉換為STRING型資料後從(d)進行輸出。
- (s)中指定了WORD型的情況下



- (s)中指定了DWORD型的情況下



- (s)中可以指定WORD型、DWORD型。不能指定BOOL型。
- SM701(輸出字數切換)為OFF的情況下，字元串的最後將儲存00H。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

- (s) 為WORD型的情況下

出錯代碼 (SD0)	內容
3401H	(s) 的資料超出0~9999的範圍時。

- (s) 為DWORD型的情況下

出錯代碼 (SD0)	內容
3401H	(s) 的資料超出0~99999999的範圍時。

32. 40 REAL型→INT型轉換

REAL_TO_INT(_E)



將REAL型資料轉換為INT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=REAL_TO_INT(s);</p> <p>[帶EN/ENO] d:=REAL_TO_INT_E(EN, ENO, s);</p>

設置數據

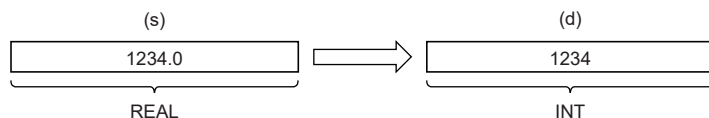
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 將(s)中輸入的REAL型的資料轉換為INT型的資料後從(d)中輸出。



- 至(s)的輸入值是REAL型的資料值且在-32768~32767的範圍內。
- 轉換後的資料將變為REAL型資料值的小數點以下第1位被四捨五入後的值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼 (SD0)	內容
3401H	(s) 中設定的單精度實數超出-32768~32767的範圍時。
3402H	<ul style="list-style-type: none">當(s)中設定了特殊數時。設定的單精度實數不在下述範圍內。 $-2^{128} < (s) \leq -2^{-126}$、$0$、$2^{-126} \leq (s) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38)設定的元件、標籤的內容為-0，在非正規化數、非數、$\pm\infty$時。

32. 41 REAL型→DINT型轉換

REAL_TO_DINT(_E)



將REAL型資料轉換為DINT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=REAL_TO_DINT(s);</p> <p>[帶EN/ENO] d:=REAL_TO_DINT_E(EN, ENO, s);</p>

設置數據

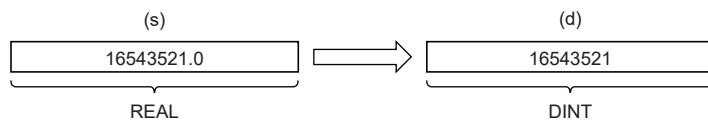
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DINT

功能

■運算處理

- 將(s)中輸入的REAL型的資料轉換為DINT型的資料後從(d)中輸出。



- 至(s)的輸入值是REAL型的資料值且在-2147483648~2147483647的範圍內。
- 轉換後的資料將變為REAL型資料值的小數點以下第1位被四捨五入後的值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼 (SD0)	內容
3401H	(s) 中設定的單精度實數超出-2147483648~2147483647的範圍時。
3402H	<ul style="list-style-type: none">當(s)中設定了特殊數時。設定的單精度實數不在下述範圍內。 $-2^{128} < (s) \leq -2^{-126}$、$0$、$2^{-126} \leq (s) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38)設定的元件、標籤的內容為-0，在非正規化數、非數、±∞時。

32.42 REAL型→LREAL型轉換

REAL_TO_LREAL(_E)



將REAL型資料轉換為LREAL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=REAL_TO_LREAL(s);</p> <p>[帶EN/ENO] d:=REAL_TO_LREAL_E(EN, ENO, s);</p>

設置數據

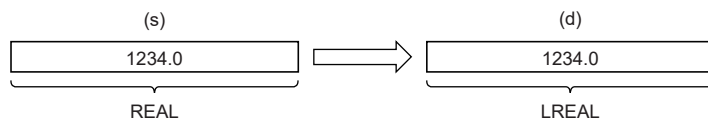
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	LREAL

功能

■運算處理

- 將(s)中輸入的REAL型的資料轉換為LREAL型的資料後從(d)中輸出。



- 至(s)的輸入值是REAL型的資料值。
- 因為REAL型資料是以32位的單精度進行處理，因此有效位數約為7位。
- 整數值超出-16777216~16777215的範圍的情況下，轉換後的值將產生誤差。(化整誤差)

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼 (SD0)	內容
3402H	(s) 的內容為-0或超出以下範圍時。 $-2^{128} < (s) \leq -2^{-126}$ 、 0 、 $2^{-126} \leq (s) < 2^{128}$ (E-3. 40282347+38~E-1. 17549435-38、0、E1. 17549435-38~E3. 40282347+38)
3403H	當 (d) 超出下述範圍時。(發生了上溢時) $ (d) < 2^{128}$

32.43 REAL型→STRING型轉換

REAL_TO_STRING(_E)



將REAL型資料轉換為STRING型(指數形式)資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=REAL_TO_STRING(s);</p> <p>[帶EN/ENO] d:=REAL_TO_STRING_E(EN, ENO, s);</p>

設置數據

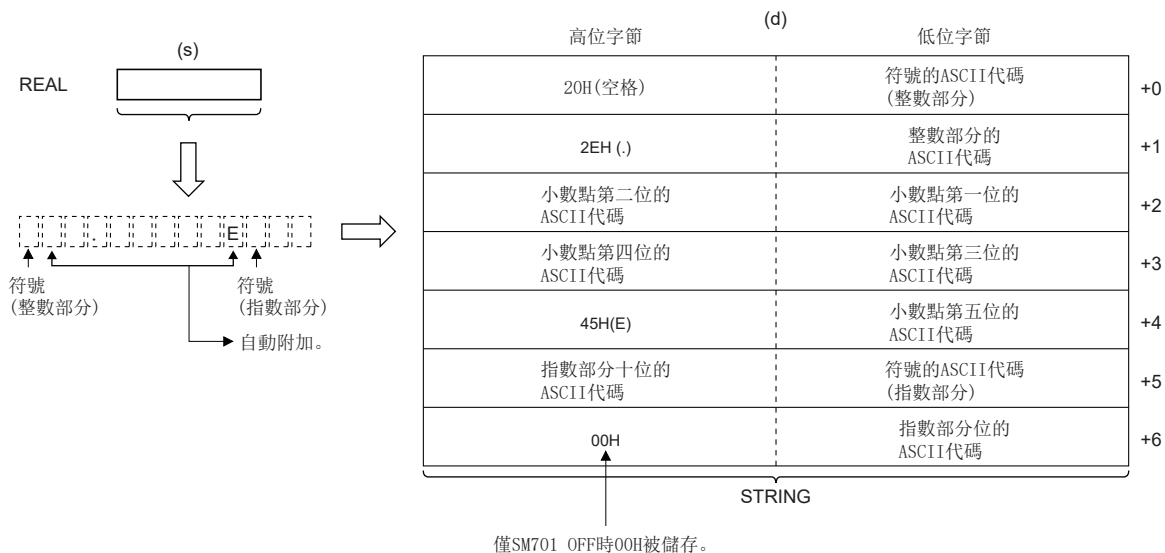
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	STRING (13)

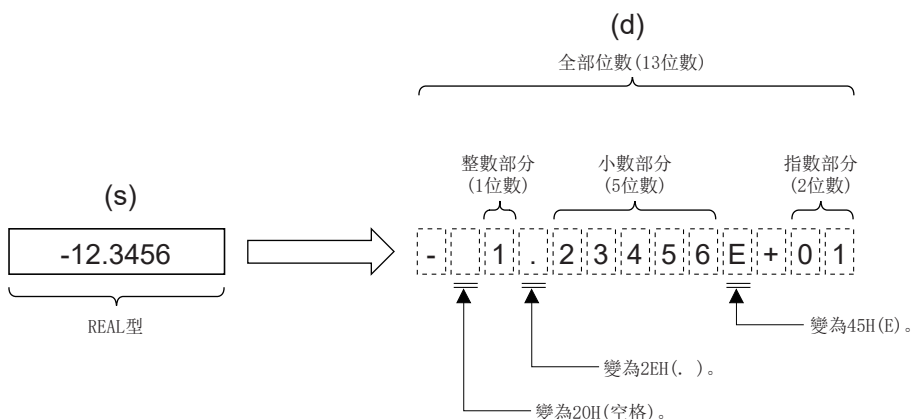
功能

■運算處理

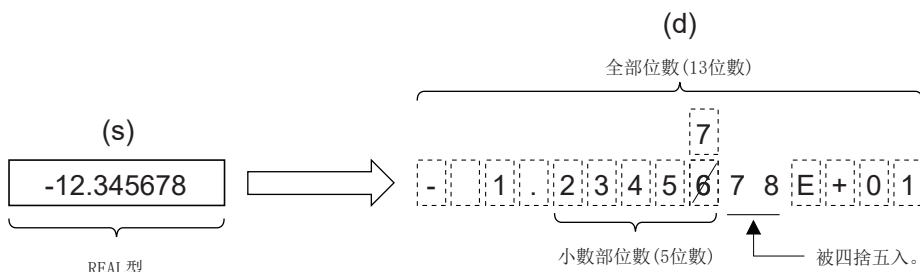
- 將(s)中輸入的REAL型的資料轉換為STRING型(指數形式)的資料後從(d)中輸出。



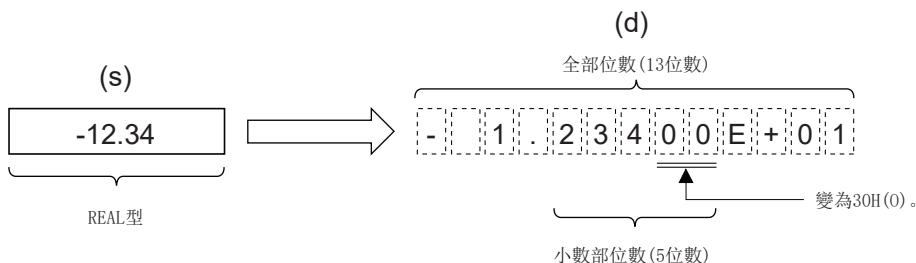
- 至 (s) 的輸入值是REAL型的資料值。
- 轉換後的字元串資料從 (d) 中按以下方式被輸出。
 - 整數部、小數部、指數部的位數是固定的。(整數部：1位、小數部：5位、指數部：2位)
 - 第2字節、第4字節、第10字節中，將分別自動儲存於“20H”(空格)、“2EH”(.)、“45H”(E)。



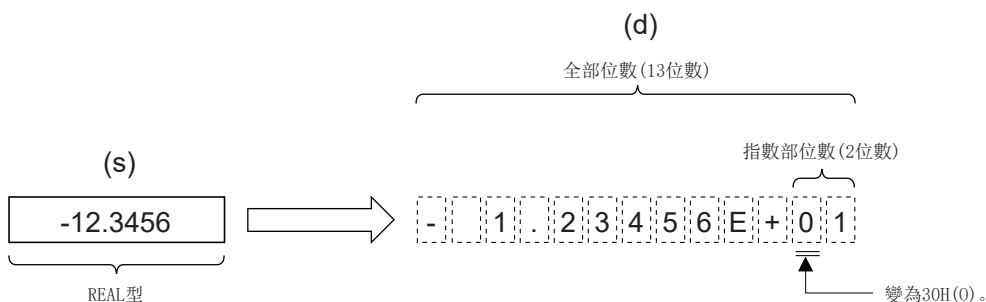
- 符號資料(整數部)中，輸入的值為正時將儲存“20H”(空格)、為負時將儲存“2DH”(-)。
- 小數部的第6位及其以後將被四捨五入。



- 有效位數較少的情況下，小數部將儲存“30H”(0)。



- 符號資料(指數部)中，指數為正時儲存“2BH”(+)、為負時儲存“2DH”(-)。
- 指數部為1位的情況下，在指數部十的位儲存“30H”(0)。



- 字元串的最後(第7字)中將自動儲存00H。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

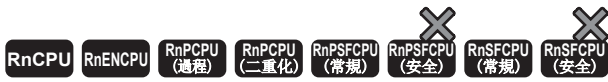
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3402H	<ul style="list-style-type: none">(s)的內容超出以下範圍時。 $-2^{128} < (s) \leq -2^{-126}$、$0$、$2^{-126} \leq (s) < 2^{128}$ (E-3. 40282347+38~E-1. 17549435-38、0、E1. 17549435-38~E3. 40282347+38)(s)的內容為-0、非正規化數、非數、$\pm\infty$時。
3406H	自(d)中指定的標籤或指定的元件編號開始至相應元件的最終元件編號為止的點數中，無法完整儲存轉換後的字元串時。

32.44 LREAL型→INT型轉換

LREAL_TO_INT(_E)



將LREAL型資料轉換為INT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=LREAL_TO_INT(s);</p> <p>[帶EN/ENO] d:=LREAL_TO_INT_E(EN, ENO, s);</p>

設置數據

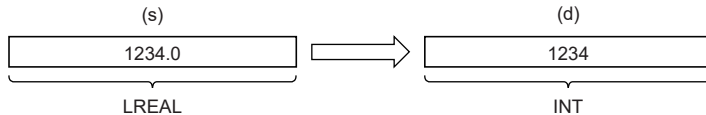
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	LREAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 將(s)中輸入的LREAL型的資料轉換為INT型的資料後從(d)中輸出。



- 至(s)的輸入值是LREAL型的資料值。
- 轉換後的資料，將變為LREAL型資料值的小數點以下第1位被四捨五入後的值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

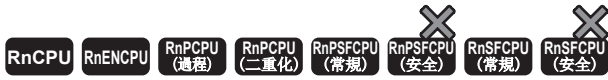
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3402H	(s)的內容為-0或超出以下範圍時。 $-2^{1024} < (s)$ 、 $(d) \leq -2^{1022}$ 、 0 、 $2^{-1022} \leq (s)$ 、 $(d) < 2^{1024}$ (E-1. 7976931348623157+308~E-2. 2250738585072014-308、0、E2. 2250738585072014-308~E1. 7976931348623157+308) (s)的內容超出-32768~32767的範圍時。

32. 45 LREAL型→DINT型轉換

LREAL_TO_DINT(_E)



將LREAL型資料轉換為DINT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=LREAL_TO_DINT(s);</p> <p>[帶EN/ENO] d:=LREAL_TO_DINT_E(EN, ENO, s);</p>

設置數據

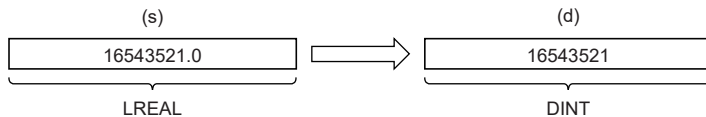
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(IN)	輸入	輸入變數	LREAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DINT

功能

■運算處理

- 將(s)中輸入的LREAL型的資料轉換為DINT型的資料後從(d)中輸出。



- 至(s)的輸入值是LREAL型的資料值。
- 轉換後的資料，將變為LREAL型資料值的小數點以下第1位被四捨五入後的值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3402H	(s)的內容為-0或超出以下範圍時。 $-2^{1024} < (s)$ 、 $(d) \leq -2^{1022}$ 、 0 、 $2^{1022} \leq (s)$ 、 $(d) < 2^{1024}$ (E-1. 7976931348623157+308~E-2. 2250738585072014-308、0、E2. 2250738585072014-308~E1. 7976931348623157+308) (s)的內容超出-2147483648~2147483647的範圍時。

32. 46 LREAL型→REAL型轉換

LREAL_TO_REAL(_E)



將LREAL型資料轉換為REAL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=LREAL_TO_REAL(s);</p> <p>[帶EN/ENO] d:=LREAL_TO_REAL_E(EN, ENO, s);</p>

設置數據

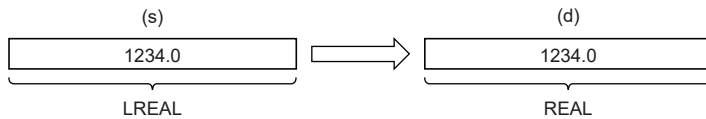
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	LREAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	REAL

功能

■運算處理

- 將(s)中輸入的LREAL型的資料轉換為REAL型的資料後從(d)中輸出。



- 至(s)的輸入值是LREAL型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

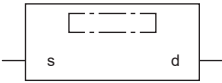
出錯代碼(SD0)	內容
3402H	(s)的內容為-0或超出以下範圍時。 $-2^{1024} < (s)$ 、 $(d) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s)$ 、 $(d) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308)
3403H	當(d)超出下述範圍時。(發生了上溢時) $ d < 2^{128}$

32. 47 TIME型→BOOL型轉換

TIME_TO_BOOL(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將TIME型資料轉換為BOOL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p> 	<p>[無EN/ENO] d:=TIME_TO_BOOL(s);</p> <p>[帶EN/ENO] d:=TIME_TO_BOOL_E(EN, ENO, s);</p>

設置數據

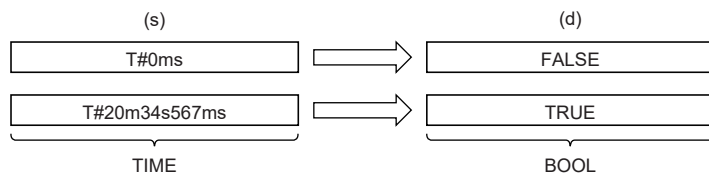
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	TIME
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	BOOL

功能

■運算處理

- 將(s)中輸入的TIME型的資料轉換為BOOL型的資料後從(d)中輸出。



■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 48 TIME型→WORD型轉換

TIME_TO_WORD(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFPCPU (常規) RnSFPCPU (安全)

將TIME型資料轉換為WORD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=TIME_TO_WORD(s);</p> <p>[帶EN/ENO] d:=TIME_TO_WORD_E(EN, ENO, s);</p>

設置數據

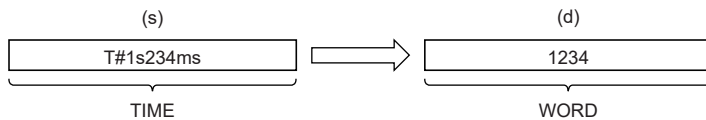
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	TIME
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	WORD

功能

■運算處理

- 將(s)中輸入的TIME型的資料轉換為WORD型的資料後從(d)中輸出。



- 至(s)的輸入值是TIME型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

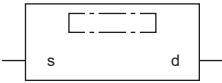
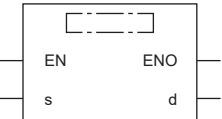
沒有運算出錯。

32. 49 TIME型→DWORD型轉換

TIME_TO_DWORD(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將TIME型資料轉換為DWORD型資料。

梯形圖、FBD/LD		ST
[無EN/ENO]		[無EN/ENO] d:=TIME_TO_DWORD(s);
[帶EN/ENO]		[帶EN/ENO] d:=TIME_TO_DWORD_E(EN, ENO, s);

設置數據

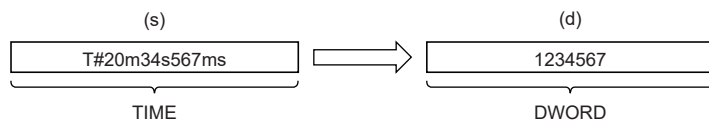
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	TIME
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DWORD

功能

■運算處理

- 將(s)中輸入的TIME型的資料轉換為DWORD型的資料後從(d)中輸出。



- 至(s)的輸入值是TIME型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 50 TIME型→INT型轉換

TIME_TO_INT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將TIME型資料轉換為INT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=TIME_TO_INT(s);</p> <p>[帶EN/ENO] d:=TIME_TO_INT_E(EN, ENO, s);</p>

設置數據

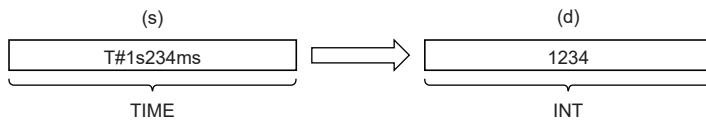
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	TIME
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 將(s)中輸入的TIME型的資料轉換為INT型的資料後從(d)中輸出。



- 至(s)的輸入值是TIME型的資料值。
- 轉換為INT型時，TIME型的高位16位(1字)的資料將被捨去。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

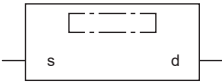
沒有運算出錯。

32.51 TIME型→DINT型轉換

TIME_TO_DINT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將TIME型資料轉換為DINT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p> 	<p>[無EN/ENO] d:=TIME_TO_DINT(s);</p> <p>[帶EN/ENO] d:=TIME_TO_DINT_E(EN, ENO, s);</p>

設置數據

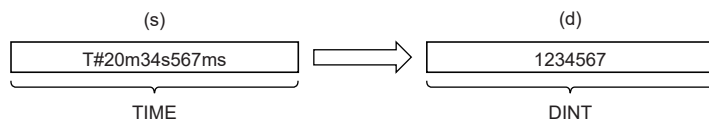
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	TIME
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出	輸出變數	DINT

功能

■運算處理

- 將(s)中輸入的TIME型的資料轉換為DINT型的資料後從(d)中輸出。



- 至(s)的輸入值是TIME型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 52 TIME型→STRING型轉換

TIME_TO_STRING(_E)



將TIME型資料轉換為STRING型資料。

梯形圖、FBD/LD		ST
[無EN/ENO]		[無EN/ENO] d:=TIME_TO_STRING(s);
[帶EN/ENO]		[帶EN/ENO] d:=TIME_TO_STRING_E(EN, ENO, s);

設置數據

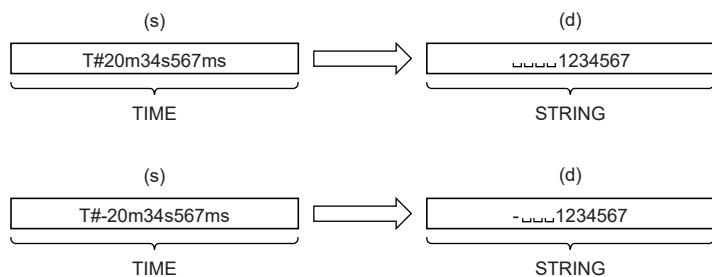
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	TIME
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	STRING STRING(11)

功能

■運算處理

- 將(s)中輸入的TIME型的資料轉換為STRING型的資料後從(d)中輸出。



- 至(s)的輸入值是TIME型的資料值。
- SM701(輸出字數切換)為OFF的情況下，字元串的最後將儲存00H。
- 輸出中儲存的運算結果如下所示。
 - 第1個字元中，BIN資料為正時儲存20H(空格)、為負時儲存2DH(-)。
 - 有效位數的左側將儲存20H(空格)。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 53 STRING型→BOOL型轉換

STRING_TO_BOOL(_E)



將STRING型資料轉換為BOOL型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=STRING_TO_BOOL(s);</p> <p>[帶EN/ENO] d:=STRING_TO_BOOL_E(EN, ENO, s);</p>

設置數據

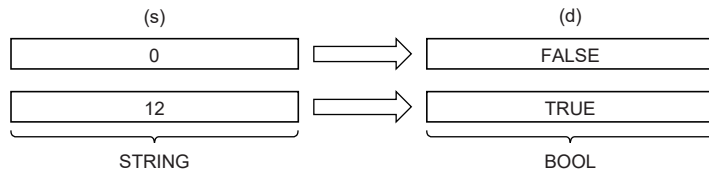
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	STRING (1)
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	BOOL

功能

■運算處理

- 將(s)中輸入的STRING型(小數點形式/指數形式)的資料轉換為BOOL型的資料後從(d)中輸出。



■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 54 STRING型→WORD型轉換

STRING_TO_WORD(_E)



將STRING型資料轉換為WORD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=STRING_TO_WORD(s);</p> <p>[帶EN/ENO] d:=STRING_TO_WORD_E(EN, ENO, s);</p>

設置數據

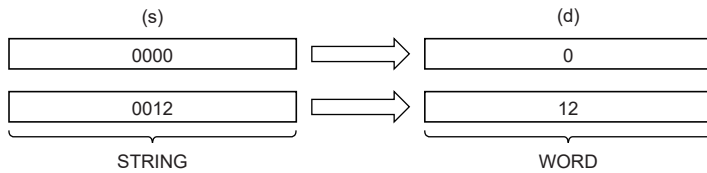
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	STRING (4)
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	WORD

功能

■運算處理

- 將(s)中輸入的STRING型的資料轉換為WORD型的資料後從(d)中輸出。



- 至(s)的輸入值為STRING型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3401H	輸入的各位的ASCII代碼超出30H~39H、41H~46H的範圍時。

32. 55 STRING型→DWORD型轉換

STRING_TO_DWORD(_E)



將STRING型資料轉換為DWORD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=STRING_TO_DWORD(s);</p> <p>[帶EN/ENO] d:=STRING_TO_DWORD_E(EN, ENO, s);</p>

設置數據

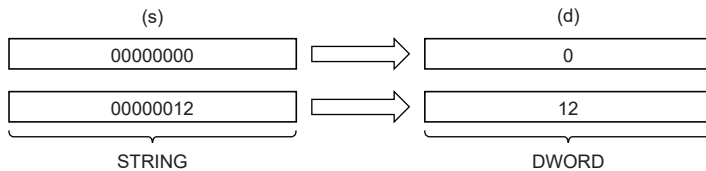
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	STRING (8)
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DWORD

功能

■運算處理

- 將(s)中輸入的STRING型的資料轉換為DWORD型的資料後從(d)中輸出。



- 至(s)的輸入值為STRING型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3401H	輸入的各位的ASCII代碼超出30H~39H、41H~46H的範圍時。

32. 56 STRING型→INT型轉換

STRING_TO_INT(_E)



將STRING型資料轉換為INT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=STRING_TO_INT(s);</p> <p>[帶EN/ENO] d:=STRING_TO_INT_E(EN, ENO, s);</p>

設置數據

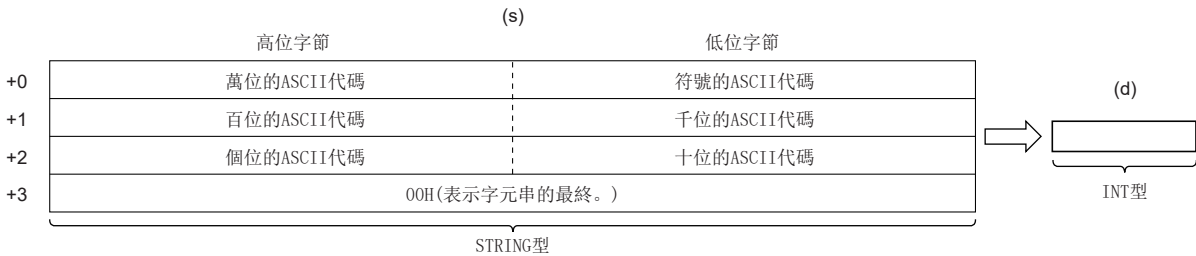
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	STRING (6)
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 將(s)中輸入的STRING型的資料轉換為INT型的資料後從(d)中輸出。



- 至(s)的輸入值為STRING型的資料值且在如下所示的範圍內。
- ASCII代碼時在30H~39H、20H、2DH、00H的範圍內
- STRING型的資料值時在-32768~32767的範圍內

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼 (SD0)	內容
3401H	當 (s) 中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none">• 各位的ASCII代碼超出30H~39H、20H、00H的範圍。• ASCII資料超出-32768~32767的範圍。

32. 57 STRING型→DINT型轉換

STRING_TO_DINT(_E)



將STRING型資料轉換為DINT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=STRING_TO_DINT(s);</p> <p>[帶EN/ENO] d:=STRING_TO_DINT_E(EN, ENO, s);</p>

設置數據

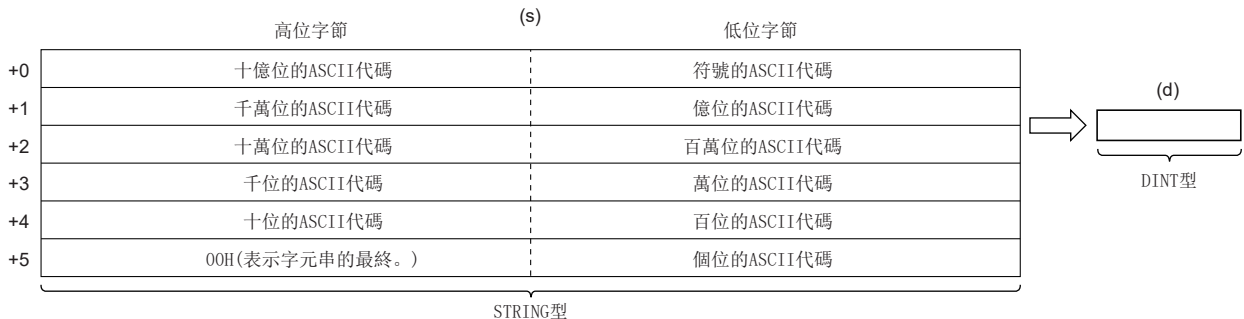
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	STRING(11)
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	DINT

功能

■運算處理

- 將(s)中輸入的STRING型的資料轉換為DINT型的資料後從(d)中輸出。



- 至(s)的輸入值為STRING型的資料值且在如下所示的範圍內。
- ASCII代碼時在30H~39H、20H、2DH、00H的範圍內
- STRING型的資料值時在-2147483648~-2147483647的範圍內

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼 (SD0)	內容
3401H	當 (s) 中輸入了不能轉換的不正確資料時。 <ul style="list-style-type: none">• 各位的ASCII代碼超出30H~39H、20H、00H的範圍。• ASCII資料超出-2147483648~2147483647的範圍。

32. 58 STRING型→BCD型轉換

STRING_TO_BCD(_E)



將STRING型資料轉換為BCD型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=STRING_TO_BCD(s);</p> <p>[帶EN/ENO] d:=STRING_TO_BCD_E(EN, ENO, s);</p>

設置數據

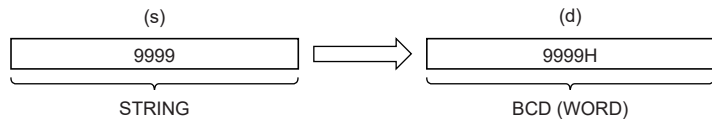
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	STRING (8)
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_BIT

功能

■運算處理

- 將(s)中輸入的STRING型4字元資料轉換為BCD型(WORD型)資料後從(d)進行輸出。



- 字元串中包含有20H(空格)的情況下，轉換時將忽略20H。
- 字元串中的20H(空格)、30H(0)也作為1個字元計數。
- 至(s)的輸入值為STRING型的資料值且在如下所示的範圍內。
ASCII代碼時在30H~39H、20H、00H的範圍內
- 輸入字元串不足4字元的情況下，字元串的後面將補充0後以4字元進行轉換。因此，希望將不足4字元的字元串進行BCD轉換的情況下，應輸入填充了0的字元串(‘1’的情況下‘0001’)。
- 字元串超過4字元的情況下，從左開始的4字元將成為轉換對象。

輸入字元串	轉換字元串	輸出(BCD型)
‘1’	‘1000’	1000H (4096D)
‘12’	‘1200’	1200H (4608D)
‘123’	‘1230’	1230H (4656D)
‘1234’	‘1234’	1234H (4660D)
‘12345’	‘1234’	1234H (4660D)

- (d)中可以指定WORD型、DWORD型。不能指定BOOL型。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

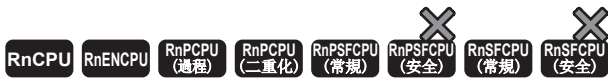
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3401H	以ASCII代碼輸入了超出30H~39H、20H、00H的範圍的字元串時。

32. 59 STRING型→REAL型轉換

STRING_TO_REAL(_E)



將STRING型資料轉換為REAL型資料。

梯形圖、FBD/LD		ST
[無EN/ENO]		[無EN/ENO] d:=STRING_TO_REAL(s);
[帶EN/ENO]		[帶EN/ENO] d:=STRING_TO_REAL_E(EN, ENO, s);

設置數據

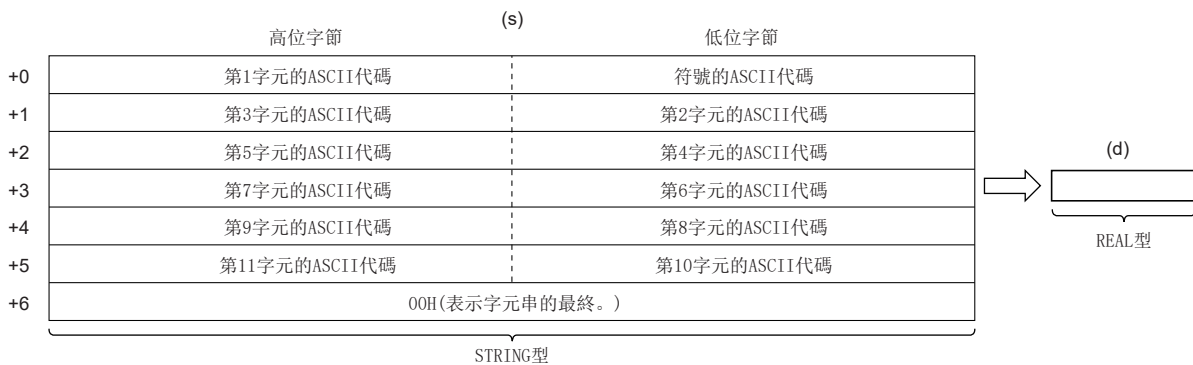
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	STRING (24)
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	REAL

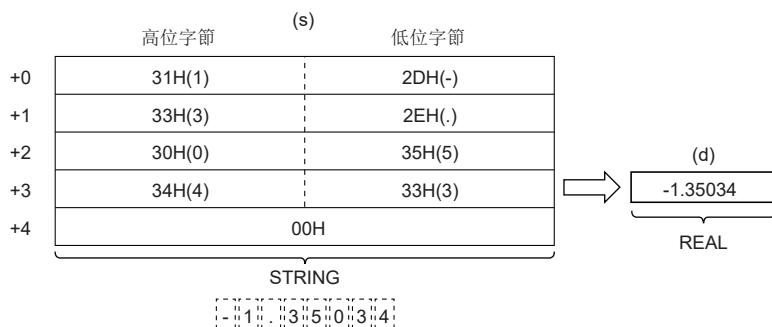
功能

■運算處理

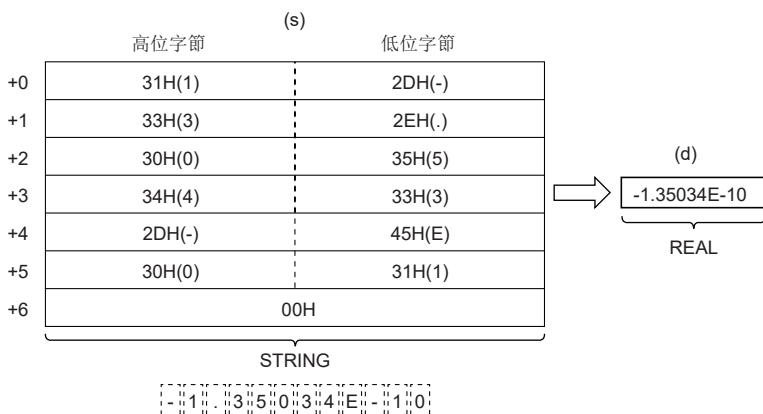
- 將(s)中輸入的STRING型(小數點形式/指數形式)的資料轉換為REAL型的資料後從(d)中輸出。



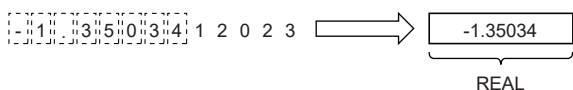
- STRING型資料可以轉換為小數點形式、指數形式。
- 小數點形式的情況下



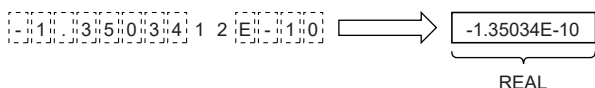
- 指數形式的情況下



- 對於STRING型資料如下所示，除了符號、小數點、指數部外的6位有效，第7位及其以後轉換時將被捨去。
- 小數點形式的情況下

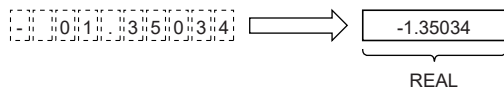


- 指數形式的情況下

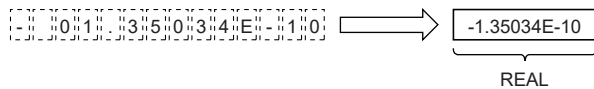


- 在小數點形式中將符號指定為2BH(+)、或省略符號時將作為正值進行轉換。此外，將符號指定為2DH(-)時將作為負值進行轉換。
- 在指數形式中將指數部的符號指定為2BH(+)、或省略符號時將作為正值進行轉換。將指數部的符號指定為2DH(-)時將作為負值進行轉換。

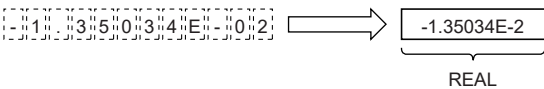
- 在STRING型資料中，最初的0以外的數值之間存在有20H(空格)或30H(0)的情況如下所示，轉換時將忽略20H、30H。
- 小數點形式的情況下



- 指數形式的情況下



- 在STRING型資料(指數形式)中“E”與數值之間存在有30H(0)的情況如下所示，轉換時將忽略30H。



- 字元串中包含有20H(空格)的情況下，轉換時將忽略20H。
- STRING型資料最多可輸入24字元。字元串中的20H(空格)、30H(0)也作為1個字元計數。
- 至(s)的輸入值為STRING型的資料值且在如下所示的範圍內。
- ASCII代碼時在30H~39H、45H、2BH、2DH、2EH、20H、00H的範圍內

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

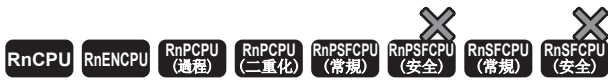
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
2820H	從(s)開始的相應元件範圍內沒有00H時。
3401H	(s)中設置了不能轉換的不正確資料時。 <ul style="list-style-type: none"> 整數部、小數部中，有30H(0)~39H(9)以外的字元 指定的字元串中至少有2個2EH(.) 指定的字元串的指數部中，有45H(E)、65H(e)、2BH(+)、2DH(-)以外的字元 指定的字元串中有多個45H(E)、65H(e)的指數部 指定的字元串中記述了至少3位的指數部的數值 指定的字元串中有多個2BH(+)、2DH(-)的指數部的符號 指定的字元串中，小數點形式的情況下整數部中有多個2BH(+)、2DH(-)的符號；指數形式的情況下尾數部中有多個2BH(+)、2DH(-)的符號 (s)及其以後的字元數為0或超過了24字元時。
3403H	當(d)超出下述範圍時。(發生了上溢時) $ d < 2^{128}$

32. 60 STRING型→TIME型轉換

STRING_TO_TIME(_E)



將STRING型資料轉換為TIME型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=STRING_TO_TIME(s);</p> <p>[帶EN/ENO] d:=STRING_TO_TIME_E(EN, ENO, s);</p>

設置數據

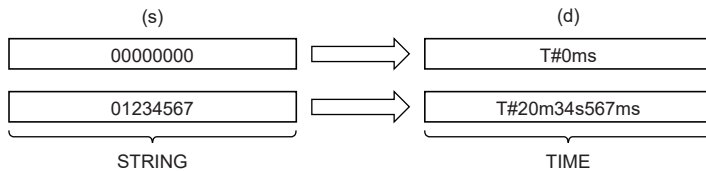
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	STRING(11)
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	TIME

功能

■運算處理

- 將(s)中輸入的STRING型的資料轉換為TIME型的資料後從(d)中輸出。



- 至(s)的輸入值為STRING型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3401H	輸入的各位的ASCII代碼超出30H~39H、20H、00H的範圍時。 輸入的ASCII資料超出以下範圍時。 -2147483648~4147483647

32. 61 位元數組→INT型轉換

BITARR_TO_INT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將透過位元數組指定的位元數轉換為INT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] <code>d:=BITARR_TO_INT(s, n);</code> [帶EN/ENO] <code>d:=BITARR_TO_INT_E(EN, ENO, s, n);</code></p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(BitArr)	輸入(要素指定也可為變數)	輸入變數	BOOL數組要素
n	只能指定4、8、12、16的常數	輸入變數	INT
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出	輸出變數	ANY16

功能

■運算處理

- 將(s)中輸入的位元數組要素作為起始將(n)中指定的位元數的資料轉換為ANY16型的資料後從(d)中輸出。
- 指定的位元數及其以上的輸出位元中將被設定為0。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 62 位元數組→DINT型轉換

BITARR_TO_DINT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCEPU (常規) RnSFCEPU (安全)

將透過位元數組指定的位元數轉換為DINT型資料。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO]</p> <pre>d:=BITARR_TO_DINT(s, n)</pre> <p>[帶EN/ENO]</p> <pre>d:=BITARR_TO_DINT_E(EN, ENO, s, n);</pre>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(BitArr)	輸入(要素指定也可為變數)	輸入變數	BOOL數組要素
n	只能指定4、8、12、16、20、24、28、32的常數	輸入變數	INT
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出	輸出變數	ANY32

功能

■運算處理

- 將(s)中輸入的位元數組要素作為起始將(n)中指定的位元數的資料轉換為ANY32型的資料後從(d)中輸出。
- 指定的位元數及其以上的輸出位元中將被設定為0。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

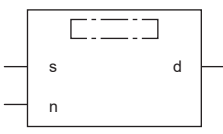
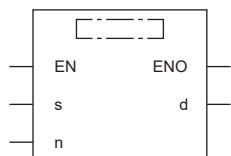
沒有運算出錯。

32. 63 INT型→位數組轉換

INT_TO_BITARR(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFPCPU (常規) RnSFPCPU (安全)

將INT型資料的低n位元輸出到位數組中。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p> 	<p>[無EN/ENO]</p> <pre>d:=INT_TO_BITARR(s, n);</pre> <p>[帶EN/ENO]</p>  <pre>d:=INT_TO_BITARR_E(EN, ENO, s, n);</pre>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s	輸入	輸入變數	ANY16
n	只能指定4、8、12、16的常數	輸入變數	INT
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出(要素指定也可為變數)	輸出變數	BOOL數組要素

功能

■運算處理

- 將(s)中指定的ANY16型的低(n)位元輸出到(d)中。
- 指定的位元數及其以上的輸出位元不更改。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

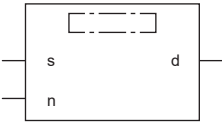
沒有運算出錯。

32. 64 DINT型→位數組轉換

DINT_TO_BITARR(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將DINT型資料的低n位元輸出到位數組中。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p> 	<p>[無EN/ENO] d:=DINT_TO_BITARR(s, n);</p> <p>[帶EN/ENO] d:=DINT_TO_BITARR_E(EN, ENO, s, n);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s	輸入	輸入變數	ANY32
n	只能指定4、8、12、16、20、24、28、32的常數	輸入變數	BOOL
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出(要素指定也可為變數)	輸出變數	BOOL數組要素

功能

■運算處理

- 將(s)中指定的ANY32型的低(n)位元輸出到(d)中。
- 指定的位元數及其以上的輸出位元不更改。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

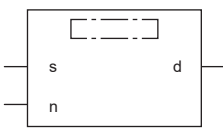
沒有運算出錯。

32. 65 位數組的複製

CPY_BITARR(_E)

RnCPU
RnENCPU
RnPCPU (過程)
RnPCPU (二重化)
RnPSFCPU (常規)
RnPSFCPU (安全)
RnSFCPU (常規)
RnSFCPU (安全)

將位數組按指定位元容量進行複製。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p> 	<p>[無EN/ENO] d:=CPY_BITARR(s, n);</p> <p>[帶EN/ENO] d:=CPY_BITARR_E(EN, ENO, s, n);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(BitArrIn)	輸入	輸入變數	BOOL數組要素
n	只能指定4、8、12、16、20、24、28、32的常數	輸入變數	INT
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出	輸出變數	BOOL數組要素

功能

■運算處理

- 將(s)中指定的位元數組的(n)位元輸出到(d)中。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

32. 66 字元標籤的指定位元讀取

GET_BIT_OF_INT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

讀取字元標籤的指定位元。

梯形圖*1		ST
<p>[無EN/ENO]</p>	<p>[帶EN/ENO]</p>	<p>[無EN/ENO] d:=GET_BIT_OF_INT(s, n);</p> <p>[帶EN/ENO] d:=GET_BIT_OF_INT_E(EN, ENO, s, n);</p>

*1 在FBD/LD中不支援。FBD/LD的情況下，應使用標籤的位元指定。

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s	輸入	輸入變數	INT
n	只能指定0至15的常數	輸入變數	INT
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸出	輸出變數	BOOL

功能

■運算處理

- 輸出(s)的第(n)位元。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

要點 

使用標籤的位元指定時，可以方便的建立與GET_BIT_OF_INT相同動作的程式。

以下示例與使用了GET_BIT_OF_INT的情況下一樣，將g_int1的位元5 (b5) 的值讀取到g_bool1中。

梯形圖的情況下



ST的情況下

```
g_bool1 := g_int1.5;
```

FBD/LD的情況下



32. 67 字標籤的指定位元寫入

SET_BIT_OF_INT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

寫入到字標籤的指定位元。

梯形圖*1		ST
<p>[無EN/ENO]</p>	<p>[帶EN/ENO]</p>	<p>[無EN/ENO] d:=SET_BIT_OF_INT(s, n);</p> <p>[帶EN/ENO] d:=SET_BIT_OF_INT_E(EN, ENO, s, n);</p>

*1 在FBD/LD中不支援。FBD/LD的情況下，應使用標籤的位元指定。

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s	輸入	輸入變數	BOOL
n	只能指定0至15的常數	輸入變數	INT
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸入輸出	輸入輸出變數	INT

功能

■運算處理

- 將(s)中指定的BOOL值寫入到(d)的第(n)位元中。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

注意事項

將SET_BIT_OF_INT(_E)透過ST使用的情况下，應建立將返回值代入變數中的程式。

```
//SET_BIT_OF_INT的返回值在代入變數之後使用。  
g_int1 := SET_BIT_OF_INT(TRUE, 0);  
g_bool1 := GET_BIT_OF_INT(g_int1, 0);
```

因為SET_BIT_OF_INT(_E)的返回值也被用於輸入輸出，因此應勿直接代入到其它指令、函數、功能塊的輸入引數中。

```
//在以下程式中，GET_BIT_OF_INT的第一引數的值將變為不定值。  
g_bool1 := GET_BIT_OF_INT( SET_BIT_OF_INT(TRUE, 0), 0);
```

出錯

沒有運算出錯。

要點

若使用標籤的位元指定，可以方便地建立與SET_BIT_OF_INT相同動作的程式。

在以下示例中，與使用了SET_BIT_OF_INT的情況下一樣，將g_int1的位元5(b5)更改為g_bool1的值。

梯形圖的情況下



ST的情況下

```
g_int1.5 := g_bool1;
```

FBD/LD的情況下



32. 68 字元標籤的指定位元複製

CPY_BIT_OF_INT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將字元標籤的指定位元複製到其它字元標籤的指定位元中。

梯形圖*1	ST
<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px;"> <p>[無EN/ENO]</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>[帶EN/ENO]</p> </div> </div>	<pre>[無EN/ENO] d:=CPY_BIT_OF_INT(s, n1, n2); [帶EN/ENO] d:=CPY_BIT_OF_INT_E(EN, ENO, s, n1, n2);</pre>

*1 在FBD/LD中不支援。FBD/LD的情況下，應使用標籤的位元指定。

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s	輸入	輸入變數	INT
n1	輸入變數的位元指定(只能指定0至15的常數)	輸入變數	INT
n2	輸出變數的位元指定(只能指定0至15的常數)	輸入變數	INT
ENO	輸出狀態(TRUE: 正常執行、FALSE: 異常或停止)	輸出變數	BOOL
d	輸入輸出	輸入輸出變數	INT

功能

■運算處理

- 將(s)中指定的字元第(n1)位元的值複製到(d)的第(n2)位元中。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

注意事項

將CPY_BIT_OF_INT(_E)透過ST使用的情况下，應建立將返回值代入變數中的程式。

```
//CPY_BIT_OF_INT的返回值在代入變數之後使用。  
g_int2 := CPY_BIT_OF_INT(g_int1,5,3);  
g_bool1 := GET_BIT_OF_INT(g_int2,3);
```

因為CPY_BIT_OF_INT(_E)的返回值也被用於輸入輸出，因此應勿直接代入到其它指令、函數、功能塊的輸入引數中。

```
//在以下程式中，GET_BIT_OF_INT的第一引數的值將變為不定值。  
g_bool1 := GET_BIT_OF_INT(CPY_BIT_OF_INT(g_int1,5,3), 3);
```

出錯

沒有運算出錯。

要點

使用標籤的位元指定時，可以方便的建立與CPY_BIT_OF_INT相同動作的程式。

以下示例中與使用了CPY_BIT_OF_INT的情況下一樣，將g_int2的位元3(b3)更改為g_int1的位元5(b5)的值。
梯形圖的情況下



ST的情況下

```
g_int2.3 := g_int1.5;
```

FBD/LD的情況下

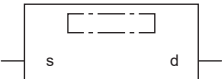


32.69 起始資料的獲取

GET_BOOL_ADDR、GET_INT_ADDR、GET_WORD_ADDR

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

將指定數組的起始資料作為BOOL型、INT型或WORD型資料進行輸出。

梯形圖、FBD/LD	ST
	<pre>d:=GET_BOOL_ADDR(s) d:=GET_INT_ADDR(s); d:=GET_WORD_ADDR(s);</pre>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
s	輸入	輸入變數	ANY
d	輸出	輸出變數	BOOL/INT/WORD

功能

■運算處理

- GET_BOOL_ADDR從(s)中指定的數組開始，將起始資料作為BOOL型資料進行輸出。
- GET_INT_ADDR從(s)中指定的數組開始，將起始資料作為INT型資料進行輸出。指定位長為32位以上的資料類型時，可獲取低位16位。
- GET_WORD_ADDR從(s)中指定的數組開始，將起始資料作為WORD型資料進行輸出。指定位長為32位以上的資料類型時，可獲取低位16位。

通用函數	輸入資料類型	輸出資料類型
GET_BOOL_ADDR	BOOL ARRAY OF BOOL	BOOL
GET_INT_ADDR	INT	INT
GET_WORD_ADDR	DINT WORD REAL TIME STRING ARRAY OF INT ARRAY OF DINT ARRAY OF WORD ARRAY OF DWORD ARRAY OF REAL ARRAY OF TIME	WORD

■運算結果

執行運算處理。從(d)輸出運算輸出值。

出錯

沒有運算出錯。

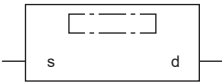
33 單數值變數函數

33.1 絕對值

ABS(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

輸出輸入值的絕對值。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p> 	<p>[無EN/ENO] d:=ABS(s);</p> <p>[帶EN/ENO] d:=ABS_E(EN, ENO, s);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_NUM*1
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_NUM*1

*1 安全程式中請勿指定REAL型/LREAL型資料。指定了REAL型/LREAL型資料的情況下，將變為出錯狀態。

功能

■運算處理

- 將(s)中輸入的INT型/DINT型/REAL/LREAL型資料的絕對值，以與(s)相同的資料類型從(d)輸出。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$$B=|A|$$

- 至(s)的輸入值為INT型/DINT型/REAL型/LREAL型的資料值。
- (s)的資料類型為INT型且輸入了-32768的情況下，從(d)輸出-32768。
- (s)的資料類型為DINT型且輸入了-2147483648的情況下，從(d)輸出-2147483648。(不變為運算出錯。此外ABS_E時，透過輸出變數ENO輸出TRUE。)

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

- (s) 為REAL型的情況下

出錯代碼 (SD0)	內容
3402H	(d) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。

- (s) 為LREAL型的情況下

出錯代碼 (SD0)	內容
3402H	(d) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。

33.2 平方根

SQRT(_E)



運算輸入值的平方根。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=SQRT(s);</p> <p>[帶EN/ENO] d:=SQRT_E(EN, ENO, s);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

- 運算(s)中輸入的REAL型/LREAL型資料的平方根，將運算結果儲存至(d)。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$$B = \sqrt{A}$$

- 至(s)的輸入值為REAL型/LREAL型的資料值且在正數的範圍內。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3405H	輸入的值為負數時。

33.3 自然對數運算

LN(_E)



輸出輸入值的自然對數運算結果。

梯形圖、FBD/LD	ST
[無EN/ENO] 	[無EN/ENO] $d := \text{LN}(s);$ [帶EN/ENO] $d := \text{LN_E}(EN, ENO, s);$

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

- 對將(s)中輸入的REAL型/LREAL型資料的自然對數“e”作為底時的對數進行運算後，從(d)輸出。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$$B = \log_e A$$

- 自然對數運算中，將底“e”作為2.71828進行運算。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3405H	輸入的值为0或負數時。

33.4 常用對數運算

LOG(_E)



輸出輸入值的常用對數(以10為底的對數)的運算結果。

梯形圖、FBD/LD	ST
<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px;"> <p>[無EN/ENO]</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>[帶EN/ENO]</p> </div> </div>	<p>[無EN/ENO] d:=LOG(s);</p> <p>[帶EN/ENO] d:=LOG_E(EN, ENO, s);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_REAL
ENO	輸出條件(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

- 對(s)中輸入的REAL型/LREAL型資料的10為底時的對數進行運算後，從(d)輸出。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$$B = \log_{10} A$$

- 至(s)的輸入值為REAL型/LREAL型的資料值。
- (s)中指定的值，僅正數可以設定。(負數不能進行運算。)
- 運算結果為-0或發生了下溢時，將運算結果以0進行輸出。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

- (s) 為REAL型的情況下

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、 $\pm\infty$ 時。
3405H	當 (s) 中設定了超出可以指定的範圍的資料時。 <ul style="list-style-type: none"> • 指定的值為負數 • 指定的值為“0”

- (s) 為LREAL型的情況下

出錯代碼 (SD0)	內容
3402H	(s) 中指定的值為-0、非正規化數、非數、 $\pm\infty$ 時。
3405H	當 (s) 中設定了超出可以指定的範圍的資料時。 <ul style="list-style-type: none"> • 指定的值為負數 • 指定的值為“0”

33.5 指數運算

EXP(_E)



輸出輸入值的指數運算結果。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=EXP (s);</p> <p>[帶EN/ENO] d:=EXP_E (EN, ENO, s);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_REAL
ENO	輸出狀態 (TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

- 對 (s) 中輸入的REAL型/LREAL型資料的指數進行運算，從 (d) 輸出。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$$B=e^A$$

- 指數運算中，將底“e”作為2.71828進行運算。
- 至 (s) 的輸入值為REAL型/LREAL型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從 (d) 輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從 (d) 中輸出的資料將變為不定值。在此情況下，應進行不使用從 (d) 中輸出的資料的程式處理。

出錯

出錯代碼 (SD0)	內容
3402H	(s) 中指定的資料為-0、非正規化數、非數、±∞時。
3403H	當 (d) 超出下述範圍時。(發生了上溢時) $ (d) < 2^{128}$

33.6 SIN運算

SIN(_E)



輸出輸入值的SIN(正弦)值。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=SIN(s);</p> <p>[帶EN/ENO] d:=SIN_E(EN, ENO, s);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

- 對(s)中輸入的REAL型資料(角度)的SIN(正弦)值進行運算後，從(d)輸出。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$B = \sin A$

- 至(s)的輸入值(角度)為REAL型的資料值。輸入值應以弧度單位(角度 $\times\pi/180$)進行輸出。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3402H	當(s)中指定的資料為-0時。

33.7 COS運算

COS(_E)



輸出輸入值的COS(餘弦)值。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=COS(s);</p> <p>[帶EN/ENO] d:=COS_E(EN, ENO, s);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

- 對(s)中輸入的REAL型資料(角度)的COS(餘弦)值進行運算後，從(d)輸出。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$$B = \cos A$$

- 至(s)的輸入值(角度)為REAL型的資料值。對於輸入值，應以弧度單位(角度 $\times\pi/180$)進行輸出。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3402H	當(s)中指定的資料為-0時。

33.8 TAN運算

TAN(_E)



輸出輸入值的TAN(正切)值。

梯形圖、FBD/LD	ST
[無EN/ENO] 	[無EN/ENO] $d := \text{TAN}(s);$ [帶EN/ENO] $d := \text{TAN_E}(EN, ENO, s);$

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

- 對(s)中輸入的REAL型資料(角度)的TAN(正切)值進行運算後，從(d)輸出。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$B = \text{TAN } A$

- 輸入的值為 $\pi/2$ 弧度、 $(3/2)\pi$ 弧度的情況下，弧度值中也將產生運算誤差，因此應注意不要發生出錯。
- 至(s)的輸入值(角度)為REAL型的資料值。對於輸入值，應以弧度單位(角度 $\times\pi/180$)進行輸出。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3402H	當(s)中指定的資料為-0時。

33.9 SIN⁻¹運算

ASIN(_E)



輸出輸入值的SIN⁻¹(反正弦)值。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=ASIN(s);</p> <p>[帶EN/ENO] d:=ASIN_E(EN, ENO, s);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

- 對(s)中輸入的REAL型資料的SIN⁻¹(反正弦)值進行運算後，從(d)輸出。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$$B = \text{SIN}^{-1} A$$

- 至(s)的輸入值為REAL型的資料值且在下述所示範圍內。

ASIN(_E): -1.0~1.0

- (d)的輸出值(角度)為弧度單位(角度×π/180)。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3402H	當(s)中指定的資料為-0時。
3405H	ASIN(_E)中輸入的值為-1.0~1.0以外時。

33. 10 \cos^{-1} 運算

ACOS(_E)



輸出輸入值的 \cos^{-1} (反餘弦)值。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=ACOS(s);</p> <p>[帶EN/ENO] d:=ACOS_E(EN, ENO, s);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

- 對(s)中輸入的REAL型資料的 \cos^{-1} (反餘正弦)值進行運算後，從(d)輸出。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$$B = \cos^{-1} A$$

- 至(s)的輸入值為REAL型的資料值且在下述所示範圍內。

$$\text{ACOS_E}: -1.0 \sim 1.0$$

- (d)的輸出值(角度)為弧度單位(角度 $\times\pi/180$)。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

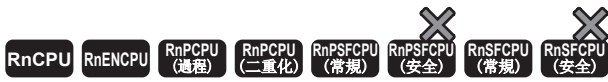
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3402H	當(s)中指定的資料為-0時。
3405H	ACOS_E中輸入的值為-1.0~1.0以外時。

33. 11 TAN^{-1} 運算

ATAN(_E)



輸出輸入值的 TAN^{-1} (反正切)值。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=ATAN(s);</p> <p>[帶EN/ENO] d:=ATAN_E(EN, ENO, s);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_REAL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

- 對(s)中輸入的REAL型資料的 TAN^{-1} (反正切)值進行運算後，從(d)輸出。
- 如果將輸入值設為A、運算輸出值設為B，其情況如下所示。

$$B=TAN^{-1} A$$

- 至(s)的輸入值為REAL型的資料值且在下述所示範圍內。

$$ATAN_E): \pm 1.17549^{-38} \sim \pm 3.40282^{+38}$$

- (d)的輸出值(角度)為弧度單位(角度 $\times\pi/180$)。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3402H	當(s)中指定的資料為-0時。

34 算術運算函數

34.1 加法運算

ADD(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

輸出輸入值的和((s1)+(s2)+...+(s28))。

梯形圖、FBD/LD*1		ST*1
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] d:=ADD(s1, s2); [帶EN/ENO] d:=ADD_E(EN, ENO, s1, s2);

*1 輸入變數s，可以在2~28的範圍內進行更改。

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1) ~ s28 (IN28)	輸入	輸入變數	ANY_NUM*1
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_NUM*1

*1 安全程式中請勿指定REAL型/LREAL型資料。指定了REAL型/LREAL型資料的情況下，將變為出錯狀態。

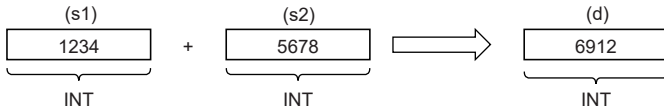
功能

■ 運算處理

- 進行 (s1) ~ (s28) 中輸入的 INT 型/DINT 型/WORD 型/DWORD 型/REAL 型/LREAL 型資料的加法運算 $((s1) + (s2) + \dots + (s28))$ ，運算結果將以與 (s) 相同的資料類型從 (d) 輸出。

例

資料類型為 INT 型的情況下



- 至 (s1) ~ (s28) 的輸入值為 INT 型/DINT 型/WORD 型/DWORD 型/REAL 型/LREAL 型的資料值。
- 運算結果發生了下溢/上溢時，按下述方式從 (d) 輸出。

資料類型	內容
INT 型	即使發生了下溢及上溢時，也不變為運算出錯。此外，ADD_E 的情況下，從 ENO 輸出 TRUE。 [例1] $32767 + 2 = -32767$ $(7FFFH) + (0002H) = (8001H)$ 由於最高位的位元變為 1，因此將變為負值。 [例2] $-32767 + (-2) = 32766$ $(8000H) + (FFFEH) = (7FFEH)$ 由於最高位的位元變為 0，因此將變為正值。
DINT 型	即使發生了下溢及上溢時，也不變為運算出錯。此外，ADD_E 的情況下，從 ENO 輸出 TRUE。 [例1] $2147483647 + 2 = -2147483647$ $(7FFFFFFFH) + (00000002H) = (80000001H)$ 由於最高位的位元變為 1，因此將變為負值。 [例2] $-2147483648 + (-2) = 2147483646$ $(80000000H) + (FFFEH) = (7FFFFFFEH)$ 由於最高位的位元變為 0，因此將變為正值。
WORD 型	即使發生了上溢時，也不變為運算出錯。此外，ADD_E 的情況下，從 ENO 輸出 TRUE。 [例] $65535 + 1 = 0$ $(FFFFH) + (0001H) = (0000H)$
DWORD 型	即使發生了上溢時，也不變為運算出錯。此外，ADD_E 的情況下，從 ENO 輸出 TRUE。 [例] $4294967295 + 1 = 0$ $(FFFFFFFFH) + (00000001H) = (00000000H)$
REAL 型	將變為運算出錯，輸出不定值。
LREAL 型	

■ 運算結果

1. 無 EN/ENO 函數

執行運算處理。從 (d) 輸出運算輸出值。

2. 帶 EN/ENO 函數

執行條件與運算結果如下所示。

執行條件	運算結果	
	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在 ENO 輸出了 FALSE 的情況下，從 (d) 中輸出的資料將變為不定值。在此情況下，應進行不使用從 (d) 中輸出的資料的程式處理。

出錯

- (s1)～(s28)為REAL型的情況下

出錯代碼 (SD0)	內容
3402H	(s1)～(s28)中指定的資料為-0、非正規化數、非數、±∞時。
	(d)中指定的資料為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生了上溢時) $ d < 2^{128}$

- (s1)～(s28)為LREAL型的情況下

出錯代碼 (SD0)	內容
3402H	(s1)～(s28)中指定的資料為-0、非正規化數、非數、±∞時。
	(d)中指定的資料為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生了上溢時) $ d < 2^{1024}$

34.2 乘法運算

MUL(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

輸出輸入值的積((s1)×(s2)×...×(s28))。

梯形圖、FBD/LD*1	ST*1
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=MUL(s1, s2);</p> <p>[帶EN/ENO] d:=MUL_E(EN, ENO, s1, s2);</p>

*1 輸入變數s，可以在2~28的範圍內進行更改。

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)~s28 (IN28)	輸入	輸入變數	ANY_NUM*1
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_NUM*1

*1 安全程式中請勿指定REAL型/LREAL型資料。指定了REAL型/LREAL型資料的情況下，將變為出錯狀態。

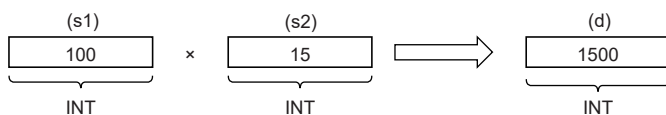
功能

■運算處理

- 進行(s1)~(s28)中輸入的INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型資料的乘法運算((s1)×(s2)×...×(s28))，運算結果將以與(s)相同的資料類型從(d)輸出。

例

資料類型為INT型的情況下



- 至(s1)~(s28)的輸入值為INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型的資料值。
- 運算結果發生了下溢/上溢時，按下述方式從(d)輸出。

資料類型	內容
INT型 WORD型	<ul style="list-style-type: none"> 即使發生了下溢及上溢時，也不變為運算出錯。此外，MUL_E的情況下，從ENO輸出TRUE。 即使運算結果超過INT型/WORD型資料範圍的情況下，也會輸出INT型/WORD型的資料。(雖然運算結果變為DINT型/DWORD型，但是輸出時以刪除了高位16位元的INT型/WORD型的資料輸出。) 運算結果超過了INT型/WORD型資料的範圍的情況下，應透過INT_TO_DINT/WORD_TO_DWORD將輸入值轉換至DINT型/DWORD型資料後再進行運算。
DINT型 DWORD型	<ul style="list-style-type: none"> 即使發生了下溢及上溢時，也不變為運算出錯。此外，MUL_E的情況下，從ENO輸出TRUE。 即使運算結果超過DINT型/DWORD型資料範圍的情況下，也會輸出DINT型/DWORD型的資料。(雖然運算結果變為64位元資料，但是輸出時以刪除了高位32位元的DINT型/DWORD型的資料輸出。) 運算結果超過了DINT型/DWORD型資料的範圍的情況下，應透過DINT_TO_REAL將輸入值轉換至REAL型資料後再進行運算。
REAL型 LREAL型	將變為運算出錯，輸出不定值。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

要點

運算結果超過了資料類型範圍的情況下，應在轉換輸入值的資料類型之後再執行運算。

出錯

- (s1) ~ (s28) 為REAL型的情況下

出錯代碼 (SD0)	內容
3402H	(s1) ~ (s28) 中指定的資料為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生了上溢時) (d) 2^{128}

- (s1) ~ (s28) 為LREAL型的情況下

出錯代碼 (SD0)	內容
3402H	(s1) ~ (s28) 中指定的資料為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生了上溢時) (d) 2^{1024}

34.3 減法運算

SUB(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

輸出輸入值的差((s1)-(s2))。

梯形圖、FBD/LD		ST
[無EN/ENO]		[無EN/ENO] d:=SUB(s1, s2); [帶EN/ENO] d:=SUB_E(EN, ENO, s1, s2);
[帶EN/ENO]		

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)、s2 (IN2)	輸入	輸入變數	ANY_NUM*1
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_NUM*1

*1 安全程式中請勿指定REAL型/LREAL型資料。指定了REAL型/LREAL型資料的情況下，將變為出錯狀態。

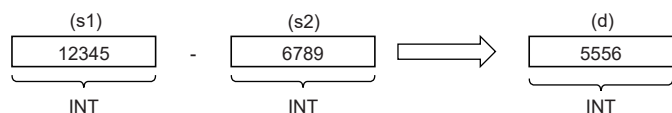
功能

■ 運算處理

- 進行 (s1)、(s2) 中輸入的 INT 型/DINT 型/WORD 型/DWORD 型/REAL 型/LREAL 型資料的減法運算 ((s1)-(s2))，運算結果將以與 (s) 相同的資料類型從 (d) 進行輸出。

例

資料類型為 INT 型的情況下



- 至 (s1) ~ (s2) 的輸入值為 INT 型/DINT 型/WORD 型/DWORD 型/REAL 型/LREAL 型的資料值。
- 運算結果發生了下溢/上溢時，按下述方式從 (d) 輸出。

資料類型	內容
INT 型	即使發生了下溢及上溢時，也不變為運算出錯。此外，SUB_E 的情況下，從 ENO 輸出 TRUE。 [例1] 32767 - (-2) = -32767 (7FFFH) - (FFFEH) = (8001H) 由於最高位的位元變為 1，因此將變為負值。 [例2] -32767 - 2 = 32766 (8000H) - (0002H) = (7FFEH) 由於最高位的位元變為 0，因此將變為正值。
DINT 型	即使發生了下溢及上溢時，也不變為運算出錯。此外，SUB_E 的情況下，從 ENO 輸出 TRUE。 [例1] 2147483647 - (-2) = -2147483647 (7FFFFFFFH) - (0000FFFEH) = (80000001H) 由於最高位的位元變為 1，因此將變為負值。 [例2] -2147483648 - 2 = 2147483646 (80000000H) - (00000002H) = (7FFFFFFEH) 由於最高位的位元變為 0，因此將變為正值。
WORD 型	即使發生了下溢時，也不變為運算出錯。此外，ADD_E 的情況下，從 ENO 輸出 TRUE。 [例] 0 - 1 = 65535 (0000H) - (0001H) = (FFFFH)
DWORD 型	即使發生了下溢時，也不變為運算出錯。此外，ADD_E 的情況下，從 ENO 輸出 TRUE。 [例] 0 - 1 = 4294967295 (00000000H) - (00000001H) = (FFFFFFFH)
REAL 型	將變為運算出錯，輸出不定值。
LREAL 型	

■ 運算結果

1. 無 EN/ENO 函數

執行運算處理。從 (d) 輸出運算輸出值。

2. 帶 EN/ENO 函數

執行條件與運算結果如下所示。

執行條件	運算結果	
	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在 ENO 輸出了 FALSE 的情況下，從 (d) 中輸出的資料將變為不定值。在此情況下，應進行不使用從 (d) 中輸出的資料的程式處理。

出錯

- (s1)、(s2)為REAL型的情況下

出錯代碼 (SD0)	內容
3402H	(s1)中指定的資料為-0、非正規化數、非數、±∞時。
	(s2)中指定的資料為-0、非正規化數、非數、±∞時。
	(d)中指定的資料為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生了上溢時) $ d < 2^{128}$

- (s1)、(s2)為LREAL型的情況下

出錯代碼 (SD0)	內容
3402H	(s1)中指定的資料為-0、非正規化數、非數、±∞時。
	(s2)中指定的資料為-0、非正規化數、非數、±∞時。
	(d)中指定的資料為-0、非正規化數、非數、±∞時。
3403H	當(d)超出下述範圍時。(發生了上溢時) $ d < 2^{1024}$

34.4 除法運算

DIV(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCEPU (常規) RnSFCEPU (安全)

輸出輸入值的商((s1)÷(s2))。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=DIV(s1, s2);</p> <p>[帶EN/ENO] d:=DIV_E(EN, ENO, s1, s2);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)	被除數	輸入變數	ANY_NUM*1
s2 (IN2)	除數	輸入變數	ANY_NUM*1
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_NUM*1

*1 安全程式中請勿指定REAL型/LREAL型資料。指定了REAL型/LREAL型資料的情況下，將變為出錯狀態。

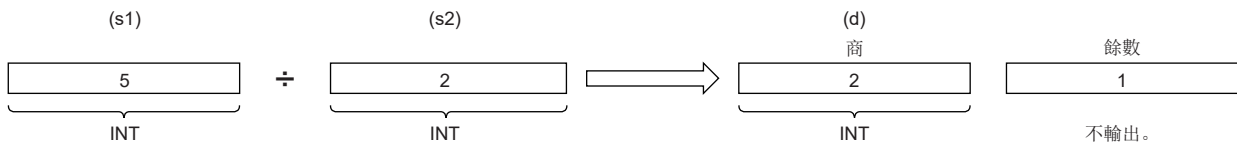
功能

■運算處理

- 進行(s1)、(s2)中輸入的INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型資料的除法運算((s1)÷(s2))，運算結果的商將以與(s)相同的資料類型從(d)進行輸出。

例

資料類型為INT型的情況下



- 至(s1)~(s2)的輸入值為INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型的資料值。(但是，輸入至(s2)的值为0以外。)

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

- (s1)、(s2)為INT型/WORD型的情況下

出錯代碼 (SD0)	內容
3400H	當 (s2) 中指定的值 (除數) 為0時。

- (s1)、(s2)為DINT型/DWORD型的情況下

出錯代碼 (SD0)	內容
3400H	當 (s2) 中指定的值 (除數) 為0時。

- (s1)、(s2)為REAL型的情況下

出錯代碼 (SD0)	內容
3400H	當 (s2) 中指定的值 (除數) 為0時。
3402H	(s1) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。 (s2) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當 (d) 超出下述範圍時。(發生了上溢時) $ d < 2^{128}$

- (s1)、(s2)為LREAL型的情況下

出錯代碼 (SD0)	內容
3400H	當 (s2) 中指定的值 (除數) 為0時。
3402H	(s1) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。 (s2) 中指定的資料為-0、非正規化數、非數、 $\pm\infty$ 時。
3403H	當 (d) 超出下述範圍時。(發生了上溢時) $ d < 2^{1024}$

34.5 餘數

MOD(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSF CPU (常規) RnSF CPU (安全)

輸出輸入值的餘數((s1)÷(s2))。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[帶EN/ENO]</p>
	<p>[無EN/ENO] 作為運算符記述。(MELSEC iQ-R 程式手冊(程式設計篇)) [帶EN/ENO] d:=MOD_E(EN, ENO, s1, s2);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)	被除數	輸入變數	ANY_INT
s2 (IN2)	除數	輸入變數	ANY_INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_INT

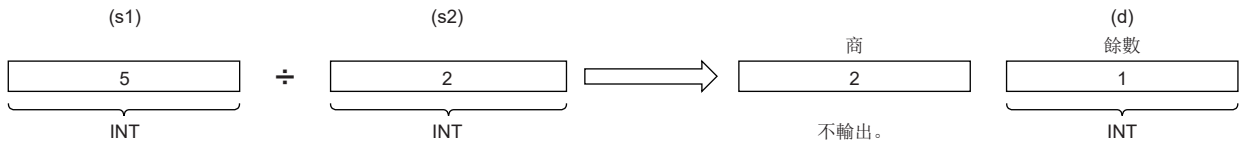
功能

■運算處理

- 進行(s1)、(s2)中輸入的INT型/DINT型/WORD型/DWORD型資料的除法運算((s1)÷(s2))，運算結果的餘將以與(s)相同的資料類型從(d)進行輸出。

例

資料類型為INT型的情況下



- 至(s1)、(s2)的輸入值為INT型/DINT型/WORD型/DWORD型的資料值。(但是，輸入至(s2)的值为0以外。)

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

- (s1)、(s2)為INT型/WORD型的情況下

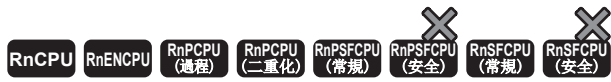
出錯代碼 (SD0)	內容
3400H	當 (s2) 中指定的值 (除數) 為0時。

- (s1)、(s2)為DINT型/DWORD型的情況下

出錯代碼 (SD0)	內容
3400H	當 (s2) 中指定的值 (除數) 為0時。

34.6 冪

EXPT(_E)



輸出輸入值的冪。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=EXPT(s1, s2);</p> <p>[帶EN/ENO] d:=EXPT_E(EN, ENO, s1, s2);</p>

設置數據

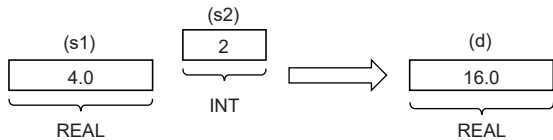
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)	基數	輸入變數	ANY_REAL
s2 (IN2)	指數	輸入變數	ANY_NUM
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_REAL

功能

■運算處理

• 將(s1)中輸入的REAL型/LREAL型資料以(s2)中指定的INT型/DINT型/REAL型/LREAL型進行冪運算，從(d)輸出運算結果。



■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
3402H	(s1)或(s2)中指定的資料為-0、非正規化數、非數、±∞時。

34.7 代入

MOVE(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFPCPU (常規) RnSFPCPU (安全)

輸出輸入值的代入。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=MOVE(s);</p> <p>[帶EN/ENO] d:=MOVE_E(EN, ENO, s);</p>
<p>[帶EN/ENO]</p>	

設置數據

■內容、類型、資料類型

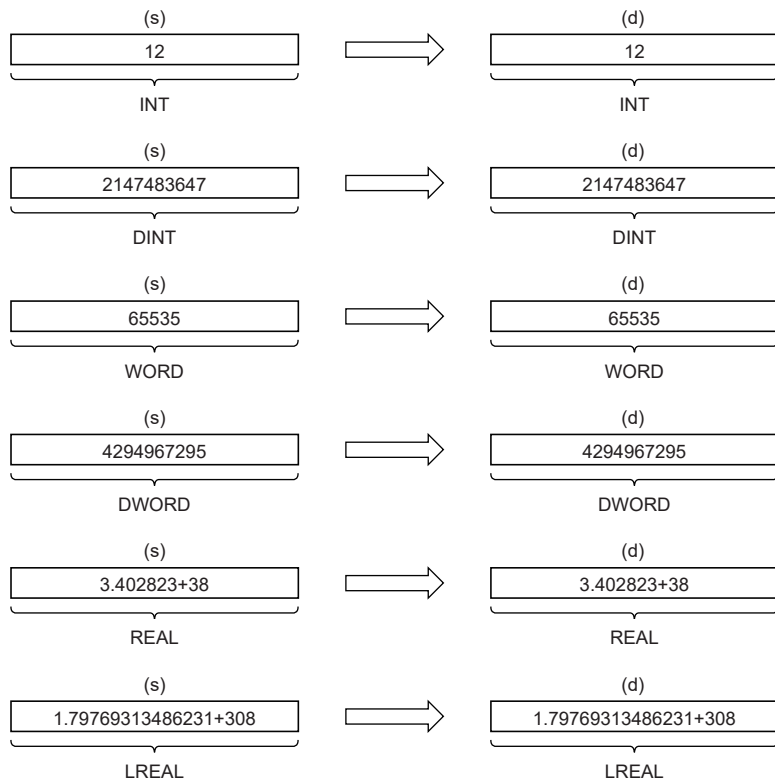
引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY*1
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY*1

*1 安全程式中請勿指定REAL型/LREAL型/STRING型資料。指定了REAL型/LREAL型/STRING型資料的情況下，將變為出錯狀態。

功能

■運算處理

- 將(s)中指定的變數的值代入至(d)中指定的變數中。
- 對於(s)、(d)，可以指定BOOL型/INT型/DINT型/WORD 型/DWORD型/REAL型/LREAL型/STRING型/TIME型/結構體型/數組型。此外，(s)與(d)僅可以指定相同的資料類型。



■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
2820H	(s)中指定的元件編號及其以後，元件/標籤記憶體的各設定區域之間不存在00H時。
3405H	(s)的字元串超過16383字元時。
3406H	(d)中指定的元件編號及其以後，相應元件/標籤記憶體的各設定區域的最終編號為止的點數中無法儲存指定的全部字元串時。

35 位移函數

35.1 n位元左移

SHL(_E)



將輸入值左移(n)位元數後輸出。

梯形圖、FBD/LD		ST
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] d:=SHL(s, n); [帶EN/ENO] d:=SHL_E(EN, ENO, s, n);

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(IN)	輸入	輸入變數	ANY_BIT
n(N)	移位位元數指定	輸入變數	ANY_BIT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_BIT

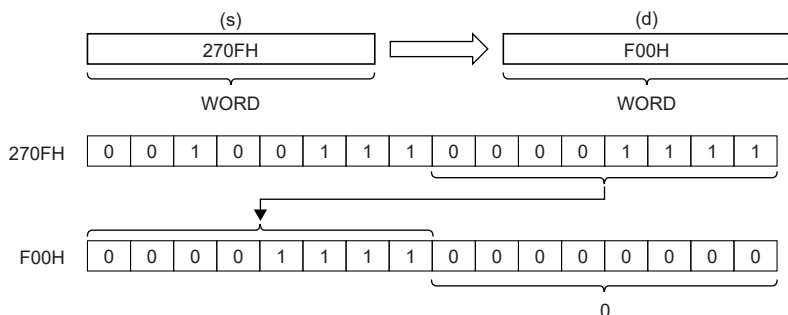
功能

■ 運算處理

- 將(s)中輸入的16位元資料/32位元資料(WORD型/DWORD型)左移(n)位元後，以與(s)相同的資料類型從(d)輸出。
- 左移的(n)位元數為輸入至(n)的值。

例

(s)的資料為16位元資料(WORD型)，至(n)的輸入值為8的情況下



- 從最低位開始的(n)位元將變為0。
- 至(s)的輸入值為16位元資料/32位元資料(WORD型/DWORD型)的資料。
- 至(n)的輸入值(移位位元數指定)為下述所示範圍內。

(s)的資料為16位元資料(WORD型)的情況下	(s)的資料為32位元資料(DWORD型)的情況下
(n)的輸入值為0~15的範圍內。 使用至(n)的輸入值的低位4位元的資料。 [例] 輸入值為6的情況下: 6 輸入值為22的情況下: 6	(n)的輸入值為0~31的範圍內。 使用至(n)的輸入值的低位5位元的資料。 [例] 輸入值為6的情況下: 6 輸入值為22的情況下: 22

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

35.2 n位元右移

SHR(_E)



將輸入值右移(n)位元數後輸出。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=SHR(s, n);</p> <p>[帶EN/ENO] d:=SHR_E(EN, ENO, s, n);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_BIT
n (N)	移位位元數指定	輸入變數	ANY_BIT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_BIT

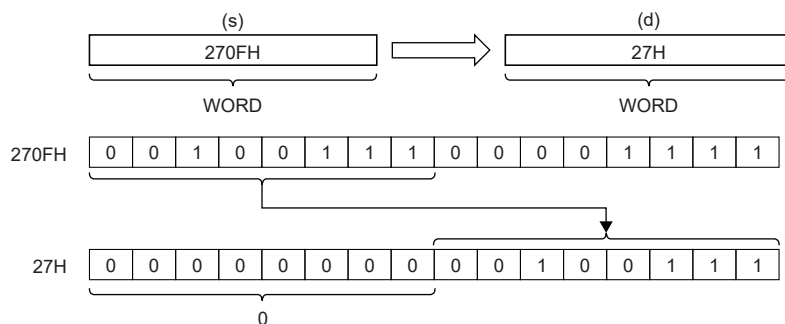
功能

■運算處理

- 將(s)中輸入的16位元資料/32位元資料(WORD型/DWORD型)右移(n)位元後，從(d)輸出與(s)相同的資料類型。
- 右移的(n)位元數時輸入至(n)的值。

例

(s)的資料為16位元資料(WORD型)，至(n)的輸入值為8的情況下



- 從最高位開始的(n)位元將變為0。
- 至(s)的輸入值為16位元資料/32位元資料(WORD型/DWORD型)的資料。
- 至(n)的輸入值(移位位元數指定)為下述所示範圍內。

(s)的資料為16位元資料(WORD型)的情況下	(s)的資料為32位元資料(DWORD型)的情況下
(n)的輸入值為0~15的範圍內。 使用至(n)的輸入值的低位4位元的資料。 [例] 輸入值為6的情況下: 6 輸入值為22的情況下: 6	(n)的輸入值為0~31的範圍內。 使用至(n)的輸入值的低位5位元的資料。 [例] 輸入值為6的情況下: 6 輸入值為22的情況下: 22

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

35.3 n位元左旋轉

ROL(_E)



將輸入值左旋轉(n)位元數後輸出。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=ROL(s, n);</p> <p>[帶EN/ENO] d:=ROL_E(EN, ENO, s, n);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_BIT
n (N)	移位位元數指定	輸入變數	ANY_BIT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_BIT

功能

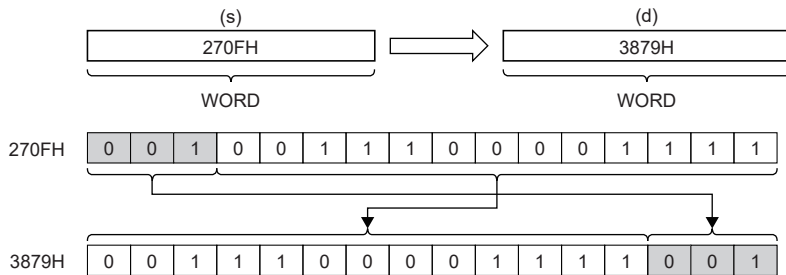
■運算處理

- 將(s)中輸入的16位元資料/32位元資料(WORD型/DWORD型)向左回轉(旋轉)(n)位元後，以與(s)相同的資料類型從(d)輸出。
- 向左回轉的(n)位元數為輸入至(n)的值。



例

(s)的資料類型為16位元資料(WORD型)，至(n)的輸入值為3的情況下。(向左回轉3位元。)



- 至(s)的輸入值為16位元資料/32位元資料(WORD型/DWORD型)的資料。
- 至(n)的輸入值(移位位元數指定)為下述所示範圍內。

(s)的資料為16位元資料(WORD型)的情況下	(s)的資料為32位元資料(DWORD型)的情況下
<p>(n)的輸入值為0~15的範圍內。 使用至(n)的輸入值的低位4位元的資料。 [例] 輸入值為6的情況下: 6 輸入值為22的情況下: 6</p>	<p>(n)的輸入值為0~31的範圍內。 使用至(n)的輸入值的低位5位元的資料。 [例] 輸入值為6的情況下: 6 輸入值為22的情況下: 22</p>

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

35.4 n位元右旋轉

ROR(_E)



將輸入值右旋轉(n)位元數後輸出。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=ROR(s, n);</p> <p>[帶EN/ENO] d:=ROR_E(EN, ENO, s, n);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANY_BIT
n (N)	移位位元數指定	輸入變數	ANY_BIT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_BIT

功能

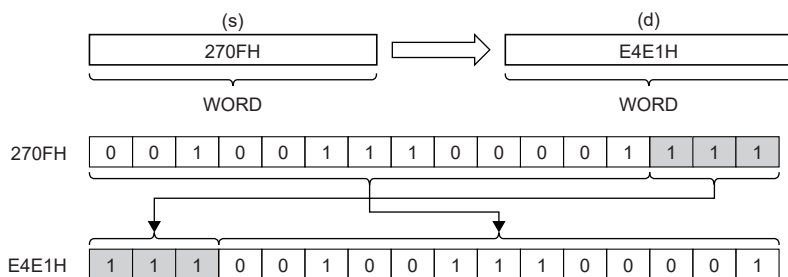
■運算處理

- 將(s)中輸入的16位元資料/32位元資料(WORD型/DWORD型)向右回轉(旋轉)(n)位元後，以與(s)相同的資料類型從(d)輸出。
- 向右旋轉的(n)位元數為輸入至(n)的值。



例

(s)的資料為16位元資料(WORD型)，至(n)的輸入值為3的情況下。(向右旋轉3位元。)



- 至(s)的輸入值為16位元資料/32位元資料(WORD型/DWORD型)的資料。
- 至(n)的輸入值(移位位元數指定)為下述所示範圍內。

(s)的資料為16位元資料(WORD型)的情況下	(s)的資料為32位元資料(DWORD型)的情況下
<p>(n)的輸入值為0~15的範圍內。 使用至(n)的輸入值的低位4位元的資料。 [例] 輸入值為6的情況下: 6 輸入值為22的情況下: 6</p>	<p>(n)的輸入值為0~31的範圍內。 使用至(n)的輸入值的低位5位元的資料。 [例] 輸入值為6的情況下: 6 輸入值為22的情況下: 22</p>

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

36 位型布爾函數

36.1 邏輯且、邏輯或、排他邏輯或

AND(_E)、OR(_E)、XOR(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

- AND(_E)：輸出輸入值的邏輯且。
- OR(_E)：輸出輸入值的邏輯或。
- XOR(_E)：輸出輸入值的排他邏輯或。

梯形圖、FBD/LD*1		ST*1
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] 作為運算符記述。(LUMELSEC iQ-R 程式手冊(程式設計篇)) [帶EN/ENO] d:=AND_E(EN, ENO, s1, s2); d:=OR_E(EN, ENO, s1, s2); d:=XOR_E(EN, ENO, s1, s2);

*1 對於輸入變數s，可以在2~28的範圍內進行更改。

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1) ~ s28 (IN28)	輸入	輸入變數	ANY_BIT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_BIT

功能

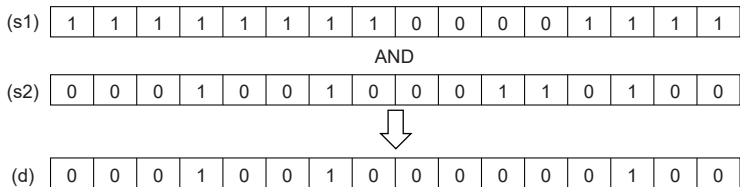
■ 運算處理

1. AND(_E)

• 對(s1)~(s28)中輸入的BOOL型/WORD型/DWORD型資料按各個位進行邏輯且的運算，運算結果中，將與(s)相同的資料類型從(d)輸出。

例

資料類型為WORD型的情況下

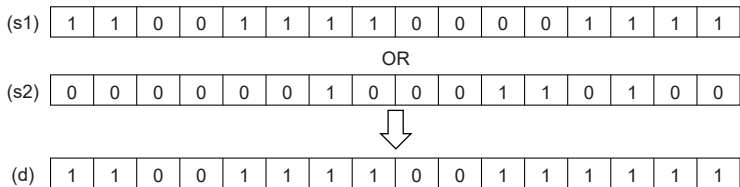


2. OR(_E)

• 對(s1)~(s28)中輸入的BOOL型/WORD型/DWORD型資料按各個位進行邏輯或的運算，運算結果中，將與(s)相同的資料類型從(d)輸出。

例

資料類型為WORD型的情況下

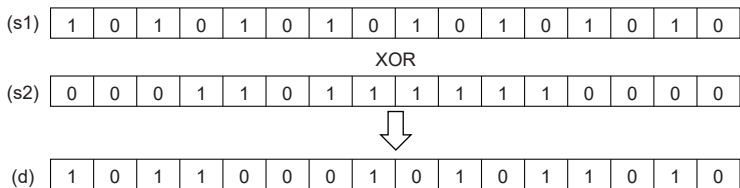


3. XOR(_E)

• 對(s1)~(s28)中輸入的BOOL型/WORD型/DWORD型資料按各個位進行排他邏輯或的運算，運算結果中，將與(s)相同的資料類型從(d)進行輸出。

例

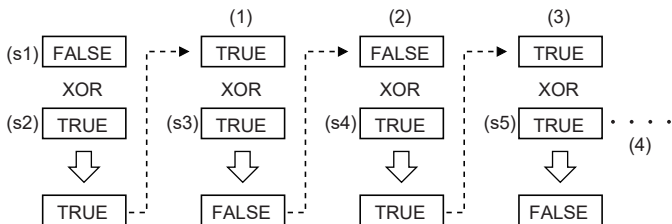
資料類型為WORD型的情況下



• (s)有3個及其以上的情況下，對於將(s1)與(s2)進行了XOR的結果，對(s3)進行XOR。此外，有(s4)的情況下，對於(s3)為進行了XOR的結果，對(s4)進行XOR。以後，僅對(s5)、(s6)與(s)的數進行XOR。

例

資料類型為BOOL型的情況下



- (1): s的數有3個的情況下
- (2): s的數有4個的情況下
- (3): s的數有5個的情況下
- (4): 以後，僅對s的數進行XOR。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

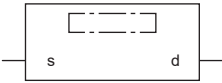
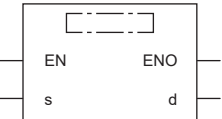
沒有運算出錯。

36.2 邏輯否定

NOT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFPCPU (常規) RnSFPCPU (安全)

輸出輸入值的邏輯否定。

梯形圖、FBD/LD		ST
[無EN/ENO] 	[帶EN/ENO] 	[無EN/ENO] 作為運算符記述。(□MELSEC iQ-R 程式手冊(程式設計篇)) [帶EN/ENO] d:=NOT_E(EN, ENO, s);

36

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(IN)*1	輸入	輸入變數	ANY_BIT*2
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_BIT*2

*1 不能使用DX。

*2 安全程式中請勿指定BOOL型資料。指定了BOOL型資料的情況下，將變為出錯狀態。

功能

■運算處理

- 對(s)中輸入的BOOL型/WORD型/DWORD型資料按各個位進行邏輯否定運算，運算結果中，將與(s)相同的資料類型從(d)輸出。

例

資料類型為WORD型的情況下

(s)	0	1	1	0	1	0	1	1	0	0	0	0	1	1	1	1
	NOT															
(d)	1	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0

- 至(s)的輸入值為BOOL型/WORD型/DWORD型的資料值。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

37 選擇函數

37.1 選擇值

SEL(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFPCPU (常規) RnSFPCPU (安全)

輸出選擇的輸入值。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=SEL(s1, s2, s3);</p> <p>[帶EN/ENO] d:=SEL_E(EN, ENO, s1, s2, s3);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (G)	輸出條件(TRUE: s3輸出、FALSE: s2輸出)	輸入變數	BOOL
s2 (IN0)	輸入	輸入變數	ANY*1
s3 (IN1)			
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY*1

*1 安全程式中請勿指定REAL型/LREAL型/STRING型資料。指定了REAL型/LREAL型/STRING型資料的情況下，將變為出錯狀態。

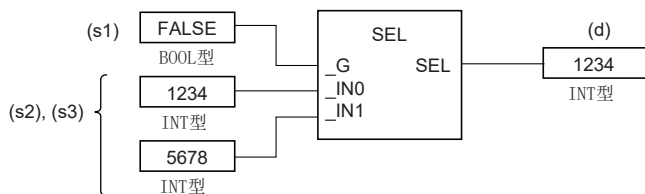
功能

■運算處理

- 按照輸入至(s1)的輸入值，將輸入至(s2)、(s3)中的其中一個值按與(s2)、(s3)相同的資料類型從(d)輸出。
- (s1)的輸入值為FALSE(=0)的情況下，從(d)輸出(s2)的輸入值。
- (s1)的輸入值為TRUE(=1)的情況下，從(d)輸出(s3)的輸入值。

例

(s2)、(s3)的資料類型為INT型的情況下(引數名的(s2)、(s3)對應於(s1)的位值(0或1)。)



- (s1)的輸入值為BOOL型的資料值。
- 至(s2)、(s3)的輸入值為BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型/結構體型/數組型的資料值。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

- (s2)、(s3)為STRING型的情況下

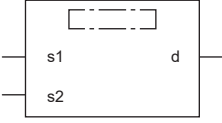
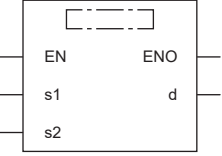
出錯代碼(SD0)	內容
2820H	自(s2)中指定的標籤或元件編號開始至相應元件的最終元件編號為止之間未設定00H時。
	自(s3)中指定的標籤或元件編號開始至相應元件的最終元件編號為止之間未設定00H時。
3406H	自(d)中指定的標籤或元件編號開始至相應元件的最終元件編號為止的點數中，指定的字元串無法全部儲存時。

37.2 最大值、最小值選擇

MAX(_E)、MIN(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

- MAX(_E)：輸出輸入值的最大值。
- MIN(_E)：輸出輸入值的最小值。

梯形圖、FBD/LD*1		ST
[無EN/ENO] 	[帶EN/ENO] 	[無EN/ENO] d:=MAX(s1, s2); d:=MIN(s1, s2); [帶EN/ENO] d:=MAX_E(EN, ENO, s1, s2); d:=MIN_E(EN, ENO, s1, s2);

*1 對於輸入變數s，可在2~28的範圍內進行更改。

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1(IN1)~s28(IN28)	輸入	輸入變數	ANY_ELEMENTARY*1
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_ELEMENTARY*1

*1 安全程式中請勿指定REAL型/LREAL型/STRING型資料。指定了REAL型/LREAL型/STRING型資料的情況下，將變為出錯狀態。

功能

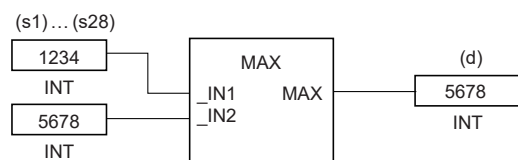
■ 運算處理

• MAX(_E)

(s1) ~ (s28) 中輸入的BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型資料的最大值中，將與(s)相同的資料類型從(d)輸出。

例

資料類型為INT型的情況下

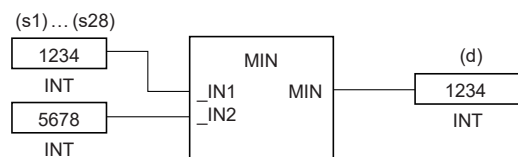


• MIN(_E)

(s1) ~ (s28) 中輸入的BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型資料的最小值中，將與(s)相同的資料類型從(d)輸出。

例

資料類型為INT型的情況下



• 至(s1) ~ (s28)的輸入值為BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型的資料值。

• 進行字元串比較的情況下，比較條件如下所示。

一致的條件：

• 所有的字元串一致的情況下

較大字元串的條件：

• 不同字元串的情況下，字元代碼較大的字元串
• 字元串的長度不同的情況下，字元串較長的字元串

較小字元串的條件：

• 不同字元串的情況下，字元代碼較小的字元串
• 字元串的長度不相同的情況下，字元串較短的字元串

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
2820H	(s1) ~ (s28) 中指定的元件編號及其以後，元件/標籤記憶體的各設定區域之間不存在00H時。
3405H	(s1) ~ (s28) 的字元串超過16383字元時。
3406H	(d) 中指定的元件編號及其以後，相應元件/標籤記憶體的各設定區域的最終編號為止的點數中無法儲存指定的全部字元串時。

37.3 上下限限位控制

LIMIT(_E)



輸出上下限限位控制的輸入值。

梯形圖、FBD/LD		ST
[無EN/ENO]		[無EN/ENO] $d := \text{LIMIT}(s1, s2, s3);$ [帶EN/ENO] $d := \text{LIMIT_E}(\text{EN}, \text{ENO}, s1, s2, s3);$

設置數據

■ 內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (MN)*1	下限限位值(最小輸出極限值)	輸入變數	ANY_ELEMENTARY
s2 (IN)*1	透過上下限限位控制來控制的輸入值	輸入變數	ANY_ELEMENTARY
s3 (MX)*1	上限限位值(最大輸出極限值)	輸入變數	ANY_ELEMENTARY
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY_ELEMENTARY

*1 不能使用DX。

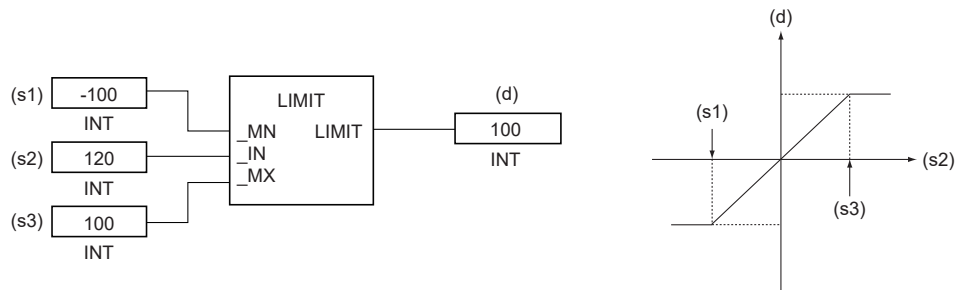
功能

■ 運算處理

- 按照輸入至(s1)、(s2)、(s3)的BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型資料，將與(s1)、(s2)、(s3)相同的資料類型從(d)輸出。
- (s2)的輸入值>(s3)的輸入值的情況下，從(d)輸出(s3)的輸入值。
- (s2)的輸入值<(s1)的輸入值的情況下，從(d)輸出(s1)的輸入值。
- (s1)的輸入值≤(s2)的輸入值≤(s3)的輸入值的情況下，從(d)輸出(s2)的輸入值。

例

資料類型為INT型的情況下



- 至(s1)、(s2)、(s3)的輸入值為BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型的資料值。(但是，(s1)的輸入值<(s3)的輸入值)
- 進行字元串比較的情況下，比較條件如下所示。
 - 一致的條件：
 - 所有的字元串一致的情況下
 - 較大字元串的條件：
 - 不同字元串的情況下，字元代碼較大的字元串
 - 字元串的長度不同的情況下，字元串較長的字元串
 - 較小字元串的條件：
 - 不同字元串的情況下，字元代碼較小的字元串
 - 字元串的長度不相同的情況下，字元串較短的字元串

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

- (s1)、(s2)、(s3) 為INT型/WORD型的情況下

出錯代碼(SD0)	內容
3405H	(s1)中指定的下限限位值大於(s2)中指定的上限限位值時。

- (s1)、(s2)、(s3) 為DINT型/DWORD型/TIME型的情況下

出錯代碼(SD0)	內容
3405H	(s1)中指定的下限限位值大於(s2)中指定的上限限位值時。

- (s1)、(s2)、(s3) 為BOOL型的情況下

出錯代碼(SD0)	內容
3405H	(s1)中指定的下限限位值大於(s3)中指定的上限限位值時。

- (s1)、(s2)、(s3) 為REAL型的情況下

出錯代碼(SD0)	內容
3402H	(s1)內容為下述範圍以外時。 $-2^{128} < (s1) \leq -2^{-126}$ 、 0 、 $2^{-126} \leq (s1) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38) (s1)中指定的資料為-0、非正規化數、非數、±∞時。
	(s2)內容為下述範圍以外時。 $-2^{128} < (s2) \leq -2^{-126}$ 、 0 、 $2^{-126} \leq (s2) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38) (s2)中指定的資料為-0、非正規化數、非數、±∞時。
	(s3)的內容為下述範圍外時。 $-2^{128} < (s3) \leq -2^{-126}$ 、 0 、 $2^{-126} \leq (s3) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38) (s3)中指定的資料為-0、非正規化數、非數、±∞時。
3405H	(s1)中指定的下限限位值大於(s3)中指定的上限限位值時。

- (s1)、(s2)、(s3) 為LREAL型的情況下

出錯代碼(SD0)	內容
3402H	(s1)內容為下述範圍以外時。 $-2^{1024} < (s1) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s1) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308) (s1)中指定的資料為-0、非正規化數、非數、±∞時。
	(s2)內容為下述範圍以外時。 $-2^{1024} < (s2) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s2) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308) (s2)中指定的資料為-0、非正規化數、非數、±∞時。
	(s3)的內容為下述範圍外時。 $-2^{1024} < (s3) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s3) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308) (s3)中指定的資料為-0、非正規化數、非數、±∞時。
3405H	(s1)中指定的下限限位值大於(s3)中指定的上限限位值時。

- (s1)、(s2)、(s3) 為STRING型的情況下

出錯代碼(SD0)	內容
2820H	自(s1)、(s2)、(s3)中指定標籤或指定的元件編號開始至相應元件的最終元件編號為止之間未設定00H時。
3405H	(s1)中指定的下限限位值大於(s3)中指定的上限限位值時。 (s1)、(s2)、(s3)中指定的字元串的字元數超過了16383字元時。
3406H	自(d)中指定的標籤或指定的元件編號開始至相應元件的最終元件編號為止的點數中，指定的字元串無法全部儲存時。

37.4 多路復用器

MUX(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCEPU (常規) RnSFCEPU (安全)

輸出多個輸入值中之一。

梯形圖、FBD/LD*1	ST*1
<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px;"> <p>[無EN/ENO]</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>[帶EN/ENO]</p> </div> </div>	<p>[無EN/ENO] <code>d:=MUX (n, s1, s2);</code> [帶EN/ENO] <code>d:=MUX_E (EN, ENO, n, s1, s2);</code></p>

*1 對於輸入變數s，可在2~28的範圍內進行更改。

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
n(K)	輸出值選擇	輸入變數	ANY_INT
s1 (IN0)~s28 (IN27)	輸入	輸入變數	ANY*1
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANY*1

*1 安全程式中請勿指定REAL型/LREAL型/STRING型資料。指定了REAL型/LREAL型/STRING型資料的情況下，將變為出錯狀態。

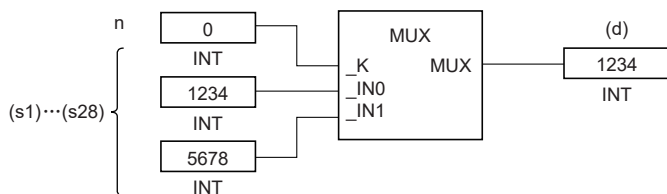
功能

■運算處理

- 按照輸入至(n)的輸入值，將輸入至(s1)~(s28)中的其中一個值，按與(s)相同的資料類型從(d)輸出。
- (n)的輸入值為0的情況下，從(d)輸出輸入至(s1)的值。
- (n)的輸入值為(n)-1的情況下，從(d)輸出輸入至(sn)的值。

例

資料類型為INT型的情況下



- (n)中輸入了(s)的針數範圍外的情況下，從(d)輸出不定值。(不變為運算出錯。此外MUX_E時，透過(ENO)輸出FALSE。)
- 至(n)的輸入值為INT型/DINT型/WORD型/DWORD型的資料值且在0~27的範圍內。(但是，在(s)的針數範圍內。)
- 至(s)的輸入值為BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型/結構體型/數組型的資料值。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
2820H	(s1)~(s28)中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在00H時。
3405H	(s1)~(s28)的字元串超過16383字元時。
3406H	(d)中指定的元件編號及其以後，相應元件/標籤記憶體各設定區域的最終編號為止的點數中無法儲存指定的全部字元串時。

38 比較函數

38.1 比較

GT(_E)、GE(_E)、EQ(_E)、LE(_E)、LT(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFPCPU (常規) RnSFPCPU (安全)

輸出輸入值的資料比較結果。

梯形圖、FBD/LD*1	ST*1
<p>[無EN/ENO]</p>	<p>[無EN/ENO]*2</p> <pre>d:=GT(s1, s2); d:=GE(s1, s2); d:=EQ(s1, s2); d:=LE(s1, s2); d:=LT(s1, s2);</pre> <p>[帶EN/ENO]</p> <pre>d:=GT_E(EN, ENO, s1, s2); d:=GE_E(EN, ENO, s1, s2); d:=EQ_E(EN, ENO, s1, s2); d:=LE_E(EN, ENO, s1, s2); d:=LT_E(EN, ENO, s1, s2);</pre>

*1 對於輸入變數s，可以在2~28的範圍內進行更改。

*2 在版本為“1.035M”及其以後的工程工具中對應。在那之前的版本中，作為運算符進行記述。(MELSEC iQ-R 程式手冊(程式設計篇))

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1) ~ s28 (IN28)*1	輸入	輸入變數	ANY_ELEMENTARY*2
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出(TRUE: 真值、FALSE: 假值)	輸出變數	BOOL

*1 不能使用DX。

*2 安全程式中請勿指定REAL型/LREAL型/STRING型資料。指定了REAL型/LREAL型/STRING型資料的情況下，將變為出錯狀態。

功能

■ 運算處理

- 進行輸入至(s)的輸入值的比較運算後，將運算結果以BOOL型從(d)進行輸出。
- GT(E)：進行 $[(s1) > (s2)] \& [(s2) > (s3)] \& \dots \& [(s)_{(n-1)} > (s)_{(n)}]$ 的比較。
 - 全部 $(s)_{(n-1)} > (s)_{(n)}$ 時，輸出TRUE。
 - 某個為 $(s)_{(n-1)} \leq (s)_{(n)}$ 時，輸出FALSE。
- GE(E)：進行 $[(s1) \geq (s2)] \& [(s2) \geq (s3)] \& \dots \& [(s)_{(n-1)} \geq (s)_{(n)}]$ 的比較。
 - 全部 $(s)_{(n-1)} \geq (s)_{(n)}$ 時，輸出TRUE。
 - 某個為 $(s)_{(n-1)} < (s)_{(n)}$ 時，輸出FALSE。
- EQ(E)：進行 $[(s1) = (s2)] \& [(s2) = (s3)] \& \dots \& [(s)_{(n-1)} = (s)_{(n)}]$ 的比較。
 - 全部 $(s)_{(n-1)} = (s)_{(n)}$ 時，輸出TRUE。
 - 某個為 $(s)_{(n-1)} \neq (s)_{(n)}$ 時，輸出FALSE。
- LE(E)：進行 $[(s1) \leq (s2)] \& [(s2) \leq (s3)] \& \dots \& [(s)_{(n-1)} \leq (s)_{(n)}]$ 的比較。
 - 全部 $(s)_{(n-1)} \leq (s)_{(n)}$ 時，輸出TRUE。
 - 某個為 $(s)_{(n-1)} > (s)_{(n)}$ 時，輸出FALSE。
- LT(E)：進行 $[(s1) < (s2)] \& [(s2) < (s3)] \& \dots \& [(s)_{(n-1)} < (s)_{(n)}]$ 的比較。
 - 全部 $(s)_{(n-1)} < (s)_{(n)}$ 時，輸出TRUE。
 - 某個為 $(s)_{(n-1)} \geq (s)_{(n)}$ 時，輸出FALSE。
- 至(s)的輸入值為INT型/DINT型/REAL型/LREAL型/BOOL型/WORD型/DWORD型/TIME型/STRING型的資料值。不能指定WSTRING型的Unicode字元串。
- 進行字元串比較的情況下，比較條件如下所示。
 - 一致的條件：
 - 所有的字元串一致的情況下
 - 較大字元串的條件：
 - 不同字元串的情況下，字元代碼較大的字元串
 - 字元串的長度不同的情況下，字元串較長的字元串
 - 較小字元串的條件：
 - 不同字元串的情況下，字元代碼較小的字元串
 - 字元串的長度不相同的情況下，字元串較短的字元串

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SDO)	內容
2820H	(s1)~(s28)中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在00H時。
3405H	(s1)~(s28)的字元串超過16383字元時。
3406H	(d)中指定的元件編號及其以後，相應元件/標籤記憶體各設定區域的最終編號為止的點數中無法儲存指定的全部字元串時。

38.2 比較

NE(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

輸出輸入值的資料比較結果。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO]*1</p> <pre>d:=NE(s1, s2);</pre> <p>[帶EN/ENO]</p> <pre>d:=NE_E(EN, ENO, s1, s2);</pre>

*1 在版本為“1.035M”及其以後的工程工具中對應。在那之前的版本中，作為運算符進行記述。(MELSEC iQ-R 程式手冊(程式設計篇))

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1、s2	輸入	輸入變數	ANY_ELEMENTARY*1
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出(TRUE: 真值、FALSE: 假值)	輸出變數	BOOL

*1 安全程式中請勿指定REAL型/LREAL型/STRING型資料。指定了REAL型/LREAL型/STRING型資料的情況下，將變為出錯狀態。

功能

■運算處理

- 進行輸入至(s)的輸入值的比較運算後，將運算結果以BOOL型從(d)進行輸出。
- NE(_E): 進行[(s1)≠(s2)]的比較。
 - (s1)≠(s2)時輸出TRUE。
 - (s1)=(s2)時輸出FALSE。
- 至(s)的輸入值為INT型/DINT型/REAL型/LREAL型/BOOL型/WORD型/DWORD型/TIME型/STRING型的資料值。不能指定WSTRING型的Unicode字元串。
- 進行字元串比較的情況下，比較條件如下所示。

一致的條件:	• 所有的字元串一致的情況下
較大字元串的條件:	• 不同字元串的情況下，字元代碼較大的字元串
	• 字元串的長度不同的情況下，字元串較長的字元串
較小字元串的條件:	• 不同字元串的情況下，字元代碼較小的字元串
	• 字元串的長度不相同的情況下，字元串較短的字元串

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

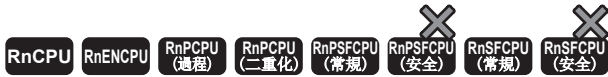
出錯

出錯代碼 (SD0)	內容
2820H	(s) 中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在00H時。
3405H	(s) 的字元串超過了16383字元時。
3406H	(d) 中指定的元件編號及其以後，相應元件/標籤記憶體各設定區域的最終編號為止的點數中無法儲存指定的全部字元串時。

39 字元串函數

39.1 字元串的長度檢測

LEN(_E)



檢測並輸出輸入的字元串的長度。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=LEN(s);</p> <p>[帶EN/ENO] d:=LEN_E(EN, ENO, s);</p>
<p>[帶EN/ENO]</p>	

設置數據

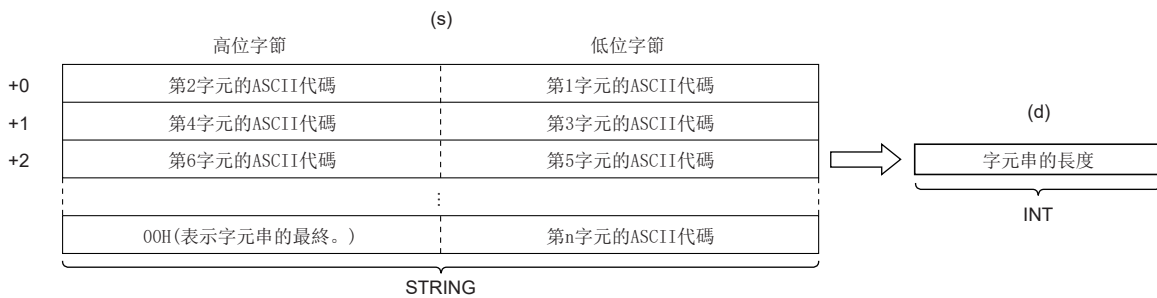
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANYSTRING_SINGLE
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 檢測(s)中輸入的字元串的長度後從(d)進行輸出。



- 至(s)的輸入值為STRING型的資料值且在0~255字節的範圍內。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

39.2 從字元串的左側、右側提取

LEFT(_E)、RIGHT(_E)



- LEFT(_E)：從輸入的字元串資料的左側開始輸出指定字元。
- RIGHT(_E)：從輸入的字元串資料的右側開始輸出指定字元。

梯形圖、FBD/LD		ST
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] d:=LEFT(s, n); d:=RIGHT(s, n); [帶EN/ENO] d:=LEFT_E(EN, ENO, s, n); d:=RIGHT_E(EN, ENO, s, n);

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s(IN)	輸入	輸入變數	ANYSTRING_SINGLE
n(L)	提取字元數指定	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANYSTRING_SINGLE

功能

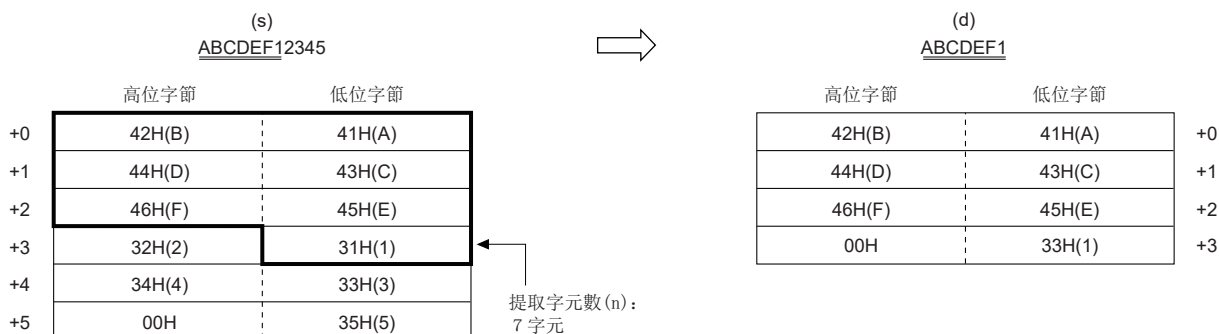
■ 運算處理

• LEFT(_E)

自輸入至(s)的字元串的左側開始將指定字元數的資料從(d)輸出。
提取的字元數根據至(n)的輸入值進行指定。

例

至(n)的輸入值為7的情況下

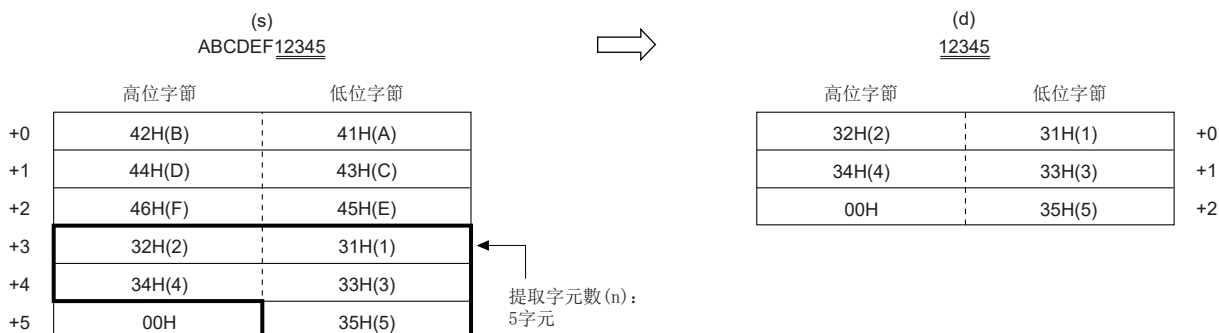


• RIGHT(_E)

自輸入至(s)的字元串的右側開始從將指定字元數的資料從(d)輸出。
提取的字元數根據至(n)的輸入值進行指定。

例

至(n)的輸入值為5的情況下



- 至(s)的輸入值為STRING型的資料值且在0~255字節的範圍內。
- 至(n)的輸入值為INT型的資料值且在0~255的範圍內。(但是，僅限不超過被輸入至(s)的字元串的字元數。)

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

39.3 字元串的提取

MID(_E)



從輸入的字元串的任意位置開始輸出指定字元。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=MID(s, n1, n2);</p> <p>[帶EN/ENO] d:=MID_E(EN, ENO, s, n1, n2);</p>

設置數據

內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANYSTRING_SINGLE
n1 (L)	提取字元數指定	輸入變數	INT
n2 (P)	提取起始位置指定	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANYSTRING_SINGLE

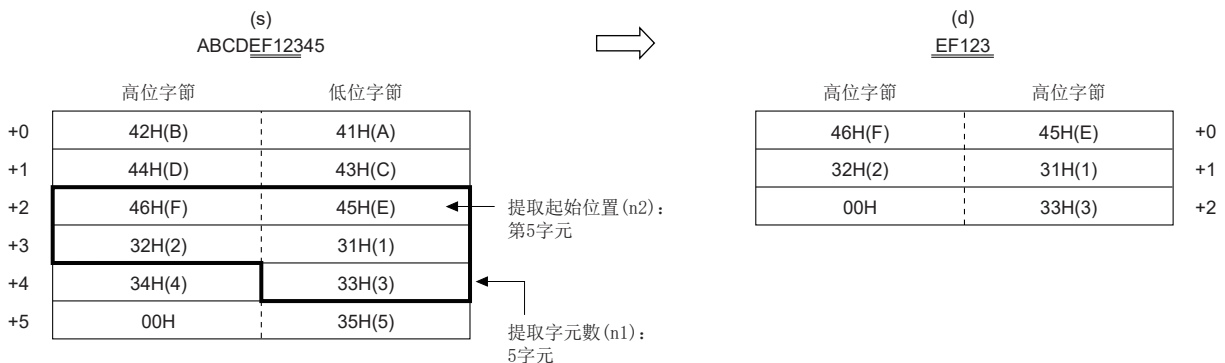
功能

運算處理

- 自輸入至(s)的字元串的任意位置開始將指定字元數的資料從(d)輸出。
- 提取的字元數根據至(n1)的輸入值進行指定。
- 提取的字元串的起始位置根據至(n2)的輸入值進行指定。

例

至(n1)、(n2)的輸入值為5的情況下



- 至(s)的輸入值為STRING型的資料值且在0~255字節的範圍內。
- 至(n1)的輸入值為INT型的資料值且在0~255的範圍內。(但是, 僅限不超過被輸入至(s)的字元串的字元數。)
- 至(n2)的輸入值為INT型的資料值且在1~255的範圍內。(但是, 僅限不超過被輸入至(s)的字元串的字元數。)

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
2820H	(s)中指定的元件編號及其以後，元件/標籤記憶體各設定區域中不存在00H時。
3405H	(s)的字元串超過了16383字元時。 (n1)、(n2)中設定了超出允許指定範圍的資料時。 <ul style="list-style-type: none">• (n1)、(n2)的值為0及其以下• (n2)的值為有效值(-1、0、1及其以上)以外• (n1)的值超過了(s)的字元數• (n1)與(n2)的加法運算後的值超過了(s)的字元數

39.4 字元串的合併

CONCAT(_E)



合併字元串後輸出。

梯形圖、FBD/LD*1	ST*1
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=CONCAT(s1, s2);</p> <p>[帶EN/ENO] d:=CONCAT_E(EN, ENO, s1, s2);</p>

*1 對於輸入變數s，可以在2~28的範圍內進行更改。

設置數據

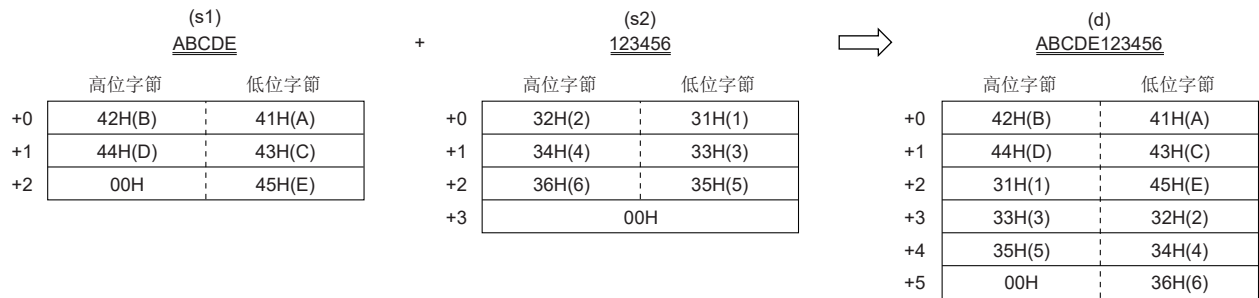
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1) ~ s28 (IN28)	輸入	輸入變數	ANYSTRING_SINGLE
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANYSTRING_SINGLE

功能

■運算處理

- 在輸入至(s1)的字元串後面合併輸入至輸入變數(s2)~(s28)的字元串、並從(d)進行輸出。
- 字元串合併時，將忽略表示(s1)的字元串的結束的00H、連接(s2)~(s28)的字元串進行合併。
- 合併的字元串超過了255字節的情況下，輸出直到255字節為止的字元串。



- 至輸入變數(s1)、(s2)~(s28)的輸入值為STRING型的資料值且在0~255字節的範圍內。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
2820H	(s1)~(s28)中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在00H時。
	(d)中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在00H時。
2821H	(s1)~(s28)中(d)中指定的字元串的儲存元件編號重複時。
3405H	(s1)~(s28)的字元串超過16383字元時。
	(s1)~(s28)的字元串為0字元時。
	(d)的字元串超過16383字元時。
3406H	(d)中指定的元件編號及其以後，相應元件/標籤記憶體各設定區域的最終編號為止的點數無法儲存全部合併的字元串時。

39.5 字元串的插入

INSERT(_E)



在字元串之間插入字元串後輸出。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO]</p> <pre>d:=INSERT(s1, s2, n);</pre> <p>[帶EN/ENO]</p> <pre>d:=INSERT_E(EN, ENO, s1, s2, n);</pre>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)、s2 (IN2)	輸入	輸入變數	ANYSTRING_SINGLE
n (P)	插入起始位置指定	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANYSTRING_SINGLE

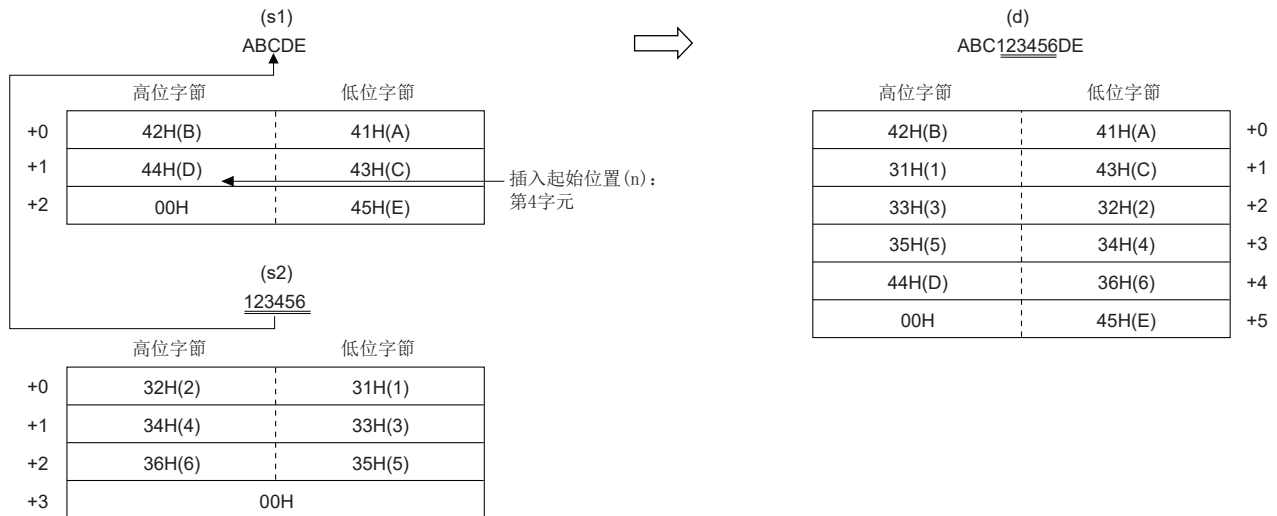
功能

■運算處理

- 將輸入至(s2)的字元串插入到從輸入至(s1)字元串的起始開始的第n字元(插入起始位置)處後，從(d)進行輸出。
- 將(s2)的字元串插入到(s1)的字元串後，表示(s2)字元串的結束的00H將忽略。
- 插入的字元串超過了255字節的情況下，輸出直到255字節為止的字元串。

例

至(n)的輸入值為4的情況下



- 至(s1)、(s2)的輸入值為STRING型的資料值且在0~255字節的範圍內。
- 至(n)的輸入值為INT型的資料值且在1~255的範圍內。(但是，僅限不超過被輸入至(s1)的字元串的字元數。)

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

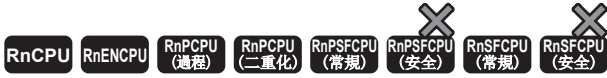
*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
2820H	(s1)~(s28)中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在00H時。
	(d)中指定的元件編號及其以後，元件/標籤記憶體各設定區域之間不存在00H時。
2821H	(s1)~(s28)中(d)中指定的字元串的儲存元件編號重複時。
3405H	(s1)~(s28)的字元串超過16383字元時。
	(s1)~(s28)的字元串為0字元時。
	(d)的字元串超過16383字元時。
3406H	(d)中指定的元件編號及其以後，相應元件/標籤記憶體各設定區域的最終編號為止的點數無法儲存全部合併的字元串時。

39.6 字元串的刪除

DELETE (_E)



刪除字元串的任意範圍後輸出。

梯形圖、FBD/LD	ST
<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px;"> <p>[無EN/ENO]</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>[帶EN/ENO]</p> </div> </div>	<p>[無EN/ENO] d:=DELETE (s, n1, n2);</p> <p>[帶EN/ENO] d:=DELETE_E (EN, ENO, s, n1, n2);</p>

設置數據

內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸入	輸入變數	ANYSTRING_SINGLE
n1 (L)	刪除字元數指定	輸入變數	INT
n2 (P)	刪除起始位置指定	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANYSTRING_SINGLE

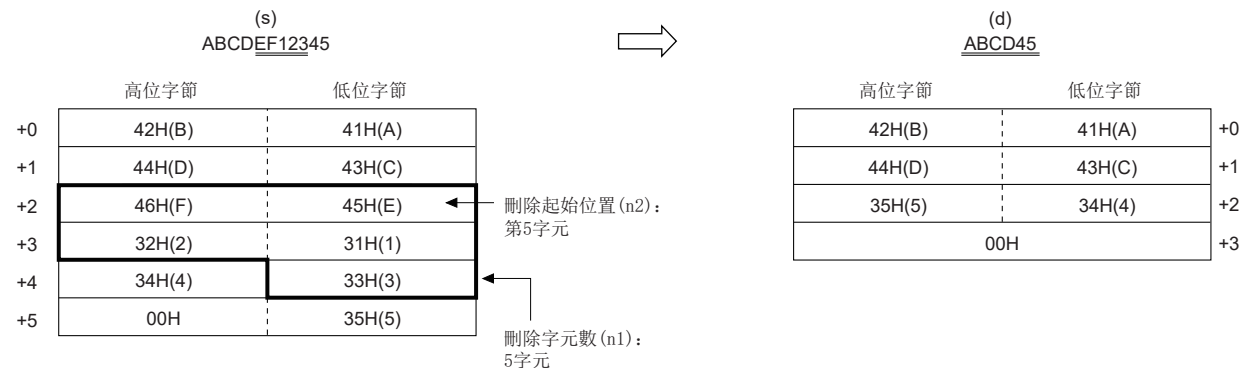
功能

運算處理

- 自輸入至(s)的字元串的任意位置開始刪除指定字元數的資料、並將剩餘的字元數從(d)輸出。
- 刪除的字元數根據至(n1)的輸入值進行指定。
- 刪除的字元串的起始位置根據至(n2)的輸入值進行指定。

例

至(n1)、(n2)的輸入值為5的情況下



- 至(s)的輸入值為STRING型的資料值且在0~255字節的範圍內。
- 至(n1)的輸入值為INT型的資料值且在0~255的範圍內。(但是，僅限不超過被輸入至(s)的字元串的字元數。)
- 至(n2)的輸入值為INT型的資料值且在1~255的範圍內。(但是，僅限不超過被輸入至(s)的字元串的字元數。)

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

出錯代碼(SD0)	內容
2820H	自(s)中指定的標籤或指定的元件編號開始至相應元件的最終元件編號為止之間未設定00H時。
3405H	當(s)中指定的字元串的字元數超過了255字元時。
	(n1)中指定的內容為0~255的範圍外時。
	(n2)中指定的內容為1~255的範圍外時。
	(n1)超過了(s)中指定的字元串的字元數時。
(n2)超過了(s)中指定的字元串的字元數時。	
3406H	自(d)中指定的標籤或指定的元件編號開始至相應元件的最終元件編號為止的點數中，刪除後的字元串無法全部儲存時。

39.7 字元串的替換

REPLACE(_E)



替換字元串的任意範圍後輸出。

梯形圖、FBD/LD		ST
<p>[無EN/ENO]</p>	<p>[帶EN/ENO]</p>	<p>[無EN/ENO] d:=REPLACE(s1, s2, n1, n2);</p> <p>[帶EN/ENO] d:=REPLACE_E(EN, ENO, s1, s2, n1, n2);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)、s2 (IN2)	輸入	輸入變數	ANYSTRING_SINGLE
n1 (L)	替換字元數指定	輸入變數	INT
n2 (P)	替換起始位置指定	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	ANYSTRING_SINGLE

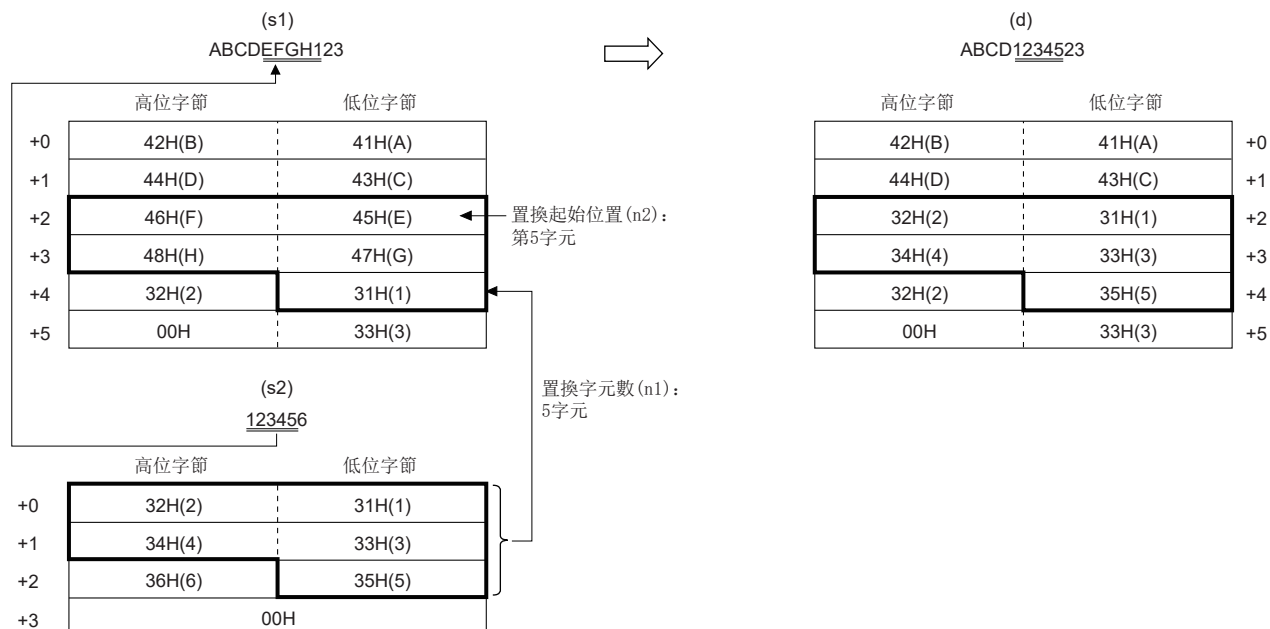
功能

■ 運算處理

- 自輸入至 (s1) 的字元串的任意位置開始，將指定字元數的資料替換為輸入至 (s2) 的字元串後從 (d) 進行輸出。
- 替換的字元數根據至 (n1) 的輸入值進行指定。
- 替換的字元串的起始位置根據至 (n2) 的輸入值進行指定。

例

至 (n1)、(n2) 的輸入值為5的情況下



- 至 (s1)、(s2) 的輸入值為STRING型的資料值且在0~255字節的範圍內。
- 至 (n1) 的輸入值為INT型的資料值且在0~255的範圍內。(但是，僅限不超過被輸入至 (s1) 的字元串的字元數。)
- 至 (n2) 的輸入值為INT型的資料值且在1~255的範圍內。(但是，僅限不超過被輸入至 (s1) 的字元串的字元數。)

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從 (d) 輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從 (d) 中輸出的資料將變為不定值。在此情況下，應進行不使用從 (d) 中輸出的資料的程式處理。

出錯

出錯代碼 (SD0)	內容
2820H	自 (s1) 中指定標籤或指定的元件編號開始至相應元件的最終元件編號為止之間未設定00H時。
	自 (s2) 中指定標籤或指定的元件編號開始至相應元件的最終元件編號為止之間未設定00H時。
3405H	(s1) 中指定的字元串的字元數超過了255字元時。
	(s2) 中指定的字元串的字元數超過了255字元時。
	(n1) 中指定的內容為0~255的範圍外時。
	(n2) 中指定的內容為1~255的範圍外時。
	(n1) 超過了 (s2) 中指定的字元串的字元數時。
	(n2) 超過了 (s1) 中指定的字元串的字元數時。
3406H	自 (d) 中指定的標籤或指定的元件編號開始至相應元件的最終元件編號為止的點數中，刪除後的字元串無法全部儲存時。

39.8 字元串的搜尋

FIND(_E)



搜尋字元串並輸出搜尋結果。

梯形圖、FBD/LD	ST
<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px;"> <p>[無EN/ENO]</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>[帶EN/ENO]</p> </div> </div>	<p>[無EN/ENO] d:=FIND(s1, s2);</p> <p>[帶EN/ENO] d:=FIND_E(EN, ENO, s1, s2);</p>

設置數據

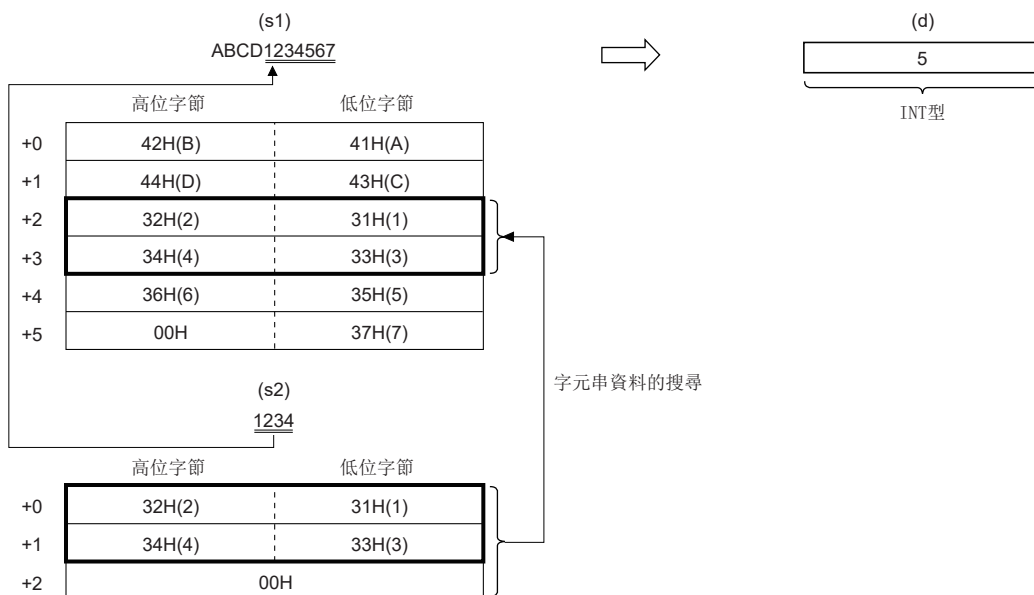
■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)、s2 (IN2)	輸入	輸入變數	ANYSTRING_SINGLE
ENO	輸出狀態(TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	INT

功能

■運算處理

- 自輸入至(s1)的字元串的最初搜尋輸入至(s2)的字元串，並將搜尋結果從(d)輸出。
- 將搜尋結果於最初搜尋的字元串的起始字元位置輸出。
- 若無法自(s1)的字元串搜尋(s2)的字元串，則輸出0。



- 至(s1)、(s2)的輸入值為STRING型的資料值且在0~255字節的範圍內。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

40 時間資料類型函數

40.1 加法運算

ADD_TIME(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

輸出輸入值 (TIME型) 的和 ((s1)+(s2))。

梯形圖、FBD/LD		ST
<p>[無EN/ENO]</p>	<p>[帶EN/ENO]</p>	<p>[無EN/ENO] d:=ADD_TIME(s1, s2);</p> <p>[帶EN/ENO] d:=ADD_TIME_E(EN, ENO, s1, s2);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)、s2 (IN2)	輸入	輸入變數	TIME
ENO	輸出狀態 (TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	TIME

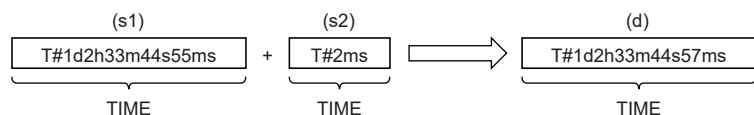
功能

■ 運算處理

- 進行 (s1)、(s2) 中輸入的 TIME 型資料的加法運算 ((s1)+(s2)) 後，將運算結果從 (d) 以 TIME 型進行輸出。

例

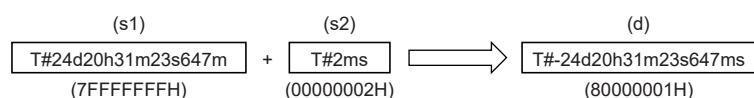
至 (s1)、(s2) 的輸入值為 T#1d2h33m44s55ms (1日2小時33分44秒55毫秒) 與 T#2ms (2毫秒) 的情況下



- 至 (s1)、(s2) 的輸入值為 TIME 型的資料值。
- 即使運算結果中發生了下溢/上溢，也不會變為運算出錯。按下述方式輸出至 (d) 中。此外，ADD_TIME_E 的情況下，從輸出變數 ENO 輸出 TRUE。

例

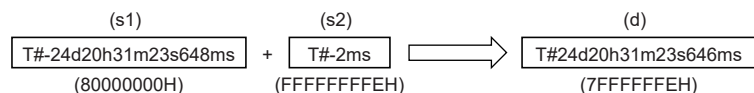
上溢



由於最高位的位元變為 1，因此變為負的時間。

例

下溢



由於最高位的位元變為 0，因此變為正的時間。

■ 運算結果

1. 無 EN/ENO 函數

執行運算處理。從 (d) 輸出運算輸出值。

2. 帶 EN/ENO 函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在 ENO 輸出了 FALSE 的情況下，從 (d) 中輸出的資料將變為不定值。在此情況下，應進行不使用從 (d) 中輸出的資料的程式處理。

出錯

沒有運算出錯。

40.2 減法運算

SUB_TIME(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSF CPU (常規) RnSF CPU (安全)

輸出輸入值 (TIME型) 的差 ((s1)-(s2))。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO] d:=SUB_TIME(s1, s2);</p> <p>[帶EN/ENO] d:=SUB_TIME_E(EN, ENO, s1, s2);</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)、s2 (IN2)	輸入	輸入變數	TIME
ENO	輸出狀態 (TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	TIME

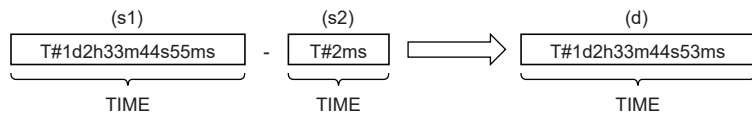
功能

■運算處理

- 進行 (s1)、(s2) 中輸入的 TIME 型資料的減法運算 ((s1)-(s2)) 後，將運算結果從 (d) 以 TIME 型進行輸出。

例

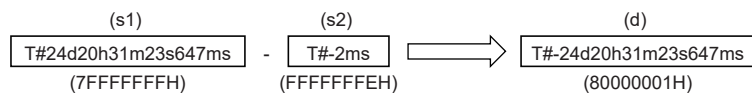
至 (s1)、(s2) 的輸入值為 T#1d2h33m44s55ms (1日2小時33分44秒55毫秒) 與 T#2ms (2毫秒) 的情況下



- 至 (s1)、(s2) 的輸入值為 TIME 型的資料值。
- 即使運算結果中發生了下溢/上溢，也不會變為運算出錯。按下述方式輸出至 (d) 中。此外，SUB_TIME_E 的情況下，從輸出變數 ENO 輸出 TRUE。

例

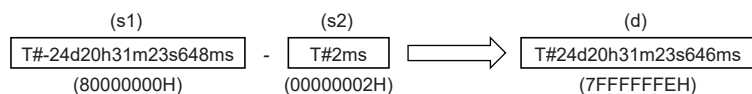
上溢



由於最高位的位元變為 1，因此變為負的時間。

例

下溢



由於最高位的位元變為 0，因此變為正的時間。

■ 運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

沒有運算出錯。

40.3 乘法運算

MUL_TIME(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCPU (常規) RnSFCPU (安全)

輸出輸入值 (TIME型) 的積 ((s1) × (s2))。

梯形圖、FBD/LD		ST
[無EN/ENO]		[無EN/ENO] d:=MUL_TIME(s1, s2); [帶EN/ENO] d:=MUL_TIME_E(EN, ENO, s1, s2);
[帶EN/ENO]		

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)	輸入	輸入變數	TIME
s2 (IN2)	輸入	輸入變數	ANY_NUM*1
ENO	輸出狀態 (TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	TIME

*1 安全程式中請勿指定INT型/REAL型/LREAL型資料。指定了INT型/REAL型/LREAL型資料的情況下，將變為出錯狀態。

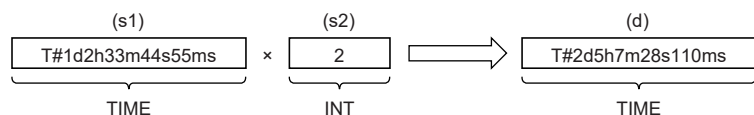
功能

■運算處理

- 進行(s1)、(s2)中輸入的TIME型資料的乘法運算((s1)×(s2))後，將運算結果從(d)以TIME型進行輸出。

例

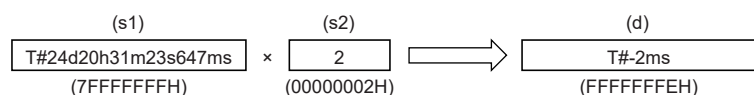
至(s1)、(s2)的輸入值為T#1d2h33m44s55ms(1日2小時33分44秒55毫秒)與2的情況下



- 至(s1)的輸入值為TIME型的資料值。
- 至(s2)的輸入值為INT型/DINT型/REAL型/LREAL型。
- 即使運算結果中發生了下溢/上溢，也不會變為運算出錯。按下述方式輸出至(d)中。此外，MUL_TIME_E的情況下，從輸出變數ENO輸出TRUE。(運算結果變為64位元資料，但是輸出在透過刪除高位32位元的時間類型資料下將被輸出。)

例

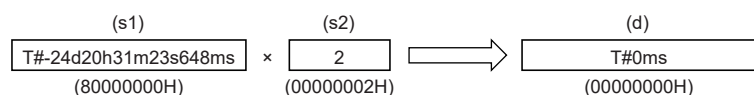
上溢



由於最高位的位元變為1，因此變為負的時間。

例

下溢



由於最高位的位元變為0，因此變為正的時間。

■運算結果

1. 無EN/ENO函數

執行運算處理。從(d)輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE(運算執行)	TRUE	運算輸出值
FALSE(運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從(d)中輸出的資料將變為不定值。在此情況下，應進行不使用從(d)中輸出的資料的程式處理。

出錯

- (s2)為LREAL型的情況下

出錯代碼(SD0)	內容
3402H	(s2)的內容為下述範圍外時。 $-2^{1024} < (s2) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s2) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308) (s2)中指定的資料為-0、非正規化數、非數、±∞時。
3405H	(s2)中設定的單精度實數資料超出-2147483648~2147483647的範圍時。

40.4 除法運算

DIV_TIME(_E)

RnCPU RnENCPU RnPCPU (過程) RnPCPU (二重化) RnPSFCPU (常規) RnPSFCPU (安全) RnSFCEPU (常規) RnSFCEPU (安全)

輸出輸入值 (TIME型) 的商 ((s1) ÷ (s2))。

梯形圖、FBD/LD	ST
<p>[無EN/ENO]</p>	<p>[無EN/ENO]</p> <pre>d:=DIV_TIME(s1, s2);</pre> <p>[帶EN/ENO]</p> <pre>d:=DIV_TIME_E(EN, ENO, s1, s2);</pre>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (IN1)	輸入	輸入變數	TIME
s2 (IN2)	輸入	輸入變數	ANY_NUM*1
ENO	輸出狀態 (TRUE: 正常、FALSE: 異常)	輸出變數	BOOL
d	輸出	輸出變數	TIME

*1 安全程式中請勿指定INT型/REAL型/LREAL型資料。指定了INT型/REAL型/LREAL型資料的情況下，將變為出錯狀態。

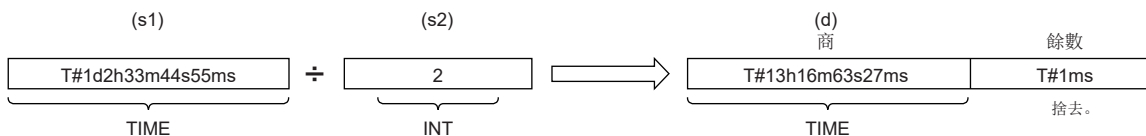
功能

■運算處理

- 進行 (s1)、(s2) 中輸入的TIME型資料的除法運算 ((s1) ÷ (s2)) 後，將運算結果從 (d) 以TIME型進行輸出。捨去餘數。

例

至 (s1)、(s2) 的輸入值為 T#1d2h33m44s55ms (1日2小時33分44秒55毫秒) 與 2 的情況下



- 至 (s1) 的輸入值為TIME型的資料值。
- 至 (s2) 的輸入值為INT型/DINT型/REAL型/LREAL型。(但是，至 (s2) 的輸入值為0以外。)

■運算結果

1. 無EN/ENO函數

執行運算處理。從 (d) 輸出運算輸出值。

2. 帶EN/ENO函數

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE*1	不定值

*1 在ENO輸出了FALSE的情況下，從 (d) 中輸出的資料將變為不定值。在此情況下，應進行不使用從 (d) 中輸出的資料的程式處理。

出錯

出錯代碼 (SD0)	內容
3400H	至 (s2) 的輸入值為0時。(除0)
• (s2) 為LREAL型的情況下	
出錯代碼 (SD0)	內容
3402H	(s2) 的內容為下述範圍外時。 $-2^{1024} < (s2) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s2) < 2^{1024}$ (E-1. 7976931348623157+308~E-2. 2250738585072014-308、0、E2. 2250738585072014-308~E1. 7976931348623157+308)
3405H	(s2) 的內容超出-2147483648~2147483647的範圍時。

第8部分 通用FB

本部分由下述章節構成。

41 雙穩態FB

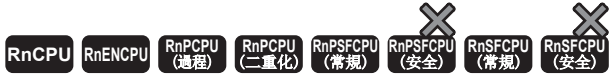
42 邊緣檢測FB

43 計數器/定時器FB

41 雙穩態FB

41.1 雙穩態FB(設定優先)

SR(_E)



判別2個輸入值，輸出1 (TRUE) 或0 (FALSE)。

梯形圖、FBD/LD		ST
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] 例項名 (S1:=s1, R:=s2, Q1:=d); [帶EN/ENO] 例項名 (EN:=en, ENO:=eno, S1:=s1, R:=s2, Q1:=d);

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (S1)	設定指令	輸入變數	BOOL
s2 (R)	重設指令	輸入變數	BOOL
ENO	輸出狀態 (TRUE: 正常、FALSE: 異常或停止)	輸出變數	BOOL
d (Q1)	輸出	輸出變數	BOOL

功能

■ 運算處理

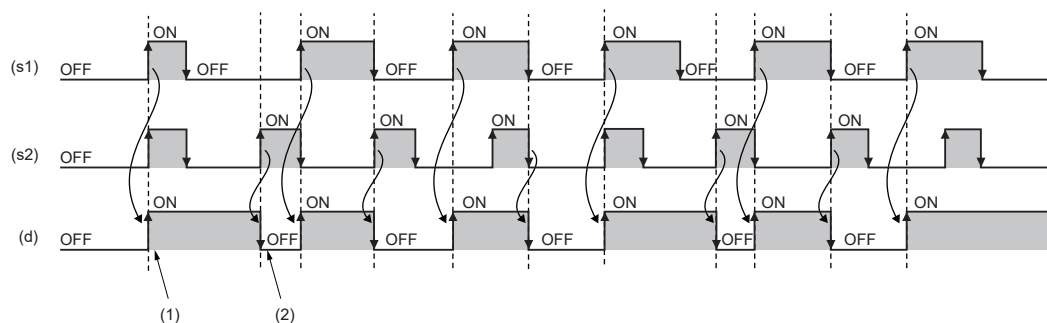
- (s1) 變為ON時，對 (d) 進行SET。(s1) 為OFF時，如果將 (s2) 設為ON，可以對 (d) 進行RESET。
- (s1) 為ON時，即使將 (s2) 設為ON，(d) 也不被RESET。

■ 運算結果

1. 無EN/ENO的FB

執行運算處理。從 (d) 輸出運算輸出值。

- 時序圖



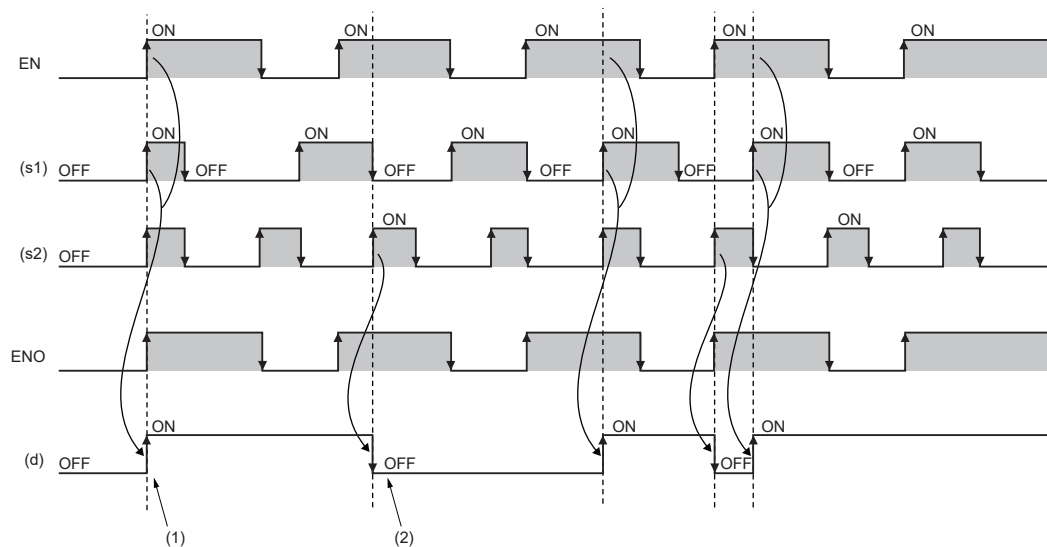
- (1) 透過 (s1) 的OFF→ON將 (d) 設為ON。
- (2) 透過 (s1)=OFF且 (s2) 的OFF→ON將 (d) 設為OFF。

2. 帶EN/ENO的FB

執行條件與運算結果如下所示。

執行條件		運算結果
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE	上次輸出值

- 時序圖



- (1) 透過EN=ON且 (s1) 的OFF→ON將 (d) 設為ON。
- (2) 透過EN=ON且 (s1)=OFF且 (s2) 的OFF→ON將 (d) 設為OFF。

出錯

沒有運算出錯。

41.2 雙穩態FB(重設優先)

RS(_E)



判別2個輸入值，輸出1 (TRUE) 或0 (FALSE)。

梯形圖、FBD/LD		ST
[無EN/ENO]		[無EN/ENO] 例項名 (S:=s1, R1:=s2, Q1:=d); [帶EN/ENO] 例項名 (EN:=en, ENO:=eno, S:=s1, R1:=s2, Q1:=d);
[帶EN/ENO]		

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (S)	設定指令	輸入變數	BOOL
s2 (R1)	重設指令	輸入變數	BOOL
ENO	輸出狀態 (TRUE: 正常、FALSE: 異常或停止)	輸出變數	BOOL
d (Q1)	輸出	輸出變數	BOOL

功能

■ 運算處理

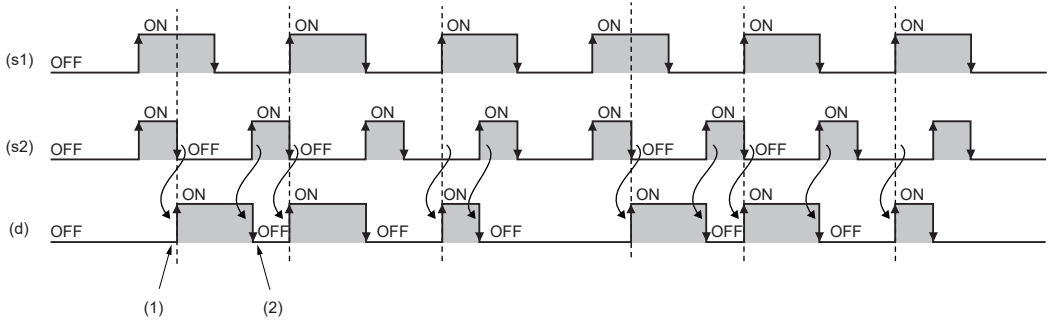
- (s1) 變為ON時，對 (d) 進行SET。如果將 (s2) 設為ON，則對 (d) 進行RESET。
- (s2) 為ON時，即使將 (s1) 設為ON，(d) 也不被SET。

■ 運算結果

1. 無EN/ENO的FB

執行運算處理。從 (d) 輸出運算輸出值。

- 時序圖



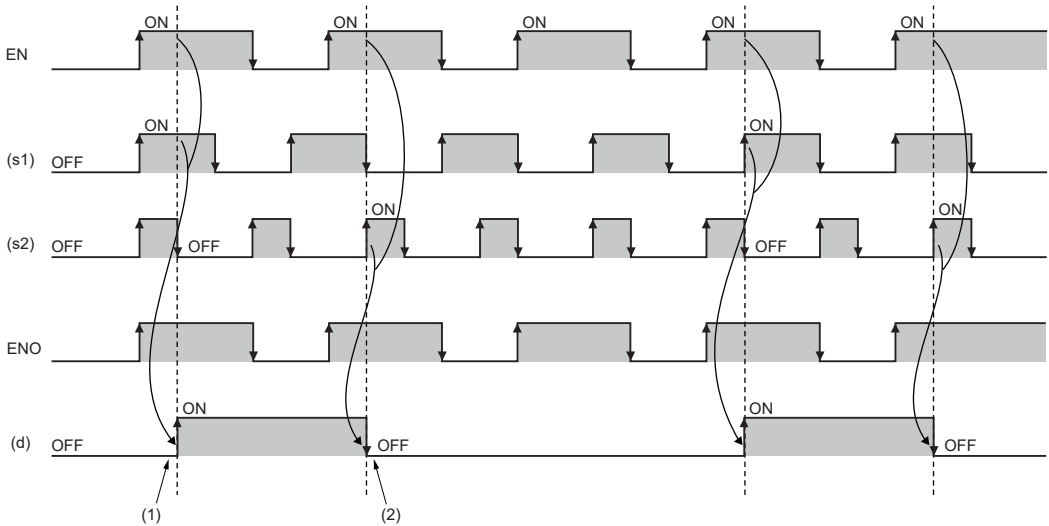
- (1) 透過 (s1)=ON 且 (s2) 的 ON→OFF 將 (d) 設為 ON。
- (2) 透過 (s2) 的 OFF→ON 將 (d) 設為 OFF。

2. 帶EN/ENO的FB

執行條件與運算結果如下所示。

執行條件		運算結果
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE	上次輸出值

- 時序圖



- (1) 透過 EN=ON 且 (s1)=ON 且 (s2) 的 ON→OFF 將 (d) 設為 ON。
- (2) 透過 EN=ON 且 (s2) 的 OFF→ON 將 (d) 設為 OFF。

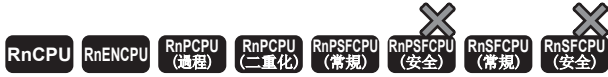
出錯

沒有運算出錯。

42 邊緣檢測FB

42.1 上升沿邊緣檢測

R_TRIG(_E)



檢測信號的上升沿後輸出脈衝信號。

梯形圖、FBD/LD		ST
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] 例項名 (CLK:=s, Q:=d); [帶EN/ENO] 例項名 (EN:= en, ENO:=eno, CLK:=s, Q:=d);

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (CLK)	上升沿邊緣檢測輸入	輸入變數	BOOL
ENO	輸出狀態(TRUE: 正常、FALSE: 異常或停止)	輸出變數	BOOL
d (Q)	輸出	輸出變數	BOOL

功能

■ 運算處理

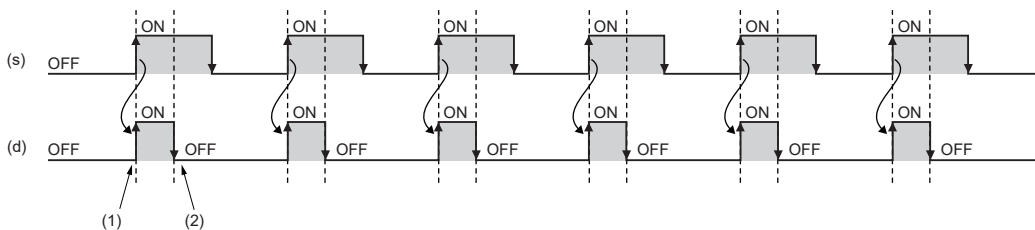
(s) 變為ON時，僅1個掃描將 (d) 設為ON。

■ 運算結果

1. 無EN/ENO的FB

執行運算處理。從 (d) 輸出運算輸出值。

• 時序圖



(1) 透過 (s) 的上升沿將 (d) 設為ON。

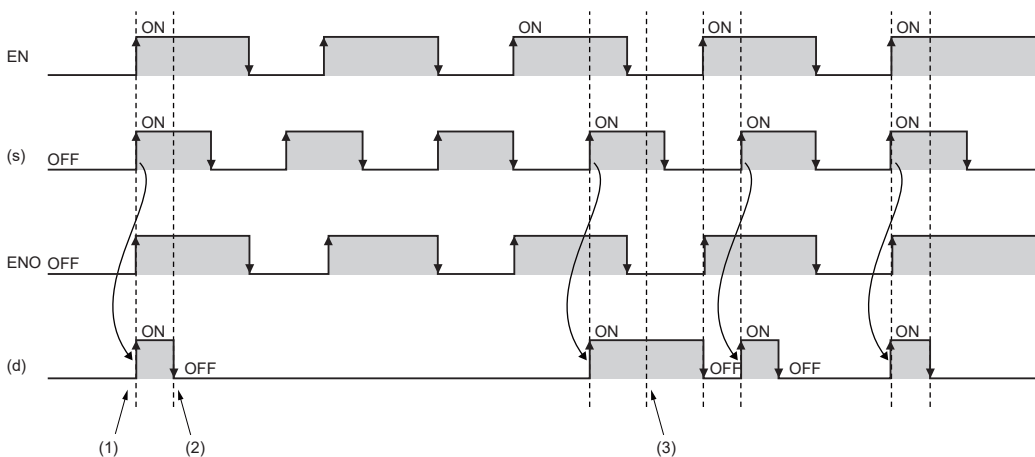
(2) 透過下個掃描將 (d) 設為OFF。

2. 帶EN/ENO的FB

執行條件與運算結果如下所示。

執行條件		運算結果
EN		(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE	上次輸出值

• 時序圖



(1) 透過EN=ON且(s)的上升沿將(d)設為ON。

(2) 透過下個掃描將(d)設為OFF。

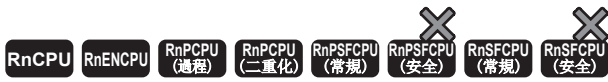
(3) EN=OFF的情況下，(d)保持上次掃描的輸出。

出錯

沒有運算出錯。

42.2 下降沿邊緣檢測

F_TRIG(_E)



檢測信號的下降沿後輸出脈衝信號。

梯形圖、FBD/LD		ST
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] 例項名 (CLK:=s, Q:=d); [帶EN/ENO] 例項名 (EN:=en, ENO:=eno, CLK:=s, Q:=d);

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (CLK)	下降沿邊緣檢測輸入	輸入變數	BOOL
ENO	輸出狀態 (TRUE: 正常、FALSE: 異常或停止)	輸出變數	BOOL
d (Q)	輸出	輸出變數	BOOL

功能

■ 運算處理

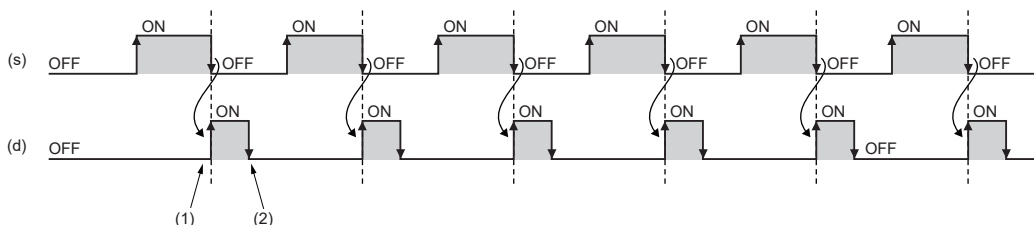
(s) 變為OFF時，僅1個掃描將 (d) 設為ON。

■ 運算結果

1. 無EN/ENO的FB

執行運算處理。從 (d) 輸出運算輸出值。

• 時序圖



(1) 透過 (s) 的下降沿將 (d) 設為ON。

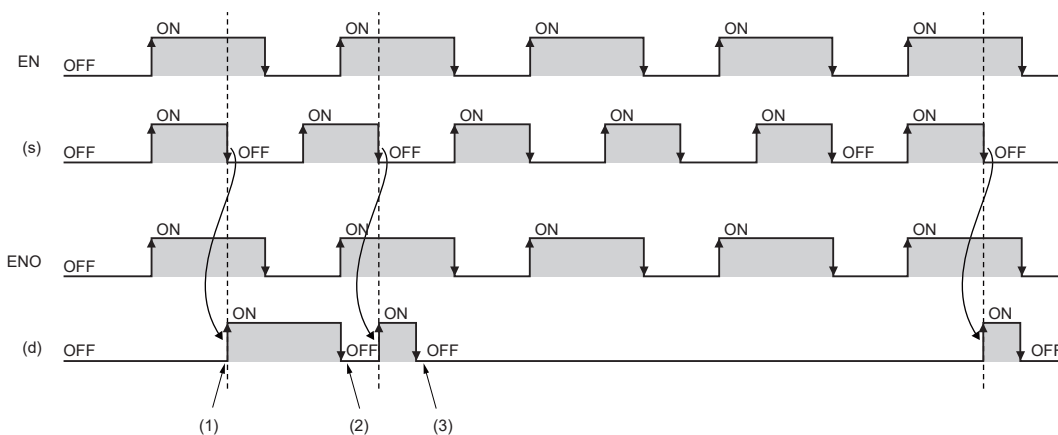
(2) 透過下個掃描將 (d) 設為OFF。

2. 帶EN/ENO的FB

執行條件與運算結果如下所示。

執行條件		運算結果
EN	ENO	(d)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE	上次輸出值

• 時序圖



(1) 透過EN=ON且(s)的下降沿將 (d) 設為ON。

(2) 透過下個掃描將 (d) 設為OFF。

(3) EN=OFF的情況下，(d) 保持上次掃描的輸出。

出錯

沒有運算出錯。

43 計數器/定時器FB

43.1 升值計數器

CTU(_E)



對信號的上升沿次數進行遞增計數。

梯形圖、FBD/LD		ST
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] 例項名 (CU:=s1, R:=s2, PV:=n, Q:=d1, CV:=d2) ; [帶EN/ENO] 例項名 (EN:=en, ENO:=eno, CU:=s1, R:=s2, PV:=n, Q:=d1, CV:=d2) ;

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (CU)	計數信號輸入	輸入變數	BOOL
s2 (R)	計數值重設	輸入變數	BOOL
n (PV)	計數最大值	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常或停止)	輸出變數	BOOL
d1 (Q)	計數完成	輸出變數	BOOL
d2 (CV)	計數值	輸出變數	INT

功能

■運算處理

1. 遞增計數

- 如果(s1)變為OFF→ON, 對(d2)進行加法計數(+1)。
- 如果(d2)到達計數器的(n), 則(d1)變為ON, 加法計數停止。
- (n)為設定計數器的最大值。如果將(s2)設為ON, 則(d1)變為OFF, (d2)被設定為0。

2. 計數最大值

(n)的有效設定範圍為0~32767。

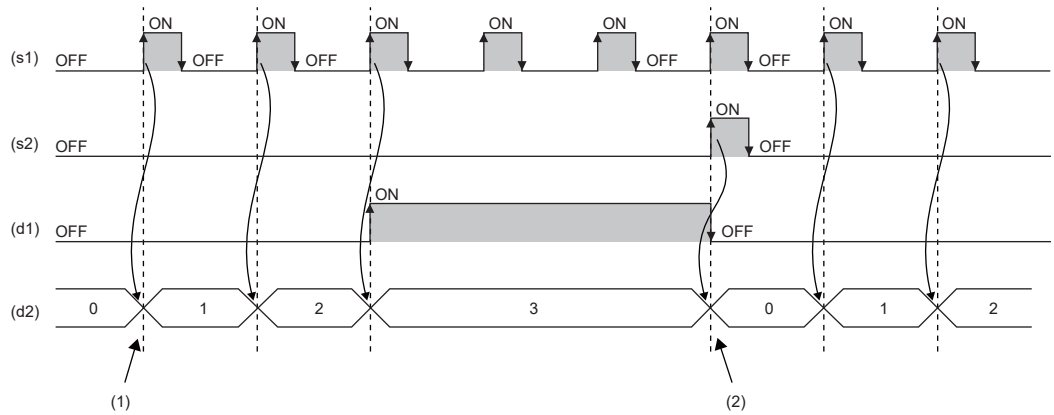
■ 運算結果

1. 無EN/ENO的FB

執行運算處理。從(d1)、(d2)中輸出運算輸出值。

• 時序圖

n=3的情況下



- (1) 透過(s1)的OFF→ON遞增計數(d2)。
- (2) 透過(s2)的OFF→ON初始化(d2)。

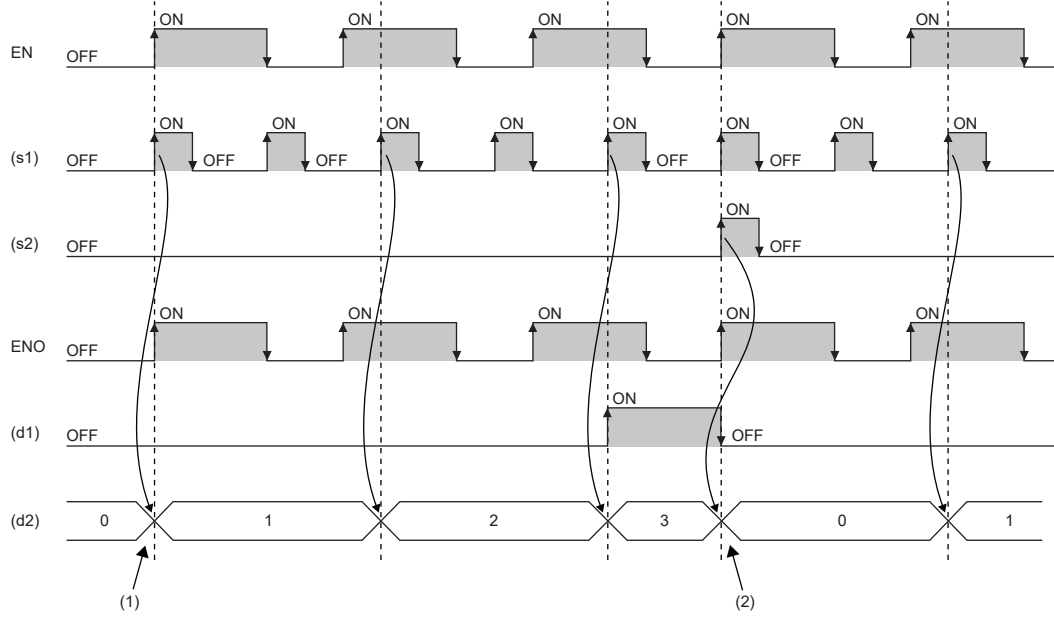
2. 帶EN/ENO的FB

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d1)、(d2)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE	上次輸出值

• 時序圖

n=3的情況下



- (1) 透過EN=ON且(s1)的OFF→ON遞增計數(d2)。
- (2) 透過EN=ON且(s2)的OFF→ON初始化(d2)。

出錯

沒有運算出錯。

43.2 降值計數器

CTD(_E)



對信號的上升沿次數進行遞減計數。

梯形圖、FBD/LD		ST
[無EN/ENO]		[無EN/ENO] 例項名 (CD:=s1, LD:=s2, PV:=n, Q:=d1, CV:=d2) ; [帶EN/ENO] 例項名 (EN:=en, ENO:=eno, CD:=s1, LD:=s2, PV:=n, Q:=d1, CV:=d2) ;
[帶EN/ENO]		

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (CD)	計數信號輸入	輸入變數	BOOL
s2 (LD)	計數值設定	輸入變數	BOOL
n (PV)	計數開始值	輸入變數	INT
ENO	輸出狀態(TRUE: 正常、FALSE: 異常或停止)	輸出變數	BOOL
d1 (Q)	計數完成	輸出變數	BOOL
d2 (CV)	計數值	輸出變數	INT

功能

■運算處理

1. 遞減計數

- 如果(s1)變為OFF→ON, 對(d2)進行減法計數(-1)。
- (d2)為0的情況下, (d1)變為ON, 減法計數停止。
- (n)設定為計數開始值。如果將(s2)設為ON, (d1)變為OFF, (n)被設定為(d2)。

2. 計數開始值

(n)的有效設定範圍為0~32767。

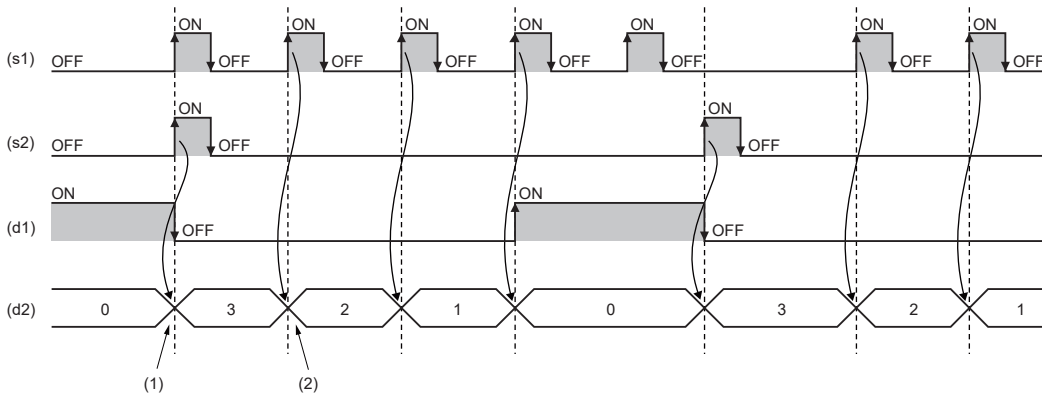
■ 運算結果

1. 無EN/ENO的FB

執行運算處理。從(d1)、(d2)中輸出運算輸出值。

• 時序圖

n=3的情況下



- (1) 透過(s2)的OFF→ON初始化(d2)。
- (2) 透過(s1)的OFF→ON遞減計數(d2)。

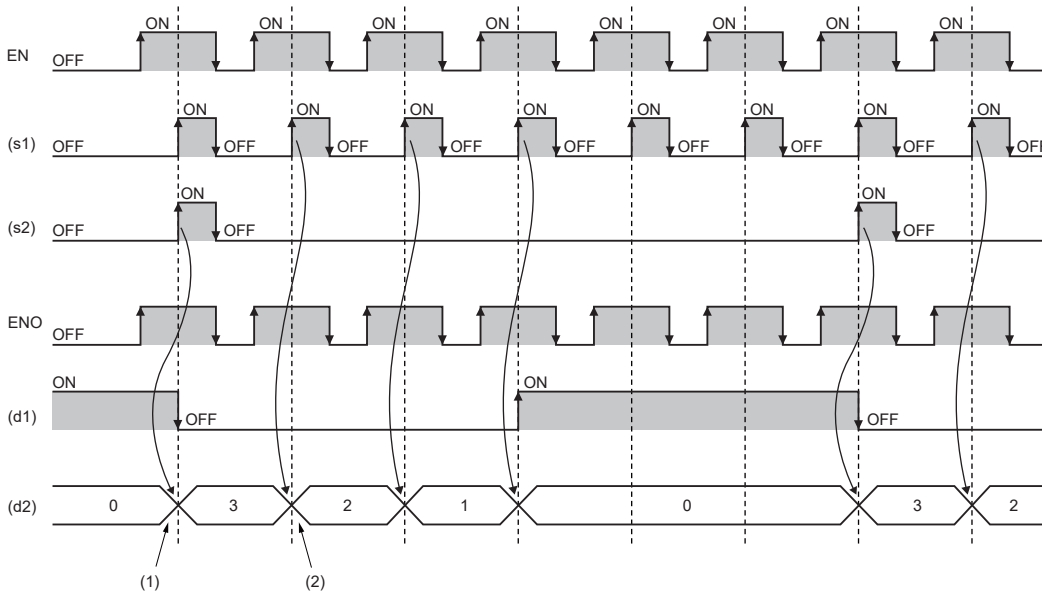
2. 帶EN/ENO的FB

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d1)、(d2)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE	上次輸出值

• 時序圖

n=3的情況下



- (1) 透過EN=ON且(s2)的OFF→ON初始化(d2)。
- (2) 透過EN=ON且(s1)的OFF→ON遞減計數(d2)。

出錯

沒有運算出錯。

43.3 升值降值計數器

CTUD(_E)



對信號的上升沿次數進行遞增/遞減計數。

梯形圖、FBD/LD		ST
<p>[無EN/ENO]</p>	<p>[帶EN/ENO]</p>	<p>[無EN/ENO] 例項名 (CU:=s1, CD:=s2, R:=s3, LD:=s4, PV:=n, QU:=d1, QD:=d2, CV:=d3) ;</p> <p>[帶EN/ENO] 例項名 (EN:=en, ENO:=eno, CU:=s1, CD:=s2, R:=s3, LD:=s4, PV:=n, QU:=d1, QD:=d2, CV:=d3) ;</p>

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s1 (CU)	遞增計數信號輸入	輸入變數	BOOL
s2 (CD)	遞減計數信號輸入	輸入變數	BOOL
s3 (R)	計數值重設	輸入變數	BOOL
s4 (LD)	計數值設定	輸入變數	BOOL
n (PV)	計數最大值/開始值	輸入變數	INT
ENO	輸出狀態 (TRUE: 正常、FALSE: 異常或停止)	輸出變數	BOOL
d1 (QU)	遞增計數完成	輸出變數	BOOL
d2 (QD)	遞減計數完成	輸出變數	BOOL
d3 (CV)	當前計數值	輸出變數	INT

功能

■ 運算處理

(n)中，在加法計數的情況下設定計數最大值，減法計數的情況下設定計數開始值。(n)的有效範圍為0~32767。

1. 遞增計數

- 如果(s1變為OFF→ON，對(d3)進行加法計數(+1)。
- 如果(d3)到達(n)，則(d1)變為ON，加法計數停止。
- 如果將(s3)設為ON，則(d1)變為OFF，(d3)被設定為0。

2. 遞減計數

- 如果(s2變為OFF→ON，對(d3)進行減法計數(-1)。
- (d3)為0的情況下，(d2)變為ON，減法計數停止。
- 如果將(s4)設為ON，則(d2)變為OFF，(n)被設定為(d3)。

3. 其他

- 如果(s1)、(s2)同時變為OFF→ON，(s1)優先對(d3)進行加法計數(+1)。
- 如果將(s3)、(s4)同時設為ON，s3優先將(d3)設定為0。

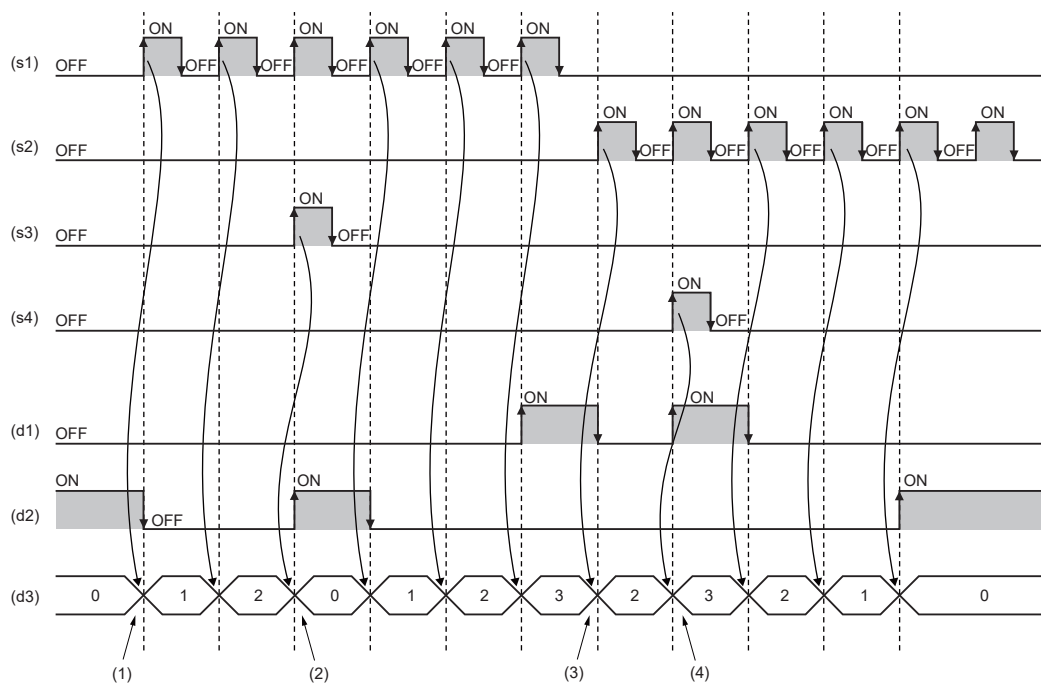
■ 運算結果

1. 無EN/ENO的FB

執行運算處理。從(d1)、(d2)、(d3)中輸出運算輸出值。

- 時序圖

n=3的情況下



- (1) 透過(s1)的OFF→ON遞增計數(d3)。
- (2) 透過(s3)的OFF→ON初始化(d3)。
- (3) 透過(s2)的OFF→ON遞減計數(d3)。
- (4) 透過(s4)的OFF→ON初始化(d3)。

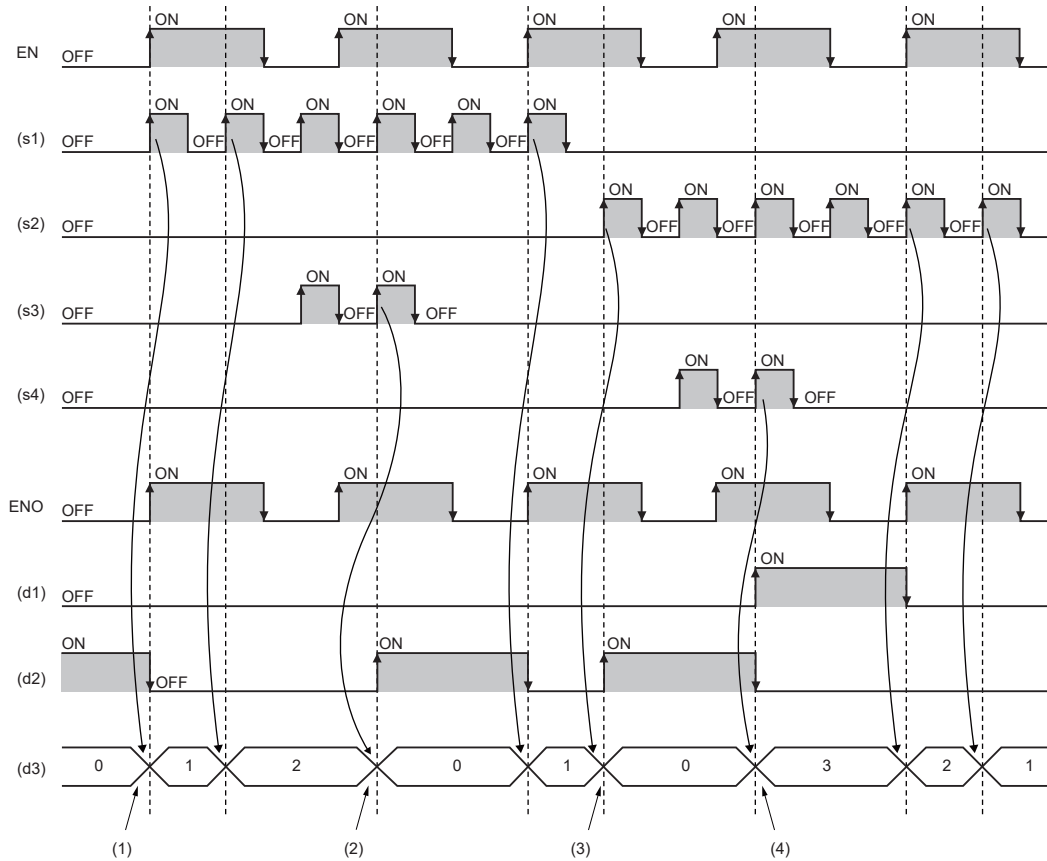
2. 帶EN/ENO的FB

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d1)、(d2)、(d3)
TRUE (運算執行)	TRUE	運算輸出值
FALSE (運算停止)	FALSE	上次輸出值

• 時序圖

n=3的情況下



- (1) 透過EN=ON且(s1)的OFF→ON遞增計數(d3)。
- (2) 透過EN=ON且(s3)的OFF→ON初始化(d3)。
- (3) 透過EN=ON且(s2)的OFF→ON遞減計數(d3)。
- (4) 透過EN=ON且(s4)的OFF→ON初始化(d3)。

出錯

沒有運算出錯。

43.4 計數器的FB

COUNTER_FB_M



執行條件成立時，執行遞增計數。

梯形圖、FBD/LD	ST
	例項名 (Coil:=s1, Preset:=s2, ValueIn:=s3, ValueOut:=d1, Status:=d2);

43

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
s1(Coil)	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s2(Preset)	計數器設定值	輸入變數	INT
s3(ValueIn)	計數器初始值	輸入變數	INT
d1(ValueOut)	計數器當前值	輸出變數	ANY16
d2(Status)	輸出	輸出變數	BOOL

功能

■ 運算處理

- 檢測(s1)的上升沿(OFF→ON)後進行計數。(s1)為ON不變的狀況下不進行計數。計數從(s3)的值開始，如果變為(s2)的值，則(d2)變為ON。當前的計數值被儲存到(d1)中。
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是，指定了負值的情況下初始值為0。
- 希望重設計數器當前值(d1)的情況下，應直接重設FB的(s1)。

例

標籤名: COUNTER_FB_M_1的情況

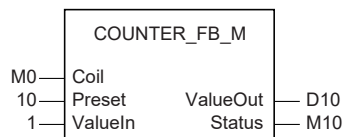
[梯形圖程式]



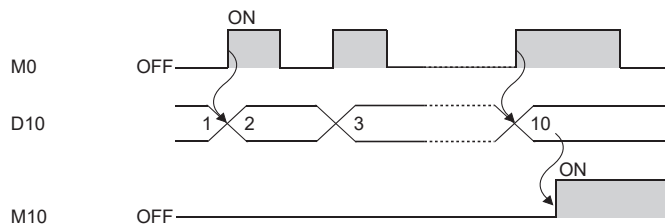
[ST程式]

```
RST(M0, COUNTER_FB_M_1.Coil)
```

[梯形圖示例]



[時序圖]



出錯

沒有運算出錯。

43.5 脈衝定時器

TP (E)



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統將不進行動作。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

在指定時間期間將信號設為ON。

梯形圖、FBD/LD		ST
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] 例項名 (IN:=s, PT:=n, Q:=d1, ET:=d2); [帶EN/ENO] 例項名 (EN:=en, ENO:=eno, IN:=s, PT:=n, Q:=d1, ET:=d2);

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	輸出開始	輸入變數	BOOL
n (PT)	輸出時間設定值	輸入變數	TIME
ENO	輸出狀態(TRUE: 正常、FALSE: 異常或停止)	輸出變數	BOOL
d1 (Q)	輸出	輸出變數	BOOL
d2 (ET)	經過時間	輸出變數	TIME

功能

■運算處理

1. 輸出

- 如果(s)變為ON，則(n)中設定的時間內將(d1)設為ON。(d2)設定(d1)變為ON後的經過時間。
- 經過時間的計數使用超長定時器。

2. 輸出結束

- 如果經過時間達到了設定時間，則將(d1)設為OFF。
- 在(d1)變為OFF後且(s)為OFF的情況下重設經過時間。
- 即使(d1)為ON時(s)變為OFF，(d1)也不被設為OFF。

3. 輸出時間的設定

(n)的有效範圍為T#1ms~T#2147483ms。但是，由於更改工程工具的定時器時限設定，有效設定範圍如下所示。

最小值	最大值
與定時器時限的超長定時器設定值[ms]相同。 但是，超長定時器設定值不滿1ms的情況下則為1ms。	為滿足下述條件的值。 但是，由於輸出時間設定值為時間型(32位元值)，最大值為包含在時間型範圍中的值。 • 輸出時間設定值[ms]≤2147483647[ms]×定時器時限的超長定時器設定值[ms] [例] • 定時器時限的超長定時器設定值為0.001ms的情況下：T#1ms~T#2147483ms • 定時器時限的超長定時器設定值為1000ms的情況下：T#1000ms~T#2147483000ms

對於(n)的設定值，(d1)變為OFF→ON(上升沿)時的值將被使用。(d1)為ON時更改了(n)的值的的情況下，更改的值在下次輸出開始時有效。

■ 運算結果

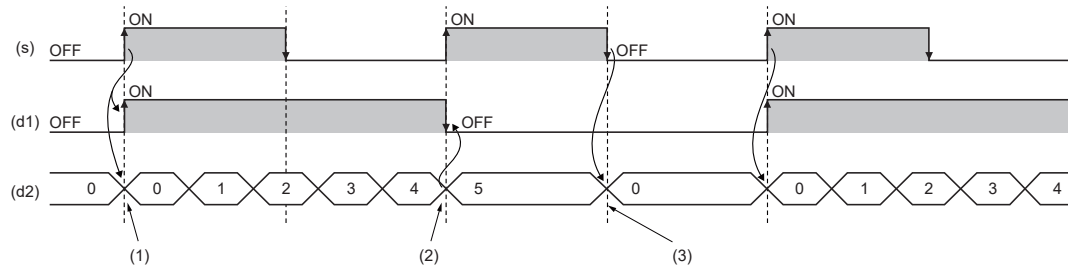
1. 無EN/ENO的FB

運算結果如下所示。

運算結果	(d1)、(d2)
無運算出錯	運算輸出值
有運算出錯	不定值

• 時序圖

n=T#5s (5秒) 的情況下



(1) 透過 (s) 的 OFF→ON 將 (d1) 設為 ON。透過 (s) 的 OFF→ON 開始 (d2) 的時間計測。

(2) 如果 (d2) 達到了 (n) 中指定的時間，則將 (d1) 設為 OFF。

(3) 透過 (s)=OFF 且 (d1)=OFF 初始化 (d2)。

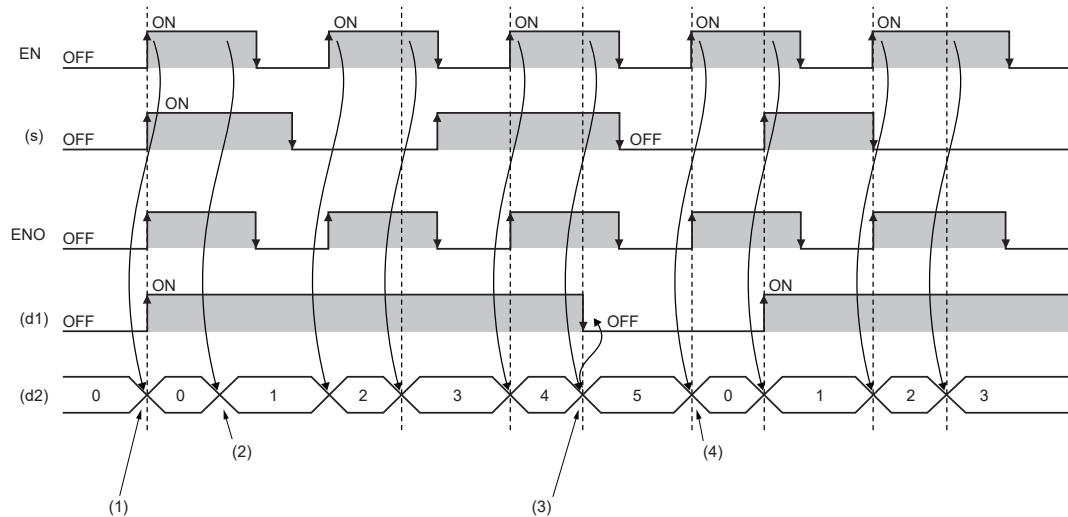
2. 帶EN/ENO的FB

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d1)、(d2)
TRUE (運算執行)	TRUE (無運算出錯)	運算輸出值
	FALSE (有運算出錯)	不定值
FALSE (運算停止)	FALSE	上次輸出值

• 時序圖

n=T#5s (5秒) 的情況下



(1) 透過 EN=ON 且 (s) 的 OFF→ON 將 (d1) 設為 ON。透過 EN=ON 且 (s) 的 OFF→ON 開始 (d2) 的時間計測。

(2) 計測開始後，透過 EN=ON 計數計測時間。

(3) 如果 (d2) 達到了 (n) 中指定的時間，則將 (d1) 設為 OFF。

(4) 透過 EN=ON 且 (s)=OFF 且 (d1)=OFF 初始化 (d2)。

出錯

出錯代碼 (SD0)	內容
3401H	輸出時間設定值超過有效範圍時。

43.6 ON延遲定時器

TON(_E)



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統將不進行動作。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

在指定時間後將信號設為ON。

梯形圖、FBD/LD		ST
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] 例項名 (IN:=s, PT:=n, Q:=d1, ET:=d2); [帶EN/ENO] 例項名 (EN:=en, ENO:=eno, IN:=s, PT:=n, Q:=d1, ET:=d2);

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	時間計測	輸入變數	BOOL
n (PT)	延遲時間設定值	輸入變數	TIME
ENO	輸出狀態(TRUE: 正常、FALSE: 異常或停止)	輸出變數	BOOL
d1 (Q)	輸出	輸出變數	BOOL
d2 (ET)	經過時間	輸出變數	TIME

功能

■運算處理

1. 輸出

- 如果(s)變為ON，則(n)中設定的時間後將(d1)設為ON。(d2)設定(d1)變為ON後的延遲經過時間。
- 如果(s)變為OFF則將(d1)設為OFF並重設延遲經過時間。
- 經過時間的計數使用超長定時器。

2. 延遲時間的設定

(n)的有效範圍為T#1ms~T#2147483ms。但是，由於更改工程工具的定時器時限設定，有效設定範圍如下所示。

最小值	最大值
與定時器時限的超長定時器設定值[ms]相同。 但是，超長定時器設定值不滿1ms的情況下則為1ms。	為滿足下述條件的值。 但是，由於延遲時間設定值為時間型(32位元值)，最大值為包含在時間型範圍中的值。 • 延遲時間設定值[ms] ≤ 2147483647[ms] × 定時器時限的超長定時器設定值[ms] [例] • 定時器時限的超長定時器設定值為0.001ms的情況下：T#1ms~T#2147483ms • 定時器時限的超長定時器設定值為1000ms的情況下：T#1000ms~T#2147483000ms

(n)的設定值使用(d)變為了OFF→ON(上升沿)時的值。(s)為ON時更改了(n)的值的的情況下，更改的值在(s)的下次上升沿時有效。

■ 運算結果

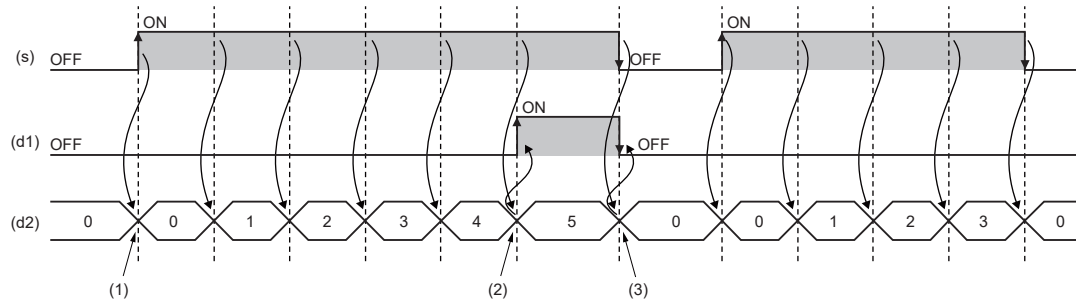
1. 無EN/ENO的FB

運算結果如下所示。

運算結果	(d1)、(d2)
無運算出錯	運算輸出值
有運算出錯	不定值

• 時序圖

n=T#5s (5秒) 的情況下



- (1) 透過 (s) 的 OFF→ON 開始 (d2) 的時間計測。
- (2) 如果 (d2) 達到了 (n) 中指定的時間，則將 (d1) 設為 ON。
- (3) 透過 (s) 的 ON→OFF 且 (d1) 的 ON→OFF，初始化 (d2)。

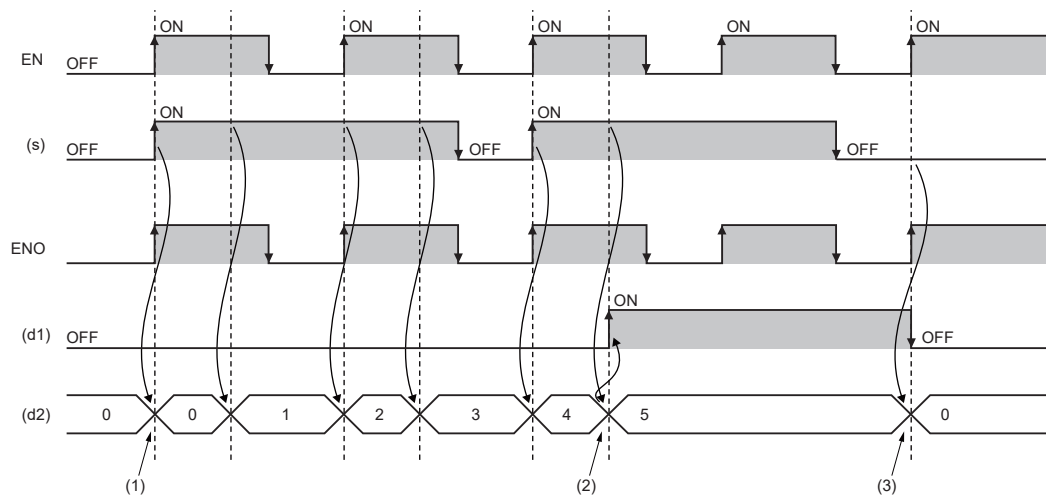
2. 帶EN/ENO的FB

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d1)、(d2)
TRUE (運算執行)	TRUE (無運算出錯)	運算輸出值
	FALSE (有運算出錯)	上次輸出值
FALSE (運算停止)	FALSE	上次輸出值

• 時序圖

n=T#5s (5秒) 的情況下



- (1) 透過 EN=ON 且 (s) 的 OFF→ON 開始 (d2) 的時間計測。
- (2) 如果 (d2) 達到了 (n) 中指定的時間，則將 (d1) 設為 ON。
- (3) 透過 EN=ON 且 (s) 的 ON→OFF 且 (d1) 的 ON→OFF，初始化 (d2)。

出錯

出錯代碼 (SD0)	內容
3401H	輸出時間設定值超過有效範圍時。

43.7 OFF延遲定時器

TOF (E)



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用的情況下，備份模式時的待機系統將不進行動作。(MELSEC iQ-R CPU模組用戶手冊(應用篇))

在指定時間後將信號設為OFF。

梯形圖、FBD/LD		ST
[無EN/ENO]	[帶EN/ENO]	[無EN/ENO] 例項名 (IN:=s, PT:=n, Q:=d1, ET:=d2); [帶EN/ENO] 例項名 (EN:=en, ENO:=eno, IN:=s, PT:=n, Q:=d1, ET:=d2);

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
EN	執行條件(TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s (IN)	時間計測	輸入變數	BOOL
n (PT)	延遲時間設定值	輸入變數	TIME
ENO	輸出狀態(TRUE: 正常、FALSE: 異常或停止)	輸出變數	BOOL
d1 (Q)	輸出	輸出變數	BOOL
d2 (ET)	經過時間	輸出變數	TIME

功能

■運算處理

1. 輸出

- 如果(s)變為ON，則將(d1)設為ON。
- 如果(s)變為ON→OFF，則在(n)中設定的時間後(d1)將設為OFF。(d2)設定(d1)變為OFF之前的經過時間。
- 經過時間的計數使用超長定時器。

2. 延遲時間的設定

(n)的有效範圍為T#1ms~T#2147483ms。但是，由於更改工程工具的定時器時限設定，有效設定範圍如下所示。

最小值	最大值
與定時器時限的超長定時器設定值[ms]相同。 但是，超長定時器設定值不滿1ms的情況下則為1ms。	為滿足下述條件的值。 但是，由於延遲時間設定值為時間型(32位元值)，最大值為包含在時間型範圍中的值。 • 延遲時間設定值[ms]≤2147483647[ms]×定時器時限的超長定時器設定值[ms] [例] • 定時器時限的超長定時器設定值為0.001ms的情況下：T#1ms~T#2147483ms • 定時器時限的超長定時器設定值為1000ms的情況下：T#1000ms~T#2147483000ms

(n)的設定值使用(s)由ON→OFF(下降沿)時的值。(s)為OFF時更改了(n)的值的的情況下，更改的值在(s)的下次下降沿時有效。

■ 運算結果

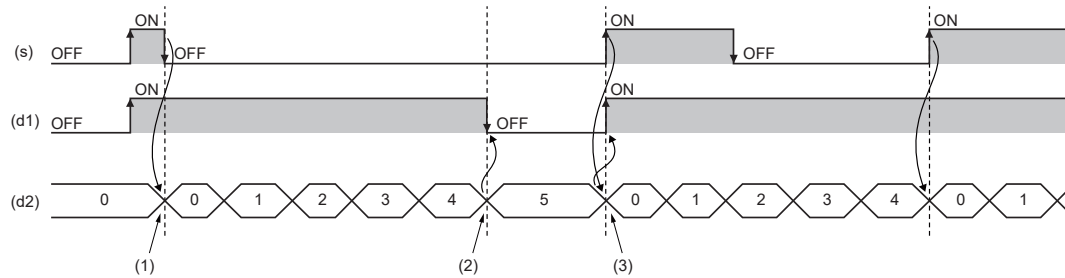
1. 無EN/ENO的FB

運算結果如下所示。

運算結果	(d1)、(d2)
無運算出錯	運算輸出值
有運算出錯	不定值

• 時序圖

n=T#5s (5秒) 的情況下



- (1) 透過 (s) 的 ON→OFF 開始 (d2) 的時間計測。
- (2) 如果 (d2) 達到了 (n) 中指定的時間，則將 (d1) 設為 ON。
- (3) 透過 (s) 的 OFF→ON 初始化 (d2)。

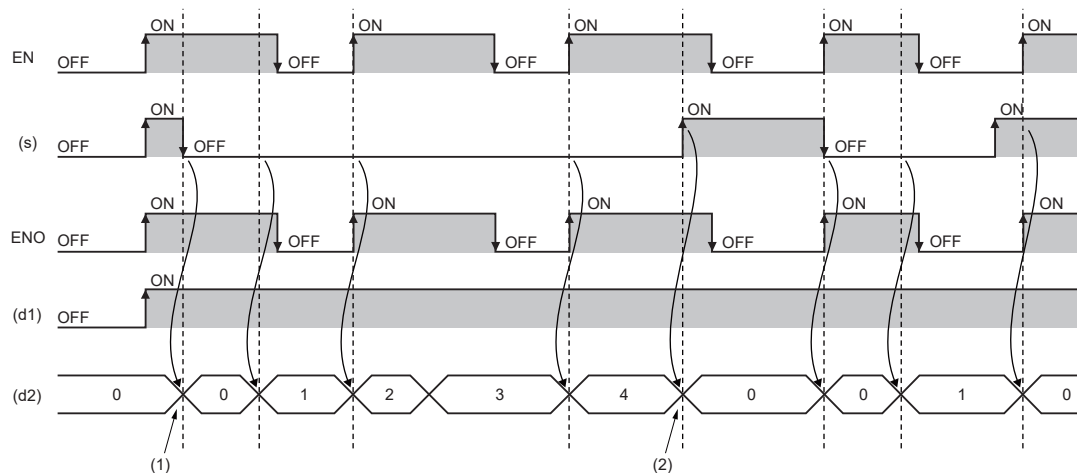
2. 帶EN/ENO的FB

執行條件與運算結果如下所示。

執行條件	運算結果	
EN	ENO	(d)
TRUE (運算執行)	TRUE (無運算出錯)	運算輸出值
	FALSE (有運算出錯)	上次輸出值
FALSE (運算停止)	FALSE	上次輸出值

• 時序圖

n=T#5s (5秒) 的情況下



- (1) 透過 EN=ON 且 (s) 的 ON→OFF 開始 (d2) 的時間計測。
- (2) 透過 EN=ON 且 (s) 的 OFF→ON 初始化 (d2)。

出錯

出錯代碼 (SD0)	內容
3401H	輸出時間設定值超過有效範圍時。

43.8 定時器FB

TIMER_□_M



• 在過程CPU(二重化模式)及SIL2過程CPU的雙系統執行程式中使用時，在系統切換時的動作中有限制。(參閱MELSEC iQ-R CPU模組用戶手冊(應用篇))
執行條件成立後，至設定的時間為止執行定時器計數。

梯形圖、FBD/LD	ST
<p>(□輸入TIMER_10_FB_M、TIMER_100_FB_M、TIMER_HIGH_FB_M、TIMER_LOW_FB_M、TIMER_CONT_FB_M、TIMER_CONTHFB_M。)</p>	例項名 (Coil:=s1, Preset:=s2, ValueIn:=s3, ValueOut:=d1, Status:=d2);

設置數據

■內容、類型、資料類型

引數	內容	類型	資料類型
s1 (Coil)	執行條件 (TRUE: 執行、FALSE: 停止)	輸入變數	BOOL
s2 (Preset)	定時器設定值	輸入變數	INT
s3 (ValueIn)	定時器初始值	輸入變數	INT
d1 (ValueOut)	定時器當前值	輸出變數	ANY16
d2 (Status)	輸出	輸出變數	BOOL

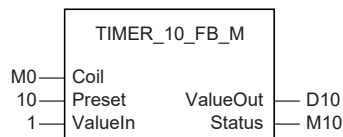
功能

■TIMER_10_FB_M

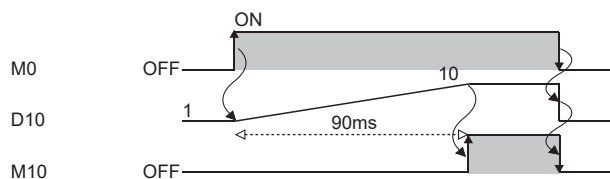
- (s1)的執行條件變為ON時，開始當前值的計測。從(s3)×10ms開始計測，直到(s2)×10ms為止到達計測值時(d2)變為ON。當前計測值被輸出到(d1)中。
- 如果(s1)的執行條件變為OFF，則當前值變為(s3)的值，(d2)也變為OFF。
- 變更了工程工具中高速定時器的計測單位(定時器時限設定)預設值的情況下，轉換/全部轉換時將發生報警。
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是，指定了負值的情況下，初始值為0。

例

[梯形圖示例]



[時序圖]

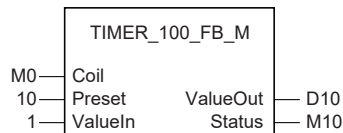


■TIMER_100_FB_M

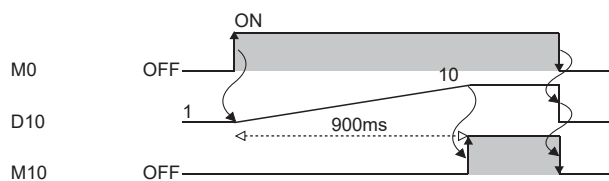
- (s1)的執行條件變為ON時，開始當前值的計測。從(s3)×100ms開始計測，直到(s2)×100ms為止到達計測值時(d2)變為ON。當前計測值被輸出到(d1)中。
- 如果(s1)的執行條件變為OFF，則當前值變為(s3)的值，(d2)也變為OFF。
- 在工程工具的預設值更改了低速定時器的計測單位(定時器時限設定)的情況下，轉換/全部轉換時將發生報警。
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是，指定了負值的情況下，初始值為0。

例

[梯形圖示例]



[時序圖]

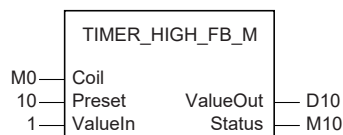


■TIMER_HIGH_FB_M

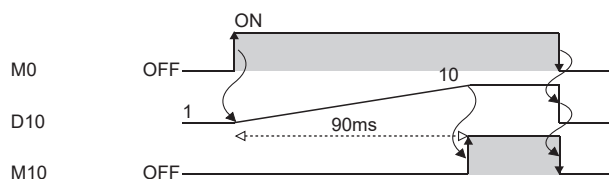
- 是計測單位為0.1~100ms的高速定時器。(s1)的執行條件變為ON時，開始當前值的計測。從(s3)×0.1~100ms(可變。透過參數設定)開始計測，直到(s2)×0.1~100ms為止達到計測值時(d2)變為ON。當前計測值被輸出到(d1)中。
- 如果(s1)的執行條件變為OFF，則當前值變為(s3)的值，(d2)也變為OFF。
- 高速定時器的計測單位(定時器時限設定)的預設值為10ms。計測單位可以在0.01ms~100ms的範圍內更改。
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是，指定了負值的情況下，初始值為0。

例

[梯形圖示例]



[時序圖]

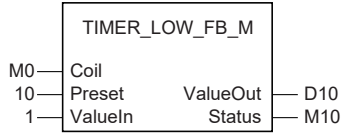


■TIMER_LOW_FB_M

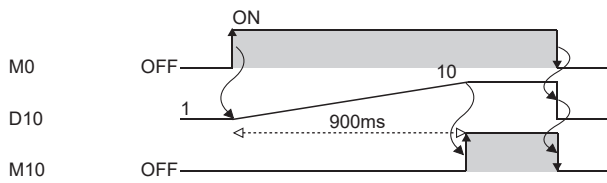
- 是計測單位為1~1000ms的低速定時器。(s1)的執行條件變為ON時，開始當前值的計測。從(s3)×1~1000ms(可變。透過參數設定)開始計測，直到(s2)×1~1000ms為止達到計測值時(d2)變為ON。當前計測值被輸出到(d1)中。
- 如果(s1)的執行條件變為OFF，則當前值變為(s3)的值，(d2)也變為OFF。
- 低速定時器的計測單位(定時器時限設定)的預設值為100ms。計測單位可以在1~1000ms內以1ms單位更改。
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是，指定了負值的情況下，初始值為0。

例

[梯形圖示例]



[時序圖]



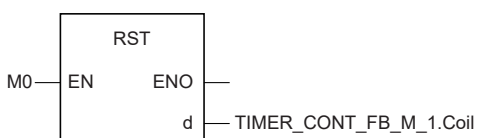
■TIMER_CONT_FB_M/TIMER_CONTHFB_M

- 是計測變數ON的時間的累計定時器。(s1)的執行條件變為ON時，開始當前值的計測。累計定時器有低速累計定時器(TIMER_CONT_FB_M)與高速累計定時器(TIMER_CONTHFB_M)兩種類型。
- 從(s3)×1~1000ms(高速累計定時器時為0.1~100ms)(可變。透過參數設定)開始計測，直到(s2)×1~1000ms(高速累計定時器時為0.1~100ms)達到計測值時(d2)變為ON。當前計測值被輸出到(d1)中。
- 即使(s1)的執行條件變為OFF，仍保持(d1)、(d2)的ON/OFF狀態。(s1)的執行條件再次變為ON時，從保持的計測值重新開始計測。
- 低速定時器(TIMER_LOW_FB_M)與高速定時器(TIMER_HIGH_FB_M)的累計定時器的計測單位(時限)相同。
 - 低速累計定時器：低速定時器
 - 高速累計定時器：高速定時器
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是，指定了負值的情況下，初始值為0。
- 希望重設累計定時器的(d1)的情況下，應直接重設FB的(s1)。

例

標籤名：TIMER_CONT_FB_M_1的情況下

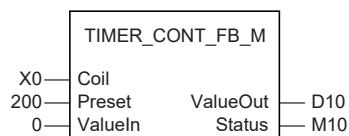
[梯形圖程式]



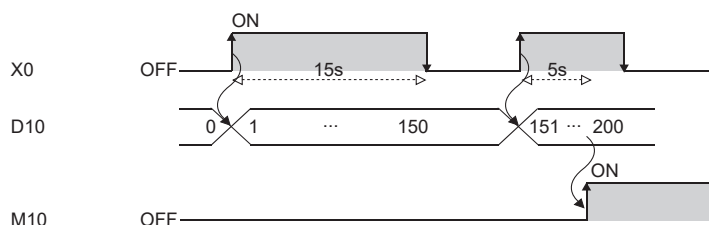
[ST程式]

RST(M0, TIMER_CONT_FB_M_1.Coil)

[梯形圖示例]



[時序圖]



出錯

沒有運算出錯。

附錄

附1 指令處理時間

各指令的處理時間如下所示。

處理時間根據源、目標的內容而有所不同，因此應將表中的值作為處理時間的大致參考。

要點

在使用檔案寄存器(R/ZR)、模組存取元件(U□\G□)、直接連結元件(J□\□)、模組標籤(僅限標籤名中附加_D的標籤)時，應將下述所示的加算時間加算至各指令處理時間。

☞ 1524頁 指令處理時間的加法運算時間

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
LD	執行時	0.03136		0.00392		0.00098	
LDI	執行時	0.03136		0.00392		0.00098	
AND	執行時	0.03136		0.00392		0.00098	
ANI	執行時	0.03136		0.00392		0.00098	
OR	執行時	0.03136		0.00392		0.00098	
ORI	執行時	0.03136		0.00392		0.00098	
LDP	執行時	0.09408		0.01176		0.00294	
LDF	執行時	0.09408		0.01176		0.00294	
ANDP	執行時	0.09408		0.01176		0.00294	
ANDF	執行時	0.09408		0.01176		0.00294	
ORP	執行時	0.09408		0.01176		0.00294	
ORF	執行時	0.09408		0.01176		0.00294	
LDPI	執行時	0.09408		0.01176		0.00294	
LDFI	執行時	0.09408		0.01176		0.00294	
ANDPI	執行時	0.09408		0.01176		0.00294	
ANDFI	執行時	0.09408		0.01176		0.00294	
ORPI	執行時	0.09408		0.01176		0.00294	
ORFI	執行時	0.09408		0.01176		0.00294	
ANB	—	0.03136		0.00392		0.00098	
ORB	—	0.03136		0.00392		0.00098	
MPS	—	0.03136		0.00392		0.00098	
MRD	—	0.03136		0.00392		0.00098	
MPP	—	0.03136		0.00392		0.00098	
INV	無執行時/執行時	0.03136		0.00392		0.00098	
MEP	無執行時/執行時	0.03136		0.00392		0.00098	
MEF	無執行時/執行時	0.03136		0.00392		0.00098	
EGP	無執行時/執行時	0.06272		0.00784		0.00196	
EGF	無執行時/執行時	0.06272		0.00784		0.00196	
OUT	無執行時/執行時	0.06272		0.00784		0.00196	
OUT(F)	無執行時	0.22272		0.02784		0.00696	
	執行時	52.400	83.100	51.400	82.100	50.900	81.600
OUT(T/ST/C)	無執行時	0.38272		0.04784		0.01196	
	執行時：計數時/時間到後	0.38272		0.04784		0.01196	
OUT(LT/LST)	無執行時	0.31872		0.03984		0.00996	
	執行時：計數時/時間到後	0.31872		0.03984		0.00996	
OUT(LC)	無執行時	0.38272		0.04784		0.01196	
	執行時：計數時/時間到後	0.38272		0.04784		0.01196	

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
SET	無執行時	0.06272		0.00784		0.00196	
	執行時：變化時/無變化時	0.06272		0.00784		0.00196	
SET (F)	無執行時	0.22272		0.02784		0.00696	
	執行時	51.800	83.200	50.800	82.200	50.300	81.700
RST	無執行時	0.06272		0.00784		0.00196	
	執行時：變化時/無變化時	0.06272		0.00784		0.00196	
RST (F)	無執行時	0.06272		0.00784		0.00196	
	執行時	17.400	30.800	16.400	29.800	15.900	29.300
RST (T/ST/C)	無執行時	0.31872		0.03984		0.00996	
	執行時	0.31872		0.03984		0.00996	
RST (LT/LST)	無執行時	0.19072		0.02384		0.00596	
	執行時	0.19072		0.02384		0.00596	
RST (LC)	無執行時	0.31872		0.03984		0.00996	
	執行時	0.31872		0.03984		0.00996	
PLS	—	0.06272		0.00784		0.00196	
PLF	—	0.06272		0.00784		0.00196	
FF	無執行時/執行時	0.06272		0.00784		0.00196	
DELTA	無執行時	0.12544		0.01568		0.00392	
	執行時	3.200	7.800	2.200	6.800	1.700	6.300
SFT	無執行時	0.12544		0.01568		0.00392	
	執行時	2.400	4.800	1.400	3.800	0.900	3.300
MC	—	0.06272		0.00784		0.00196	
MCR	—	0.06272		0.00784		0.00196	
PHASE	導通時	0.59584		0.09800		0.06272	
	非導通時/無處理時	0.59584		0.09800		0.06076	
PHASECHG	—	0.18816		0.02352		0.01176	
PHASEEND	—	0.12544		0.01568		0.00784	
FEND	—	應確認以下手冊的END處理時間的項目。					
END	—	□ MELSEC iQ-R CPU模組用戶手冊(應用篇)					
STOP	—	—		—		—	
NOP	—	0.03136		0.00392		0.00098	
LD=	導通時/非導通時	0.18816		0.02352		0.00588	
LD<>	導通時/非導通時	0.18816		0.02352		0.00588	
LD>	導通時/非導通時	0.18816		0.02352		0.00588	
LD<=	導通時/非導通時	0.18816		0.02352		0.00588	
LD<	導通時/非導通時	0.18816		0.02352		0.00588	
LD>=	導通時/非導通時	0.18816		0.02352		0.00588	
AND=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
AND<>	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
AND>	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
AND<=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
AND<	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
AND>=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
OR<>	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR>	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR<=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR<	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR>=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
LD=_U	導通時/非導通時	0.18816		0.02352		0.00588	
LD<>_U	導通時/非導通時	0.18816		0.02352		0.00588	
LD>_U	導通時/非導通時	0.18816		0.02352		0.00588	
LD<=_U	導通時/非導通時	0.18816		0.02352		0.00588	
LD<_U	導通時/非導通時	0.18816		0.02352		0.00588	
LD>=_U	導通時/非導通時	0.18816		0.02352		0.00588	
AND=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
AND<>_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
AND>_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
AND<=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
AND<_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
AND>=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR<>_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR>_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR<=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR<_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
OR>=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
LDD=	導通時/非導通時	0.18816		0.02352		0.00588	
LDD<>	導通時/非導通時	0.18816		0.02352		0.00588	
LDD>	導通時/非導通時	0.18816		0.02352		0.00588	
LDD<=	導通時/非導通時	0.18816		0.02352		0.00588	
LDD<	導通時/非導通時	0.18816		0.02352		0.00588	
LDD>=	導通時/非導通時	0.18816		0.02352		0.00588	
ANDD=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDD<>	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
ANDD>	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDD<=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDD<	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDD>=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD<>	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD>	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD<=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD<	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD>=	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
LDD=_U	導通時/非導通時	0.18816		0.02352		0.00588	
LDD<>_U	導通時/非導通時	0.18816		0.02352		0.00588	
LDD>_U	導通時/非導通時	0.18816		0.02352		0.00588	
LDD<=_U	導通時/非導通時	0.18816		0.02352		0.00588	
LDD<_U	導通時/非導通時	0.18816		0.02352		0.00588	
LDD>=_U	導通時/非導通時	0.18816		0.02352		0.00588	
ANDD=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDD<>_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDD>_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDD<=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDD<_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDD>=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD<>_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD>_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD<=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD<_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORD>=_U	無執行時	0.12544		0.01568		0.00392	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
CMP	—	4.400	6.900	3.400	5.900	2.900	5.400

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
CMP_U	—	4.400	6.900	3.400	5.900	2.900	5.400
DCMP	—	4.400	6.900	3.400	5.900	2.900	5.400
DCMP_U	—	4.400	6.900	3.400	5.900	2.900	5.400
ZCP	—	4.800	7.700	3.800	6.700	3.300	6.200
ZCP_U	—	4.800	7.700	3.800	6.700	3.300	6.200
DZCP	—	4.800	7.700	3.800	6.700	3.300	6.200
DZCP_U	—	4.800	7.700	3.800	6.700	3.300	6.200
BKCMP=	(n)=1	5.100	12.000	4.100	11.000	3.600	10.500
	(n)=96	18.500	24.900	17.500	23.900	17.000	23.400
BKCMP<>	(n)=1	5.200	11.400	4.200	10.400	3.700	9.900
	(n)=96	18.900	25.100	17.900	24.100	17.400	23.600
BKCMP>	(n)=1	5.100	12.000	4.100	11.000	3.600	10.500
	(n)=96	18.300	26.100	17.300	25.100	16.800	24.600
BKCMP<=	(n)=1	5.000	11.800	4.000	10.800	3.500	10.300
	(n)=96	19.000	24.800	18.000	23.800	17.500	23.300
BKCMP<	(n)=1	5.100	11.800	4.100	10.800	3.600	10.300
	(n)=96	19.000	26.000	18.000	25.000	17.500	24.500
BKCMP>=	(n)=1	5.100	11.500	4.100	10.500	3.600	10.000
	(n)=96	19.200	25.600	18.200	24.600	17.700	24.100
BKCMP=_U	(n)=1	5.100	11.800	4.100	10.800	3.600	10.300
	(n)=96	18.400	25.400	17.400	24.400	16.900	23.900
BKCMP<>_U	(n)=1	5.200	11.300	4.200	10.300	3.700	9.800
	(n)=96	18.800	25.100	17.800	24.100	17.300	23.600
BKCMP>_U	(n)=1	5.000	11.700	4.000	10.700	3.500	10.200
	(n)=96	18.400	25.100	17.400	24.100	16.900	23.600
BKCMP<=_U	(n)=1	5.100	11.600	4.100	10.600	3.600	10.100
	(n)=96	18.200	24.800	17.200	23.800	16.700	23.300
BKCMP<_U	(n)=1	5.200	11.600	4.200	10.600	3.700	10.100
	(n)=96	18.300	25.600	17.300	24.600	16.800	24.100
BKCMP>=_U	(n)=1	5.100	11.400	4.100	10.400	3.600	9.900
	(n)=96	18.500	26.900	17.500	25.900	17.000	25.400
DBKCMPE=	(n)=1	5.200	11.900	4.200	10.900	3.700	10.400
	(n)=96	18.500	25.000	17.500	24.000	17.000	23.500
DBKCMPE<>	(n)=1	5.200	11.500	4.200	10.500	3.700	10.000
	(n)=96	18.900	25.200	17.900	24.200	17.400	23.700
DBKCMPE>	(n)=1	5.100	11.600	4.100	10.600	3.600	10.100
	(n)=96	18.900	25.600	17.900	24.600	17.400	24.100
DBKCMPE<=	(n)=1	5.300	11.600	4.300	10.600	3.800	10.100
	(n)=96	18.600	25.100	17.600	24.100	17.100	23.600
DBKCMPE<	(n)=1	5.200	11.400	4.200	10.400	3.700	9.900
	(n)=96	18.800	25.300	17.800	24.300	17.300	23.800
DBKCMPE>=	(n)=1	5.100	11.700	4.100	10.700	3.600	10.200
	(n)=96	18.600	24.400	17.600	23.400	17.100	22.900
DBKCMPE=_U	(n)=1	5.200	11.700	4.200	10.700	3.700	10.200
	(n)=96	18.500	25.100	17.500	24.100	17.000	23.600
DBKCMPE<>_U	(n)=1	5.200	11.500	4.200	10.500	3.700	10.000
	(n)=96	18.900	25.100	17.900	24.100	17.400	23.600
DBKCMPE>_U	(n)=1	5.200	11.700	4.200	10.700	3.700	10.200
	(n)=96	25.800	32.300	24.800	31.300	24.300	30.800
DBKCMPE<=_U	(n)=1	5.200	11.700	4.200	10.700	3.700	10.200
	(n)=96	25.400	31.800	24.400	30.800	23.900	30.300

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
DBKCMPL<_U	(n)=1	5.300	11.500	4.300	10.500	3.800	10.000
	(n)=96	25.800	32.500	24.800	31.500	24.300	31.000
DBKCMPL>=_U	(n)=1	5.300	12.000	4.300	11.000	3.800	10.500
	(n)=96	25.900	32.500	24.900	31.500	24.400	31.000
+ (s) (d)	執行時	0.18816		0.02352		0.00588	
+ (s1) (s2) (d)	執行時	0.18816		0.02352		0.00588	
+_U (s) (d)	執行時	0.18816		0.02352		0.00588	
+_U (s1) (s2) (d)	執行時	0.18816		0.02352		0.00588	
- (s) (d)	執行時	0.18816		0.02352		0.00588	
- (s1) (s2) (d)	執行時	0.18816		0.02352		0.00588	
-_U (s) (d)	執行時	0.18816		0.02352		0.00588	
-_U (s1) (s2) (d)	執行時	0.18816		0.02352		0.00588	
D+ (s) (d)	執行時	0.18816		0.02352		0.00588	
D+ (s1) (s2) (d)	執行時	0.18816		0.02352		0.00588	
D+_U (s) (d)	執行時	0.18816		0.02352		0.00588	
D+_U (s1) (s2) (d)	執行時	0.18816		0.02352		0.00588	
D- (s) (d)	執行時	0.18816		0.02352		0.00588	
D- (s1) (s2) (d)	執行時	0.18816		0.02352		0.00588	
D-_U (s) (d)	執行時	0.18816		0.02352		0.00588	
D-_U (s1) (s2) (d)	執行時	0.18816		0.02352		0.00588	
*	執行時	0.37632		0.04704		0.01176	
*_U	執行時	0.37632		0.04704		0.01176	
/	執行時	0.62720		0.07840		0.01960	
/_U	執行時	0.62720		0.07840		0.01960	
D*	執行時	0.62720		0.07840		0.01960	
D*_U	執行時	0.62720		0.07840		0.01960	
D/	執行時	0.94080		0.11760		0.02940	
D/_U	執行時	0.94080		0.11760		0.02940	
B+ (s) (d)	執行時	2.800	5.200	1.800	4.200	1.300	3.700
B+ (s1) (s2) (d)	執行時	3.300	6.600	2.300	5.600	1.800	5.100
B- (s) (d)	執行時	2.800	5.300	1.800	4.300	1.300	3.800
B- (s1) (s2) (d)	執行時	3.400	6.500	2.400	5.500	1.900	5.000
DB+ (s) (d)	執行時	3.500	7.700	2.500	6.700	2.000	6.200
DB+ (s1) (s2) (d)	執行時	3.600	7.600	2.600	6.600	2.100	6.100
DB- (s) (d)	執行時	3.500	7.600	2.500	6.600	2.000	6.100
DB- (s1) (s2) (d)	執行時	3.600	7.400	2.600	6.400	2.100	5.900
B*	執行時	2.900	6.400	1.900	5.400	1.400	4.900
B/	執行時	3.000	6.300	2.000	5.300	1.500	4.800
DB*	執行時	3.900	9.500	2.900	8.500	2.400	8.000
DB/	執行時	3.700	9.400	2.700	8.400	2.200	7.900
BK+	(n)=1	5.000	9.600	4.000	8.600	3.500	8.100
	(n)=96	18.700	23.000	17.700	22.000	17.200	21.500
BK+_U	(n)=1	4.700	9.800	3.700	8.800	3.200	8.300
	(n)=96	19.000	23.100	18.000	22.100	17.500	21.600
BK-	(n)=1	4.600	9.700	3.600	8.700	3.100	8.200
	(n)=96	18.900	22.900	17.900	21.900	17.400	21.400
BK-_U	(n)=1	4.700	9.800	3.700	8.800	3.200	8.300
	(n)=96	19.000	23.000	18.000	22.000	17.500	21.500
DBK+	(n)=1	4.800	8.400	3.800	7.400	3.300	6.900
	(n)=96	18.900	23.400	17.900	22.400	17.400	21.900

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
DBK+ _U	(n)=1	4.500	9.300	3.500	8.300	3.000	7.800
	(n)=96	18.900	23.400	17.900	22.400	17.400	21.900
DBK-	(n)=1	4.500	9.400	3.500	8.400	3.000	7.900
	(n)=96	18.900	23.500	17.900	22.500	17.400	22.000
DBK- _U	(n)=1	4.500	9.400	3.500	8.400	3.000	7.900
	(n)=96	18.900	23.500	17.900	22.500	17.400	22.000
INC	執行時	0.12544		0.01568		0.00392	
INC_U	執行時	0.12544		0.01568		0.00392	
DEC	執行時	0.12544		0.01568		0.00392	
DEC_U	執行時	0.12544		0.01568		0.00392	
DINC	執行時	0.12544		0.01568		0.00392	
DINC_U	執行時	0.12544		0.01568		0.00392	
DDEC	執行時	0.12544		0.01568		0.00392	
DDEC_U	執行時	0.12544		0.01568		0.00392	
WAND(s) (d)	執行時	0.12544		0.01568		0.00392	
WAND(s1) (s2) (d)	執行時	0.12544		0.01568		0.00392	
DAND(s) (d)	執行時	0.12544		0.01568		0.00392	
DAND(s1) (s2) (d)	執行時	0.12544		0.01568		0.00392	
BKAND	(n)=1	4.800	9.900	3.800	8.900	3.300	8.400
	(n)=96	19.000	23.900	18.000	22.900	17.500	22.400
WOR(s) (d)	執行時	0.12544		0.01568		0.00392	
WOR(s1) (s2) (d)	執行時	0.12544		0.01568		0.00392	
DOR(s) (d)	執行時	0.12544		0.01568		0.00392	
DOR(s1) (s2) (d)	執行時	0.12544		0.01568		0.00392	
BKOR	(n)=1	4.800	9.800	3.800	8.800	3.300	8.300
	(n)=96	19.000	24.300	18.000	23.300	17.500	22.800
WXOR(s) (d)	執行時	0.12544		0.01568		0.00392	
WXOR(s1) (s2) (d)	執行時	0.12544		0.01568		0.00392	
DXOR(s) (d)	執行時	0.12544		0.01568		0.00392	
DXOR(s1) (s2) (d)	執行時	0.12544		0.01568		0.00392	
BKXOR	(n)=1	4.800	9.800	3.800	8.800	3.300	8.300
	(n)=96	19.000	23.900	18.000	22.900	17.500	22.400
WXNR(s) (d)	執行時	0.12544		0.01568		0.00392	
WXNR(s1) (s2) (d)	執行時	0.12544		0.01568		0.00392	
DXNR(s) (d)	執行時	0.12544		0.01568		0.00392	
DXNR(s1) (s2) (d)	執行時	0.12544		0.01568		0.00392	
BKXNR	(n)=1	4.800	9.700	3.800	8.700	3.300	8.200
	(n)=96	19.400	24.300	18.400	23.300	17.900	22.800
BSET	(n)=1	0.09408		0.01176		0.00294	
	(n)=15	0.09408		0.01176		0.00294	
BRST	(n)=1	0.09408		0.01176		0.00294	
	(n)=15	0.09408		0.01176		0.00294	
TEST	執行時	0.38400		0.04800		0.0120	
DTEST	執行時	0.60800		0.07600		0.0190	
BKRST	(n)=1	3.000	3.500	1.100	2.300	0.900	2.000
	(n)=96	4.600	5.300	1.600	2.500	1.200	2.200
SFR	(n)=1	0.28224		0.03528		0.00882	
	(n)=15	0.28224		0.03528		0.00882	
SFL	(n)=1	0.28224		0.03528		0.00882	
	(n)=15	0.28224		0.03528		0.00882	

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
BSFR	(n)=1	3.000	3.500	1.100	2.000	0.800	1.800
	(n)=96	6.200	6.700	1.900	3.000	1.500	2.600
BSFL	(n)=1	3.000	3.500	1.100	2.200	0.800	1.900
	(n)=96	6.200	6.700	1.900	3.000	1.500	2.600
DSFR	(n)=1	2.600	4.800	1.600	3.800	1.100	3.300
	(n)=96	9.700	12.300	8.700	11.300	8.200	10.800
DSFL	(n)=1	2.600	4.900	1.600	3.900	1.100	3.400
	(n)=96	9.700	12.200	8.700	11.200	8.200	10.700
DDSFR	(n)=1	3.700	5.300	2.700	4.300	2.200	5.400
	(n)=96	4.500	6.700	3.500	5.700	3.200	6.800
DDSFL	(n)=1	3.600	5.100	2.600	4.100	2.200	5.400
	(n)=96	4.600	6.600	3.600	5.600	3.200	6.800
ESFR	(n)=1	3.700	5.300	2.700	4.300	2.200	5.400
	(n)=96	4.600	6.800	3.600	5.800	3.200	6.800
ESFL	(n)=1	3.700	5.000	2.700	4.000	2.200	5.400
	(n)=96	4.700	6.600	3.700	5.600	3.200	6.800
EDSFR	(n)=1	3.800	5.700	2.800	4.700	2.800	5.800
	(n)=96	5.100	7.500	4.100	6.500	4.300	7.800
EDSFL	(n)=1	3.800	5.600	2.800	4.600	2.800	5.800
	(n)=96	5.100	7.500	4.100	6.500	4.300	7.800
SFTBR	(n1)=16、(n2)=1	7.600	8.200	1.900	2.900	0.900	2.100
	(n1)=16、(n2)=15	7.600	8.200	1.900	2.900	0.900	2.100
SFTR	(n1)=16、(n2)=1	5.900	12.800	4.900	11.800	4.400	11.300
	(n1)=16、(n2)=15	5.900	12.800	4.900	11.800	4.400	11.300
SFTBL	(n1)=16、(n2)=1	7.600	8.200	1.900	2.900	0.900	2.100
	(n1)=16、(n2)=15	7.600	8.200	1.900	2.900	0.900	2.100
SFTL	(n1)=16、(n2)=1	5.900	12.800	4.900	11.800	4.400	11.300
	(n1)=16、(n2)=15	5.900	12.800	4.900	11.800	4.400	11.300
SFTWR	(n1)=16、(n2)=1	3.900	6.900	2.900	5.900	2.400	5.400
	(n1)=16、(n2)=15	3.900	7.200	2.900	6.200	2.400	5.700
WSFR	(n1)=16、(n2)=1	6.600	11.100	5.600	10.100	5.100	9.600
	(n1)=16、(n2)=15	6.600	11.100	5.600	10.100	5.100	9.600
SFTWL	(n1)=16、(n2)=1	3.900	7.300	2.900	6.300	2.400	5.800
	(n1)=16、(n2)=15	3.900	6.900	2.900	5.900	2.400	5.400
WSFL	(n1)=16、(n2)=1	6.600	11.100	5.600	10.100	5.100	9.600
	(n1)=16、(n2)=15	6.600	11.100	5.600	10.100	5.100	9.600
SFTDWR	(n1)=16、(n2)=1	5.200	7.500	4.200	6.500	3.900	7.700
	(n1)=16、(n2)=15	5.300	7.600	4.300	6.600	3.900	7.700
DWSFTR	(n1)=16、(n2)=1	6.300	11.200	5.300	10.200	5.100	11.500
	(n1)=16、(n2)=15	6.300	11.100	5.300	10.100	5.100	11.500
SFTDWL	(n1)=16、(n2)=1	5.200	7.500	4.200	6.500	3.900	7.700
	(n1)=16、(n2)=15	5.200	7.500	4.200	6.500	3.900	7.700
DWSFTL	(n1)=16、(n2)=1	6.700	11.400	5.700	10.400	5.100	11.500
	(n1)=16、(n2)=15	6.800	11.300	5.800	10.300	5.100	11.500
SFTR	(n1)=16、(n2)=1	5.300	7.500	4.300	6.500	3.900	7.700
	(n1)=16、(n2)=15	5.300	7.600	4.300	6.600	3.900	7.700
ESFTR	(n1)=16、(n2)=1	6.400	11.200	5.400	10.200	5.100	11.500
	(n1)=16、(n2)=15	6.300	11.200	5.300	10.200	5.100	11.500
SFTL	(n1)=16、(n2)=1	5.200	7.500	4.200	6.500	3.900	7.700
	(n1)=16、(n2)=15	5.200	7.500	4.200	6.500	3.900	7.700

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
ESFTL	(n1)=16、(n2)=1	6.800	11.300	5.800	10.300	5.100	11.500
	(n1)=16、(n2)=15	6.800	11.300	5.800	10.300	5.100	11.500
SFTEDR	(n1)=16、(n2)=1	5.100	7.600	4.100	6.600	4.100	7.900
	(n1)=16、(n2)=15	5.200	7.700	4.200	6.700	4.100	7.900
EDSFTR	(n1)=16、(n2)=1	6.300	11.300	5.300	10.300	5.800	11.800
	(n1)=16、(n2)=15	6.400	11.200	5.400	10.200	5.800	11.800
SFTEDL	(n1)=16、(n2)=1	5.200	7.400	4.200	6.400	4.100	7.900
	(n1)=16、(n2)=15	5.300	7.500	4.300	6.500	4.100	7.900
EDSFTL	(n1)=16、(n2)=1	6.400	11.200	5.400	10.200	5.800	11.800
	(n1)=16、(n2)=15	6.500	11.200	5.500	10.200	5.800	11.800
BCD	執行時	0.34496		0.04312		0.01078	
DBCD	執行時	0.59584		0.07448		0.01862	
BIN	執行時	0.21952		0.02744		0.00686	
DBIN	執行時	0.21952		0.02744		0.00686	
FLT2INT	(s)=0	0.21952		0.02744		0.00686	
	(s)=32766.5	0.21952		0.02744		0.00686	
FLT2UINT	(s)=0	0.21952		0.02744		0.00686	
	(s)=65534.5	0.21952		0.02744		0.00686	
FLT2DINT	(s)=0	0.21952		0.02744		0.00686	
	(s)=1234567890.3	0.21952		0.02744		0.00686	
FLT2UDINT	(s)=0	0.21952		0.02744		0.00686	
	(s)=1234567890.3	0.21952		0.02744		0.00686	
DBL2INT	(s)=0	2.900	4.900	1.900	3.900	1.400	3.400
	(s)=32766.5	3.000	5.600	2.000	4.600	1.500	4.100
DBL2UINT	(s)=0	2.800	5.100	1.800	4.100	1.300	3.600
	(s)=65534.5	3.000	5.500	2.000	4.500	1.500	4.000
DBL2DINT	(s)=0	2.900	4.900	1.900	3.900	1.400	3.400
	(s)=1234567890.3	2.900	5.700	1.900	4.700	1.400	4.200
DBL2UDINT	(s)=0	3.000	4.900	2.000	3.900	1.500	3.400
	(s)=1234567890.3	3.000	5.600	2.000	4.600	1.500	4.100
INT2UINT	執行時	0.09408		0.01176		0.00294	
INT2DINT	執行時	0.09408		0.01176		0.00294	
INT2UDINT	執行時	0.09408		0.01176		0.00294	
UINT2INT	執行時	0.09408		0.01176		0.00294	
UINT2DINT	執行時	0.09408		0.01176		0.00294	
UINT2UDINT	執行時	0.09408		0.01176		0.00294	
DINT2INT	執行時	0.09408		0.01176		0.00294	
DINT2UINT	執行時	0.09408		0.01176		0.00294	
DINT2UDINT	執行時	0.09408		0.01176		0.00294	
UDINT2INT	執行時	0.09408		0.01176		0.00294	
UDINT2UINT	執行時	0.09408		0.01176		0.00294	
UDINT2DINT	執行時	0.09408		0.01176		0.00294	
GRY	執行時	0.15680		0.01960		0.00490	
GRY_U	執行時	0.15680		0.01960		0.00490	
DGRY	執行時	0.15680		0.01960		0.00490	
DGRY_U	執行時	0.15680		0.01960		0.00490	
GBIN	執行時	0.15680		0.01960		0.00490	
GBIN_U	執行時	0.15680		0.01960		0.00490	
DGBIN	執行時	0.15680		0.01960		0.00490	
DGBIN_U	執行時	0.15680		0.01960		0.00490	

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外のCPU模組	
		最小	最大	最小	最大	最小	最大
BKBCD	(n)=1	3.900	9.300	2.900	8.300	2.400	7.800
	(n)=96	22.200	27.200	21.200	26.200	20.700	25.700
BKBIN	(n)=1	3.800	8.900	2.800	7.900	2.300	7.400
	(n)=96	18.600	23.400	17.600	22.400	17.100	21.900
DABIN	(s)=1	4.500	12.300	3.500	11.300	3.000	10.800
	(s)=-32768	4.500	12.300	3.500	11.300	3.000	10.800
DABIN_U	(s)=1	4.500	12.300	3.500	11.300	3.000	10.800
	(s)=65535	4.500	12.300	3.500	11.300	3.000	10.800
DDABIN	(s)=1	4.900	12.600	3.900	11.600	3.400	11.100
	(s)=-2147483648	4.900	12.600	3.900	11.600	3.400	11.100
DDABIN_U	(s)=1	4.900	12.600	3.900	11.600	3.400	11.100
	(s)=4294967295	4.900	12.600	3.900	11.600	3.400	11.100
HABIN	(s)=1	4.400	10.200	3.400	9.200	2.900	8.700
	(s)=FFFFH	4.400	10.200	3.400	9.200	2.900	8.700
DHABIN	(s)=1	4.600	10.500	3.600	9.500	3.100	9.000
	(s)=FFFFFFFFH	4.600	10.500	3.600	9.500	3.100	9.000
DABCD	(s)=1	4.300	10.400	3.300	9.400	2.800	8.900
	(s)=9999	4.300	10.400	3.300	9.400	2.800	8.900
DDABCD	(s)=1	4.500	10.600	3.500	9.600	3.000	9.100
	(s)=99999999	4.500	10.600	3.500	9.600	3.000	9.100
VAL	—	5.300	13.400	4.300	12.400	3.800	11.900
VAL_U	—	5.600	13.400	4.600	12.400	4.100	11.900
DVAL	—	6.400	14.600	5.400	13.600	4.900	13.100
DVAL_U	—	6.700	14.100	5.700	13.100	5.200	12.600
ASC2INT	(n)=1	4.100	8.900	3.100	7.900	2.600	7.400
	(n)=96	9.800	15.200	8.800	14.200	8.300	13.700
EMOD	—	4.500	9.700	3.500	8.700	3.000	8.200
NEG	執行時	0.15680		0.01960		0.00490	
DNEG	執行時	0.15680		0.01960		0.00490	
DECO	(n)=2	3.900	7.400	2.900	6.400	2.400	5.900
	(n)=8	3.900	7.800	2.900	6.800	2.400	6.300
ENCO	(n)=2、M1=ON	4.000	7.900	3.000	6.900	2.500	6.400
	(n)=2、M4=ON	4.000	7.900	3.000	6.900	2.500	6.400
	(n)=8、M1=ON	5.000	9.300	4.000	8.300	3.500	7.800
	(n)=8、M256=ON	3.900	7.700	2.900	6.700	2.400	6.200
SEG	執行時	1.900	3.100	0.900	2.100	0.400	1.600
DIS	(n)=1	3.700	5.700	2.700	4.700	2.200	4.200
	(n)=4	3.700	5.800	2.700	4.800	2.200	4.300
UNI	(n)=1	3.900	6.800	2.900	5.800	2.400	5.300
	(n)=4	4.000	6.700	3.000	5.700	2.500	5.200
NDIS	執行時	3.700	6.500	2.700	5.500	2.200	5.000
NUNI	執行時	3.700	6.500	2.700	5.500	2.200	5.000
WTOB	(n)=1	4.000	7.000	3.000	6.000	2.500	5.500
	(n)=96	14.500	17.500	13.500	16.500	13.000	16.000
BTOW	(n)=1	4.100	7.000	3.100	6.000	2.600	5.500
	(n)=96	11.300	14.100	10.300	13.100	9.800	12.600
MOV	—	0.06272		0.00784		0.00196	
DMOV	—	0.06272		0.00784		0.00196	
CML	—	0.06272		0.00784		0.00196	
DCML	—	0.06272		0.00784		0.00196	

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
SMOV	SM773=OFF	5.000	10.000	4.000	9.000	3.500	8.500
	SM773=ON	4.700	7.600	3.700	6.600	3.200	6.100
CMLB	—	0.06272		0.00784		0.00196	
BMOV	(n)=1	4.800	5.000	1.500	1.600	1.100	1.300
	(n)=96	6.600	6.700	2.500	2.700	2.000	2.200
BMOVL	(n)=1	5.000	5.500	1.700	2.900	1.300	2.500
	(n)=96	6.900	7.400	2.700	4.100	2.200	3.600
FMOV	(n)=1	3.800	4.000	1.100	1.400	0.700	1.000
	(n)=96	5.700	6.000	2.300	2.500	1.800	2.000
FMOVL	(n)=1	4.100	4.600	1.300	2.300	1.000	2.000
	(n)=96	6.000	6.600	2.400	3.700	2.000	3.300
DFMOV	(n)=1	4.000	4.500	1.300	2.300	1.000	2.000
	(n)=96	6.200	6.600	2.600	3.800	2.200	3.400
DFMOVL	(n)=1	4.200	4.600	1.400	2.400	1.000	2.000
	(n)=96	6.300	6.700	2.700	4.000	2.200	3.500
XCH	—	3.000	3.400	1.100	1.700	0.900	1.500
DXCH	—	3.100	3.400	1.200	1.800	0.900	1.500
BXCH	(n)=1	4.200	8.300	3.200	7.300	2.700	6.800
	(n)=96	18.400	22.200	17.400	21.200	16.900	20.700
SWAP	—	2.700	3.700	1.700	2.700	1.200	2.200
DSWAP	—	2.700	3.700	1.700	2.700	1.200	2.200
MOVB	—	0.06272		0.00784		0.00196	
BLKMOVB	(n)=1	4.600	9.100	3.600	8.100	3.100	7.600
	(n)=96	5.100	10.100	4.100	9.100	3.600	8.600
ROR	(n)=1	0.28224		0.03528		0.00882	
	(n)=15	0.28224		0.03528		0.00882	
RCR	(n)=1	0.28224		0.03528		0.00882	
	(n)=15	0.28224		0.03528		0.00882	
DROR	(n)=1	0.28224		0.03528		0.00882	
	(n)=31	0.28224		0.03528		0.00882	
DRCR	(n)=1	0.28224		0.03528		0.00882	
	(n)=31	0.28224		0.03528		0.00882	
ROL	(n)=1	0.28224		0.03528		0.00882	
	(n)=15	0.28224		0.03528		0.00882	
RCL	(n)=1	0.28224		0.03528		0.00882	
	(n)=15	0.28224		0.03528		0.00882	
DROL	(n)=1	0.28224		0.03528		0.00882	
	(n)=31	0.28224		0.03528		0.00882	
DRCL	(n)=1	0.28224		0.03528		0.00882	
	(n)=31	0.28224		0.03528		0.00882	
CJ	—	4.000	4.700	1.300	2.300	1.000	2.000
SCJ	—	1.300	2.400	1.100	2.200	1.000	2.000
JMP	—	4.100	4.900	1.400	2.500	1.000	2.100
GOEND	—	28.80000		3.60000		0.9000	
DI	—	6.000	8.300	3.700	6.000	2.800	4.200
DI(s)	—	6.600	8.900	3.800	6.100	3.000	4.200
EI	—	6.000	11.000	4.600	10.500	4.100	9.200
IMASK	—	3.500	4.000	1.300	2.000	1.000	1.800
SIMASK	—	3.100	3.700	1.000	2.000	0.800	1.700
IRET	—	3.200	3.700	2.200	2.700	1.700	2.200
WDT	—	6.400	17.900	5.400	16.900	4.900	16.400

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
FOR	—	0.06272		0.00784		0.00196	
NEXT	—	1.25440		0.15680		0.03920	
BREAK	—	5.900	11.500	4.900	10.500	4.400	10.000
CALL Pn	局部指針	3.700	4.300	1.100	1.800	0.800	1.500
	全局指針	8.200	17.700	4.900	15.500	3.900	13.500
CALL Pn(s1)~(s5)	局部指針	18.900	42.100	17.900	41.100	17.400	40.600
RET	返回至本程式	6.40000		0.80000		0.2000	
	返回至其它程式	64.00000		8.00000		2.0000	
FCALL Pn	局部指針	5.500	6.200	1.800	2.500	1.000	2.000
	全局指針	14.000	34.000	9.000	30.000	4.300	22.900
FCALL Pn(s1)~(s5)	局部指針	17.900	39.300	16.900	38.300	16.400	37.800
ECALL Pn	局部指針、檔案名="P1"	90.000	145.500	83.500	134.000	74.700	118.400
ECALL Pn(s1)~(s5)	局部指針、檔案名="P1"	96.000	149.900	95.000	148.900	94.500	148.400
EFCALL Pn	局部指針、檔案名="P1"	85.000	130.000	82.000	127.000	72.400	114.300
EFCALL Pn(s1)~(s5)	局部指針、檔案名="P1"	92.600	133.700	91.600	132.700	91.100	132.200
XCALL Pn	—	6.800	20.700	4.500	19.000	3.800	16.700
FIFR	資料儲存數=1	3.600	6.300	2.600	5.300	2.100	4.800
	資料儲存數=96	8.600	11.800	7.600	10.800	7.100	10.300
FPOP	資料儲存數=1	3.500	6.100	2.500	5.100	2.000	4.600
	資料儲存數=96	3.600	6.100	2.600	5.100	2.100	4.600
FIFW	資料儲存數=0	3.600	6.300	2.600	5.300	2.100	4.800
	資料儲存數=96	3.600	6.200	2.600	5.200	2.100	4.700
FINS	資料儲存數=0	4.000	7.300	3.000	6.300	2.500	5.800
	資料儲存數=96	10.700	15.200	9.700	14.200	9.200	13.700
FDEL	資料儲存數=1	3.800	7.500	2.800	6.500	2.300	6.000
	資料儲存數=96	8.900	13.000	7.900	12.000	7.400	11.500
S.DEVLD	—	5.800	8.800	4.800	7.800	4.300	7.300
SP.DEVST	—	21.700	26.300	20.700	25.300	20.200	24.800
SP.FREAD	—	38.300	44.300	37.300	43.300	36.800	42.800
SP.FWRITE	—	43.000	48.900	42.000	47.900	41.500	47.400
SP.FDELETE	—	26.700	30.200	25.700	29.200	25.200	28.700
SP.FCOPY	—	31.100	34.300	30.100	33.300	29.600	32.800
SP.FMOVE	—	31.100	34.300	30.100	33.300	29.600	32.800
SP.FRENAME	—	32.500	35.900	31.500	34.900	31.000	34.400
SP.FSTATUS	—	28.700	32.000	27.700	31.000	27.200	30.500
LEDR	未發生自我診斷出錯時	3.600	6.700	2.600	5.700	2.100	5.200
	發生自我診斷出錯時(發生繼續運行型出錯、報警器ON(1個)時)	16.500	27.400	15.500	26.400	15.000	25.900
PALERT(P)	—	—	—	—	—	77.100	146.000
PABORT	—	—	—	—	—	—	—
LD\$=	導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
LD\$<>	導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
LD\$>	導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
LD\$<=	導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
LD\$<	導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
LD\$>=	導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
AND\$=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
AND\$<	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
AND\$>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
AND\$<=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
AND\$<	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
AND\$>=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
OR\$=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
OR\$<>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
OR\$>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
OR\$<=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
OR\$<	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
OR\$>=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.100	6.000	2.100	5.000	1.600	4.500
\$+ (s) (d)	執行時	3.900	8.200	2.900	7.200	2.400	6.700
\$+ (s1) (s2) (d)	執行時	4.400	9.700	3.400	8.700	2.900	8.200
\$MOV	0字元	3.900	10.700	2.900	9.700	2.400	9.200
	32字元	5.900	13.600	4.900	12.600	4.400	12.100
\$MOV_WS	0字元	3.900	11.600	2.900	10.600	2.400	10.100
	32字元	7.600	15.700	6.600	14.700	6.100	14.200
BINDA	(s)=1	4.100	7.400	3.100	6.400	2.600	5.900
	(s)=-32768	4.500	8.300	3.500	7.300	3.000	6.800
BINDA_U	(s)=1	4.100	7.400	3.100	6.400	2.600	5.900
	(s)=65535	4.500	8.300	3.500	7.300	3.000	6.800
DBINDA	(s)=1	4.200	7.600	3.200	6.600	2.700	6.100
	(s)=-2147483648	4.600	8.900	3.600	7.900	3.100	7.400
DBINDA_U	(s)=1	4.200	7.600	3.200	6.600	2.700	6.100
	(s)=4294967295	4.600	8.900	3.600	7.900	3.100	7.400
BINHA	(s)=1	4.100	7.000	3.100	6.000	2.600	5.500
	(s)=FFFFH	4.100	7.100	3.100	6.100	2.600	5.600
DBINHA	(s)=1	4.100	6.800	3.100	5.800	2.600	5.300
	(s)=FFFFFFFH	4.100	7.200	3.100	6.200	2.600	5.700
STR	—	4.300	8.800	3.300	7.800	2.800	7.300
STR_U	—	4.400	8.800	3.400	7.800	2.900	7.300
DSTR	—	4.700	8.900	3.700	7.900	3.200	7.400
DSTR_U	—	4.800	9.400	3.800	8.400	3.300	7.900
BCDDA	(s)=1	4.100	7.700	3.100	6.700	2.600	6.200
	(s)=9999	4.200	7.800	3.200	6.800	2.700	6.300
DBCDDA	(s)=1	4.100	7.900	3.100	6.900	2.600	6.400
	(s)=99999999	4.200	8.200	3.200	7.200	2.700	6.700
ESTR	—	6.200	18.600	5.200	17.600	4.7000	17.1000
INT2ASC	(n)=1	4.100	8.400	3.100	7.400	2.600	6.900
	(n)=96	7.600	12.500	6.600	11.500	6.100	11.000

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
WS2SJIS	字元數=1	4.800	11.800	3.800	10.800	3.300	10.300
	字元數=96	54.000	60.700	53.000	59.700	52.500	59.200
SJIS2WS	字元數=1	4.600	11.400	3.600	10.400	3.100	9.900
	字元數=96	49.800	56.700	48.800	55.700	48.300	55.200
SJIS2WSB	字元數=1	4.700	11.300	3.700	10.300	3.200	9.800
	字元數=96	49.800	56.600	48.800	55.600	48.300	55.100
LEN	1字元	2.900	5.300	1.900	4.300	1.400	3.800
	96字元	11.100	13.400	10.100	12.400	9.600	11.900
RIGHT	獲取字元數=1	4.700	12.700	3.700	11.700	3.200	11.200
	獲取字元數=96	17.300	24.900	16.300	23.900	15.800	23.400
LEFT	獲取字元數=1	4.700	12.600	3.700	11.600	3.200	11.100
	獲取字元數=96	17.200	24.700	16.200	23.700	15.700	23.200
MIDR	—	5.100	13.700	4.100	12.700	3.600	12.200
MIDW	—	5.700	13.800	4.700	12.800	4.200	12.300
INSTR	無一致	9.000	16.500	8.000	15.500	7.500	15.000
	有一致：起始	6.800	14.500	5.800	13.500	5.300	13.000
	有一致：最後	9.000	16.800	8.000	15.800	7.500	15.300
STRINS	(s)=128、(d)=40、(n)=1	19.400	28.700	18.400	27.700	17.900	27.200
	(s)=128、(d)=40、(n)=48	21.900	31.600	20.900	30.600	20.400	30.100
STRDEL	(s)=128、(d)=40、(n)=1	17.300	24.500	16.300	23.500	15.800	23.000
	(s)=128、(d)=40、(n)=48	15.100	22.400	14.100	21.400	13.600	20.900
LDE=	導通時/非導通時	0.18816		0.02352		0.00588	
LDE<>	導通時/非導通時	0.18816		0.02352		0.00588	
LDE>	導通時/非導通時	0.18816		0.02352		0.00588	
LDE<=	導通時/非導通時	0.18816		0.02352		0.00588	
LDE<	導通時/非導通時	0.18816		0.02352		0.00588	
LDE>=	導通時/非導通時	0.18816		0.02352		0.00588	
ANDE=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDE<>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDE>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDE<=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDE<	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ANDE>=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORE=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORE<>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORE>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORE<=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORE<	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	
ORE>=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	0.18816		0.02352		0.00588	

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
LDED=	導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
LDED<>	導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
LDED>	導通時/非導通時	3.300	5.900	2.300	4.900	1.800	4.400
LDED<=	導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
LDED<	導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
LDED>=	導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ANDED=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ANDED<>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ANDED>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ANDED<=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ANDED<	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ANDED>=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ORED=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ORED<>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ORED>	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ORED<=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ORED<	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ORED>=	無執行時	0.18816		0.02352		0.00588	
	執行時：導通時/非導通時	3.400	6.000	2.400	5.000	1.900	4.500
ECMP	—	4.400	7.400	3.400	6.400	2.900	5.900
EDCMP	—	4.600	8.900	3.600	7.900	3.100	7.400
EZCP	—	4.800	8.100	3.800	7.100	3.300	6.600
EDZCP	—	5.200	10.100	4.200	9.100	3.700	8.600
E+ (s) (d)	(s)=0、(d)=0	0.31360		0.03920		0.0098	
	(s)=2 ¹²⁷ 、(d)=2 ¹²⁷	0.31360		0.03920		0.0098	
E+ (s1) (s2) (d)	(s1)=0、(s2)=0	0.31360		0.03920		0.0098	
	(s1)=2 ¹²⁷ 、(s2)=2 ¹²⁷	0.31360		0.03920		0.0098	
E- (s) (d)	(s)=0、(d)=0	0.31360		0.03920		0.0098	
	(s)=2 ¹²⁷ 、(d)=2 ¹²⁷	0.31360		0.03920		0.0098	
E- (s1) (s2) (d)	(s1)=0、(s2)=0	0.31360		0.03920		0.0098	
	(s1)=2 ¹²⁷ 、(s2)=2 ¹²⁷	0.31360		0.03920		0.0098	
ED+ (s) (d)	(s)=0、(d)=0	3.100	7.300	2.100	6.300	1.600	5.800
	(s)=2 ¹⁰²³ 、(d)=2 ¹⁰²³	3.500	8.500	2.500	7.500	2.000	7.000
ED+ (s1) (s2) (d)	(s1)=0、(s2)=0	3.500	7.200	2.500	6.200	2.000	5.700
	(s1)=2 ¹⁰²³ 、(s2)=2 ¹⁰²³	3.700	8.500	2.700	7.500	2.200	7.000
ED- (s) (d)	(s)=0、(d)=0	3.300	7.000	2.300	6.000	1.800	5.500
	(s)=2 ¹⁰²³ 、(d)=2 ¹⁰²³	3.400	7.200	2.400	6.200	1.900	5.700
ED- (s1) (s2) (d)	(s1)=0、(s2)=0	3.500	7.000	2.500	6.000	2.000	5.500
	(s1)=2 ¹⁰²³ 、(s2)=2 ¹⁰²³	3.600	7.400	2.600	6.400	2.100	5.900

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
E*	(s1)=0、(s2)=0	0.31360		0.03920		0.0098	
	(s1)=2 ¹²⁷ 、(s2)=2 ¹²⁷	0.31360		0.03920		0.0098	
E/	(s1)=2 ¹²⁷ 、(s2)=2 ¹²⁷	18.18880		2.27360		0.5684	
ED*	(s1)=0、(s2)=0	3.500	7.400	2.500	6.400	2.000	5.900
	(s1)=2 ¹⁰²³ 、(s2)=2 ¹⁰²³	3.800	8.800	2.800	7.800	2.300	7.300
ED/	(s1)=2 ¹⁰²³ 、(s2)=2 ¹⁰²³	3.800	8.800	2.800	7.800	2.300	7.300
INT2FLT	(s)=0	0.21952		0.02744		0.00686	
	(s)=7FFFH	0.21952		0.02744		0.00686	
UINT2FLT	(s)=0	0.21952		0.02744		0.00686	
	(s)=FFFFH	0.21952		0.02744		0.00686	
DINT2FLT	(s)=0	0.21952		0.02744		0.00686	
	(s)=7FFFFFFFH	0.21952		0.02744		0.00686	
UDINT2FLT	(s)=0	0.21952		0.02744		0.00686	
	(s)=FFFFFFFFH	0.21952		0.02744		0.00686	
DBL2FLT	—	3.100	5.500	2.100	4.500	1.600	4.000
INT2DBL	(s)=0	2.900	4.600	1.900	3.600	1.400	3.100
	(s)=7FFFH	2.900	4.600	1.900	3.600	1.400	3.100
UINT2DBL	(s)=0	2.900	4.800	1.900	3.800	1.400	3.300
	(s)=FFFFH	2.900	4.700	1.900	3.700	1.400	3.200
DINT2DBL	(s)=0	2.900	4.600	1.900	3.600	1.400	3.100
	(s)=7FFFFFFFH	2.900	4.600	1.900	3.600	1.400	3.100
UDINT2DBL	(s)=0	2.900	4.700	1.900	3.700	1.400	3.200
	(s)=FFFFFFFFH	2.900	4.700	1.900	3.700	1.400	3.200
FLT2DBL	—	2.900	6.100	1.900	5.100	1.400	4.600
EVAL	小數點形式全2位指定	5.200	13.800	4.200	12.800	3.700	12.300
	位數形式全6位指定	5.600	13.500	4.600	12.500	4.100	12.000
EREXP	—	4.600	10.900	3.600	9.900	3.100	9.400
ENEG	(d)=0	2.700	3.700	1.700	2.700	1.200	2.200
	(d)=-1.0	2.800	4.400	1.800	3.400	1.300	2.900
EDNEG	(d)=0	2.800	5.800	1.800	4.800	1.300	4.300
	(d)=-1.0	2.600	5.900	1.600	4.900	1.100	4.400
EMOV	—	0.06272		0.00784		0.00196	
EDMOV	—	0.06272		0.00784		0.00196	
SIN	—	2.900	5.800	1.900	4.800	1.400	4.300
COS	—	2.900	5.700	1.900	4.700	1.400	4.200
TAN	—	2.900	5.700	1.900	4.700	1.400	4.200
ASIN	—	2.900	6.000	1.900	5.000	1.400	4.500
ACOS	—	2.900	6.000	1.900	5.000	1.400	4.500
ATAN	—	2.900	4.900	1.900	3.900	1.400	3.400
SIND	—	3.900	14.200	2.900	13.200	2.400	12.700
COSD	—	3.800	14.200	2.800	13.200	2.300	12.700
TAND	—	4.300	15.400	3.300	14.400	2.800	13.900
ASIND	—	3.900	12.000	2.900	11.000	2.400	10.500
ACOSD	—	3.700	11.200	2.700	10.200	2.200	9.700
ATAND	—	3.500	10.700	2.500	9.700	2.000	9.200
BSIN	—	4.100	10.800	3.100	9.800	2.600	9.300
BCOS	—	4.100	10.400	3.100	9.400	2.600	8.900
BTAN	—	4.200	11.100	3.200	10.100	2.700	9.600
BASIN	—	3.900	8.800	2.900	7.800	2.400	7.300
BACOS	—	4.000	8.900	3.000	7.900	2.500	7.400
BATAN	—	3.900	8.900	2.900	7.900	2.400	7.400

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
RAD	—	2.900	4.200	1.900	3.200	1.400	2.700
DEG	—	2.900	4.400	1.900	3.400	1.400	2.900
RADD	—	3.100	8.800	2.100	7.800	1.600	7.300
DEGD	—	3.100	8.600	2.100	7.600	1.600	7.100
ESQRT	—	2.600	4.400	1.600	3.400	1.100	2.900
EDSQRT	—	3.200	8.700	2.200	7.700	1.700	7.200
EXP	(s)=-10	2.900	6.000	1.900	5.000	1.400	4.500
	(s)=1	3.000	6.100	2.000	5.100	1.500	4.600
EXPD	(s)=-10	3.700	12.000	2.700	11.000	2.200	10.500
	(s)=1	3.600	11.800	2.600	10.800	2.100	10.300
LOG	(s)=1	2.800	5.600	1.800	4.600	1.300	4.100
	(s)=10	2.900	6.200	1.900	5.200	1.400	4.700
LOGD	(s)=1	3.300	10.100	2.300	9.100	1.800	8.600
	(s)=10	3.800	12.400	2.800	11.400	2.300	10.900
BSQRT	(s)=0	3.000	4.600	2.000	3.600	1.500	3.100
	(s)=9999	3.700	8.400	2.700	7.400	2.200	6.900
BDSQRT	(s)=0	2.900	4.100	1.900	3.100	1.400	2.600
	(s)=99999999	3.600	7.500	2.600	6.500	2.100	6.000
POW	(s1)=1.23E+5、(s2)=3.45E+0	4.500	10.400	3.500	9.400	3.000	8.900
POWD	(s1)=1.23E+5、(s2)=3.45E+0	5.900	20.800	4.900	19.800	4.400	19.300
LOG10	(s)=1.23E+20	3.000	6.200	2.000	5.200	1.500	4.700
LOG10D	(s)=1.23E+20	3.900	14.400	2.900	13.400	2.400	12.900
EMAX	(n)=1	4.000	6.600	3.000	5.600	2.500	5.100
	(n)=96	16.200	19.400	15.200	18.400	14.700	17.900
EDMAX	(n)=1	4.200	8.400	3.200	7.400	2.700	6.900
	(n)=96	29.200	34.000	28.200	33.000	27.700	32.500
EMIN	(n)=1	4.000	6.600	3.000	5.600	2.500	5.100
	(n)=96	16.200	19.400	15.200	18.400	14.700	17.900
EDMIN	(n)=1	4.200	8.300	3.200	7.300	2.700	6.800
	(n)=96	28.700	33.900	27.700	32.900	27.200	32.400
RND	—	2.300	3.200	1.300	2.200	0.800	1.700
SRND	—	2.600	3.300	1.600	2.300	1.100	1.800
ZPUSH(d)	—	3.300	3.700	1.200	1.900	0.900	1.700
ZPUSH(s)(d)	Z、LZ全範圍	6.300	6.900	2.000	3.700	1.600	3.200
	Z全範圍(預設設定: 20點)	6.300	6.900	2.000	3.500	1.600	3.100
	LZ全範圍(預設設定: 2點)	5.100	5.800	2.000	3.600	1.300	3.100
ZPOP(d)	—	3.300	3.800	1.800	2.700	0.900	1.800
ZPOP(s)(d)	Z、LZ全範圍	5.800	6.700	1.900	3.700	1.500	3.200
	Z全範圍(預設設定: 20點)	5.800	6.600	1.900	3.800	1.500	3.100
	LZ全範圍(預設設定: 2點)	4.900	5.700	1.800	3.800	1.300	3.100
LIMIT	—	3.700	4.300	1.600	2.800	1.000	1.800
LIMIT_U	—	3.700	4.300	1.600	2.800	1.000	1.800
DLIMIT	—	3.700	4.300	1.600	2.800	1.000	1.800
DLIMIT_U	—	3.700	4.300	1.600	2.800	1.000	1.800
BAND	—	3.300	4.600	2.300	3.600	1.800	3.100
BAND_U	—	3.300	4.600	2.300	3.600	1.800	3.100
DBAND	—	3.400	4.500	2.400	3.500	1.900	3.000
DBAND_U	—	3.400	4.400	2.400	3.400	1.900	2.900
ZONE	—	3.300	4.500	2.300	3.500	1.800	3.000
ZONE_U	—	3.300	4.500	2.300	3.500	1.800	3.000
DZONE	—	3.400	4.400	2.400	3.400	1.900	2.900

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
DZONE_U	—	3.400	4.500	2.400	3.500	1.900	3.000
SCL	SM755=ON、點No. 1<(s1)<點No. 2	4.800	10.400	3.800	9.400	3.300	8.900
	SM755=ON、點No. 9<(s1)<點No. 10	4.800	10.400	3.800	9.400	3.300	8.900
	SM755=OFF、點No. 1<(s1)<點No. 2	4.600	10.200	3.600	9.200	3.100	8.700
	SM755=OFF、點No. 9<(s1)<點No. 10	5.200	10.400	4.200	9.400	3.700	8.900
SCL_U	SM755=ON、點No. 1<(s1)<點No. 2	4.700	10.200	3.700	9.200	3.200	8.700
	SM755=ON、點No. 9<(s1)<點No. 10	4.700	10.300	3.700	9.300	3.200	8.800
	SM755=OFF、點No. 1<(s1)<點No. 2	4.500	10.000	3.500	9.000	3.000	8.500
	SM755=OFF、點No. 9<(s1)<點No. 10	4.900	10.400	3.900	9.400	3.400	8.900
DACL	SM755=ON、點No. 1<(s1)<點No. 2	4.800	11.400	3.800	10.400	3.300	9.900
	SM755=ON、點No. 9<(s1)<點No. 10	4.800	11.300	3.800	10.300	3.300	9.800
	SM755=OFF、點No. 1<(s1)<點No. 2	4.500	10.800	3.500	9.800	3.000	9.300
	SM755=OFF、點No. 9<(s1)<點No. 10	5.100	11.200	4.100	10.200	3.600	9.700
DACL_U	SM755=ON、點No. 1<(s1)<點No. 2	4.700	11.700	3.700	10.700	3.200	10.200
	SM755=ON、點No. 9<(s1)<點No. 10	4.700	11.700	3.700	10.700	3.200	10.200
	SM755=OFF、點No. 1<(s1)<點No. 2	4.400	11.000	3.400	10.000	2.900	9.500
	SM755=OFF、點No. 9<(s1)<點No. 10	5.000	11.400	4.000	10.400	3.500	9.900
SCL2	SM755=ON、點No. 1<(s1)<點No. 2	4.800	10.700	3.800	9.700	3.300	9.200
	SM755=ON、點No. 9<(s1)<點No. 10	4.800	10.600	3.800	9.600	3.300	9.100
	SM755=OFF、點No. 1<(s1)<點No. 2	4.800	10.200	3.800	9.200	3.300	8.700
	SM755=OFF、點No. 9<(s1)<點No. 10	5.000	10.300	4.000	9.300	3.500	8.800
SCL2_U	SM755=ON、點No. 1<(s1)<點No. 2	4.700	10.300	3.700	9.300	3.200	8.800
	SM755=ON、點No. 9<(s1)<點No. 10	4.800	10.400	3.800	9.400	3.300	8.900
	SM755=OFF、點No. 1<(s1)<點No. 2	4.400	10.000	3.400	9.000	2.900	8.500
	SM755=OFF、點No. 9<(s1)<點No. 10	5.000	10.500	4.000	9.500	3.500	9.000
DACL2	SM755=ON、點No. 1<(s1)<點No. 2	4.800	11.500	3.800	10.500	3.300	10.000
	SM755=ON、點No. 9<(s1)<點No. 10	4.800	11.500	3.800	10.500	3.300	10.000
	SM755=OFF、點No. 1<(s1)<點No. 2	4.500	11.000	3.500	10.000	3.000	9.500
	SM755=OFF、點No. 9<(s1)<點No. 10	4.800	11.600	3.800	10.600	3.300	10.100
DACL2_U	SM755=ON、點No. 1<(s1)<點No. 2	4.700	11.600	3.700	10.600	3.200	10.100
	SM755=ON、點No. 9<(s1)<點No. 10	4.800	11.600	3.800	10.600	3.300	10.100
	SM755=OFF、點No. 1<(s1)<點No. 2	4.400	11.100	3.400	10.100	2.900	9.600
	SM755=OFF、點No. 9<(s1)<點No. 10	5.000	11.500	4.000	10.500	3.500	10.000
UDCNT1	—	2.300	3.100	1.300	2.100	0.800	1.600
UDCNT2	—	2.300	3.100	1.300	2.100	0.800	1.600
TTMR	—	3.500	7.400	2.500	6.400	2.000	5.900
STMR	—	4.200	9.600	3.200	8.600	2.700	8.100
ROTC	—	6.300	10.100	5.300	9.100	4.800	8.600
RAMPQ	—	4.900	9.700	3.900	8.700	3.400	8.200
SPD	—	2.200	3.100	1.200	2.100	0.700	1.600
PLSY	—	2.300	2.900	1.300	1.900	0.800	1.400
PWM	—	2.300	2.900	1.300	1.900	0.800	1.400
MTR	—	4.900	13.100	3.900	12.100	3.400	11.600
CCD	SM772=OFF、(n)=1	4.800	8.600	3.800	7.600	3.300	7.100
	SM772=OFF、(n)=96	13.200	17.000	12.200	16.000	11.700	15.500
	SM772=ON、(n)=1	4.800	8.600	3.800	7.600	3.300	7.100
	SM772=ON、(n)=96	13.200	17.000	12.200	16.000	11.700	15.500
SERDATA	(n)=1: 全部一致	4.200	8.200	3.200	7.200	2.700	6.700
	(n)=1: 全部不一致	4.200	8.100	3.200	7.100	2.700	6.600
	(n)=96: 全部一致	11.200	16.500	10.200	15.500	9.700	15.000
	(n)=96: 全部不一致	11.200	16.600	10.200	15.600	9.700	15.100

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
DSERDATA	(n)=1; 全部一致	4.500	8.800	3.500	7.800	3.000	7.300
	(n)=1; 全部不一致	4.500	8.800	3.500	7.800	3.000	7.300
	(n)=96; 全部一致	16.000	20.400	15.000	19.400	14.500	18.900
	(n)=96; 全部不一致	16.000	20.400	15.000	19.400	14.500	18.900
SERMM	(n)=1、無相同資料	5.600	10.800	4.600	9.800	4.100	9.300
	(n)=1、相同資料的個數=1	5.600	10.800	4.600	9.800	4.100	9.300
	(n)=96、無相同資料	14.600	19.700	13.600	18.700	13.100	18.200
	(n)=96、相同資料的個數=96	14.600	19.700	13.600	18.700	13.100	18.200
DSERMM	(n)=1、無相同資料	5.600	10.800	4.600	9.800	4.100	9.300
	(n)=1、相同資料的個數=1	5.600	10.800	4.600	9.800	4.100	9.300
	(n)=96、無相同資料	14.600	19.700	13.600	18.700	13.100	18.200
	(n)=96、相同資料的個數=96	14.600	19.700	13.600	18.700	13.100	18.200
SUM	(s)=0	5.000	5.500	1.600	2.200	1.300	1.900
	(s)=FFFFH	5.000	5.500	1.600	2.200	1.300	1.900
DSUM	(s)=0	7.600	8.000	2.000	2.700	1.700	2.400
	(s)=FFFFFFFFH	7.500	7.900	2.000	2.700	1.700	2.400
BON	(s)=0	3.900	6.400	2.900	5.400	2.400	4.900
	(s)=FFFFH	3.900	6.400	2.900	5.400	2.400	4.900
DBON	(s)=0	3.900	6.400	2.900	5.400	2.400	4.900
	(s)=FFFFFFFFH	3.900	6.400	2.900	5.400	2.400	4.900
MAX	(n)=1	3.900	6.500	2.900	5.500	2.400	5.000
	(n)=96	11.000	13.800	10.000	12.800	9.500	12.300
MAX_U	(n)=1	4.000	7.600	3.000	6.600	2.500	6.100
	(n)=96	12.200	16.200	11.200	15.200	10.700	14.700
DMAX	(n)=1	4.000	6.800	3.000	5.800	2.500	5.300
	(n)=96	18.900	21.900	17.900	20.900	17.400	20.400
DMAX_U	(n)=1	4.000	7.300	3.000	6.300	2.500	5.800
	(n)=96	11.800	15.600	10.800	14.600	10.300	14.100
MIN	(n)=1	3.900	6.500	2.900	5.500	2.400	5.000
	(n)=96	11.000	13.900	10.000	12.900	9.500	12.400
MIN_U	(n)=1	4.000	7.400	3.000	6.400	2.500	5.900
	(n)=96	12.200	15.900	11.200	14.900	10.700	14.400
DMIN	(n)=1	4.000	6.900	3.000	5.900	2.500	5.400
	(n)=96	19.000	22.000	18.000	21.000	17.500	20.500
DMIN_U	(n)=1	3.900	7.400	2.900	6.400	2.400	5.900
	(n)=96	11.800	15.500	10.800	14.500	10.300	14.000
SORTD	(n)=1、(s1)=1	4.600	7.100	3.600	6.100	3.100	5.600
	(n)=96、(s1)=16	9.900	13.400	8.900	12.400	8.400	11.900
SORTD_U	(n)=1、(s1)=1	4.600	7.100	3.600	6.100	3.100	5.600
	(n)=96、(s1)=16	10.300	14.400	9.300	13.400	8.800	12.900
DSORTD	(n)=1、(s1)=1	4.500	7.000	3.500	6.000	3.000	5.500
	(n)=96、(s1)=16	11.400	15.000	10.400	14.000	9.900	13.500
DSORTD_U	(n)=1、(s1)=1	4.600	7.300	3.600	6.300	3.100	5.800
	(n)=96、(s1)=16	12.600	17.300	11.600	16.300	11.100	15.800
SORTTBL	(n1)=1、(n2)=1	13.100	15.500	12.100	14.500	11.600	14.000
	(n1)=32、(n2)=6	13.100	15.500	12.100	14.500	11.600	14.000
SORTTBL_U	(n1)=1、(n2)=1	13.100	15.500	12.100	14.500	11.600	14.000
	(n1)=32、(n2)=6	13.100	15.500	12.100	14.500	11.600	14.000
SORTTBL2	(n1)=1、(n2)=1	13.100	15.500	12.100	14.500	11.600	14.000
	(n1)=32、(n2)=6	13.100	15.500	12.100	14.500	11.600	14.000

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
SORTTBL2_U	(n1)=1、(n2)=1	13.100	15.500	12.100	14.500	11.600	14.000
	(n1)=32、(n2)=6	13.100	15.500	12.100	14.500	11.600	14.000
DSORTTBL2	(n1)=1、(n2)=1	13.100	15.500	12.100	14.500	11.600	14.000
	(n1)=32、(n2)=6	13.100	15.500	12.100	14.500	11.600	14.000
DSORTTBL2_U	(n1)=1、(n2)=1	13.100	15.500	12.100	14.500	11.600	14.000
	(n1)=32、(n2)=6	13.100	15.500	12.100	14.500	11.600	14.000
WSUM	(n)=1	3.700	4.100	1.300	2.200	0.900	1.800
	(n)=96	24.500	25.000	6.000	6.700	4.700	5.600
WSUM_U	(n)=1	3.600	4.000	1.200	2.200	0.900	1.800
	(n)=96	16.100	16.600	5.500	6.300	4.700	5.600
DWSUM	(n)=1	3.800	8.500	2.800	7.500	2.300	7.000
	(n)=96	10.900	15.600	9.900	14.600	9.400	14.100
DWSUM_U	(n)=1	3.900	9.300	2.900	8.300	2.400	7.800
	(n)=96	11.000	16.400	10.000	15.400	9.500	14.900
MEAN	(n)=1	3.400	6.400	2.400	5.400	1.900	4.900
	(n)=96	6.600	10.300	5.600	9.300	5.100	8.800
MEAN_U	(n)=1	3.400	6.300	2.400	5.300	1.900	4.800
	(n)=96	6.200	10.200	5.200	9.200	4.700	8.700
DMEAN	(n)=1	4.100	9.300	3.100	8.300	2.600	7.800
	(n)=96	10.400	15.700	9.400	14.700	8.900	14.200
DMEAN_U	(n)=1	3.700	9.000	2.700	8.000	2.200	7.500
	(n)=96	9.800	14.900	8.800	13.900	8.300	13.400
SQRT	—	3.300	4.800	2.300	3.800	1.800	3.300
DSQRT	—	3.300	4.800	2.300	3.800	1.800	3.300
CRC	SM772=OFF、(n)=1	4.700	9.400	3.700	8.400	3.200	7.900
	SM772=OFF、(n)=96	13.700	20.300	12.700	19.300	12.200	18.800
	SM772=ON、(n)=1	4.700	9.400	3.700	8.400	3.200	7.900
	SM772=ON、(n)=96	13.700	20.300	12.700	19.300	12.200	18.800
DBOPEN	執行時(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	18.300	20.900
	執行時(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	19.300	24.500
DBCLOSE	執行時(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	11.500	13.100
	執行時(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	13.300	18.300
DBINSERT	(s3)=1(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	25.400	28.900
	(s3)=16(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	46.400	50.100
	(s3)=1(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	23.300	29.100
	(s3)=16(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	40.200	45.800
DBUPDATE	(s3)=1(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	31.700	34.700
	(s3)=16(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	49.300	52.900
	(s3)=1(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	29.900	35.900
	(s3)=16(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	46.900	53.200

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
DBSELECT	(s3)=1(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	32.100	35.300
	(s3)=16(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	51.600	55.000
	(s3)=1(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	32.800	38.300
	(s3)=16(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	48.900	55.200
DBDELETE	(s3)=1(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	26.000	29.900
	(s3)=2(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	27.400	31.200
	(s3)=1(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	24.800	30.900
	(s3)=2(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	26.600	32.500
DBIMPORT	執行時(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	16.200	19.200
	執行時(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	17.800	22.400
DBEXPORT	執行時(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	16.000	19.000
	執行時(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	17.300	22.500
DBTRANS	執行時(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	11.400	13.000
	執行時(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	12.800	18.500
DBCMMIT	執行時(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	11.600	13.000
	執行時(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	12.800	18.500
DBROLBAK	執行時(韌體版本低於“28”的可程式控制器CPU)	—	—	—	—	11.500	12.900
	執行時(韌體版本為“28”及其以後的可程式控制器CPU)	—	—	—	—	12.800	18.400
RSET	—	3.300	7.800	2.300	6.800	1.800	6.300
QDRSET	—	66.600	93.600	65.600	92.600	65.100	92.100
ZRRDB	—	3.100	4.400	2.100	3.400	1.600	2.900
ZRWRB	—	3.200	4.700	2.200	3.700	1.700	3.200
ADRSET	—	2.600	3.500	1.600	2.500	1.100	2.000
DATERD	—	4.900	12.200	3.900	11.200	3.400	10.700
DATEWR	—	13.700	38.900	12.700	37.900	12.200	37.400
DATE+	無進位	4.800	8.300	3.800	7.300	3.300	6.800
	有進位	4.900	8.000	3.900	7.000	3.400	6.500
DATE-	無進位	4.900	8.000	3.900	7.000	3.400	6.500
	有進位	4.900	8.100	3.900	7.100	3.400	6.600
TIME2SEC	—	3.400	5.200	2.400	4.200	1.900	3.700
SEC2TIME	—	3.300	5.400	2.300	4.400	1.800	3.900
DATE2SEC	(s)+0=2067、(s)+1=12、(s)+2=31、(s)+3=23、(s)+4=59、(s)+5=59	4.700	8.600	3.700	7.600	3.300	7.400
DATE2SEC_U	(s)+0=2099、(s)+1=12、(s)+2=31、(s)+3=23、(s)+4=59、(s)+5=59	4.700	8.600	3.700	7.600	3.300	7.400
SEC2DATE	(s)=2145916799	4.700	12.200	3.700	11.200	3.100	10.800
SEC2DATE_U	(s)=3155759999	4.700	12.200	3.700	11.200	3.100	10.800

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
LDDT=	與指定的日期比較：導通時	4.400	11.000	3.400	10.000	2.900	9.500
	與指定的日期比較：非導通時	4.400	10.900	3.400	9.900	2.900	9.400
	與當前的日期比較：導通時	6.200	16.500	5.200	15.500	4.700	15.000
	與當前的日期比較：非導通時	6.300	16.600	5.300	15.600	4.800	15.100
LDDT<>	與指定的日期比較：導通時	4.100	10.800	3.100	9.800	2.600	9.300
	與指定的日期比較：非導通時	4.100	10.900	3.100	9.900	2.600	9.400
	與當前的日期比較：導通時	6.100	17.100	5.100	16.100	4.600	15.600
	與當前的日期比較：非導通時	6.300	17.000	5.300	16.000	4.800	15.500
LDDT>	與指定的日期比較：導通時	4.200	11.000	3.200	10.000	2.700	9.500
	與指定的日期比較：非導通時	4.200	11.200	3.200	10.200	2.700	9.700
	與當前的日期比較：導通時	6.100	17.000	5.100	16.000	4.600	15.500
	與當前的日期比較：非導通時	6.100	16.800	5.100	15.800	4.600	15.300
LDDT<=	與指定的日期比較：導通時	4.200	11.400	3.200	10.400	2.700	9.900
	與指定的日期比較：非導通時	4.300	11.300	3.300	10.300	2.800	9.800
	與當前的日期比較：導通時	6.300	16.100	5.300	15.100	4.800	14.600
	與當前的日期比較：非導通時	6.400	15.900	5.400	14.900	4.900	14.400
LDDT<	與指定的日期比較：導通時	4.300	11.400	3.300	10.400	2.800	9.900
	與指定的日期比較：非導通時	4.200	11.400	3.200	10.400	2.700	9.900
	與當前的日期比較：導通時	6.300	16.500	5.300	15.500	4.800	15.000
	與當前的日期比較：非導通時	6.300	16.400	5.300	15.400	4.800	14.900
LDDT>=	與指定的日期比較：導通時	4.200	11.300	3.200	10.300	2.700	9.800
	與指定的日期比較：非導通時	4.300	11.300	3.300	10.300	2.800	9.800
	與當前的日期比較：導通時	6.500	16.000	5.500	15.000	5.000	14.500
	與當前的日期比較：非導通時	6.400	16.200	5.400	15.200	4.900	14.700
ANDDT=	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	3.900	11.500	2.900	10.500	2.400	10.000
	與指定的日期比較：非導通時	4.000	11.600	3.000	10.600	2.500	10.100
	與當前的日期比較：導通時	6.100	17.400	5.100	16.400	4.600	15.900
	與當前的日期比較：非導通時	6.000	17.400	5.000	16.400	4.500	15.900
ANDDT<>	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.400	10.900	3.400	9.900	2.900	9.400
	與指定的日期比較：非導通時	4.400	10.900	3.400	9.900	2.900	9.400
	與當前的日期比較：導通時	6.300	17.300	5.300	16.300	4.800	15.800
	與當前的日期比較：非導通時	6.400	17.300	5.400	16.300	4.900	15.800
ANDDT>	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.200	11.200	3.200	10.200	2.700	9.700
	與指定的日期比較：非導通時	4.200	11.500	3.200	10.500	2.700	10.000
	與當前的日期比較：導通時	6.200	17.100	5.200	16.100	4.700	15.600
	與當前的日期比較：非導通時	6.200	17.100	5.200	16.100	4.700	15.600
ANDDT<=	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.200	11.300	3.200	10.300	2.700	9.800
	與指定的日期比較：非導通時	4.300	11.400	3.300	10.400	2.800	9.900
	與當前的日期比較：導通時	6.400	16.100	5.400	15.100	4.900	14.600
	與當前的日期比較：非導通時	6.400	16.000	5.400	15.000	4.900	14.500
ANDDT<	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.100	10.900	3.100	9.900	2.600	9.400
	與指定的日期比較：非導通時	4.100	11.100	3.100	10.100	2.600	9.600
	與當前的日期比較：導通時	6.300	17.700	5.300	16.700	4.800	16.200
	與當前的日期比較：非導通時	6.400	16.700	5.400	15.700	4.900	15.200

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
ANDDT>=	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.200	11.400	3.200	10.400	2.700	9.900
	與指定的日期比較：非導通時	4.200	11.200	3.200	10.200	2.700	9.700
	與當前的日期比較：導通時	6.400	16.100	5.400	15.100	4.900	14.600
	與當前的日期比較：非導通時	6.400	16.400	5.400	15.400	4.900	14.900
ORDT=	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.200	11.300	3.200	10.300	2.700	9.800
	與指定的日期比較：非導通時	4.100	11.300	3.100	10.300	2.600	9.800
	與當前的日期比較：導通時	6.300	17.100	5.300	16.100	4.800	15.600
	與當前的日期比較：非導通時	6.300	16.900	5.300	15.900	4.800	15.400
ORDT<>	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.300	11.500	3.300	10.500	2.800	10.000
	與指定的日期比較：非導通時	4.200	11.600	3.200	10.600	2.700	10.100
	與當前的日期比較：導通時	6.200	17.400	5.200	16.400	4.700	15.900
	與當前的日期比較：非導通時	6.300	17.100	5.300	16.100	4.800	15.600
ORDT>	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.300	11.500	3.300	10.500	2.800	10.000
	與指定的日期比較：非導通時	4.200	11.500	3.200	10.500	2.700	10.000
	與當前的日期比較：導通時	6.300	17.100	5.300	16.100	4.800	15.600
	與當前的日期比較：非導通時	6.300	17.200	5.300	16.200	4.800	15.700
ORDT<=	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.200	11.300	3.200	10.300	2.700	9.800
	與指定的日期比較：非導通時	4.100	11.500	3.100	10.500	2.600	10.000
	與當前的日期比較：導通時	6.400	16.500	5.400	15.500	4.900	15.000
	與當前的日期比較：非導通時	6.400	16.500	5.400	15.500	4.900	15.000
ORDT<	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.300	11.700	3.300	10.700	2.800	10.200
	與指定的日期比較：非導通時	4.200	11.600	3.200	10.600	2.700	10.100
	與當前的日期比較：導通時	6.300	17.000	5.300	16.000	4.800	15.500
	與當前的日期比較：非導通時	6.400	16.900	5.400	15.900	4.900	15.400
ORDT>=	無執行時	0.18816		0.02352		0.00588	
	與指定的日期比較：導通時	4.300	11.000	3.300	10.000	2.800	9.500
	與指定的日期比較：非導通時	4.300	11.100	3.300	10.100	2.800	9.600
	與當前的日期比較：導通時	6.200	16.000	5.200	15.000	4.700	14.500
	與當前的日期比較：非導通時	6.200	16.400	5.200	15.400	4.700	14.900
LDTM=	與指定的時間比較：導通時	4.400	11.000	3.400	10.000	2.900	9.500
	與指定的時間比較：非導通時	4.400	11.000	3.400	10.000	2.900	9.500
	與當前的時間比較：導通時	6.200	17.100	5.200	16.100	4.700	15.600
	與當前的時間比較：非導通時	6.200	16.400	5.200	15.400	4.700	14.900
LDTM<>	與指定的時間比較：導通時	4.100	10.700	3.100	9.700	2.600	9.200
	與指定的時間比較：非導通時	4.100	10.800	3.100	9.800	2.600	9.300
	與當前的時間比較：導通時	6.400	16.000	5.400	15.000	4.900	14.500
	與當前的時間比較：非導通時	6.300	16.000	5.300	15.000	4.800	14.500
LDTM>	與指定的時間比較：導通時	4.100	10.700	3.100	9.700	2.600	9.200
	與指定的時間比較：非導通時	4.100	10.800	3.100	9.800	2.600	9.300
	與當前的時間比較：導通時	6.300	16.100	5.300	15.100	4.800	14.600
	與當前的時間比較：非導通時	6.400	16.200	5.400	15.200	4.900	14.700
LDTM<=	與指定的時間比較：導通時	4.100	10.700	3.100	9.700	2.600	9.200
	與指定的時間比較：非導通時	4.200	10.700	3.200	9.700	2.700	9.200
	與當前的時間比較：導通時	6.400	16.000	5.400	15.000	4.900	14.500
	與當前的時間比較：非導通時	6.400	16.100	5.400	15.100	4.900	14.600

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
LDTM<	與指定的時間比較：導通時	4.100	10.800	3.100	9.800	2.600	9.300
	與指定的時間比較：非導通時	4.100	10.800	3.100	9.800	2.600	9.300
	與當前的時間比較：導通時	6.200	16.300	5.200	15.300	4.700	14.800
	與當前的時間比較：非導通時	6.400	16.200	5.400	15.200	4.900	14.700
LDTM>=	與指定的時間比較：導通時	4.000	10.600	3.000	9.600	2.500	9.100
	與指定的時間比較：非導通時	4.100	10.700	3.100	9.700	2.600	9.200
	與當前的時間比較：導通時	6.400	16.000	5.400	15.000	4.900	14.500
	與當前的時間比較：非導通時	6.300	16.100	5.300	15.100	4.800	14.600
ANDTM=	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	3.900	10.600	2.900	9.600	2.400	9.100
	與指定的時間比較：非導通時	4.000	10.700	3.000	9.700	2.500	9.200
	與當前的時間比較：導通時	6.400	16.100	5.400	15.100	4.900	14.600
	與當前的時間比較：非導通時	6.400	16.200	5.400	15.200	4.900	14.700
ANDTM<>	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	3.900	10.600	2.900	9.600	2.400	9.100
	與指定的時間比較：非導通時	4.100	10.900	3.100	9.900	2.600	9.400
	與當前的時間比較：導通時	6.400	16.300	5.400	15.300	4.900	14.800
	與當前的時間比較：非導通時	6.400	16.300	5.400	15.300	4.900	14.800
ANDTM>	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	4.100	10.900	3.100	9.900	2.600	9.400
	與指定的時間比較：非導通時	4.100	11.000	3.100	10.000	2.600	9.500
	與當前的時間比較：導通時	6.300	16.200	5.300	15.200	4.800	14.700
	與當前的時間比較：非導通時	6.400	16.300	5.400	15.300	4.900	14.800
ANDTM<=	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	4.100	10.900	3.100	9.900	2.600	9.400
	與指定的時間比較：非導通時	4.100	10.900	3.100	9.900	2.600	9.400
	與當前的時間比較：導通時	6.400	16.500	5.400	15.500	4.900	15.000
	與當前的時間比較：非導通時	6.400	16.400	5.400	15.400	4.900	14.900
ANDTM<	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	4.100	11.000	3.100	10.000	2.600	9.500
	與指定的時間比較：非導通時	4.100	11.100	3.100	10.100	2.600	9.600
	與當前的時間比較：導通時	6.400	16.200	5.400	15.200	4.900	14.700
	與當前的時間比較：非導通時	6.400	16.300	5.400	15.300	4.900	14.800
ANDTM>=	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	4.100	10.900	3.100	9.900	2.600	9.400
	與指定的時間比較：非導通時	4.100	10.900	3.100	9.900	2.600	9.400
	與當前的時間比較：導通時	6.400	16.400	5.400	15.400	4.900	14.900
	與當前的時間比較：非導通時	6.300	16.400	5.300	15.400	4.800	14.900
ORTM=	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	4.200	10.700	3.200	9.700	2.700	9.200
	與指定的時間比較：非導通時	4.200	10.600	3.200	9.600	2.700	9.100
	與當前的時間比較：導通時	6.500	16.400	5.500	15.400	5.000	14.900
	與當前的時間比較：非導通時	6.500	16.300	5.500	15.300	5.000	14.800
ORTM<>	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	4.200	10.700	3.200	9.700	2.700	9.200
	與指定的時間比較：非導通時	4.200	10.700	3.200	9.700	2.700	9.200
	與當前的時間比較：導通時	6.500	16.500	5.500	15.500	5.000	15.000
	與當前的時間比較：非導通時	6.500	16.500	5.500	15.500	5.000	15.000

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
ORTM>	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	4.200	10.800	3.200	9.800	2.700	9.300
	與指定的時間比較：非導通時	4.300	10.800	3.300	9.800	2.800	9.300
	與當前的時間比較：導通時	6.600	16.600	5.600	15.600	5.100	15.100
	與當前的時間比較：非導通時	6.600	16.600	5.600	15.600	5.100	15.100
ORTM<=	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	4.200	10.700	3.200	9.700	2.700	9.200
	與指定的時間比較：非導通時	4.100	10.800	3.100	9.800	2.600	9.300
	與當前的時間比較：導通時	6.500	16.400	5.500	15.400	5.000	14.900
	與當前的時間比較：非導通時	6.400	16.500	5.400	15.500	4.900	15.000
ORTM<	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	4.200	10.900	3.200	9.900	2.700	9.400
	與指定的時間比較：非導通時	4.200	10.900	3.200	9.900	2.700	9.400
	與當前的時間比較：導通時	6.500	16.300	5.500	15.300	5.000	14.800
	與當前的時間比較：非導通時	6.400	16.400	5.400	15.400	4.900	14.900
ORTM>=	無執行時	0.18816		0.02352		0.00588	
	與指定的時間比較：導通時	4.200	10.700	3.200	9.700	2.700	9.200
	與指定的時間比較：非導通時	4.000	10.700	3.000	9.700	2.500	9.200
	與當前的時間比較：導通時	6.400	16.400	5.400	15.400	4.900	14.900
	與當前的時間比較：非導通時	6.400	16.400	5.400	15.400	4.900	14.900
TCMP	—	5.700	9.300	4.700	8.300	4.200	7.800
TZCP	—	6.200	10.500	5.200	9.500	4.700	9.000
S. DATERD	—	5.000	13.100	4.000	12.100	3.500	11.600
S. DATE+	無進位	5.000	9.100	4.000	8.100	3.500	7.600
	有進位	5.100	9.100	4.100	8.100	3.600	7.600
S. DATE-	無借位	5.100	9.500	4.100	8.500	3.600	8.000
	有借位	5.000	9.300	4.000	8.300	3.500	7.800
DUTY	—	3.900	8.100	2.900	7.100	2.400	6.600
TIMCHK	—	3.700	6.300	2.700	5.300	2.200	4.800
HOURM	—	4.400	7.700	3.400	6.700	2.900	6.200
DHOURM	—	4.400	7.700	3.400	6.700	2.900	6.200
RFS (X)	(n)=1	7.000	17.000	6.000	16.000	5.500	15.500
	(n)=64	14.500	35.300	13.500	34.300	13.000	33.800
RFS (Y)	(n)=1	6.200	16.400	5.200	15.400	4.700	14.900
	(n)=64	10.500	30.800	9.500	29.800	9.000	29.300
COM	僅選擇I/O更新時	8.000	21.200	7.000	20.200	6.500	19.700
	僅選擇CC-Link IE控制網路更新時(管理站側)	17.900	47.200	16.900	46.200	16.400	45.700
	僅選擇CC-Link IE現場網路更新時(主站側)	16.100	41.500	15.100	40.500	14.600	40.000
	僅選擇MELSECNET/H網路更新時(管理站側)	20.200	51.000	19.200	50.000	18.700	49.500
	僅選擇CC-Link IE控制網路更新時(普通站側)	18.300	45.800	17.300	44.800	16.800	44.300
	僅選擇CC-Link IE現場網路更新時(本地站側)	16.500	40.100	15.500	39.100	15.000	38.600
	僅選擇MELSECNET/H網路更新時(普通站側)	20.600	49.600	19.600	48.600	19.100	48.100
	僅選擇CC-Link IE現場網路Basic更新時	9.200	21.900	8.200	20.900	7.7	20.4
	僅選擇智能功能模組時	8.300	13.000	7.300	12.000	6.800	11.500
	使用了多CPU系統的CPU緩衝記憶體更新(END時)	5.300	13.600	4.300	12.600	3.800	12.100
	多CPU系統的組外的輸入/輸出的獲取 輸入：64點+輸出：64點	5.300	14.200	4.300	13.200	3.800	12.700
	元件/標籤存取服務處理時 (與工程工具、GOT或其它外部設備的通訊)	8.600	25.100	7.600	24.100	7.100	23.600

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
S. ZCOM	僅選擇CC-Link IE TSN更新時(主站側)	30.800	64.200	29.800	63.200	29.300	62.700
	僅選擇CC-Link IE控制網路更新時(管理站側)	25.400	57.200	24.400	56.200	23.900	55.700
	僅選擇CC-Link IE現場網路更新時(主站側)	23.100	51.500	22.100	50.500	21.600	50.000
	僅選擇MELSECNET/H網路更新時(管理站側)	28.700	61.400	27.700	60.400	27.200	59.900
	僅選擇CC-Link IE TSN更新時(本地站側)	31.200	62.800	30.200	61.800	29.700	61.300
	僅選擇CC-Link IE控制網路更新時(普通站側)	25.800	55.800	24.800	54.800	24.300	54.300
	僅選擇CC-Link IE現場網路更新時(本地站側)	23.500	50.100	22.500	49.100	22.000	48.600
	僅選擇MELSECNET/H網路更新時(普通站側)	29.100	60.000	28.100	59.000	27.600	58.500
FROM	緩衝記憶體讀取、(n)=1	8.100	8.600	4.400	5.000	3.700	4.300
	緩衝記憶體讀取、(n)=1000	39.500	57.100	38.500	56.100	38.000	55.600
	本機的CPU緩衝記憶體讀取、(n)=1	4.200	4.600	1.600	2.000	1.100	1.500
	本機的CPU緩衝記憶體讀取、(n)=320	15.600	28.500	14.600	27.500	14.100	27.000
	其它機號的CPU緩衝記憶體讀取、(n)=1	8.000	8.500	4.400	5.000	3.700	4.300
	其它機號的CPU緩衝記憶體讀取、(n)=320	23.400	41.000	22.400	40.000	21.900	39.500
DFROM	緩衝記憶體讀取、(n)=1	8.100	8.600	4.500	6.200	4.000	5.500
	緩衝記憶體讀取、(n)=500	40.900	70.900	39.900	69.900	39.400	69.400
	本機的CPU緩衝記憶體讀取、(n)=1	4.100	4.700	1.500	3.200	1.200	2.800
	本機的CPU緩衝記憶體讀取、(n)=320	23.400	44.400	22.400	43.400	21.900	42.900
	其它機號的CPU緩衝記憶體讀取、(n)=1	8.000	8.700	4.300	6.200	3.800	5.500
	其它機號的CPU緩衝記憶體讀取、(n)=320	33.800	67.800	32.800	66.800	32.300	66.300
TO	緩衝記憶體寫入、(n)=1	6.600	7.100	2.600	3.200	2.100	2.700
	緩衝記憶體寫入、(n)=1000	46.500	67.000	45.500	66.000	45.000	65.500
	至本機的CPU緩衝記憶體的寫入、(n)=1	4.200	4.300	1.300	1.600	0.900	1.200
	至本機的CPU緩衝記憶體的寫入、(n)=320	12.700	28.100	11.700	27.100	11.200	26.600
DTO	緩衝記憶體寫入、(n)=1	6.800	7.400	2.800	4.500	2.400	4.000
	緩衝記憶體寫入、(n)=500	45.900	72.500	44.900	71.500	44.400	71.000
	至本機的CPU緩衝記憶體的寫入、(n)=1	4.200	4.600	1.300	2.900	1.000	2.600
	至本機的CPU緩衝記憶體的寫入、(n)=320	17.000	39.400	16.000	38.400	15.500	37.900
FROMD	緩衝記憶體讀取、(n)=1	8.400	8.900	4.500	6.400	4.000	5.700
	緩衝記憶體讀取、(n)=1000	42.500	65.100	41.500	64.100	41.000	63.600
	本機的CPU緩衝記憶體讀取、(n)=1	4.400	4.900	1.700	3.300	1.400	2.900
	本機的CPU緩衝記憶體讀取、(n)=320	17.000	36.200	16.000	35.200	15.500	34.700
	其它機號的CPU緩衝記憶體讀取、(n)=1	8.400	9.000	4.500	6.400	4.000	5.700
	其它機號的CPU緩衝記憶體讀取、(n)=320	26.000	49.100	25.000	48.100	24.500	47.600
DFROMD	緩衝記憶體讀取、(n)=1	8.300	8.900	4.500	6.400	4.000	5.700
	緩衝記憶體讀取、(n)=500	42.900	71.900	41.900	70.900	41.400	70.400
	本機的CPU緩衝記憶體讀取、(n)=1	4.100	4.700	1.500	3.200	1.200	2.800
	本機的CPU緩衝記憶體讀取、(n)=320	23.900	46.800	22.900	45.800	22.400	45.300
	其它機號的CPU緩衝記憶體讀取、(n)=1	8.200	8.800	4.400	6.300	3.900	5.600
	其它機號的CPU緩衝記憶體讀取、(n)=320	35.500	64.300	34.500	63.300	34.000	62.800
TOD	緩衝記憶體寫入、(n)=1	6.800	7.400	2.700	4.600	2.300	4.100
	緩衝記憶體寫入、(n)=1000	48.600	77.500	47.600	76.500	47.100	76.000
	至本機的CPU緩衝記憶體的寫入、(n)=1	4.300	4.800	1.500	3.000	1.200	2.700
	至本機的CPU緩衝記憶體的寫入、(n)=320	15.200	39.100	14.200	38.100	13.700	37.600
DTOD	緩衝記憶體寫入、(n)=1	6.800	7.400	2.700	4.600	2.300	4.100
	緩衝記憶體寫入、(n)=500	47.800	82.900	46.800	81.900	46.300	81.400
	至本機的CPU緩衝記憶體的寫入、(n)=1	3.900	4.700	1.300	3.200	1.000	2.800
	至本機的CPU緩衝記憶體的寫入、(n)=320	18.700	48.000	17.700	47.000	17.200	46.500
TYPERD	—	12.000	26.900	11.000	25.900	10.500	25.400

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
UNIINFRD	(n)=1	14.900	21.500	13.900	20.500	13.400	20.000 (使用過程CPU時: 22.900)
	(n)=16	27.600	34.600	26.600	33.600	26.100	33.100 (使用過程CPU時: 36.800)
S. RTREAD	—	3.900	7.400	2.900	6.400	2.400	5.900
S. RTWRITE	—	15.000	45.300	14.000	44.300	13.500	43.800
LOGTRG	—	57.200	76.000	56.200	75.000	55.700	74.500
LOGTRGR	—	12.500	20.100	11.500	19.100	11.000	18.600
DATATRG	採集對象元件點數=32點	—	—	—	—	65.000	65.000
SP. SOCOPE	TCP Active	17.100	35.300	16.100	34.300	15.600	33.800
	TCP Unpassive	17.100	35.300	16.100	34.300	15.600	33.800
	TCP Fullpassive	17.100	35.300	16.100	34.300	15.600	33.800
	UDP	17.100	35.300	16.100	34.300	15.600	33.800
SP. SOCCLOSE	透過TCP本機執行	16.800	35.500	15.800	34.500	15.300	34.000
	透過TCP 對象設備執行	16.800	35.500	15.800	34.500	15.300	34.000
	UDP	16.800	35.500	15.800	34.500	15.300	34.000
SP. SOCRCV	TCP 資料量最小(1字節)	16.500	38.200	15.500	37.200	15.000	36.700
	TCP 資料量最大(10238字節)	16.500	38.200	15.500	37.200	15.000	36.700
	UDP 資料量最小(1字節)	16.700	38.300	15.700	37.300	15.200	36.800
	UDP 資料量最大(10238字節)	16.700	38.300	15.700	37.300	15.200	36.800
S. SOCRCVS	TCP 資料量最小(1字節)	17.300	33.500	16.300	32.500	15.800	32.000
	TCP 資料量最大(10238字節)	142.500	181.500	141.500	180.500	141.000	180.000
	UDP 資料量最小(1字節)	17.300	33.500	16.300	32.500	15.800	32.000
	UDP 資料量最大(10238字節)	146.500	183.500	145.500	182.500	145.000	182.000
SP. SOCSND	TCP 資料量最小(1字節)	19.500	36.900	18.500	35.900	18.000	35.400
	TCP 資料量最大(10238字節)	156.500	181.500	155.500	180.500	155.000	180.000
	UDP 資料量最小(1字節)	19.500	36.900	18.500	35.900	18.000	35.400
	UDP 資料量最大(10238字節)	156.500	181.500	155.500	180.500	155.000	180.000
SP. SOCCINF	—	4.600	10.700	3.600	9.700	3.100	9.200
SP. SOCCSET	—	4.800	16.000	3.800	15.000	3.300	14.500
SP. SOCRMODE	普通模式→固定長模式的切換	6.300	15.700	5.300	14.700	4.800	14.200
	固定長模式→普通模式的切換	6.300	15.400	5.300	14.400	4.800	13.900
S. SOCRDATA	資料量最小(1字元)	5.700	11.600	4.700	10.600	4.200	10.100
	資料量最大(5120字元)	115.900	164.300	114.900	163.300	114.400	162.800
SP. ECPRTCL	—	34.700	41.300	33.700	40.300	33.200	39.800
SP. SLMPSND	“Read(請求: 0401H)” (字元單位讀出): 讀出點數=1點	17.900	42.900	16.900	41.900	16.400	47.400
SP. FTPPUT	檔案名+伺服器路徑字元串=32字元	77.500	81.500	76.500	80.500	76.000	85.000
	檔案名+伺服器路徑字元串=64字元	94.500	98.500	93.500	97.500	93.000	102.000
SP. FTPGET	檔案名+伺服器路徑字元串=32字元	77.500	81.500	76.500	80.500	76.000	85.000
	檔案名+伺服器路徑字元串=64字元	94.500	98.500	93.500	97.500	93.000	102.000
PSTOP	檔案名: “P1”	44.500	79.500	43.500	78.500	43.000	78.000
POFF	檔案名: “P1”	43.800	78.500	42.800	77.500	42.300	77.000
PSCAN	檔案名: “P1”	45.200	79.700	44.200	78.700	43.700	78.200
PID	—	5.400	11.700	4.400	10.700	3.900	10.200
S. PIDINIT	1環路	5.000	8.500	4.000	7.500	3.500	7.000
	32環路	54.800	58.400	53.800	57.400	53.300	56.900

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
S. PIDCONT	1環路(首次)	18.100	19.600	17.100	18.600	16.600	18.100
	1環路(第2次及其以後)	15.400	21.000	14.400	20.000	13.900	19.500
	32環路(首次)	206.500	210.200	205.500	209.200	205.000	208.700
	32環路(第2次及其以後)	194.800	215.600	193.800	214.600	193.300	214.100
S. PIDSTOP	1環路	2.600	4.100	1.600	3.100	1.100	2.600
S. PIDRUN	1環路	3.000	4.200	2.000	3.200	1.500	2.700
S. PIDPRMW	1環路	5.200	8.000	4.200	7.000	3.700	6.500
PIDINIT	1環路	4.500	8.200	3.500	7.200	3.000	6.700
	32環路	38.500	42.000	37.500	41.000	37.000	40.500
PIDCONT	1環路(首次)	18.100	19.400	17.100	18.400	16.600	17.900
	1環路(第2次及其以後)	15.000	16.600	14.000	15.600	13.500	15.100
	32環路(首次)	201.100	203.100	200.100	202.100	199.600	201.600
	32環路(第2次及其以後)	192.300	203.200	191.300	202.200	190.800	201.700
PIDSTOP	1環路	2.700	4.000	1.700	3.000	1.200	2.500
PIDRUN	1環路	2.700	4.000	1.700	3.000	1.200	2.500
PIDPRMW	1環路	4.500	8.000	3.500	7.000	3.000	6.500
D. DDRD	讀取資料點數=1	76.800	123.000	75.800	122.000	75.300	121.500
	讀取資料點數=16	77.000	123.000	76.000	122.000	75.500	121.500
	讀取資料點數=96	79.500	125.500	78.500	124.500	78.000	124.000
	讀取資料點數=8192	154.600	198.300	153.600	197.300	153.100	196.800
D. DDWR	讀取資料點數=1	75.400	122.600	74.400	121.600	73.900	121.100
	讀取資料點數=16	75.800	122.700	74.800	121.700	74.300	121.200
	讀取資料點數=96	75.900	122.900	74.900	121.900	74.400	121.400
	讀取資料點數=8192	76.000	122.900	75.000	121.900	74.500	121.400
M. DDRD	寫入資料點數=1	66.200	113.900	65.200	112.900	64.700	112.400
	寫入資料點數=16	67.200	114.200	66.200	113.200	65.700	112.700
	寫入資料點數=96	66.900	114.600	65.900	113.600	65.400	113.100
	寫入資料點數=8192	146.700	189.700	145.700	188.700	145.200	188.200
M. DDWR	寫入資料點數=1	65.100	112.900	64.100	111.900	63.600	111.400
	寫入資料點數=16	65.100	113.900	64.100	112.900	63.600	112.400
	寫入資料點數=96	65.400	115.000	64.400	114.000	63.900	113.500
	寫入資料點數=8192	66.200	115.100	65.200	114.100	64.700	113.600
LD [S□]	導通時/非導通時	1.500	1.900	0.700	1.200	0.500	1.100
LD [BL□\S□]	導通時/非導通時	4.300	6.500	3.300	5.500	2.800	5.000
LD [BL□]	導通時/非導通時	3.300	5.100	2.300	4.100	1.800	3.600
AND [S□]	導通時/非導通時	1.500	1.900	0.700	1.200	0.500	1.100
AND [BL□\S□]	導通時/非導通時	4.300	6.500	3.300	5.500	2.800	5.000
AND [BL□]	導通時/非導通時	3.300	5.100	2.300	4.100	1.800	3.600
OR [S□]	導通時/非導通時	1.500	1.900	0.700	1.200	0.500	1.100
OR [BL□\S□]	導通時/非導通時	4.300	6.500	3.300	5.500	2.800	5.000
OR [BL□]	導通時/非導通時	3.300	5.100	2.300	4.100	1.800	3.600
LDI [S□]	導通時/非導通時	1.500	1.900	0.700	1.200	0.500	1.100
LDI [BL□\S□]	導通時/非導通時	4.300	6.500	3.300	5.500	2.800	5.000
LDI [BL□]	導通時/非導通時	3.300	5.100	2.300	4.100	1.800	3.600
ANI [S□]	導通時/非導通時	1.500	1.900	0.700	1.200	0.500	1.100
ANI [BL□\S□]	導通時/非導通時	4.300	6.500	3.300	5.500	2.800	5.000
ANI [BL□]	導通時/非導通時	3.300	5.100	2.300	4.100	1.800	3.600
ORI [S□]	導通時/非導通時	1.500	1.900	0.700	1.200	0.500	1.100
ORI [BL□\S□]	導通時/非導通時	4.300	6.500	3.300	5.500	2.800	5.000
ORI [BL□]	導通時/非導通時	3.300	5.100	2.300	4.100	1.800	3.600
MOV(P) [K4S□]	—	4.100	7.700	3.100	6.700	2.600	6.200

指令名	條件	處理時間(μs)					
		R00CPU、R01CPU		R02CPU		左述以外的CPU模組	
		最小	最大	最小	最大	最小	最大
MOV (P) [BL□\K4S□]	—	5.300	10.700	4.300	9.700	3.800	9.200
DMOV (P) [K8S□]	—	4.100	7.700	3.100	6.700	2.600	6.200
DMOV (P) [BL□\K8S□]	—	5.300	10.700	4.300	9.700	3.800	9.200
BMOV (P) [K4S□]	(n)=1	5.700	15.600	4.700	14.600	4.200	14.100
	(n)=96	6.400	16.600	5.400	15.600	4.900	15.100
BMOV (P) [BL□\K4S□]	(n)=1	7.100	19.100	6.100	18.100	5.600	17.600
	(n)=96	7.800	19.700	6.800	18.700	6.300	18.200
SET [S□]	無變化時	4.900	9.900	3.900	8.900	3.400	8.400
	變化時	7.100	15.100	5.900	14.100	5.100	13.600
SET [BL□\S□]	無變化時	4.900	9.900	3.900	8.900	3.400	8.400
	變化時	6.700	15.100	5.300	14.100	4.700	13.600
SET [BL□]	無變化時	3.300	5.000	2.300	4.000	1.800	3.500
	變化時	4.600	5.000	3.200	4.000	2.700	3.500
RST [S□]	無變化時	3.800	7.900	2.800	6.900	2.300	6.400
	變化時	6.200	9.000	5.100	8.000	4.400	7.500
RST [BL□\S□]	無變化時	3.800	7.900	2.800	6.900	2.300	6.400
	變化時	5.800	9.000	4.700	8.000	4.200	7.500
RST [BL□]	無變化時	3.900	5.900	2.900	4.900	2.400	4.400
	變化時	4.800	5.900	3.400	4.900	2.900	4.400
PAUSE [BL□]	—	3.300	4.600	2.300	3.600	1.800	3.100
RSTART [BL□]	—	3.300	4.600	2.300	3.600	1.800	3.100
BRSET [S□]	—	3.600	5.000	2.600	4.000	2.100	3.500
SP. CONTSW	SM1646 = ON時	—	—	—	—	10.600	16.000
	SM1646 = OFF時	—	—	—	—	137.200	169.700
DCONTSW	允許→不允許	—	—	—	—	1.400	1.900
ECONTSW	不允許→允許	—	—	—	—	1.400	2.000
CONTWR	寫入資料點數=1	—	—	—	—	18.300	21.200
	寫入資料點數=16	—	—	—	—	18.700	21.400
	寫入資料點數=96	—	—	—	—	19.200	21.900
	寫入資料點數=1024	—	—	—	—	27.200	29.900
SP. SIDRD	—	—	—	—	30.800	37.400	

指令處理時間的加法運算時間

在使用檔案寄存器 (R/ZR)、模組存取元件 (U□\G□) 及模組標籤 (標籤名中附加_D的標籤)、直接連結元件 (J□\□) 及模組標籤 (標籤名中附加_D的連結元件) 時，應將下述所示的加算時間加算至各指令處理時間。

元件名	資料類型	元件指定位置	加法運算時間 (μs)		
			R04CPU、R04ENCPU	R08CPU、R08ENCPU、R16CPU、R16ENCPU、R32CPU、R32ENCPU、R120CPU、R120ENCPU	
檔案寄存器 (R/ZR)	未安裝擴展SRAM卡盒及無電池選項卡盒時	位元	源	0.074	0.043
			目標	0.023	0.023
		字元	源	0.074	0.043
			目標	0.023	0.023
		雙字元	源	0.148	0.085
			目標	0.044	0.044
	安裝擴展SRAM卡盒或無電池選項卡盒時	位元	源	0.099	0.099
			目標	0.028	0.028
		字元	源	0.099	0.099
			目標	0.028	0.028
		雙字元	源	0.198	0.198
			目標	0.054	0.054
模組存取元件 (U□\G□) 及模組標籤 (標籤名附加_D的標籤)	位元	源	13.000	13.000	
		目標	14.000	14.000	
	字元	源	13.000	13.000	
		目標	14.000	14.000	
	雙字元	源	13.000	13.000	
		目標	14.000	14.000	
直接連結元件 (J□\□) 及模組標籤 (標籤名中附加_D的連結元件)	位元	源	51.000	51.000	
		目標	52.000	52.000	
	字元	源	51.000	51.000	
		目標	52.000	52.000	
	雙字元	源	51.000	51.000	
		目標	52.000	52.000	

附2 基本步數及子集處理的可否

各指令的基本步數及子集處理的可否如下所示。

指令名	基本步數	子集可否
LD	1	○
LDI	1	○
AND	1	○
ANI	1	○
OR	1	○
ORI	1	○
LDP	2	○
LDF	2	○
ANDP	2	○
ANDF	2	○
ORP	2	○
ORF	2	○
LDPI	2	○
LDFI	2	○
ANDPI	2	○
ANDFI	2	○
ORPI	2	○
ORFI	2	○
ANB	1	—
ORB	1	—
MPS	1	—
MRD	1	—
MPP	1	—
INV	1	—
MEP	1	—
MEF	1	—
EGP	1	—
EGF	1	—
OUT	1	○
OUT T/ST	4	—
OUT LT/LST	2	—
OUT C	4	—
OUT LC	4	—
OUT F	2	—
OUTH T/ST	4	—
SET	1	○
RST	1	○
SET F	3	—
RST F	3	—
PLS	2	○
PLF	2	○
FF	2	○
DELTA	2	—
DELTAP	3	—
SFT	2	—
SFTP	3	—
MC	2	○
MCR	1	—
PHASE	19*1	—

指令名	基本步數	子集可否
PHASECHG	6	—
PHASEEND	4	—
FEND	2	—
END	2	—
STOP	1	—
NOP	1	—
LD=	3	○
LD<>	3	○
LD>	3	○
LD<=	3	○
LD<	3	○
LD>=	3	○
AND=	3	○
AND<>	3	○
AND>	3	○
AND<=	3	○
AND<	3	○
AND>=	3	○
OR=	3	○
OR<>	3	○
OR>	3	○
OR<=	3	○
OR<	3	○
OR>=	3	○
LD=_U	3	○
LD<>_U	3	○
LD>_U	3	○
LD<=_U	3	○
LD<_U	3	○
LD>=_U	3	○
AND=_U	3	○
AND<>_U	3	○
AND>_U	3	○
AND<=_U	3	○
AND<_U	3	○
AND>=_U	3	○
OR=_U	3	○
OR<>_U	3	○
OR>_U	3	○
OR<=_U	3	○
OR<_U	3	○
OR>=_U	3	○
LDD=	3	○
LDD<>	3	○
LDD>	3	○
LDD<=	3	○
LDD<	3	○
LDD>=	3	○
ANDD=	3	○
ANDD<>	3	○
ANDD>	3	○
ANDD<=	3	○
ANDD<	3	○

指令名	基本步數	子集可否
ANDD>=	3	○
ORD=	3	○
ORD<>	3	○
ORD>	3	○
ORD<=	3	○
ORD<	3	○
ORD>=	3	○
LDD=_U	3	○
LDD<>_U	3	○
LDD>_U	3	○
LDD<=_U	3	○
LDD<_U	3	○
LDD>=_U	3	○
ANDD=_U	3	○
ANDD<>_U	3	○
ANDD>_U	3	○
ANDD<=_U	3	○
ANDD<_U	3	○
ANDD>=_U	3	○
ORD=_U	3	○
ORD<>_U	3	○
ORD>_U	3	○
ORD<=_U	3	○
ORD<_U	3	○
ORD>=_U	3	○
CMP	4	—
CMPP	5	—
CMP_U	4	—
CMPP_U	5	—
DCMP	4	—
DCMPP	5	—
DCMP_U	4	—
DCMPP_U	5	—
ZCP	5	—
ZCPP	6	—
ZCP_U	5	—
ZCPP_U	6	—
DZCP	5	—
DZCPP	6	—
DZCP_U	5	—
DZCPP_U	6	—
BKCMPP=	5	—
BKCMPP<>	5	—
BKCMPP>	5	—
BKCMPP<=	5	—
BKCMPP<	5	—
BKCMPP>=	5	—
BKCMPP=P	6	—
BKCMPP<>P	6	—
BKCMPP>P	6	—
BKCMPP<=P	6	—
BKCMPP<P	6	—
BKCMPP>=P	6	—

指令名	基本步數	子集可否
BKMP=_U	5	—
BKMP<>_U	5	—
BKMP>_U	5	—
BKMP<=_U	5	—
BKMP<_U	5	—
BKMP>=_U	5	—
BKMP=P_U	6	—
BKMP<>P_U	6	—
BKMP>P_U	6	—
BKMP<=P_U	6	—
BKMP<P_U	6	—
BKMP>=P_U	6	—
DBKMP=	5	—
DBKMP<>	5	—
DBKMP>	5	—
DBKMP<=	5	—
DBKMP<	5	—
DBKMP>=	5	—
DBKMP=P	6	—
DBKMP<>P	6	—
DBKMP>P	6	—
DBKMP<=P	6	—
DBKMP<P	6	—
DBKMP>=P	6	—
DBKMP=_U	5	—
DBKMP<>_U	5	—
DBKMP>_U	5	—
DBKMP<=_U	5	—
DBKMP<_U	5	—
DBKMP>=_U	5	—
DBKMP=P_U	6	—
DBKMP<>P_U	6	—
DBKMP>P_U	6	—
DBKMP<=P_U	6	—
DBKMP<P_U	6	—
DBKMP>=P_U	6	—
+ (s) (d)	3	○
+P (s) (d)	4	○
+ (s1) (s2) (d)	3	○
+P (s1) (s2) (d)	4	○
+_U (s) (d)	3	○
+P_U (s) (d)	4	○
+_U (s1) (s2) (d)	3	○
+P_U (s1) (s2) (d)	4	○
- (s) (d)	3	○
-P (s) (d)	4	○
- (s1) (s2) (d)	3	○
-P (s1) (s2) (d)	4	○
-_U (s) (d)	3	○
-P_U (s) (d)	4	○
-_U (s1) (s2) (d)	3	○
-P_U (s1) (s2) (d)	4	○
D+ (s) (d)	3	○

指令名	基本步數	子集可否
D+P(s) (d)	4	○
D+ (s1) (s2) (d)	3	○
D+P(s1) (s2) (d)	4	○
D+_U(s) (d)	3	○
D+P_U(s) (d)	4	○
D+_U(s1) (s2) (d)	3	○
D+P_U(s1) (s2) (d)	4	○
D-(s) (d)	3	○
D-P(s) (d)	4	○
D- (s1) (s2) (d)	3	○
D-P(s1) (s2) (d)	4	○
D-_U(s) (d)	3	○
D-P_U(s) (d)	4	○
D-_U(s1) (s2) (d)	3	○
D-P_U(s1) (s2) (d)	4	○
*	3	○
*P	4	○
*_U	3	○
*P_U	4	○
/	3	○
/P	4	○
/_U	3	○
/P_U	4	○
D*	3	○
D*P	4	○
D*_U	3	○
D*P_U	4	○
D/	3	○
D/P	4	○
D/_U	3	○
D/P_U	4	○
B+(s) (d)	3	○
B+P(s) (d)	4	○
B+ (s1) (s2) (d)	4	—
B+P(s1) (s2) (d)	5	—
B-(s) (d)	3	○
B-P(s) (d)	4	○
B- (s1) (s2) (d)	4	—
B-P(s1) (s2) (d)	5	—
DB+(s) (d)	3	—
DB+P(s) (d)	4	—
DB+(s1) (s2) (d)	4	—
DB+P(s1) (s2) (d)	5	—
DB-(s) (d)	3	—
DB-P(s) (d)	4	—
DB-(s1) (s2) (d)	4	—
DB-P(s1) (s2) (d)	5	—
B*	4	○
B*P	5	○
B/	4	○
B/P	5	○
DB*	4	—
DB*P	5	—

指令名	基本步數	子集可否
DB/	4	—
DB/P	5	—
BK+	5	—
BK+P	6	—
BK+_U	5	—
BK+P_U	6	—
BK-	5	—
BK-P	6	—
BK-_U	5	—
BK-P_U	6	—
DBK+	5	—
DBK+P	6	—
DBK+_U	5	—
DBK+P_U	6	—
DBK-	5	—
DBK-P	6	—
DBK-_U	5	—
DBK-P_U	6	—
INC	2	○
INCP	3	○
INC_U	2	○
INCP_U	3	○
DEC	2	○
DECP	3	○
DEC_U	2	○
DECP_U	3	○
DINC	2	○
DINCP	3	○
DINC_U	2	○
DINCP_U	3	○
DDEC	2	○
DDECP	3	○
DDEC_U	2	○
DDECP_U	3	○
WAND (s) (d)	3	○
WANDP (s) (d)	4	○
WAND (s1) (s2) (d)	3	○
WANDP (s1) (s2) (d)	4	○
DAND (s) (d)	3	○
DANDP (s) (d)	4	○
DAND (s1) (s2) (d)	3	○
DANDP (s1) (s2) (d)	4	○
BKAND	5	—
BKANDP	6	—
WOR (s) (d)	3	○
WORP (s) (d)	4	○
WOR (s1) (s2) (d)	3	○
WORP (s1) (s2) (d)	4	○
DOR (s) (d)	3	○
DORP (s) (d)	4	○
DOR (s1) (s2) (d)	3	○
DORP (s1) (s2) (d)	4	○
BKOR	5	—

指令名	基本步數	子集可否
BKORP	6	—
WXOR (s) (d)	3	○
WXORP (s) (d)	4	○
WXOR (s1) (s2) (d)	3	○
WXORP (s1) (s2) (d)	4	○
DXOR (s) (d)	3	○
DXORP (s) (d)	4	○
DXOR (s1) (s2) (d)	3	○
DXORP (s1) (s2) (d)	4	○
BKXOR	5	—
BKXORP	6	—
WXNR (s) (d)	3	○
WXNRP (s) (d)	4	○
WXNR (s1) (s2) (d)	3	○
WXNRP (s1) (s2) (d)	4	○
DXNR (s) (d)	3	○
DXNRP (s) (d)	4	○
DXNR (s1) (s2) (d)	3	○
DXNRP (s1) (s2) (d)	4	○
BKXNR	5	—
BKXNRP	6	—
BSET	3	○
BSETP	4	○
BRST	3	○
BRSTP	4	○
TEST	4	○
TESTP	5	○
DTEST	5	○
DTESTP	6	○
BKRST	3	○
BKRSTP	4	○
SFR	4	○
SFRP	5	○
SFL	4	○
SFLP	5	○
BSFR	3	○
BSFRP	4	○
BSFL	3	○
BSFLP	4	○
DSFR	3	○
DSFRP	4	○
DSFL	3	○
DSFLP	4	○
DDSFR	3	—
DDSFRP	4	—
DDSFL	3	—
DDSFLP	4	—
ESFR	3	—
ESFRP	4	—
ESFL	3	—
ESFLP	4	—
EDSFR	3	—
EDSFRP	4	—

指令名	基本步數	子集可否
EDSFL	3	—
EDSFLP	4	—
SFTBR	4	○
SFTBRP	5	○
SFTBL	4	○
SFTBLP	5	○
SFTWR	4	○
SFTWRP	5	○
SFTWL	4	○
SFTWLP	5	○
SFTDWR	4	—
SFTDWRP	5	—
DWSFTR	5	—
DWSFTRP	6	—
SFTDWL	4	—
SFTDWLP	5	—
DWSFTL	5	—
DWSFTLP	6	—
SFTER	4	—
SFTERP	5	—
ESFTR	5	—
ESFTRP	6	—
SFTEL	4	—
SFTELP	5	—
ESFTL	5	—
ESFTLP	6	—
SFTEDR	4	—
SFTEDRP	5	—
EDSFTR	5	—
EDSFTRP	6	—
SFTEDL	4	—
SFTEDLP	5	—
EDSFTL	5	—
EDSFTLP	6	—
BCD	2	○
BCDP	3	○
DBCD	2	○
DBCDP	3	○
BIN	2	○
BINP	3	○
DBIN	2	○
DBINP	3	○
FLT2INT	2	○
FLT2INTP	3	○
FLT2UINT	2	○
FLT2UINTP	3	○
FLT2DINT	2	○
FLT2DINTP	3	○
FLT2UDINT	2	○
FLT2UDINTP	3	○
DBL2INT	3	—
DBL2INTP	4	—
DBL2UINT	3	—

指令名	基本步數	子集可否
DBL2UINTP	4	—
DBL2DINT	3	—
DBL2DINTP	4	—
DBL2UDINT	3	—
DBL2UDINTP	4	—
INT2UINT	3	○
INT2UINTP	4	○
INT2DINT	3	○
INT2DINTP	4	○
INT2UDINT	3	○
INT2UDINTP	4	○
UINT2INT	3	○
UINT2INTP	4	○
UINT2DINT	3	○
UINT2DINTP	4	○
UINT2UDINT	3	○
UINT2UDINTP	4	○
DINT2INT	2	○
DINT2INTP	3	○
DINT2UINT	3	○
DINT2UINTP	4	○
DINT2UDINT	3	○
DINT2UDINTP	4	○
UDINT2INT	3	○
UDINT2INTP	4	○
UDINT2UINT	3	○
UDINT2UINTP	4	○
UDINT2DINT	3	○
UDINT2DINTP	4	○
GRY	2	○
GRYP	3	○
GRY_U	2	○
GRYP_U	3	○
DGRY	2	○
DGRYP	3	○
DGRY_U	2	○
DGRYP_U	3	○
GBIN	2	○
GBINP	3	○
GBIN_U	2	○
GBINP_U	3	○
DGBIN	2	○
DGBINP	3	○
DGBIN_U	2	○
DGBINP_U	3	○
BKBCD	4	—
BKBCDP	5	—
BKBIN	4	—
BKBINP	5	—
DABIN	3	—
DABINP	4	—
DABIN_U	3	—
DABINP_U	4	—

指令名	基本步數	子集可否
DDABIN	3	—
DDABINP	4	—
DDABIN_U	3	—
DDABINP_U	4	—
HABIN	3	—
HABINP	4	—
DHABIN	3	—
DHABINP	4	—
DABCD	3	—
DABCDP	4	—
DDABCD	3	—
DDABCDP	4	—
VAL	4	—
VALP	5	—
VAL_U	4	—
VALP_U	5	—
DVAL	4	—
DVALP	5	—
DVAL_U	4	—
DVALP_U	5	—
ASC2INT	4	—
ASC2INTP	5	—
EMOD	4	—
EMODP	5	—
NEG	2	○
NEGP	3	○
DNEG	2	○
DNEGP	3	○
DECO	4	—
DECOP	5	—
ENCO	4	—
ENCOP	5	—
SEG	3	○
SEGP	4	○
DIS	4	—
DISP	5	—
UNI	4	—
UNIP	5	—
NDIS	4	—
NDISP	5	—
NUNI	4	—
NUNIP	5	—
WTOB	4	—
WTOBP	5	—
BTOW	4	—
BTOWP	5	—
MOV	2	○
MOVP	3	○
DMOV	2	○
DMOVP	3	○
CML	2	○
CMLP	3	○
DCML	2	○

指令名	基本步數	子集可否
DCMLP	3	○
SMOV	6	—
SMOV	7	—
CMLB	3	○
CMLBP	4	○
BMOV	4	○
BMOV	5	○
BMOVL	4	○
BMOVLP	5	○
FMOV	4	○
FMOV	5	○
FMOVL	4	○
FMOVLP	5	○
DFMOV	4	○
DFMOV	5	○
DFMOVL	4	○
DFMOVLP	5	○
XCH	3	○
XCHP	4	○
DXCH	3	○
DXCHP	4	○
BXCH	4	—
BXCHP	5	—
SWAP	2	—
SWAPP	3	—
DSWAP	3	—
DSWAPP	4	—
MOVB	2	○
MOVBP	3	○
BLKMOVB	4	—
BLKMOVBP	5	—
ROR	4	○
RORP	5	○
RCR	4	○
RCRP	5	○
DROR	4	○
DRORP	5	○
DRCR	4	○
DRCRP	5	○
ROL	4	○
ROLP	5	○
RCL	4	○
RCLP	5	○
DROL	4	○
DROLP	5	○
DRCL	4	○
DRCLP	5	○
CJ	4	—
SCJ	5	—
JMP	4	—
GOEND	1	—
DI	1	—
DI (s)	2	○

指令名	基本步數	子集可否
EI	1	○
IMASK	2	○
SIMASK	3	○
IRET	1	—
WDT	1	—
WDTP	2	—
FOR	2	○
NEXT	1	—
BREAK	4	—
BREAKP	5	—
CALL (P)	3	—
CALL (P) (s1)	4	—
CALL (P) (s1) (s2)	5	—
CALL (P) (s1) (s2) (s3)	6	—
CALL (P) (s1) (s2) (s3) (s4)	7	—
CALL (P) (s1) (s2) (s3) (s4) (s5)	8	—
CALLP (P)	4	—
CALLP (P) (s1)	5	—
CALLP (P) (s1) (s2)	6	—
CALLP (P) (s1) (s2) (s3)	7	—
CALLP (P) (s1) (s2) (s3) (s4)	8	—
CALLP (P) (s1) (s2) (s3) (s4) (s5)	9	—
RET	1	—
FCALL (P)	3	—
FCALL (P) (s1)	4	—
FCALL (P) (s1) (s2)	5	—
FCALL (P) (s1) (s2) (s3)	6	—
FCALL (P) (s1) (s2) (s3) (s4)	7	—
FCALL (P) (s1) (s2) (s3) (s4) (s5)	8	—
FCALLP (P)	4	—
FCALLP (P) (s1)	5	—
FCALLP (P) (s1) (s2)	6	—
FCALLP (P) (s1) (s2) (s3)	7	—
FCALLP (P) (s1) (s2) (s3) (s4)	8	—
FCALLP (P) (s1) (s2) (s3) (s4) (s5)	9	—
ECALL (檔案名) (P)	4+檔案名的字元數	—
ECALL (檔案名) (P) (s1)	5+檔案名的字元數	—
ECALL (檔案名) (P) (s1) (s2)	6+檔案名的字元數	—
ECALL (檔案名) (P) (s1) (s2) (s3)	7+檔案名的字元數	—
ECALL (檔案名) (P) (s1) (s2) (s3) (s4)	8+檔案名的字元數	—
ECALL (檔案名) (P) (s1) (s2) (s3) (s4) (s5)	9+檔案名的字元數	—
ECALLP (檔案名) (P)	5+檔案名的字元數	—
ECALLP (檔案名) (P) (s1)	6+檔案名的字元數	—
ECALLP (檔案名) (P) (s1) (s2)	7+檔案名的字元數	—
ECALLP (檔案名) (P) (s1) (s2) (s3)	8+檔案名的字元數	—
ECALLP (檔案名) (P) (s1) (s2) (s3) (s4)	9+檔案名的字元數	—
ECALLP (檔案名) (P) (s1) (s2) (s3) (s4) (s5)	10+檔案名的字元數	—
EFCALL (檔案名) (P)	4+檔案名的字元數	—
EFCALL (檔案名) (P) (s1)	5+檔案名的字元數	—
EFCALL (檔案名) (P) (s1) (s2)	6+檔案名的字元數	—
EFCALL (檔案名) (P) (s1) (s2) (s3)	7+檔案名的字元數	—
EFCALL (檔案名) (P) (s1) (s2) (s3) (s4)	8+檔案名的字元數	—
EFCALL (檔案名) (P) (s1) (s2) (s3) (s4) (s5)	9+檔案名的字元數	—

指令名	基本步數	子集可否
EFCALLP(檔案名)(P)	5+檔案名的字元數	—
EFCALLP(檔案名)(P)(s1)	6+檔案名的字元數	—
EFCALLP(檔案名)(P)(s1)(s2)	7+檔案名的字元數	—
EFCALLP(檔案名)(P)(s1)(s2)(s3)	8+檔案名的字元數	—
EFCALLP(檔案名)(P)(s1)(s2)(s3)(s4)	9+檔案名的字元數	—
EFCALLP(檔案名)(P)(s1)(s2)(s3)(s4)(s5)	10+檔案名的字元數	—
XCALL(P)	4	—
XCALL(P)(s1)	5	—
XCALL(P)(s1)(s2)	6	—
XCALL(P)(s1)(s2)(s3)	7	—
XCALL(P)(s1)(s2)(s3)(s4)	8	—
XCALL(P)(s1)(s2)(s3)(s4)(s5)	9	—
FIFR	3	—
FIFRP	4	—
FPOP	3	—
FPOPP	4	—
FIFW	3	—
FIFWP	4	—
FINS	4	—
FINSP	5	—
FDEL	4	—
FDELP	5	—
S.DEVLD	4	—
SP.DEVLD	5	—
SP.DEVST	6	—
SP.FREAD	8	—
SP.FWRITE	8	—
SP.FDELETE	7	—
SP.FCOPY	9	—
SP.FMOVE	9	—
SP.FRENAME	8	—
SP.FSTATUS	8	—
LEDR	2	—
PALERT	2	—
PALERTP	3	—
PABORT	2	—
LD\$=	3	○
LD\$<>	3	○
LD\$>	3	○
LD\$<=	3	○
LD\$<	3	○
LD\$>=	3	○
AND\$=	3	○
AND\$<>	3	○
AND\$>	3	○
AND\$<=	3	○
AND\$<	3	○
AND\$>=	3	○
OR\$=	3	○
OR\$<>	3	○
OR\$>	3	○
OR\$<=	3	○
OR\$<	3	○

指令名	基本步數	子集可否
OR\$>=	3	○
\$+ (s) (d)	3	○
\$+P (s) (d)	4	○
\$+ (s1) (s2) (d)	4	○
\$+P (s1) (s2) (d)	5	○
\$MOV	3	—
\$MOVP	4	—
\$MOV_WS	3	—
\$MOVP_WS	4	—
BINDA	3	—
BINDAP	4	—
BINDA_U	3	—
BINDAP_U	4	—
DBINDA	3	—
DBINDAP	4	—
DBINDA_U	3	—
DBINDAP_U	4	—
BINHA	3	—
BINHAP	4	—
DBINHA	3	—
DBINHAP	4	—
STR	4	—
STRP	5	—
STR_U	4	—
STRP_U	5	—
DSTR	4	—
DSTRP	5	—
DSTR_U	4	—
DSTRP_U	5	—
BCDDA	3	—
BCDDAP	4	—
DBCDDA	3	—
DBCDDAP	4	—
ESTR	4	—
ESTRP	5	—
INT2ASC	4	—
INT2ASCP	5	—
WS2SJIS	3	—
WS2SJISP	4	—
SJIS2WS	3	—
SJIS2WSP	4	—
SJIS2WSB	3	—
SJIS2WSBP	4	—
LEN	3	○
LENP	4	○
RIGHT	4	—
RIGHTP	5	—
LEFT	4	—
LEFTP	5	—
MIDR	4	—
MIDRP	5	—
MIDW	4	—
MIDWP	5	—

指令名	基本步數	子集可否
INSTR	5	—
INSTRP	6	—
STRINS	4	—
STRINSP	5	—
STRDEL	4	—
STRDELP	5	—
LDE=	3	○
LDE<>	3	○
LDE>	3	○
LDE<=	3	○
LDE<	3	○
LDE>=	3	○
ANDE=	3	○
ANDE<>	3	○
ANDE>	3	○
ANDE<=	3	○
ANDE<	3	○
ANDE>=	3	○
ORE=	3	○
ORE<>	3	○
ORE>	3	○
ORE<=	3	○
ORE<	3	○
ORE>=	3	○
LDED=	3	—
LDED<>	3	—
LDED>	3	—
LDED<=	3	—
LDED<	3	—
LDED>=	3	—
ANDED=	3	—
ANDED<>	3	—
ANDED>	3	—
ANDED<=	3	—
ANDED<	3	—
ANDED>=	3	—
ORED=	3	—
ORED<>	3	—
ORED>	3	—
ORED<=	3	—
ORED<	3	—
ORED>=	3	—
ECMP	4	—
ECMPP	5	—
EDCMP	4	—
EDCMPP	5	—
EZCP	5	—
EZCPP	6	—
EDZCP	5	—
EDZCPP	6	—
E+ (s) (d)	3	○
E+P (s) (d)	4	○
E+ (s1) (s2) (d)	3	○

指令名	基本步數	子集可否
E+P (s1) (s2) (d)	4	○
E- (s) (d)	3	○
E-P (s) (d)	4	○
E- (s1) (s2) (d)	3	○
E-P (s1) (s2) (d)	4	○
ED+ (s) (d)	3	—
ED+P (s) (d)	4	—
ED+ (s1) (s2) (d)	4	—
ED+P (s1) (s2) (d)	5	—
ED- (s) (d)	3	—
ED-P (s) (d)	4	—
ED- (s1) (s2) (d)	4	—
ED-P (s1) (s2) (d)	5	—
E*	3	○
E*P	4	○
E/	3	○
E/P	4	○
ED*	4	—
ED*P	5	—
ED/	4	—
ED/P	5	—
INT2FLT	2	○
INT2FLTP	3	○
UINT2FLT	2	○
UINT2FLTP	3	○
DINT2FLT	2	○
DINT2FLTP	3	○
UDINT2FLT	2	○
UDINT2FLTP	3	○
DBL2FLT	3	—
DBL2FLTP	4	—
INT2DBL	3	—
INT2DBLP	4	—
UINT2DBL	3	—
UINT2DBLP	4	—
DINT2DBL	3	—
DINT2DBLP	4	—
UDINT2DBL	3	—
UDINT2DBLP	4	—
FLT2DBL	3	—
FLT2DBLP	4	—
EVAL	3	—
EVALP	4	—
EREXP	4	—
EREXPP	5	—
ENEG	2	—
ENEGP	3	—
EDNEG	2	—
EDNEGP	3	—
EMOV	2	○
EMOVP	3	○
EDMOV	2	○
EDMOVP	3	○

指令名	基本步數	子集可否
SIN	3	—
SINP	4	—
COS	3	—
COSP	4	—
TAN	3	—
TANP	4	—
ASIN	3	—
ASINP	4	—
ACOS	3	—
ACOSP	4	—
ATAN	3	—
ATANP	4	—
SIND	3	—
SINDP	4	—
COSD	3	—
COSDP	4	—
TAND	3	—
TANDP	4	—
ASIND	3	—
ASINDP	4	—
ACOSD	3	—
ACOSDP	4	—
ATAND	3	—
ATANDP	4	—
BSIN	3	—
BSINP	4	—
BCOS	3	—
BCOSP	4	—
BTAN	3	—
BTANP	4	—
BASIN	3	—
BASINP	4	—
BACOS	3	—
BACOSP	4	—
BATAN	3	—
BATANP	4	—
RAD	3	—
RADP	4	—
DEG	3	—
DEGP	4	—
RADD	3	—
RADDP	4	—
DEGD	3	—
DEGDP	4	—
ESQRT	3	—
ESQRTP	4	—
EDSQRT	3	—
EDSQRTP	4	—
EXP	3	—
EXPP	4	—
EXPD	3	—
EXPDP	4	—
LOG	3	—

指令名	基本步數	子集可否
LOGP	4	—
LOGD	3	—
LOGDP	4	—
BSQRT	3	—
BSQRTP	4	—
BDSQRT	3	—
BDSQRTP	4	—
POW	4	—
POWP	5	—
POWD	4	—
POWDP	5	—
LOG10	3	—
LOG10P	4	—
LOG10D	3	—
LOG10DP	4	—
EMAX	4	—
EMAXP	5	—
EDMAX	4	—
EDMAXP	5	—
EMIN	4	—
EMINP	5	—
EDMIN	4	—
EDMINP	5	—
RND	2	—
RNDP	3	—
SRND	2	—
SRNDP	3	—
ZPUSH(d)	2	○
ZPUSHP(d)	3	○
ZPUSH(s)(d)	3	○
ZPUSHP(s)(d)	4	○
ZPOP(d)	2	○
ZPOPP(d)	3	○
ZPOP(s)(d)	3	○
ZPOPP(s)(d)	4	○
LIMIT	5	○
LIMITP	6	○
LIMIT_U	5	○
LIMITP_U	6	○
DLIMIT	5	○
DLIMITP	6	○
DLIMIT_U	5	○
DLIMITP_U	6	○
BAND	5	—
BANDP	6	—
BAND_U	5	—
BANDP_U	6	—
DBAND	5	—
DBANDP	6	—
DBAND_U	5	—
DBANDP_U	6	—
ZONE	5	—
ZONEP	6	—

指令名	基本步數	子集可否
ZONE_U	5	—
ZONEP_U	6	—
DZONE	5	—
DZONEP	6	—
DZONE_U	5	—
DZONEP_U	6	—
SCL	4	—
SCLP	5	—
SCL_U	4	—
SCLP_U	5	—
DSCL	4	—
DSCLP	5	—
DSCL_U	4	—
DSCLP_U	5	—
SCL2	4	—
SCL2P	5	—
SCL2_U	4	—
SCL2P_U	5	—
DSCL2	4	—
DSCL2P	5	—
DSCL2_U	4	—
DSCL2P_U	5	—
UDCNT1	5	—
UDCNT2	5	—
TMR	4	—
STMR	4	—
ROTC	5	—
RAMPQ	7	—
SPD	5	—
PLSY	5	—
PWM	5	—
MTR	5	—
CCD	4	—
CCDP	5	—
SERDATA	5	—
SERDATAP	6	—
DSERDATA	5	—
DSERDATAP	6	—
SERMM	6	—
SERMMP	7	—
DSERMM	6	—
DSERMMP	7	—
SUM	3	○
SUMP	4	○
DSUM	3	○
DSUMP	4	○
BON	5	—
BONP	6	—
DBON	5	—
DBONP	6	—
MAX	4	—
MAXP	5	—
MAX_U	4	—

指令名	基本步數	子集可否
MAXP_U	5	—
DMAX	4	—
DMAXP	5	—
DMAX_U	4	—
DMAXP_U	5	—
MIN	4	—
MINP	5	—
MIN_U	4	—
MINP_U	5	—
DMIN	4	—
DMINP	5	—
DMIN_U	4	—
DMINP_U	5	—
SORTD	7	—
SORTD_U	7	—
DSORTD	7	—
DSORTD_U	7	—
SORTTBL	7	—
SORTTBL_U	7	—
SORTTBL2	7	—
SORTTBL2_U	7	—
DSORTTBL2	7	—
DSORTTBL2_U	7	—
WSUM	4	○
WSUMP	5	○
WSUM_U	4	○
WSUMP_U	5	○
DWSUM	4	—
DWSUMP	5	—
DWSUM_U	4	—
DWSUMP_U	5	—
MEAN	4	—
MEANP	5	—
MEAN_U	4	—
MEANP_U	5	—
DMEAN	4	—
DMEANP	5	—
DMEAN_U	4	—
DMEANP_U	5	—
SQRT	4	—
SQRTP	5	—
DSQRT	4	—
DSQRTP	5	—
CRC	4	—
CRCP	5	—
DBOPEN	5	—
DBOPENP	6	—
DBCLOSE	4	—
DBCLOSEP	5	—
DBINSERT	7	—
DBINSERTP	8	—
DBUPDATE	8	—
DBUPDATEP	9	—

指令名	基本步數	子集可否
DBSELECT	8	—
DBSELECTP	9	—
DBDELETE	6	—
DBDELETEP	7	—
DBIMPORT	4	—
DBIMPORTP	5	—
DBEXPORT	4	—
DBEXPORTP	5	—
DBTRANS	4	—
DBTRANSP	5	—
DBCOMMIT	4	—
DBCOMMITP	5	—
DBROLBAK	4	—
DBROLBAKP	5	—
RSET	2	—
RSETP	3	—
QDRSET	2+檔案名的字元數	—
QDRSETP	3+檔案名的字元數	—
ZRRDB	3	—
ZRRDBP	4	—
ZRWRB	3	—
ZRWRBP	4	—
ADRSET	3	—
ADRSETP	4	—
DATERD	2	—
DATERDP	3	—
DATEWR	2	—
DATEWRP	3	—
DATE+	4	—
DATE+P	5	—
DATE-	4	—
DATE-P	5	—
TIME2SEC	3	—
TIME2SECP	4	—
SEC2TIME	3	—
SEC2TIMEP	4	—
DATE2SEC	3	—
DATE2SECP	4	—
DATE2SEC_U	3	—
DATE2SECP_U	4	—
SEC2DATE	3	—
SEC2DATEP	4	—
SEC2DATE_U	3	—
SEC2DATEP_U	4	—
LDDT=	4	—
LDDT<>	4	—
LDDT>	4	—
LDDT<=	4	—
LDDT<	4	—
LDDT>=	4	—
ANDDT=	4	—
ANDDT<>	4	—
ANDDT>	4	—

指令名	基本步數	子集可否
ANDDT<=	4	—
ANDDT<	4	—
ANDDT>=	4	—
ORDT=	4	—
ORDT<>	4	—
ORDT>	4	—
ORDT<=	4	—
ORDT<	4	—
ORDT>=	4	—
LDTM=	4	—
LDTM<>	4	—
LDTM>	4	—
LDTM<=	4	—
LDTM<	4	—
LDTM>=	4	—
ANDTM=	4	—
ANDTM<>	4	—
ANDTM>	4	—
ANDTM<=	4	—
ANDTM<	4	—
ANDTM>=	4	—
ORTM=	4	—
ORTM<>	4	—
ORTM>	4	—
ORTM<=	4	—
ORTM<	4	—
ORTM>=	4	—
TCMP	6	—
TCMPP	7	—
TZCP	5	—
TZCPP	6	—
S. DATERD	2	—
SP. DATERD	3	—
S. DATE+	4	—
SP. DATE+	5	—
S. DATE-	4	—
SP. DATE-	5	—
DUTY	5	—
TIMCHK	5	—
HOURM	5	—
DHOURM	5	—
RFS	3	—
RFSP	4	—
COM	1	—
COMP	2	—
S. ZCOM	2	—
SP. ZCOM	3	—
FROM	5	○
FROMP	6	○
DFROM	5	○
DFROMP	6	○
TO	5	○
TOP	6	○

指令名	基本步數	子集可否
DT0	5	○
DTOP	6	○
FROMD	5	○
FROMDP	6	○
DFROMD	5	○
DFROMDP	6	○
TOD	5	○
TODP	6	○
DTOD	5	○
DTODP	6	○
TYPED	3	—
TYPEDP	4	—
UNIINFRD	4	—
UNIINFRDP	5	—
S. RTREAD	3	—
SP. RTREAD	4	—
S. RTWRITE	3	—
SP. RTWRITE	4	—
LOGTRG	3	—
LOGTRGR	3	—
DATATRG	3	—
SP. SOCOPEN	5	—
SP. SOCCLOSE	6	—
SP. SOCRCV	7	—
S. SOCRCVS	4	—
SP. SOCSND	7	—
SP. SOCCINF	6	—
SP. SOCCSET	5	—
SP. SOCRMDE	5	—
S. SOCRDATA	6	—
SP. SOCRDATA	7	—
SP. ECPRTCL	7	—
SP. SLMPSND	6	—
SP. FTPPUT	7	—
SP. FTPGET	7	—
PSTOP	2+檔案名的字元數	—
PSTOPP	3+檔案名的字元數	—
POFF	2+檔案名的字元數	—
POFFP	3+檔案名的字元數	—
PSCAN	2+檔案名的字元數	—
PSCANP	3+檔案名的字元數	—
PID	5	—
S. PIDINIT	2	—
SP. PIDINIT	3	—
S. PIDCONT	2	—
SP. PIDCONT	3	—
S. PIDSTOP	2	—
SP. PIDSTOP	3	—
S. PIDRUN	2	—
SP. PIDRUN	3	—
S. PIDPRMW	3	—
SP. PIDPRMW	4	—
PIDINIT	2	—

指令名	基本步數	子集可否
PIDINITP	3	—
PIDCONT	2	—
PIDCONTP	3	—
PIDSTOP	2	—
PIDSTOPP	3	—
PIDRUN	2	—
PIDRUNP	3	—
PIDPRMW	3	—
PIDPRMWP	4	—
D. DDRD	9	—
DP. DDRD	10	—
D. DDWR	9	—
DP. DDWR	10	—
M. DDRD	9	—
MP. DDRD	10	—
M. DDWR	9	—
MP. DDWR	10	—
LD [S□]	2	—
LD [BL□\S□]	3	—
LDI [S□]	2	—
LDI [BL□\S□]	3	—
AND [S□]	2	—
AND [BL□\S□]	3	—
ANI [S□]	2	—
ANI [BL□\S□]	3	—
OR [S□]	2	—
OR [BL□\S□]	3	—
ORI [S□]	2	—
ORI [BL□\S□]	3	—
LD [BL□]	2	—
LDI [BL□]	2	—
AND [BL□]	2	—
ANI [BL□]	2	—
OR [BL□]	2	—
ORI [BL□]	2	—
MOV [K4S□]	3	—
MOV [BL□\K4S□]	4	—
MOVP [K4S□]	4	—
MOVP [BL□\K4S□]	5	—
DMOV [K8S□]	3	—
DMOV [BL□\K8S□]	4	—
DMOVP [K8S□]	4	—
DMOVP [BL□\K8S□]	5	—
BMOV [K4S□]	4	—
BMOV [BL□\K4S□]	5	—
BMOVP [K4S□]	5	—
BMOVP [BL□\K4S□]	6	—
SET [BL□]	2	—
RST [BL□]	2	—
PAUSE [BL□]	2	—
RSTART [BL□]	2	—
SET [S□]	2	—
SET [BL□\S□]	3	—

指令名	基本步數	子集可否
RST [S□]	2	—
RST [BL□\S□]	3	—
BRSET	2	—
SP. CONTSW	4	—
DCONTSW	1	—
ECONTSW	1	—
CONTWR	5	—
CONTWRP	6	—
SP. SIDRD	6	—

*1 PHASE指令將加入2步，因此在PHASE指令後的梯形圖程式上使用指定步No. 功能時，需要指定考慮了加入2步(合計15步)的步No.。
 (☞ 198頁 階段處理指令)

附3 PID的3常數的求出方法

PID運算指令的自動調諧功能有限位週期法和步響應法兩個類型的方法。以下說明每個方法的概要。

限位週期法的概要

對求出輸入值的振幅(a)、振動週期(τ, τ_{on})，根據以下“動作特性和3常數”的公式，計算出比例增益(K_p)、積分時間(T_I)、微分時間(T_D)的方法的限位週期法進行說明。

■限位週期法

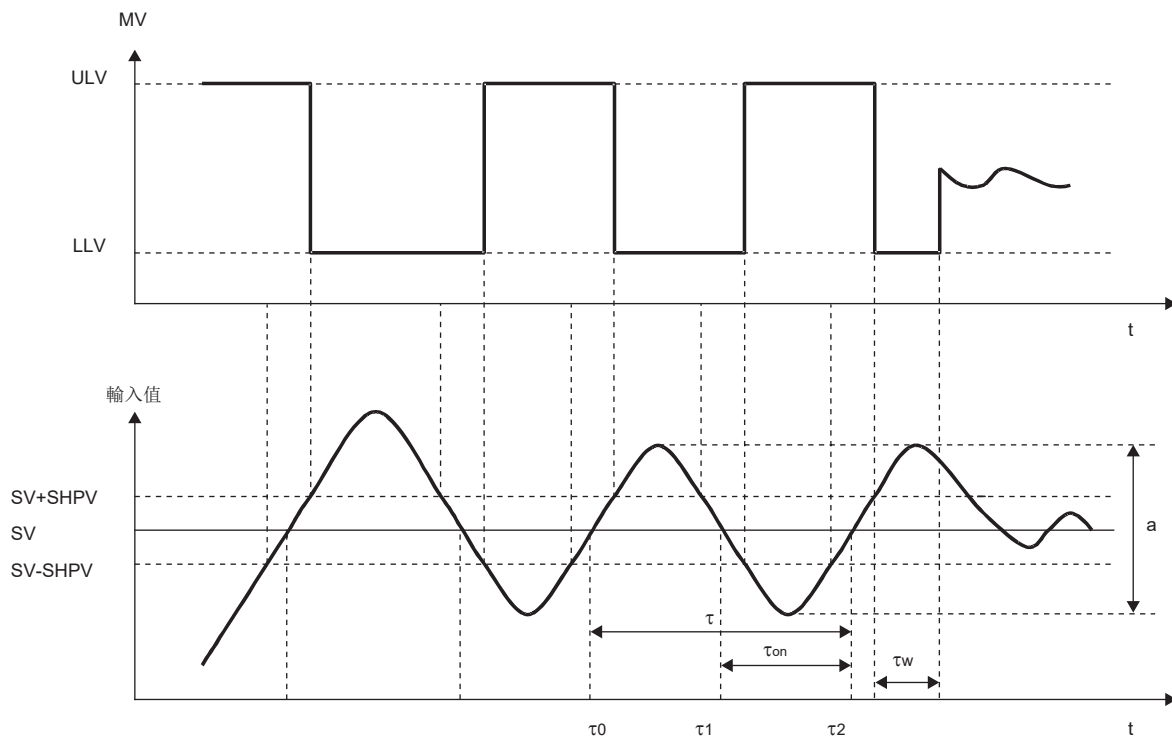
是進行二位值控制(根據偏差切換並輸出輸出上限值(ULV)和輸出下限值(LLV))時，測定輸入值的變化、求出PID的3常數的方法。

■動作特性(逆動作的示例)

調諧週期完成後，τ_w之間輸出值(MV)保持輸出下限值(LLV)並移動到普通的PID控制。

τ根據 $\tau = (50 + K_w) / 100 \times (\tau - \tau_{on})$ 求出，等待設定參數(K_w)可以設定為(s3)+28。

(設定範圍K_w=-50~32717[%]，指定異常範圍時作為τ_w=0動作)



ULV: 輸出上限值
 LLV: 輸出下限值
 SV: 目標值
 t: 時間
 SHPV: PV值開始(滯後)幅

■動作特性和3常數

控制形態	比例增益(K _p) [%]	積分時間(T _I) [×100ms]	微分時間(T _D) [×10ms]
僅比例控制(P動作)	$\frac{1}{a} (ULV-LLV) \times 100$	—	—
PI控制(PI動作)	$\frac{0.9}{a} (ULV-LLV) \times 100$	$33 \times \tau_{on} \left(1 - \frac{\tau_{on}}{\tau}\right)$	—
PID控制(PID動作)	$\frac{1.2}{a} (ULV-LLV) \times 100$	$20 \times \tau_{on} \left(1 - \frac{\tau_{on}}{\tau}\right)$	$50 \times \tau_{on} \left(1 - \frac{\tau_{on}}{\tau}\right)$

步響應法的概要

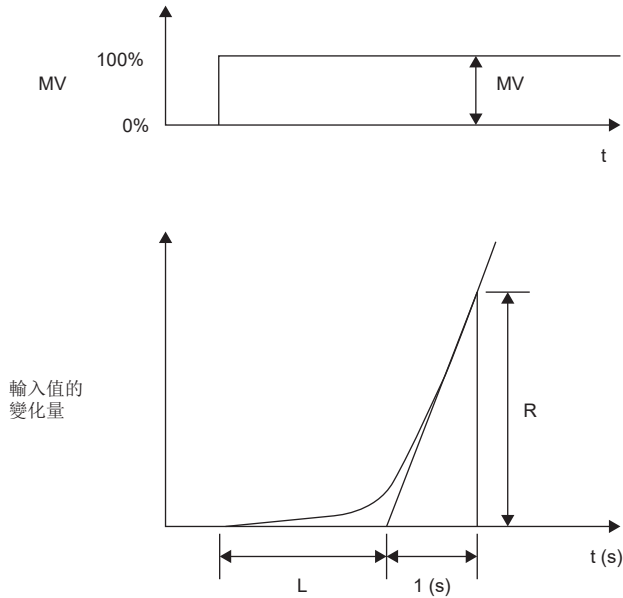
對求出PID的3常數(比例增益(K_p)、積分時間(T_I)、微分時間(T_D))的最佳值的方法的步響應法進行說明。

■步響應法

是對於控制系統，透過給予0→100%*1的步狀的輸出，並從輸入變化求出的動作特性(最大傾斜(R)、空載時間(L))求出PID的3常數的方法。

*1 步狀的輸出，即使是0→75%或0→50%也可以求出。

■動作特性



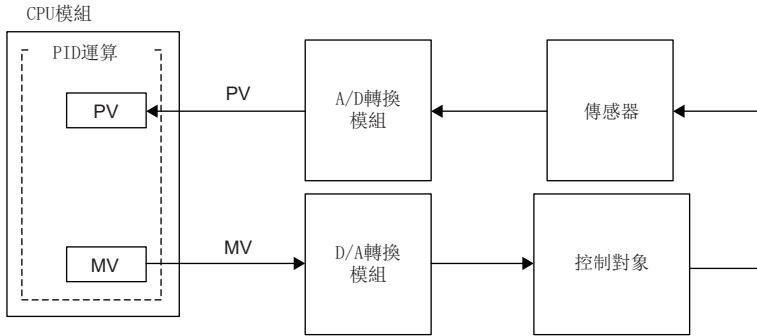
MV: 輸出值
L: 空載時間
R: 最大傾斜
t: 時間

■動作特性和3常數

控制形態	比例增益(K_p) [%]	積分時間(T_I) [$\times 100\text{ms}$]	微分時間(T_D) [$\times 10\text{ms}$]
僅比例控制(P動作)	$\frac{1}{RL} \times MV \times 100$	—	—
PI控制(PI動作)	$\frac{0.9}{RL} \times MV \times 100$	33L	—
PID控制(PID動作)	$\frac{1.2}{RL} \times MV \times 100$	20L	50L

附4 PID運算程式示例

以下介紹使用了PID運算指令的PID運算程式示例。
程式示例的系統配置如下所示。

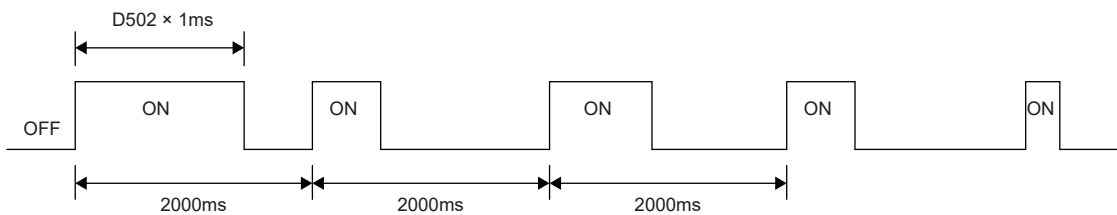


- X2: 自動調諧指令
- X3: PID控制指令
- Y20: 出錯顯示
- Y21: 加熱器輸出

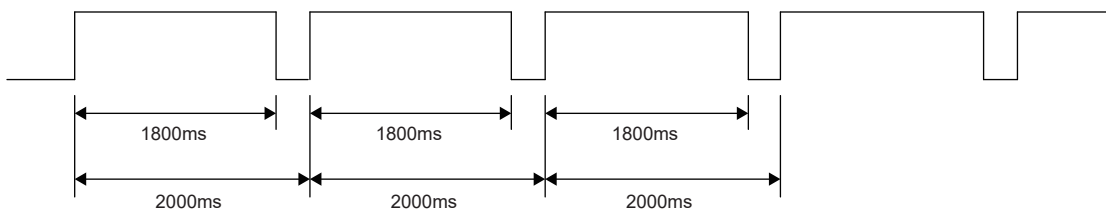
項目		自動調諧中	PID控制中
目標值	(s1)	500 (+50°C)	500 (+50°C)
控制用資料	採樣時間 (T _S)	3000ms	500ms
	輸入濾波器 (α)	70%	70%
	微分增益 (K _D)	0%	0%
	輸出值上限	2000 (2秒)	2000 (2秒)
	輸出值下限	0	0
	動作設定 (ACT)	輸入變化量報警 (s3)+1 位元1	無
	輸出變化量報警 (s3)+1 位元2	無	無
	輸出值上下限設定 (s3)+1 位元5	有	有
輸出值	(d)	1800	根據運算

輸出值的動作

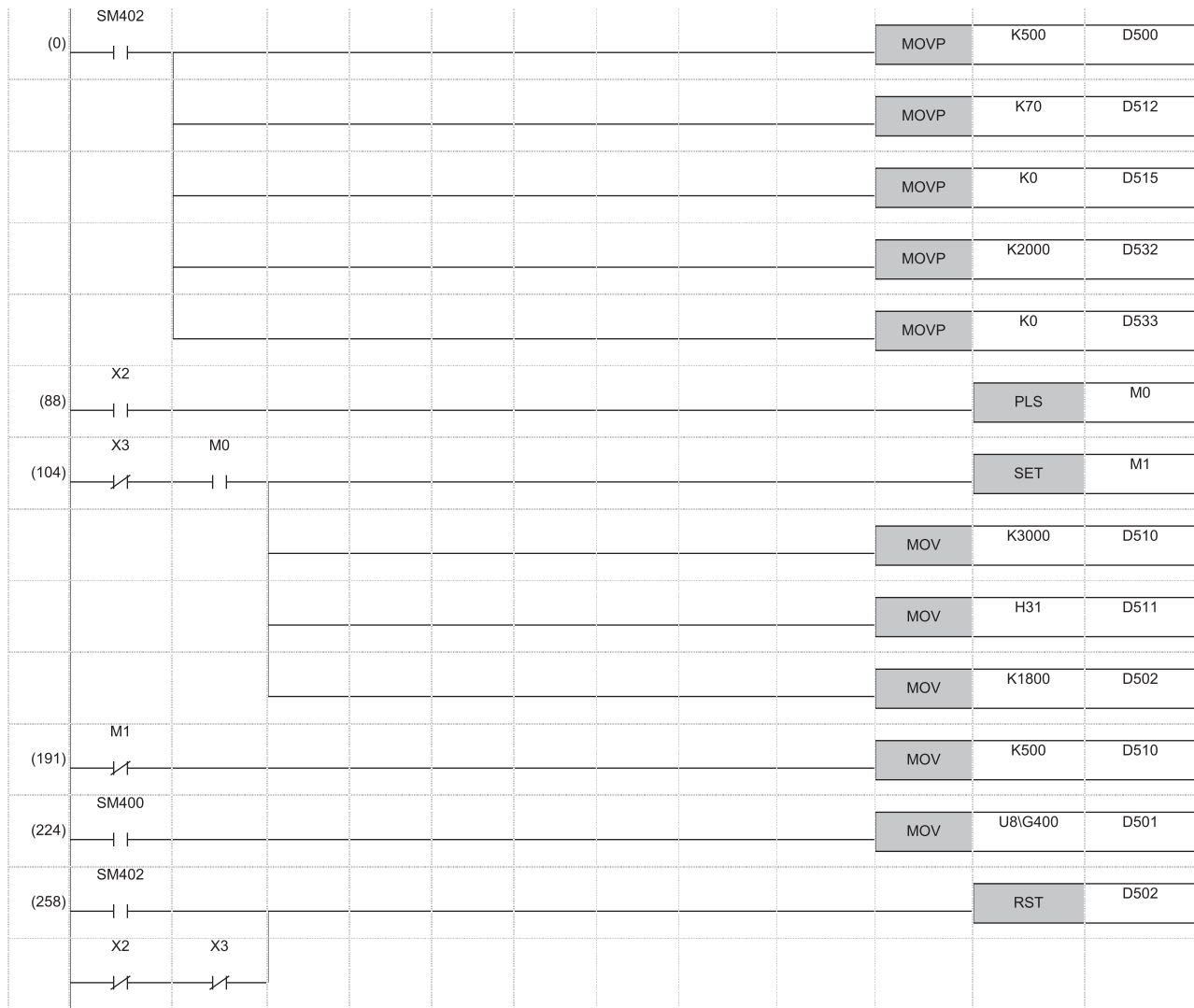
■PID控制時



■自動調諧時：最大輸出90%的情況下

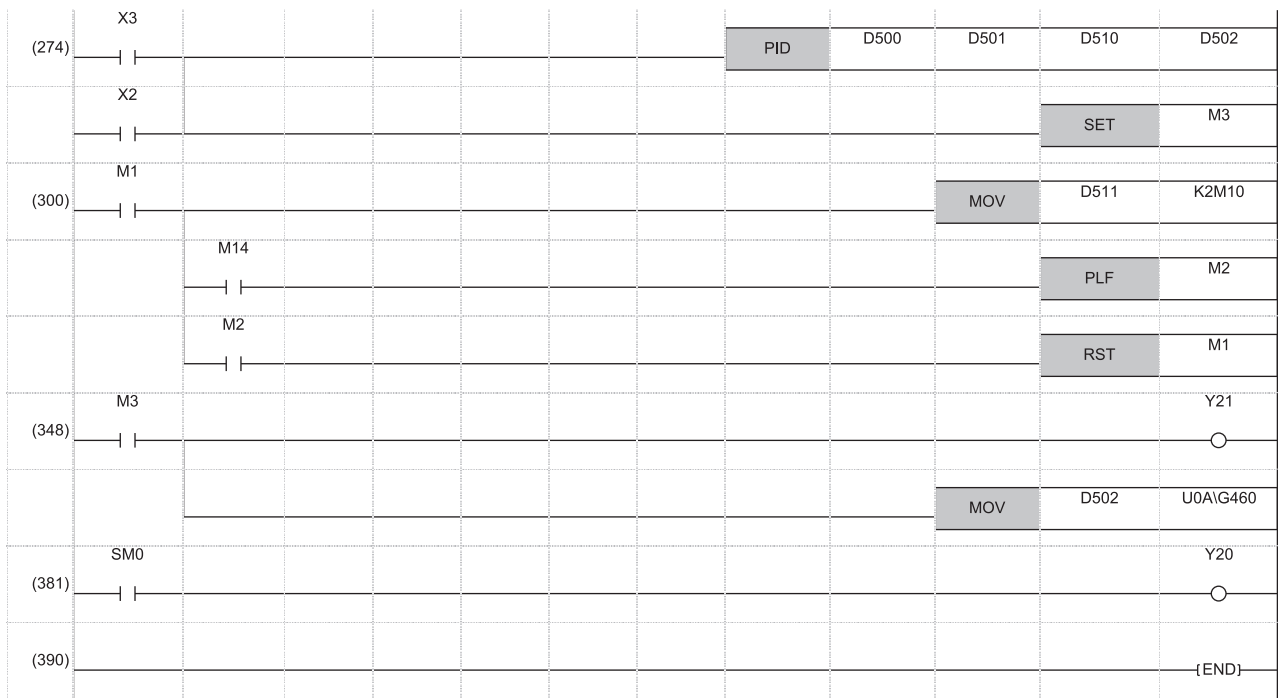


自動調諧(步響應法)+PID控制的程式示例



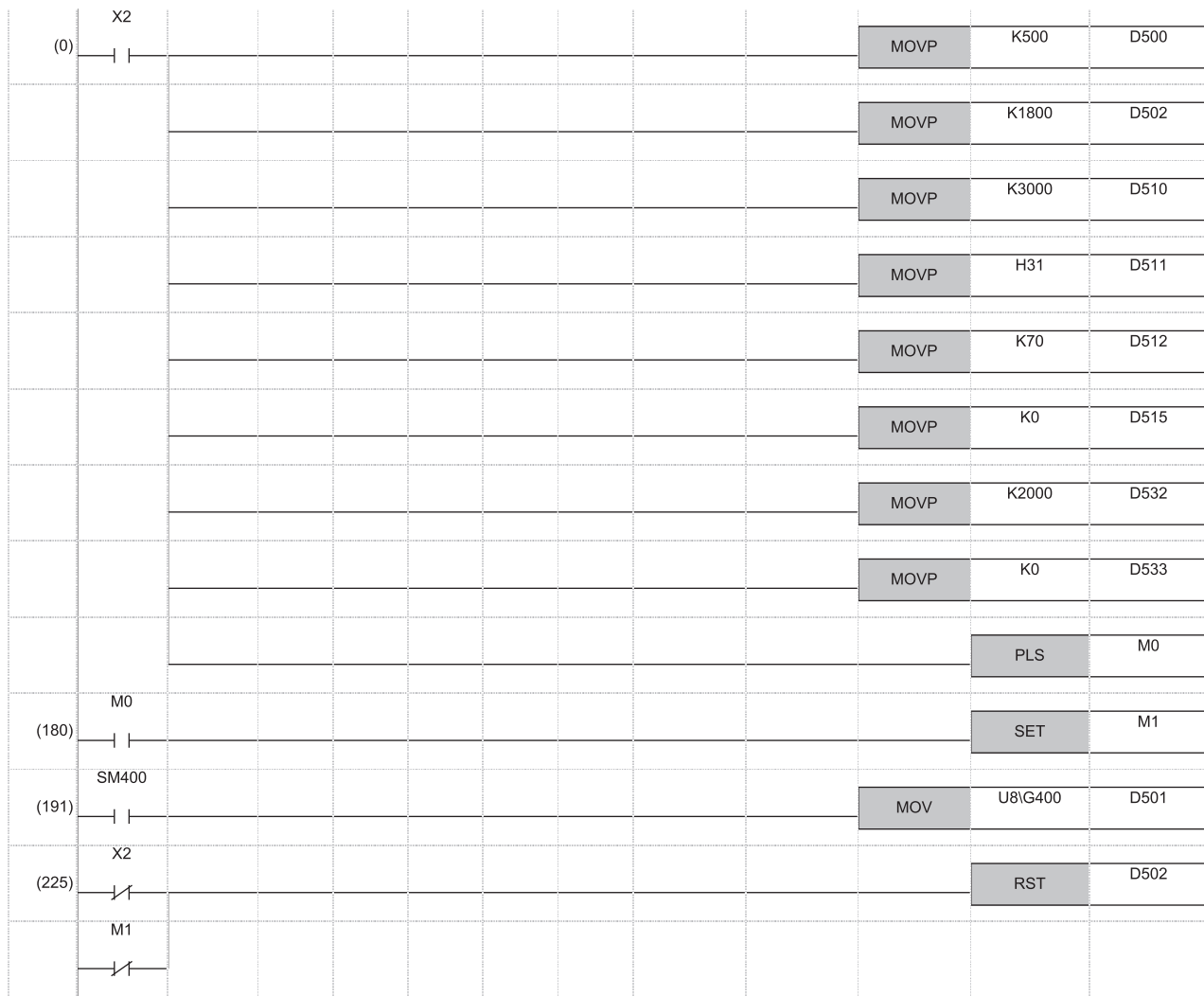
- (0) 目標值設定<50°C>
輸入濾波器(α)設定
微分增益(K_D)設定<0%>
輸出值上限設定<2秒ON>
輸出值下限設定<0秒ON>
- (88) 自動調諧開始
- (104) 自動調諧動作標誌
自動調諧用採樣時間(T_S)設定<3秒>
動作設定(ACT)自動調諧開始
自動調諧用輸出值設定<1.8秒ON>
- (191) 普通動作時的採樣時間(T_S)設定<500ms>
- (224) 根據A/D轉換模組在輸入輸出資料區域中設定測定值(PV)
- (258) PID動作初始化

附

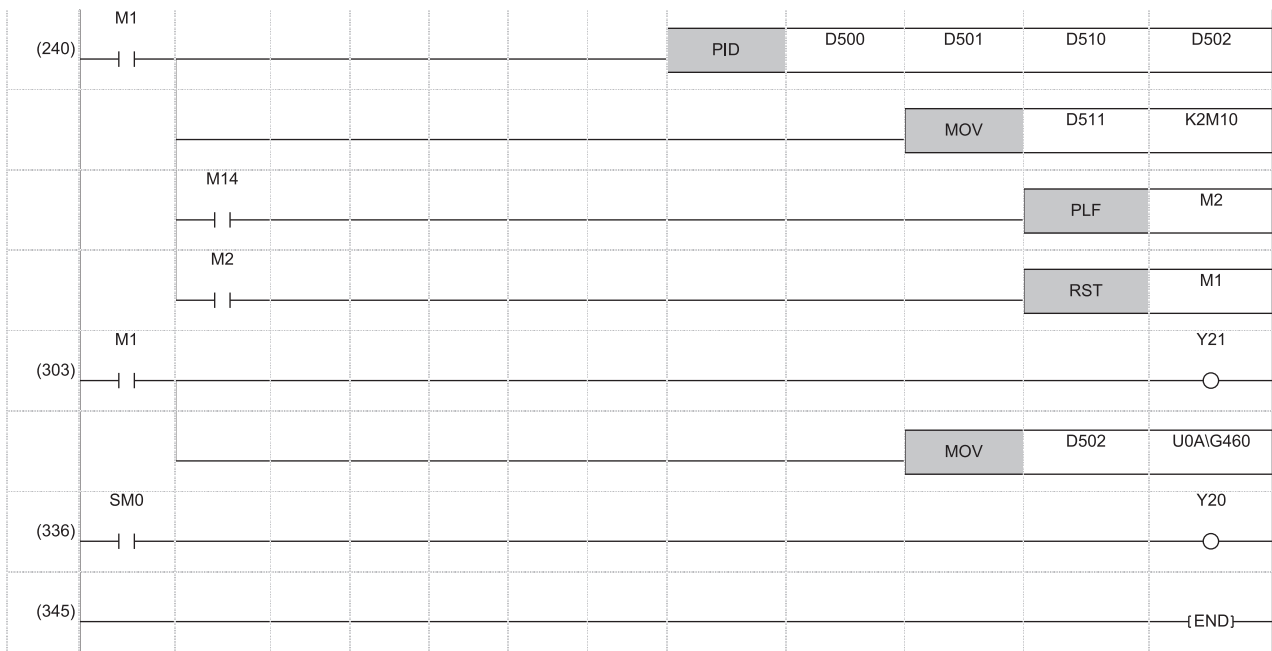


- (274) 執行PID指令時
PID動作中
- (300) 自動調諧動作確認
自動調諧結束
轉移到普通動作
- (348) 加熱器輸出
寫入輸出值(MV)至D/A轉換模組
- (381) 出錯發生

只有自動調諧(步響應法)的程式示例



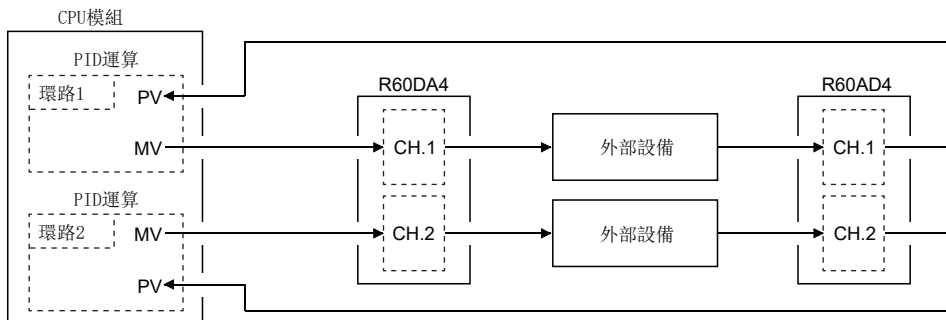
- (0) 目標值設定<50℃>
 自動調諧用輸出值設定<1.8秒ON>
 採樣時間(T_S)設定<3秒>
 動作設定(ACT)自動調諧開始
 輸入濾波器(α)設定
 微分增益(K_D)設定<0%>
 輸出值上限設定<2秒ON>
 輸出值下限設定<0秒ON>
 自動調諧開始
- (180) PID指令動作
- (191) 根據A/D轉換模組在輸入輸出資料區域中設定測定值(PV)
- (225) PID動作初始化



- (240) 執行PID指令時
自動調諧動作確認
自動調諧結束
轉移到普通動作
- (303) 加熱器輸出
寫入輸出值(MV)至D/A轉換模組
- (336) 出錯發生

附5 PID控制程式示例

以下介紹使用了PID控制指令的PID控制程式示例。
程式示例的系統配置如下所示。



R60AD4的輸入輸出編號：X/Y80～X/Y8F，R60DA4的輸入輸出編號：X/YA0～X/YAF

透過自動模式進行的PID控制程式示例

以下為將從R60AD4獲取的數位值作為測定值(PV)進行PID運算、再將PID運算求出的操作量(MV)透過R60DA4進行輸出，最後對外部設備進行控制的程式示例。

程式條件如下所示。

項目		不完全微分	完全微分
進行PID運算的環路數		2個	
採樣週期		1秒	
設定PID控制用資料的元件	通用資料	D500～D501	
	環路1用資料	D502～D515	D502～D511
	環路2用資料	D516～D529	D512～D521
設定輸入輸出資料的元件	通用資料	D600～D609	
	環路1用資料	D610～D632	D610～D627
	環路2用資料	D633～D655	D628～D645
設定值(SV)	環路1	600	
	環路2	1000	
PID控制的開始/中止時使用的元件	PID控制開始指令	X0	
	PID控制中止指令	X1	
R60AD4、R60DA4的數位值		0～2000	

(462)	SM402					S.PIDINIT	D500
(465)	X0A0					MOV	K0 U0AIG500
						MOV	K0 U0AIG700
						SET	Y0A9
(476)	X0A0	X0A9	Y0A9			RST	Y0A9
(480)	SM402					MOV	K0 D600
						MOV	K600 D610
						MOV	K0 D615
						MOV	K1000 D633
						MOV	K0 D638
(553)	X0					SET	M0
(566)	X80	M0				MOV	U8IG400 D611
						MOV	U8IG600 D634
(642)	M0					S.PIDCONT	D600
(652)	X0A0	M0				SET	Y0A1
						SET	Y0A2
						MOV	D612 U0AIG460
						MOV	D635 U0AIG660
(766)	X1					RST	M0
(777)							{ END }

- (462) 對D500~D529中設定的PID控制用資料進行設定。
(465)~(476) 將R60DA4設定為允許輸出。
(480) 在輸入輸出資料中設定通用資料。
對環路1設定輸入輸出資料。
對環路2設定輸入輸出資料。

(214)	SM402									MOV	K1	D512
										MOV	K100	D513
										MOV	K100	D514
										MOV	K30000	D515
										MOV	K0	D516
										MOV	K0	D517
										MOV	K0	D518
										MOV	K2000	D519
										MOV	K2000	D520
										MOV	K2000	D521
(386)	SM402									PIDINIT		D500
(389)	X0A0									MOV	K0	U0AIG500
										MOV	K0	U0AIG700
										SET		Y0A9
(400)	X0A0	X0A9	Y0A9							RST		Y0A9

- (214) 對環路2設定PID控制用資料。
(386) 對D500~D521中設定的PID控制用資料進行設定。
(389)~(400) 將R60DA4設定為允許輸出。

(404)	SM402					MOV	K0	D600
						MOV	K600	D610
						MOV	K0	D615
						MOV	K1000	D628
						MOV	K0	D633
(477)	X0					SET	M0	
(490)	X80	M0				MOV	U8\G400	D611
						MOV	U8\G600	D629
(566)	M0					PIDCONT	D600	
(576)	X0A0	M0				SET	Y0A1	
						SET	Y0A2	
						MOV	D612	U0A\G460
						MOV	D630	U0A\G660
(690)	X1					RST	M0	
(701)								[END]

- (404) 在輸入輸出資料中設定通用資料。
 對環路1設定輸入輸出資料。
 對環路2設定輸入輸出資料。

模式切換時的程式示例

是在切換自動模式、手動模式的同時進行PID運算的程式示例。

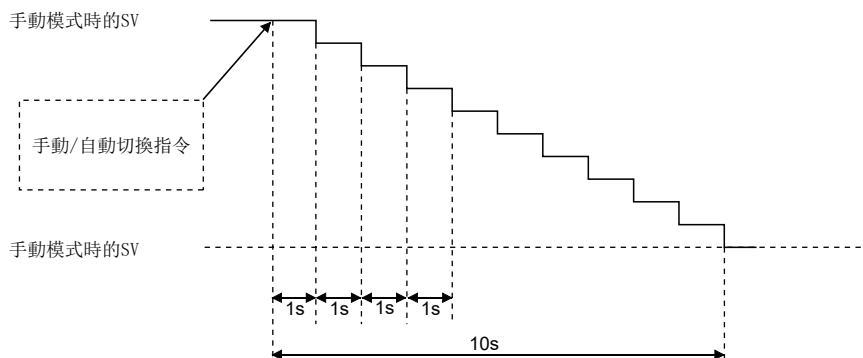
程式條件如下所示。

項目		不完全微分	完全微分
進行PID運算的環路數		1個	
採樣週期		1秒	
設定PID控制用資料的元件	通用資料	D500~D501	
	環路1用資料	D502~D515	D502~D511
設定輸入輸出資料的元件	通用資料	D600~D609	
	環路1用資料	D610~D630	D610~D627
透過外部數位開關設定的設定值 (SV)，及手動模式時的操作量 (MV)	設定值 (SV)	X30~X3F	
	手動模式時的操作量 (MV)	X20~X2F	
PID控制的開始/中止時使用的元件	PID控制開始指令	X0	
	PID控制中止指令	X1	
自動/手動切換指令中使用的元件	設定值 (SV) 設定指令	X3	
	手動模式時的操作量 (MV) 設定指令	X4	
	自動/手動切換指令	X6 (OFF: 自動模式、ON: 手動模式)	
R60AD4、R60DA4的數位值		0~2000	
SM792、SM794		OFF	

要點

從自動模式切換為手動模式時，PID運算時自動將設定值 (SV) 改寫為測定值 (PV)，因此從手動模式返回為自動模式的情況下，將設定值 (SV) 改寫為切換為手動模式前自動模式控制時的值。

但是，不對設定值 (SV) 進行1次改寫，而是如下所示分為10次逐步進行。



設定值 (SV) 改寫中的運算方法如下所示。

$$\frac{\left(SV_m \right) - \left(SV_a \right)}{10} = \boxed{\text{減法運算值}} \cdots \cdots \boxed{\text{餘數}}$$

SV_m : 手動模式時的SV

SV_a : 自動模式時的SV

對於設定值 (SV) 的減法運算，每1秒對上式的減法運算值進行減法運算。在第1次的減法運算時同時減去餘數。

不完全微分的程式示例

(0)	SM402				MOV	K1	D500
					MOV	K1	D501
					MOV	K0	D502
					MOV	K100	D503
					MOV	K100	D504
					MOV	K30000	D505
					MOV	K0	D506
					MOV	K0	D507
					MOV	K0	D508
					MOV	K2000	D509
					MOV	K2000	D510
					MOV	K2000	D511
					MOV	K0	D512
					MOV	K800	D513
					MOV	K0	D514
					MOV	K0	D515
(252)	SM402				S.PIDINT		D500
(255)	X0A0				MOV	K0	U0AIG500
					MOV	K0	U0AIG700
					SET		Y0A9
(266)	X0A0	X0A9	Y0A9		RST		Y0A9

- (0) 在PID控制用資料中設定通用資料。
對環路1設定PID控制用資料。
- (252) 對D500~D515中設定的PID控制用資料進行設定。
- (255) 將R60DA4設定為允許輸出。

完全微分的程式示例

(0)	SM402										MOV	K1	D500
											MOV	K1	D501
											MOV	K0	D502
											MOV	K100	D503
											MOV	K100	D504
											MOV	K30000	D505
											MOV	K0	D506
											MOV	K0	D507
											MOV	K0	D508
											MOV	K2000	D509
											MOV	K2000	D510
											MOV	K2000	D511
(214)	SM402											PIDINIT	D500
(217)	X0A0										MOV	K0	U0AIG500
											MOV	K0	U0AIG700
												SET	Y0A9
(228)	X0A0	X0A9	Y0A9									RST	Y0A9
(232)	SM402										MOV	K0	D600

- (0) ~ (214) 在PID控制用資料中設定通用資料。
對環路1設定PID控制用資料。
- (217) ~ (232) 對D500~D511中設定的PID控制用資料進行設定。
將R60DA4設定為允許輸出。

附6 從其他格式工程的替換

從GX Works2格式工程的替換

將GX Works2格式工程當成GX Works3工程使用的情況下，下述的指令將會自動替換成相容用指令/函數/功能塊(FB)。*1

GX Works3只要是“1.047Z”及以後的版本皆可替換相容用指令/函數。

GX Works3只要是“1.055H”及以後的版本皆可替換相容用FB。

*1 引數的資料類型，其順序與GX Works2相同。

相容用指令(基本指令/應用指令)

於ST(包含內嵌ST)、結構化梯形圖/FBD的程式中所使用的基本指令/應用指令被替換為相容用指令的。

GX Works2指令名	相容用指令名	使用程式		參閱目標
		ST	FBD	
*	MULTI_M2	×	○	245頁 *(P) (U)
*P	MULTIP_M2	×	○	
+	PLUS_M2	×	○	231頁 +(P) (U) [操作數為3個的情況下]
+P	PLUSP_M2	×	○	
-	MINUS_M2	×	○	235頁 -(P) (U) [操作數為3個的情況下]
-P	MINUSP_M2	×	○	
/	DIVISION_M2	×	○	247頁 /(P) (U)
/P	DIVISIONP_M2	×	○	
ABRST1_M	Z_ABRST1_M2	○	○	MELSEC iQ-R 程式手冊(模組專用指令篇)
ABRST2_M	Z_ABRST2_M2	○	○	
ABRST3_M	Z_ABRST3_M2	○	○	
ABRST4_M	Z_ABRST4_M2	○	○	
ACOS	ACOS_M2	○	○	1399頁 ACOS (E)
ACOS_MD		○	○	
ACOS_E_MD		○	○	
ACOSD	ACOSD_M2	○	○	
ACOSD_MD		○	○	
ACOSD_E_MD		○	○	
AND<	AND_LT_M2	×	○	212頁 LD□ (U)、AND□ (U)、OR□ (U)
AND<=	AND_LE_M2	×	○	
AND<>	AND_NE_M2	×	○	
AND=	AND_EQ_M2	×	○	
AND>	AND_GT_M2	×	○	
AND>=	AND_GE_M2	×	○	
AND_EQ_M	AND_EQ_M2	○	○	
AND_GE_M	AND_GE_M2	○	○	
AND_GT_M	AND_GT_M2	○	○	
AND_LE_M	AND_LE_M2	○	○	
AND_LT_M	AND_LT_M2	○	○	
AND_NE_M	AND_NE_M2	○	○	

GX Works2指令名	相容用指令名	使用程式		參閱目標
		ST	FBD	
ANDD<	ANDD_LT_M2	×	○	214頁 LDD□(_U)、ANDD□(_U)、ORD□(_U)
ANDD<=	ANDD_LE_M2	×	○	
ANDD<>	ANDD_NE_M2	×	○	
ANDD=	ANDD_EQ_M2	×	○	
ANDD>	ANDD_GT_M2	×	○	
ANDD>=	ANDD_GE_M2	×	○	
ANDD_EQ_M	ANDD_EQ_M2	○	○	
ANDD_GE_M	ANDD_GE_M2	○	○	
ANDD_GT_M	ANDD_GT_M2	○	○	
ANDD_LE_M	ANDD_LE_M2	○	○	
ANDD_LT_M	ANDD_LT_M2	○	○	
ANDD_NE_M	ANDD_NE_M2	○	○	
ASIN	ASIN_M2	○	○	1398頁 ASIN(_E)
ASIN_MD		○	○	
ASIN_E_MD		○	○	
ASIND	ASIND_M2	○	○	
ASIND_MD		○	○	
ASIND_E_MD		○	○	
ATAN	ATAN_M2	○	○	1400頁 ATAN(_E)
ATAN_MD		○	○	
ATAN_E_MD		○	○	
ATAND	ATAND_M2	○	○	
ATAND_MD		○	○	
ATAND_E_MD		○	○	
BAND	BAND_M2	○	○	685頁 BAND(P)(_U)
BAND_MD		○	○	
BANDP	BANDP_M2	○	○	
BAND_P_MD		○	○	
BDSQR	BDSQR_M2	○	○	955頁 BDSQRT(P)
BDSQRP	BDSQRP_M2	○	○	
BIDIN_M	G_BIDIN_M2	○	○	MELSEC iQ-R 程式手冊(模組專用指令篇)
BIDINP_M	GP_BIDIN_M2	○	○	
BIDOUT_M	G_BIDOUT_M2	○	○	
BIDOUTP_M	GP_BIDOUT_M2	○	○	
BINDA	BINDA_M2	○	○	774頁 BINDA(P)(_U)
BINDA_MD		○	○	
BINDA_K_MD		○	○	
BINDA_S_MD		○	○	
BINDAP	BINDAP_M2	○	○	
BINDA_P_MD		○	○	
BINDA_P_S_MD		○	○	
BINDA_K_P_MD		○	○	
BK+	BKPLUS_M2	×	○	273頁 BK+(P)(_U)
BK+P	BKPLUSP_M2	×	○	
BK-	BKMINUS_M2	×	○	275頁 BK-(P)(_U)
BK-P	BKMINUSP_M2	×	○	

GX Works2指令名	相容用指令名	使用程式		參閱目標	
		ST	FBD		
BKCMPC	BKCMPC_LT_M2	×	○	224頁 BKCMPC□(P)(U)	
BKCMPC<P	BKCMPC_LTP_M2	×	○		
BKCMPC<=	BKCMPC_LE_M2	×	○		
BKCMPC<=P	BKCMPC_LEP_M2	×	○		
BKCMPC<>	BKCMPC_NE_M2	×	○		
BKCMPC<>P	BKCMPC_NEP_M2	×	○		
BKCMPC=	BKCMPC_EQ_M2	×	○		
BKCMPC=P	BKCMPC_EQP_M2	×	○		
BKCMPC>	BKCMPC_GT_M2	×	○		
BKCMPC>P	BKCMPC_GTP_M2	×	○		
BKCMPC>=	BKCMPC_GE_M2	×	○		
BKCMPC>=P	BKCMPC_GEP_M2	×	○		
BKCMPC_EQ_M	BKCMPC_EQ_M2	○	○		
BKCMPC_EQP_M	BKCMPC_EQP_M2	○	○		
BKCMPC_GE_M	BKCMPC_GE_M2	○	○		
BKCMPC_GEP_M	BKCMPC_GEP_M2	○	○		
BKCMPC_GT_M	BKCMPC_GT_M2	○	○		
BKCMPC_GTP_M	BKCMPC_GTP_M2	○	○		
BKCMPC_LE_M	BKCMPC_LE_M2	○	○		
BKCMPC_LEP_M	BKCMPC_LEP_M2	○	○		
BKCMPC_LT_M	BKCMPC_LT_M2	○	○		
BKCMPC_LTP_M	BKCMPC_LTP_M2	○	○		
BKCMPC_NE_M	BKCMPC_NE_M2	○	○		
BKCMPC_NEP_M	BKCMPC_NEP_M2	○	○		
BKMINUS_M	BKMINUS_M2	○	○		275頁 BK-(P)(U)
BKMINUSP_M	BKMINUSP_M2	○	○		273頁 BK+(P)(U)
BKPLUS_M	BKPLUS_M2	○	○		
BKPLUSP_M	BKPLUSP_M2	○	○	339頁 BKRST(P)	
BKRST	BKRST_M2	○	○		
BKRST_M		○	○		
BKRSTP	BKRSTP_M2	○	○		
BKRSTP_M		○	○	953頁 BSQR(P)	
BSQR	BSQR_M2	○	○		
BSQR_P	BSQR_P_M2	○	○		
BUFRVCV_M	ZP_BUFRVCV_M2	○	○	MELSEC iQ-R 程式手冊(模組專用指令篇)	
BUFSND_M	ZP_BUFSND_M2	○	○		
CLOSE_M	ZP_CLOSE_M2	○	○		
COS	COS_M2	○	○	1396頁 COS(U)	
COS_MD		○	○		
COS_E_MD		○	○		
COSD	COSD_M2	○	○		
COSD_MD		○	○		
COSD_E_MD		○	○		
CSET_M	ZP_CSET_M2	○	○	MELSEC iQ-R 程式手冊(模組專用指令篇)	
CSET_P_M		○	○		
D*	DMULTI_M2	×	○	249頁 D*(P)(U)	
D*P	DMULTIP_M2	×	○	239頁 D+(P)(U) [操作數為3個的情況下]	
D+	DPLUS_M2	×	○		
D+P	DPLUSP_M2	×	○		
D-	DMINUS_M2	×	○	243頁 D-(P)(U) [操作數為3個的情況下]	
D-P	DMINUSP_M2	×	○		

GX Works2指令名	相容用指令名	使用程式		參閱目標	
		ST	FBD		
D/	DDIVISION_M2	×	○	251頁 D/(P) (U)	
D/P	DDIVISIONP_M2	×	○		
DABIN	DABIN_M2	○	○	465頁 DABIN(P) (U)	
DABIN_MD		○	○		
DABIN_S_MD		○	○		
DABINP	DABINP_M2	○	○		
DABIN_P_MD		○	○		
DABIN_P_S_MD		○	○		
DB*	DBMULTI_M2	×	○		269頁 DB*(P)
DB*P	DBMULTIP_M2	×	○		
DBAND	DBAND_M2	○	○	687頁 DBAND(P) (U)	
DBAND_MD		○	○		
DBANDP	DBANDP_M2	○	○		
DBAND_P_MD		○	○		
DBINDA	DBINDA_M2	○	○	778頁 DBINDA(P) (U)	
DBINDA_MD		○	○		
DBINDA_K_MD		○	○		
DBINDA_S_MD		○	○		
DBINDAP	DBINDAP_M2	○	○		
DBINDA_P_MD		○	○		
DBINDA_P_S_MD		○	○		
DBINDA_K_P_MD		○	○		
DBK+	DBKPLUS_M2	×	○	277頁 DBK+(P) (U)	
DBK+P	DBKPLUSP_M2	×	○		
DBK-	DBKMINUS_M2	×	○	280頁 DBK-(P) (U)	
DBK-P	DBKMINUSP_M2	×	○		
DBKCM<	DBKCM_LT_M2	×	○	226頁 DBKCM□(P) (U)	
DBKCM<P	DBKCM_LTP_M2	×	○		
DBKCM<=	DBKCM_LE_M2	×	○		
DBKCM<=P	DBKCM_LEP_M2	×	○		
DBKCM<>	DBKCM_NE_M2	×	○		
DBKCM<>P	DBKCM_NEP_M2	×	○		
DBKCM=	DBKCM_EQ_M2	×	○		
DBKCM=P	DBKCM_EQP_M2	×	○		
DBKCM>	DBKCM_GT_M2	×	○		
DBKCM>P	DBKCM_GTP_M2	×	○		
DBKCM>=	DBKCM_GE_M2	×	○		
DBKCM>=P	DBKCM_GEP_M2	×	○		
DBKCM_EQ_M	DBKCM_EQ_M2	○	○		
DBKCM_EQP_M	DBKCM_EQP_M2	○	○		
DBKCM_GE_M	DBKCM_GE_M2	○	○		
DBKCM_GEP_M	DBKCM_GEP_M2	○	○		
DBKCM_GT_M	DBKCM_GT_M2	○	○		
DBKCM_GTP_M	DBKCM_GTP_M2	○	○		
DBKCM_LE_M	DBKCM_LE_M2	○	○		
DBKCM_LEP_M	DBKCM_LEP_M2	○	○		
DBKCM_LT_M	DBKCM_LT_M2	○	○		
DBKCM_LTP_M	DBKCM_LTP_M2	○	○		
DBKCM_NE_M	DBKCM_NE_M2	○	○		
DBKCM_NEP_M	DBKCM_NEP_M2	○	○		
DBKMINUS_M	DBKMINUS_M2	○	○		280頁 DBK-(P) (U)
DBKMINUSP_M	DBKMINUSP_M2	○	○		

GX Works2指令名	相容用指令名	使用程式		參閱目標
		ST	FBD	
DBKPLUS_M	DBKPLUS_M2	○	○	277頁 DBK+(P) (_U)
DBKPLUSP_M	DBKPLUSP_M2	○	○	
DBMULTI_M	DBMULTI_M2	○	○	269頁 DB*(P)
DBMULTIP_M	DBMULTIP_M2	○	○	
DDABIN	DDABIN_M2	○	○	468頁 DDABIN(P) (_U)
DDABIN_MD		○	○	
DDABIN_S_MD		○	○	
DDABINP	DDABINP_M2	○	○	
DDABIN_P_MD		○	○	
DDABIN_P_S_MD		○	○	
DDEC	DDEC_M2	○	○	289頁 DDEC(P) (_U)
DDEC_M		○	○	
DDECP	DDECP_M2	○	○	
DDECP_M		○	○	
DDIVID_3_M	DDIVISION_M2	○	○	251頁 D/(P) (_U)
DDIVIDP_3_M	DDIVISIONP_M2	○	○	
DEC	DEC_M2	○	○	285頁 DEC(P) (_U)
DEC_M		○	○	
DECP	DECP_M2	○	○	
DECP_M		○	○	
DGBIN	DGBIN_M2	○	○	459頁 DGBIN(P) (_U)
DGBIN_M		○	○	
DGBINP	DGBINP_M2	○	○	
DGBINP_M		○	○	
DGRY	DGRY_M2	○	○	455頁 DGRY(P) (_U)
DGRY_M		○	○	
DGRYP	DGRYP_M2	○	○	
DGRYP_M		○	○	
DINC	DINC_M2	○	○	287頁 DINC(P) (_U)
DINC_M		○	○	
DINCP	DINCP_M2	○	○	
DINCP_M		○	○	
DIVID_3_M	DIVISION_M2	○	○	247頁 /(P) (_U)
DIVIDP_3_M	DIVISIONP_M2	○	○	
DLIMIT	DLIMIT_M2	○	○	1432頁 LIMIT(_E)
DLIMIT_MD		○	○	
DLIMITP	DLIMITP_M2	○	○	683頁 DLIMIT(P) (_U)
DLIMIT_P_MD		○	○	
DMAX	DMAX_M2	○	○	1430頁 MAX(_E)、MIN(_E)
DMAX_M		○	○	
DMAXP	DMAXP_M2	○	○	719頁 DMAX(P) (_U)
DMAXP_M		○	○	
DMEAN	DMEAN_M2	○	○	747頁 DMEAN(P) (_U)
DMEAN_M		○	○	
DMEANP	DMEANP_M2	○	○	
DMEANP_M		○	○	
DMIN	DMIN_M2	○	○	1430頁 MAX(_E)、MIN(_E)
DMIN_M		○	○	
DMINP	DMINP_M2	○	○	723頁 DMIN(P) (_U)
DMINP_M		○	○	
DMINUS_M	DMINUS_2_M2	○	○	241頁 D-(P) (_U) [操作數為2個的情況下]
DMINUS_3_M	DMINUS_M2	○	○	243頁 D-(P) (_U) [操作數為3個的情況下]

GX Works2指令名	相容用指令名	使用程式		參閱目標
		ST	FBD	
DMINUSP_M	DMINUSP_2_M2	○	○	241頁 D-(P) (_U) [操作數為2個的情況下]
DMINUSP_3_M	DMINUSP_M2	○	○	243頁 D-(P) (_U) [操作數為3個的情況下]
DMULTI_3_M	DMULTI_M2	○	○	249頁 D*(P) (_U)
DMULTIP_3_M	DMULTIP_M2	○	○	
DPLUS_M	DPLUS_2_M2	○	○	237頁 D+(P) (_U) [操作數為2個的情況下]
DPLUS_3_M	DPLUS_M2	○	○	239頁 D+(P) (_U) [操作數為3個的情況下]
DPLUSP_M	DPLUSP_2_M2	○	○	237頁 D+(P) (_U) [操作數為2個的情況下]
DPLUSP_3_M	DPLUSP_M2	○	○	239頁 D+(P) (_U) [操作數為3個的情況下]
DROL	DROL_M2	○	○	611頁 DROL(P)、DRCL(P)
DROL_M		○	○	
DROR	DROR_M2	○	○	609頁 DROR(P)、DRCR(P)
DROR_M		○	○	
DSCL	DSCL_M2	○	○	696頁 DSCL(P) (_U)
DSCLP	DSCLP_M2	○	○	
DSCL2	DSCL2_M2	○	○	701頁 DSCL2(P) (_U)
DSCL2P	DSCL2P_M2	○	○	
DSORT	DSORT_M2	○	○	727頁 DSORTD (_U)
DSORT_M		○	○	
DSTR	DSTR_M2	○	○	794頁 DSTR(P) (_U)
DSTR_MD		○	○	
DSTR_K_MD		○	○	
DSTR_S_MD		○	○	
DSTRP	DSTRP_M2	○	○	
DSTR_P_MD		○	○	
DSTR_P_S_MD		○	○	
DSTR_K_P_MD		○	○	
DTO	DTO_M2	○	○	1113頁 TO(P)、DTO(P)
DTO_M		○	○	
DTOP	DTOP_M2	○	○	
DTOP_M		○	○	
DVAL	DVAL_M2	○	○	485頁 DVAL(P) (_U)
DVAL_MD		○	○	
DVAL_S_MD		○	○	
DVALP	DVALP_M2	○	○	
DVAL_P_MD		○	○	
DVAL_P_S_MD		○	○	
		○	○	
DWSUM	DWSUM_M2	○	○	743頁 DWSUM(P) (_U)
DWSUM_M		○	○	
DWSUMP	DWSUMP_M2	○	○	
DWSUMP_M		○	○	
DZONE	DZONE_M2	○	○	691頁 DZONE(P) (_U)
DZONE_MD		○	○	
DZONEP	DZONEP_M2	○	○	
DZONE_P_MD		○	○	
EREXP	EREXP_M2	○	○	891頁 EREXP(P)
EREXP_M		○	○	
EREXPP	EREXPP_M2	○	○	
EREXPP_M		○	○	
ERRCLR_M	ZP_ERRCLR_M2	○	○	MELSEC iQ-R 程式手冊(模組專用指令篇)
ERRRD_M	ZP_ERRRD_M2	○	○	

GX Works2指令名	相容用指令名	使用程式		參閱目標	
		ST	FBD		
EXP	EXP_M2	○	○	1394頁 EXP(_E)	
EXP_MD		○	○		
EXP_E_MD		○	○		
EXPD	EXPD_M2	○	○		
EXPD_MD		○	○		
EXPD_E_MD		○	○		
G_BIDIN	G_BIDIN_M2	○	○	MELSEC iQ-R 程式手冊(模組專用指令篇)	
G_BIDOUT	G_BIDOUT_M2	○	○		
G_CCPASET	G_CCPASET_M2	○	○		
G_CPRTCL	G_CPRTCL_M2	○	○		
G_GETE	G_GETE_M2	○	○		
G_INPUT	G_INPUT_M2	○	○		
G_OGLOAD	G_OGLOAD_M2	○	○		
G_OGSTOR	G_OGSTOR_M2	○	○		
G_ONDEMAND	G_ONDEMAND_M2	○	○		
G_OUTPUT	G_OUTPUT_M2	○	○		
G_PRR	G_PRR_M2	○	○		
G_PUTE	G_PUTE_M2	○	○		
G_RDMSG	G_RDMSG_M2	○	○		
G_REQ	G_REQ_M2	○	○		
G_RIRCV	G_RIRCV_M2	○	○		
G_RIRD	G_RIRD_M2	○	○		
G_RISEND	G_RISEND_M2	○	○		
G_RITO	G_RITO_M2	○	○		
G_RIWT	G_RIWT_M2	○	○		
G_RLPASET	G_RLPASET_M2	○	○		
G_SPBUSY	G_SPBUSY_M2	○	○		
GBIN	GBIN_M2	○	○		457頁 GBIN(P)(_U)
GBIN_M		○	○		
GBINP	GBINP_M2	○	○		
GBINP_M		○	○		

附

GX Works2指令名	相容用指令名	使用程式		參閱目標
		ST	FBD	
GETE_M	G_GETE_M2	○	○	MELSEC iQ-R 程式手冊 (模組專用指令篇)
GETEP_M	GP_GETE_M2	○	○	
GP_BIDIN	GP_BIDIN_M2	○	○	
GP_BIDOUT	GP_BIDOUT_M2	○	○	
GP_CCPASET	GP_CCPASET_M2	○	○	
GP_CPRTCL	GP_CPRTCL_M2	○	○	
GP_ECPRTCL	GP_ECPRTCL_M2	○	○	
GP_GETE	GP_GETE_M2	○	○	
GP_OGLOAD	GP_OGLOAD_M2	○	○	
GP_OGSTOR	GP_OGSTOR_M2	○	○	
GP_ONDEMAND	GP_ONDEMAND_M2	○	○	
GP_OUTPUT	GP_OUTPUT_M2	○	○	
GP_PRR	GP_PRR_M2	○	○	
GP_PUTE	GP_PUTE_M2	○	○	
GP_RDMSG	GP_RDMSG_M2	○	○	
GP_RECV	GP_RECV_M2	○	○	
GP_REQ	GP_REQ_M2	○	○	
GP_RIRCV	GP_RIRCV_M2	○	○	
GP_RIRD	GP_RIRD_M2	○	○	
GP_RISEND	GP_RISEND_M2	○	○	
GP_RITO	GP_RITO_M2	○	○	
GP_RIWT	GP_RIWT_M2	○	○	
GP_RLPASET	GP_RLPASET_M2	○	○	
GP_SEND	GP_SEND_M2	○	○	
GP_SPBUSY	GP_SPBUSY_M2	○	○	
GRY	GRY_M2	○	○	453頁 GRY(P) (_U)
GRY_M		○	○	
GRYP	GRYP_M2	○	○	
GRYP_M		○	○	
INC	INC_M2	○	○	283頁 INC(P) (_U)
INC_M		○	○	
INCP	INCP_M2	○	○	
INCP_M		○	○	
INPUT_M	G_INPUT_M2	○	○	MELSEC iQ-R 程式手冊 (模組專用指令篇)
LD<	LD_LT_M2	×	○	212頁 LD□(_U)、AND□(_U)、OR□(_U)
LD<=	LD_LE_M2	×	○	
LD<>	LD_NE_M2	×	○	
LD=	LD_EQ_M2	×	○	
LD>	LD_GT_M2	×	○	
LD>=	LD_GE_M2	×	○	
LD_EQ_M	LD_EQ_M2	○	○	
LD_GE_M	LD_GE_M2	○	○	
LD_GT_M	LD_GT_M2	○	○	
LD_LE_M	LD_LE_M2	○	○	
LD_LT_M	LD_LT_M2	○	○	
LD_NE_M	LD_NE_M2	○	○	

GX Works2指令名	相容用指令名	使用程式		參閱目標
		ST	FBD	
LDD<	LDD_LT_M2	×	○	214頁 LDD□(_U)、ANDD□(_U)、ORD□(_U)
LDD<=	LDD_LE_M2	×	○	
LDD<>	LDD_NE_M2	×	○	
LDD=	LDD_EQ_M2	×	○	
LDD>	LDD_GT_M2	×	○	
LDD>=	LDD_GE_M2	×	○	
LDD_EQ_M	LDD_EQ_M2	○	○	
LDD_GE_M	LDD_GE_M2	○	○	
LDD_GT_M	LDD_GT_M2	○	○	
LDD_LE_M	LDD_LE_M2	○	○	
LDD_LT_M	LDD_LT_M2	○	○	
LDD_NE_M	LDD_NE_M2	○	○	
LEFT	LEFT_M2	○	○	
LEFT_M		○	○	
LEN	LEN_M2	○	○	1441頁 LEN(_E)
LEN_MD		○	○	
LEN_S		○	○	
LEN_S_MD		○	○	
LIMIT	LIMIT_M2	○	○	681頁 LIMIT(P)(_U)
LIMIT_MD		○	○	
LIMITP	LIMITP_M2	○	○	
LIMIT_P_MD		○	○	
LOG	LOG_M2	○	○	1392頁 LOG(_E)
LOG_MD		○	○	
LOG_E_MD		○	○	
LOGD	LOGD_M2	○	○	
LOGD_MD		○	○	
LOGD_E_MD		○	○	
MAX	MAX_M2	○	○	1430頁 MAX(_E)、MIN(_E)
MAX_M		○	○	
MAXP	MAXP_M2	○	○	717頁 MAX(P)(_U)
MAXP_M		○	○	719頁 DMAX(P)(_U)
MEAN	MEAN_M2	○	○	745頁 MEAN(P)(_U)
MEAN_M		○	○	
MEANP	MEANP_M2	○	○	
MEANP_M		○	○	
MIN	MIN_M2	○	○	1430頁 MAX(_E)、MIN(_E)
MIN_M		○	○	
MINP	MINP_M2	○	○	721頁 MIN(P)(_U)
MINP_M		○	○	
MINUS_M	MINUS_2_M2	○	○	233頁 -(P)(_U) [操作數為2個的情況下]
MINUS_3_M	MINUS_M2	○	○	235頁 -(P)(_U) [操作數為3個的情況下]
MINUSP_M	MINUSP_2_M2	○	○	233頁 -(P)(_U) [操作數為2個的情況下]
MINUSP_3_M	MINUSP_M2	○	○	235頁 -(P)(_U) [操作數為3個的情況下]
MULTI_3_M	MULTI_M2	○	○	245頁 *(P)(_U)
MULTIP_3_M	MULTIP_M2	○	○	

GX Works2指令名	相容用指令名	使用程式		參閱目標
		ST	FBD	
OGLOAD_U_M	G_OGLOAD_M2	○	○	MELSEC iQ-R 程式手冊 (模組專用指令篇)
OGLOAD_UP_M	GP_OGLOAD_M2	○	○	
OGSTOR_U_M	G_OGSTOR_M2	○	○	
OGSTOR_UP_M	GP_OGSTOR_M2	○	○	
ONDEMAND_M	G_ONDEMAND_M2	○	○	
ONDEMANDP_M	GP_ONDEMAND_M2	○	○	
OPEN_M	ZP_OPEN_M2	○	○	
OR<	OR_LT_M2	×	○	212頁 LD□(U)、AND□(U)、OR□(U)
OR<=	OR_LE_M2	×	○	
OR<>	OR_NE_M2	×	○	
OR=	OR_EQ_M2	×	○	
OR>	OR_GT_M2	×	○	
OR>=	OR_GE_M2	×	○	
OR_EQ_M	OR_EQ_M2	○	○	
OR_GE_M	OR_GE_M2	○	○	
OR_GT_M	OR_GT_M2	○	○	
OR_LE_M	OR_LE_M2	○	○	
OR_LT_M	OR_LT_M2	○	○	
OR_NE_M	OR_NE_M2	○	○	
ORD<	ORD_LT_M2	×	○	
ORD<=	ORD_LE_M2	×	○	
ORD<>	ORD_NE_M2	×	○	
ORD=	ORD_EQ_M2	×	○	
ORD>	ORD_GT_M2	×	○	
ORD>=	ORD_GE_M2	×	○	
ORD_EQ_M	ORD_EQ_M2	○	○	
ORD_GE_M	ORD_GE_M2	○	○	
ORD_GT_M	ORD_GT_M2	○	○	
ORD_LE_M	ORD_LE_M2	○	○	
ORD_LT_M	ORD_LT_M2	○	○	
ORD_NE_M	ORD_NE_M2	○	○	
OUTPUT_M	G_OUTPUT_M2	○	○	MELSEC iQ-R 程式手冊 (模組專用指令篇)
OUTPUTP_M	GP_OUTPUT_M2	○	○	
PEWRT_P_M	ZP_PEWRT_M2	○	○	
PINIT_M	ZP_PINIT_M2	○	○	
PINIT_P_M		○	○	
PLUS_M	PLUS_2_M2	○	○	229頁 +(P)(U) [操作數為2個的情況下]
PLUS_3_M	PLUS_M2	○	○	231頁 +(P)(U) [操作數為3個的情況下]
PLUSP_M	PLUSP_2_M2	○	○	229頁 +(P)(U) [操作數為2個的情況下]
PLUSP_3_M	PLUSP_M2	○	○	231頁 +(P)(U) [操作數為3個的情況下]

GX Works2指令名	相容用指令名	使用程式		參閱目標	
		ST	FBD		
PRR_M	G_PRR_M2	○	○	MELSEC iQ-R 程式手冊 (模組專用指令篇)	
PRRP_M	GP_PRR_M2	○	○		
PSTRT1_P_M	ZP_PSTRT1_M2	○	○		
PSTRT2_P_M	ZP_PSTRT2_M2	○	○		
PSTRT3_P_M	ZP_PSTRT3_M2	○	○		
PSTRT4_P_M	ZP_PSTRT4_M2	○	○		
PUTE_M	G_PUTE_M2	○	○		
PUTEP_M	GP_PUTE_M2	○	○		
RDMSG_MD	G_RDMSG_M2	○	○		
RDMSG_P_MD	GP_RDMSG_M2	○	○		
RECV_UP_M	GP_RECV_M2	○	○		
RECV_P_M		○	○		
RECVS_U_M	Z_RECVS_M2	○	○		
REMT0_P_MD	ZP_REMT0_M2	○	○		
REQ_M	G_REQ_M2	○	○		
REQ_U_M		○	○		
REQ_UP_M	GP_REQ_M2	○	○		
REQP_M		○	○		
RIGHT	RIGHT_M2	○	○	1443頁 LEFT(_E)、RIGHT(_E)	
RIGHT_M		○	○		
RIRCV_MD	G_RIRCV_M2	○	○	MELSEC iQ-R 程式手冊 (模組專用指令篇)	
RIRCV_P_MD	GP_RIRCV_M2	○	○		
RIRD_MD	G_RIRD_M2	○	○		
RIRD_P_MD	GP_RIRD_M2	○	○		
RISEND_MD	G_RISEND_M2	○	○		
RISEND_P_MD	GP_RISEND_M2	○	○		
RITO_MD	G_RITO_M2	○	○		
RITO_P_MD	GP_RITO_M2	○	○		
RIWT_MD	G_RIWT_M2	○	○		
RIWT_P_MD	GP_RIWT_M2	○	○		
RLPASET_MD	G_RLPASET_M2	○	○		
RLPASET_P_MD	GP_RLPASET_M2	○	○		
ROL	ROL_M2	○	○		606頁 ROL(P)、RCL(P)
ROL_M		○	○		
ROR	ROR_M2	○	○		603頁 ROR(P)、RCR(P)
ROR_M		○	○		
SCL	SCL_M2	○	○		693頁 SCL(P) (_U)
SCLP	SCLP_M2	○	○		699頁 SCL2(P) (_U)
SCL2	SCL2_M2	○	○		
SCL2P	SCL2P_M2	○	○		
SEND_4_P_M	GP_SEND_M2	○	○	MELSEC iQ-R 程式手冊 (模組專用指令篇)	
SEND_UP_M		○	○		
SIN	SIN_M2	○	○	1395頁 SIN(_E)	
SIN_MD		○	○		
SIN_E_MD		○	○		
SIND	SIND_M2	○	○		
SIND_MD		○	○		
SIND_E_MD		○	○		
SORT	SORT_M2	○	○	725頁 SORTD(_U)	
SORT_M		○	○		
SPBUSY_MD	G_SPBUSY_M2	○	○	MELSEC iQ-R 程式手冊 (模組專用指令篇)	
SPBUSY_P_MD	GP_SPBUSY_M2	○	○		

GX Works2指令名	相容用指令名	使用程式		參閱目標	
		ST	FBD		
STR	STR_M2	○	○	791頁 STR(P) (_U)	
STR_MD		○	○		
STR_K_MD		○	○		
STR_S_MD		○	○		
STRP	STRP_M2	○	○		
STR_P_MD		○	○		
STR_P_S_MD		○	○		
STR_K_P_MD		○	○		
SWAP	SWAP_M2	×	○	543頁 SWAP(P)	
SWAP_MD		×	○		
SWAP_P_MD	SWAPP_M2	×	○		
SWAPP		×	○		
TAN	TAN_M2	○	○	1397頁 TAN(_E)	
TAN_MD		○	○		
TAN_E_MD		○	○		
TAND	TAND_M2	○	○		
TAND_MD		○	○		
TAND_E_MD		○	○		
TEACH1_P_M	ZP_TEACH1_M2	○	○	MELSEC iQ-R 程式手冊(模組專用指令篇)	
TEACH2_P_M	ZP_TEACH2_M2	○	○		
TEACH3_P_M	ZP_TEACH3_M2	○	○		
TEACH4_P_M	ZP_TEACH4_M2	○	○		
TO	TO_M2	○	○	1113頁 TO(P)、DTO(P)	
TO_M		○	○		
TOP	TOP_M2	○	○		
TOP_M		○	○		
UINI_M	ZP_UINI_M2	○	○		MELSEC iQ-R 程式手冊(模組專用指令篇)
UINI_U_M	Z_UINI_M2	○	○		
UINI_UP_M	ZP_UINI_M2	○	○		
VAL	VAL_M2	○	○	482頁 VAL(P) (_U)	
VAL_MD		○	○		
VAL_S_MD		○	○		
VALP	VALP_M2	○	○		
VAL_P_MD		○	○		
VAL_P_S_MD		○	○		
WSUM	WSUM_M2	○	○	741頁 WSUM(P) (_U)	
WSUM_M		○	○		
WSUMP	WSUMP_M2	○	○		
WSUMP_M		○	○		
Z_ABRST1	Z_ABRST1_M2	○	○	MELSEC iQ-R 程式手冊(模組專用指令篇)	
Z_ABRST2	Z_ABRST2_M2	○	○		
Z_ABRST3	Z_ABRST3_M2	○	○		
Z_ABRST4	Z_ABRST4_M2	○	○		
Z_RECVS	Z_RECVS_M2	○	○		
Z_UINI	Z_UINI_M2	○	○		
ZONE	ZONE_M2	○	○	689頁 ZONE(P) (_U)	
ZONE_MD		○	○		
ZONEP	ZONEP_M2	○	○		
ZONE_P_MD		○	○		

GX Works2指令名	相容用指令名	使用程式		參閱目標
		ST	FBD	
ZP_BUFRVC	ZP_BUFRVC_M2	○	○	MELSEC iQ-R 程式手冊 (模組專用指令篇)
ZP_BUFSND	ZP_BUFSND_M2	○	○	
ZP_CLOSE	ZP_CLOSE_M2	○	○	
ZP_CSET	ZP_CSET_M2	○	○	
ZP_ERRCLR	ZP_ERRCLR_M2	○	○	
ZP_ERRRD	ZP_ERRRD_M2	○	○	
ZP_OPEN	ZP_OPEN_M2	○	○	
ZP_PFWRT	ZP_PFWRT_M2	○	○	
ZP_PINIT	ZP_PINIT_M2	○	○	
ZP_PSTR1	ZP_PSTR1_M2	○	○	
ZP_PSTR2	ZP_PSTR2_M2	○	○	
ZP_PSTR3	ZP_PSTR3_M2	○	○	
ZP_PSTR4	ZP_PSTR4_M2	○	○	
ZP_REMTO	ZP_REMTO_M2	○	○	
ZP_TEACH1	ZP_TEACH1_M2	○	○	
ZP_TEACH2	ZP_TEACH2_M2	○	○	
ZP_TEACH3	ZP_TEACH3_M2	○	○	
ZP_TEACH4	ZP_TEACH4_M2	○	○	
ZP_UINI	ZP_UINI_M2	○	○	

相容用函數(通用函數)

ST(包含內嵌ST)、於結構化梯形圖/FBD的程式中所使用的通用函數是可以替換為相容用指令的。

GX Works2函數名	相容用函數名	使用程式		參閱目標
		ST	FBD	
ABS_E	ABS_E_M2	○	×	1388頁 ABS(_E)
ACOS_E	ACOS_E_M2	○	×	1399頁 ACOS(_E)
ADD_E	ADD_E_M2	○	×	1401頁 ADD(_E)
ADD_TIME_E	ADD_TIME_E_M2	○	×	1458頁 ADD_TIME(_E)
AND_E	AND_E_M2	○	×	1424頁 AND(_E)、OR(_E)、XOR(_E)
ASIN_E	ASIN_E_M2	○	×	1398頁 ASIN(_E)
ATAN_E	ATAN_E_M2	○	×	1400頁 ATAN(_E)
BCD_TO_DINT_E	BCD_TO_DINT_E_M2	○	×	1339頁 BCD_TO_DINT(_E)
BCD_TO_INT_E	BCD_TO_INT_E_M2	○	×	1337頁 BCD_TO_INT(_E)
BCD_TO_STR_E	BCD_TO_STR_E_M2	○	×	1342頁 BCD_TO_STRING(_E)
BITARR_TO_DINT_E	BITARR_TO_DINT_E_M2	○	×	1377頁 BITARR_TO_DINT(_E)
BITARR_TO_INT_E	BITARR_TO_INT_E_M2	○	×	1376頁 BITARR_TO_INT(_E)
BOOL_TO_DINT_E	BOOL_TO_DINT_E_M2	○	×	1295頁 BOOL_TO_DINT(_E)
BOOL_TO_DWORD_E	BOOL_TO_DWORD_E_M2	○	×	1293頁 BOOL_TO_DWORD(_E)
BOOL_TO_INT_E	BOOL_TO_INT_E_M2	○	×	1294頁 BOOL_TO_INT(_E)
BOOL_TO_STR_E	BOOL_TO_STR_E_M2	○	×	1297頁 BOOL_TO_STRING(_E)
BOOL_TO_TIME_E	BOOL_TO_TIME_E_M2	○	×	1296頁 BOOL_TO_TIME(_E)
BOOL_TO_WORD_E	BOOL_TO_WORD_E_M2	○	×	1292頁 BOOL_TO_WORD(_E)
CONCAT_E	CONCAT_E_M2	○	×	1447頁 CONCAT(_E)
COS_E	COS_E_M2	○	×	1396頁 COS(_E)
CPY_BITARR_E	CPY_BITARR_E_M2	○	×	1380頁 CPY_BITARR(_E)
CPY_BIT_OF_INT_E	CPY_BIT_OF_INT_E_M2	○	×	1385頁 CPY_BIT_OF_INT(_E)
DELETE_E	DELETE_E_M2	○	×	1451頁 DELETE(_E)
DINT_TO_BCD_E	DINT_TO_BCD_E_M2	○	×	1330頁 DINT_TO_BCD(_E)
DINT_TO_BITARR_E	DINT_TO_BITARR_E_M2	○	×	1379頁 DINT_TO_BITARR(_E)
DINT_TO_BOOL_E	DINT_TO_BOOL_E_M2	○	×	1325頁 DINT_TO_BOOL(_E)
DINT_TO_DWORD_E	DINT_TO_DWORD_E_M2	○	×	1328頁 DINT_TO_DWORD(_E)
DINT_TO_INT_E	DINT_TO_INT_E_M2	○	×	1329頁 DINT_TO_INT(_E)
DINT_TO_LREAL_E	DINT_TO_LREAL_E_M2	○	×	1333頁 DINT_TO_LREAL(_E)
DINT_TO_REAL_E	DINT_TO_REAL_E_M2	○	×	1332頁 DINT_TO_REAL(_E)
DINT_TO_STR_E	DINT_TO_STR_E_M2	○	×	1335頁 DINT_TO_STRING(_E)
DINT_TO_TIME_E	DINT_TO_TIME_E_M2	○	×	1334頁 DINT_TO_TIME(_E)
DINT_TO_WORD_E	DINT_TO_WORD_E_M2	○	×	1326頁 DINT_TO_WORD(_E)
DIV_E	DIV_E_M2	○	×	1409頁 DIV(_E)
DIV_TIME_E	DIV_TIME_E_M2	○	×	1464頁 DIV_TIME(_E)
DWORD_TO_BOOL_E	DWORD_TO_BOOL_E_M2	○	×	1305頁 DWORD_TO_BOOL(_E)
DWORD_TO_DINT_E	DWORD_TO_DINT_E_M2	○	×	1310頁 DWORD_TO_DINT(_E)
DWORD_TO_INT_E	DWORD_TO_INT_E_M2	○	×	1308頁 DWORD_TO_INT(_E)
DWORD_TO_STR_E	DWORD_TO_STR_E_M2	○	×	1312頁 DWORD_TO_STRING(_E)
DWORD_TO_TIME_E	DWORD_TO_TIME_E_M2	○	×	1311頁 DWORD_TO_TIME(_E)
DWORD_TO_WORD_E	DWORD_TO_WORD_E_M2	○	×	1306頁 DWORD_TO_WORD(_E)
EQ_E	EQ_E_M2	○	×	1437頁 GT(_E)、GE(_E)、EQ(_E)、LE(_E)、LT(_E)
EXP_E	EXP_E_M2	○	×	1394頁 EXP(_E)
EXPT_E	EXPT_E_M2	○	×	1413頁 EXPT(_E)
FIND_E	FIND_E_M2	○	×	1456頁 FIND(_E)
GE_E	GE_E_M2	○	×	1437頁 GT(_E)、GE(_E)、EQ(_E)、LE(_E)、LT(_E)
GET_BIT_OF_INT_E	GET_BIT_OF_INT_E_M2	○	×	1381頁 GET_BIT_OF_INT(_E)
GT_E	GT_E_M2	○	×	1437頁 GT(_E)、GE(_E)、EQ(_E)、LE(_E)、LT(_E)
INSERT_E	INSERT_E_M2	○	×	1449頁 INSERT(_E)

GX Works2函數名	相容用函數名	使用程式		參閱目標
		ST	FBD	
INT_TO_BCD_E	INT_TO_BCD_E_M2	○	×	1318頁 INT_TO_BCD(_E)
INT_TO_BITARR_E	INT_TO_BITARR_E_M2	○	×	1378頁 INT_TO_BITARR(_E)
INT_TO_BOOL_E	INT_TO_BOOL_E_M2	○	×	1313頁 INT_TO_BOOL(_E)
INT_TO_DINT_E	INT_TO_DINT_E_M2	○	×	1317頁 INT_TO_DINT(_E)
INT_TO_DWORD_E	INT_TO_DWORD_E_M2	○	×	1315頁 INT_TO_DWORD(_E)
INT_TO_LREAL_E	INT_TO_LREAL_E_M2	○	×	1321頁 INT_TO_LREAL(_E)
INT_TO_REAL_E	INT_TO_REAL_E_M2	○	×	1320頁 INT_TO_REAL(_E)
INT_TO_STR_E	INT_TO_STR_E_M2	○	×	1323頁 INT_TO_STRING(_E)
INT_TO_TIME_E	INT_TO_TIME_E_M2	○	×	1322頁 INT_TO_TIME(_E)
INT_TO_WORD_E	INT_TO_WORD_E_M2	○	×	1314頁 INT_TO_WORD(_E)
LE_E	LE_E_M2	○	×	1437頁 GT(_E)、GE(_E)、EQ(_E)、LE(_E)、LT(_E)
LEFT_E	LEFT_E_M2	○	×	1443頁 LEFT(_E)、RIGHT(_E)
LEN_E	LEN_E_M2	○	×	1441頁 LEN(_E)
LIMITATION_E	LIMITATION_E_M2	○	×	1432頁 LIMIT(_E)
LN_E	LN_E_M2	○	×	1391頁 LN(_E)
LOG_E	LOG_E_M2	○	×	1392頁 LOG(_E)
LREAL_TO_DINT_E	LREAL_TO_DINT_E_M2	○	×	1354頁 LREAL_TO_DINT(_E)
LREAL_TO_INT_E	LREAL_TO_INT_E_M2	○	×	1353頁 LREAL_TO_INT(_E)
LREAL_TO_REAL_E	LREAL_TO_REAL_E_M2	○	×	1355頁 LREAL_TO_REAL(_E)
LT_E	LT_E_M2	○	×	1437頁 GT(_E)、GE(_E)、EQ(_E)、LE(_E)、LT(_E)
M_REAL_TO_REAL_E	M_REAL_TO_REAL_E_M2	○	×	—
MAXIMUM_E	MAXIMUM_E_M2	○	×	1430頁 MAX(_E)、MIN(_E)
MID_E	MID_E_M2	○	×	1445頁 MID(_E)
MINIMUM_E	MINIMUM_E_M2	○	×	1430頁 MAX(_E)、MIN(_E)
MOD_E	MOD_E_M2	○	×	1411頁 MOD(_E)
MOVE_E	MOVE_E_M2	○	×	1414頁 MOVE(_E)
MUL_E	MUL_E_M2	○	×	1404頁 MUL(_E)
MUL_TIME_E	MUL_TIME_E_M2	○	×	1462頁 MUL_TIME(_E)
MUX_E	MUX_E_M2	○	×	1435頁 MUX(_E)
NE_E	NE_E_M2	○	×	1439頁 NE(_E)
NOT_E	NOT_E_M2	○	×	1427頁 NOT(_E)
OR_E	OR_E_M2	○	×	1424頁 AND(_E)、OR(_E)、XOR(_E)
REAL_TO_DINT_E	REAL_TO_DINT_E_M2	○	×	1346頁 REAL_TO_DINT(_E)
REAL_TO_INT_E	REAL_TO_INT_E_M2	○	×	1344頁 REAL_TO_INT(_E)
REAL_TO_LREAL_E	REAL_TO_LREAL_E_M2	○	×	1348頁 REAL_TO_LREAL(_E)
REAL_TO_M_REAL_E	REAL_TO_M_REAL_E_M2	○	×	—
REAL_TO_STR_E	REAL_TO_STR_E_M2	○	×	1350頁 REAL_TO_STRING(_E)
REPLACE_E	REPLACE_E_M2	○	×	1453頁 REPLACE(_E)
RIGHT_E	RIGHT_E_M2	○	×	1443頁 LEFT(_E)、RIGHT(_E)
ROL_E	ROL_E_M2	○	×	1420頁 ROL(_E)
ROR_E	ROR_E_M2	○	×	1422頁 ROR(_E)
SEL_E	SEL_E_M2	○	×	1428頁 SEL(_E)
SET_BIT_OF_INT_E	SET_BIT_OF_INT_E_M2	○	×	1383頁 SET_BIT_OF_INT(_E)
SHL_E	SHL_E_M2	○	×	1416頁 SHL(_E)
SHR_E	SHR_E_M2	○	×	1418頁 SHR(_E)
SIN_E	SIN_E_M2	○	×	1395頁 SIN(_E)
SQRT_E	SQRT_E_M2	○	×	1390頁 SQRT(_E)
STR_TO_BCD_E	STR_TO_BCD_E_M2	○	×	1370頁 STRING_TO_BCD(_E)
STR_TO_BOOL_E	STR_TO_BOOL_E_M2	○	×	1363頁 STRING_TO_BOOL(_E)
STR_TO_DINT_E	STR_TO_DINT_E_M2	○	×	1368頁 STRING_TO_DINT(_E)
STR_TO_DWORD_E	STR_TO_DWORD_E_M2	○	×	1365頁 STRING_TO_DWORD(_E)
STR_TO_INT_E	STR_TO_INT_E_M2	○	×	1366頁 STRING_TO_INT(_E)

GX Works2函數名	相容用函數名	使用程式		參閱目標
		ST	FBD	
STR_TO_REAL_E	STR_TO_REAL_E_M2	○	×	1372頁 STRING_TO_REAL(_E)
STR_TO_TIME_E	STR_TO_TIME_E_M2	○	×	1375頁 STRING_TO_TIME(_E)
STR_TO_WORD_E	STR_TO_WORD_E_M2	○	×	1364頁 STRING_TO_WORD(_E)
SUB_E	SUB_E_M2	○	×	1406頁 SUB(_E)
SUB_TIME_E	SUB_TIME_E_M2	○	×	1460頁 SUB_TIME(_E)
TAN_E	TAN_E_M2	○	×	1397頁 TAN(_E)
TIME_TO_BOOL_E	TIME_TO_BOOL_E_M2	○	×	1356頁 TIME_TO_BOOL(_E)
TIME_TO_DINT_E	TIME_TO_DINT_E_M2	○	×	1360頁 TIME_TO_DINT(_E)
TIME_TO_DWORD_E	TIME_TO_DWORD_E_M2	○	×	1358頁 TIME_TO_DWORD(_E)
TIME_TO_INT_E	TIME_TO_INT_E_M2	○	×	1359頁 TIME_TO_INT(_E)
TIME_TO_STR_E	TIME_TO_STR_E_M2	○	×	1361頁 TIME_TO_STRING(_E)
TIME_TO_WORD_E	TIME_TO_WORD_E_M2	○	×	1357頁 TIME_TO_WORD(_E)
WORD_TO_BOOL_E	WORD_TO_BOOL_E_M2	○	×	1298頁 WORD_TO_BOOL(_E)
WORD_TO_DINT_E	WORD_TO_DINT_E_M2	○	×	1301頁 WORD_TO_DINT(_E)
WORD_TO_DWORD_E	WORD_TO_DWORD_E_M2	○	×	1299頁 WORD_TO_DWORD(_E)
WORD_TO_INT_E	WORD_TO_INT_E_M2	○	×	1300頁 WORD_TO_INT(_E)
WORD_TO_STR_E	WORD_TO_STR_E_M2	○	×	1304頁 WORD_TO_STRING(_E)
WORD_TO_TIME_E	WORD_TO_TIME_E_M2	○	×	1303頁 WORD_TO_TIME(_E)
XOR_E	XOR_E_M2	○	×	1424頁 AND(_E)、OR(_E)、XOR(_E)

相容用FB(通用FB)

ST(包含內嵌ST)、於結構化梯形圖/FBD的程式中所使用的通用FB可替換為相容用FB。

GX Works2 FB名	相容用FB名	使用程式		參閱目標
		ST	FBD	
TOF_HIGH	TOF_HIGH_M2*1	○	○	1489頁 TOF(_E)
TOF_HIGH_E	TOF_HIGH_E_M2*1	○	○	
TON_HIGH	TON_HIGH_M2*1	○	○	1487頁 TON(_E)
TON_HIGH_E	TON_HIGH_E_M2*1	○	○	
TP_HIGH	TP_HIGH_M2*1	○	○	1485頁 TP(_E)
TP_HIGH_E	TP_HIGH_E_M2*1	○	○	
CTD	CTD_M2*2	○	○	1478頁 CTD(_E)
CTD_E	CTD_E_M2*2	○	○	
CTU	CTU_M2*2	○	○	1476頁 CTU(_E)
CTU_E	CTU_E_M2*2	○	○	
CTUD	CTUD_M2*2	○	○	1480頁 CTUD(_E)
CTUD_E	CTUD_E_M2*2	○	○	
F_TRIG	F_TRIG_M2*2	○	○	1474頁 F_TRIG(_E)
F_TRIG_E	F_TRIG_E_M2*2	○	○	
RS	RS_M2*2	○	○	1470頁 RS(_E)
RS_E	RS_E_M2*2	○	○	
R_TRIG	R_TRIG_M2*2	○	○	1472頁 R_TRIG(_E)
R_TRIG_E	R_TRIG_E_M2*2	○	○	
SR	SR_M2*2	○	○	1468頁 SR(_E)
SR_E	SR_E_M2*2	○	○	

*1 與GX Works2 FB的動作一部分有所不同。

*2 當成GX Works3工程使用的情況下，只有輸入標籤將會自動替換。

■動作的差異


相容用FB	內容
TOF_HIGH_M2 TOF_LOW_M2	<p>OFF延遲定時器時間*1的有效範圍依超長定時器的設定而有所不同。由於OFF延遲定時器時間為時間型(32位元值)，最大值为包含在時間型範圍中的值。</p> <p>■最小值</p> <ul style="list-style-type: none"> 與超長定時器的設定值(ms)相同。超長定時器的設定值不滿1ms的情況下則為1ms。 <p>■最大值</p> <ul style="list-style-type: none"> OFF延遲定時器時間(ms) ≤ 2147483647(ms) × 超長定時器的設定值(ms) <p>OFF延遲定時器時間超過有效範圍的情況下，將會發生運算出錯。(出錯代碼：3401H) 發生運算出錯時，應重新審視超長定時器的設定值。</p>
TON_HIGH_M2 TON_LOW_M2	<p>ON延遲定時器時間*1的有效範圍依超長定時器的設定而有所不同。由於ON延遲定時器時間為時間型(32位元值)，最大值为包含在時間型範圍中的值。</p> <p>■最小值</p> <ul style="list-style-type: none"> 與超長定時器的設定值(ms)相同。超長定時器的設定值不滿1ms的情況下則為1ms。 <p>■最大值</p> <ul style="list-style-type: none"> ON延遲定時器時間(ms) ≤ 2147483647(ms) × 超長定時器的設定值(ms) <p>ON延遲定時器時間超過有效範圍的情況下，將會發生運算出錯。(出錯代碼：3401H) 發生運算出錯時，應重新審視超長定時器的設定值。</p>
TP_HIGH_M2 TP_LOW_M2	<p>脈衝寬度時間*1的有效範圍依超長定時器的設定而有所不同。由於脈衝寬度時間為時間型(32位元值)，最大值为包含在時間型範圍中的值。</p> <p>■最小值</p> <ul style="list-style-type: none"> 與超長定時器的設定值(ms)相同。超長定時器的設定值不滿1ms的情況下則為1ms。 <p>■最大值</p> <ul style="list-style-type: none"> 脈衝寬度時間(ms) ≤ 2147483647(ms) × 超長定時器的設定值(ms) <p>脈衝寬度時間超過有效範圍的情況下，將會發生運算出錯。(出錯代碼：3401H) 發生運算出錯時，應重新審視超長定時器的設定值。</p>

*1 關於OFF延遲定時器時間/ON延遲定時器時間/脈衝寬度時間，請參閱下述手冊。

📖 GX Developer Version 1 Programming Manual

從PX Developer格式工程的替換

將PX Developer格式工程當成GX Works3工程使用的情況下，一部分的指令將會自動替換成相容用FB。
有關詳細內容，請參閱下述手冊。

 MELSEC iQ-R 程式手冊 (過程控制FB/指令篇)

索引

數字

- 16位元資料(字元資料) 38
- 32位元資料(雙字元資料) 41

A

- A系統 22

B

- B系統 22

C

- CC-Link IE內建乙太網路介面模組 23
- CC-Link IE控制網路搭載模組 23
- CC-Link IE現場網路主站・本地站搭載模組 23

P

- PID指令的種類 1192
- PID控制指令 1207

R

- RnCPU 23
- RnENCPU 23
- RnENCPU(網路部分) 23
- RnPCPU 23

S

- SFC程式用指令 1247
- SIL2功能模組 22
- SIL2過程CPU 22
- SLMP 23

二畫

- 二重化系統 22
- 二重化系統用指令 1274
- 二重化擴展基板配置 1133

三畫

- 子集處理 50

四畫

- 元件數/傳送數/資料數/字元串數等的數值(n) 31
- 內建乙太網路功能用指令 1145
- 分離模式 22
- 手冊閱讀方法 25
- 支援SLMP的設備 23

五畫

- 可程式控制器CPU 23
- 外部設備 23
- 目標(d) 30

六畫

- 多CPU之間專用指令 1237
- 字元串資料 47
- 安全CPU 22
- 安全元件 22
- 安全功能模組 22
- 安全控制 22
- 安全通訊 22
- 安全程式 22
- 安全週期處理 22
- 安全標籤 22, 23

七畫

- 位元資料 36

九畫

- 待機系統 22
- 指令的構成 30
- 指令處理時間 1495

十畫

- 記錄功能 1143

十一畫

- 基本指令 212
- 執行條件 49
- 常規CPU 23
- 常規元件 22
- 常規通訊 22
- 常規程式 22
- 控制系統 22
- 通用FB 1467
- 通用函數 1291

十二畫

- 備份模式 22
- 單精度實數資料 44
- 順控程式指令 148

十三畫

- 源(s) 30
- 過程CPU 23
- 過程CPU(二重化模式) 22
- 過程CPU(過程模式) 22

十四畫

- 實數資料(浮點資料) 44
- 遠程起始模組 23

十五畫

- 標籤 22
- 緩衝記憶體 22
- 請求報文 22

十六畫

操作數 23

十七畫

應用指令 551

十八畫

雙精度實數資料 45

二十一畫

響應報文 22

指令索引

符號

-(P) (_U)	233, 235
*(P) (_U)	245
/(P) (_U)	247
+(P) (_U)	229, 231
\$+(P)	766, 768
\$MOV (P)	770
\$MOV (P)_WS	772

A

ABS (_E)	1388
ABS_E_M2	1584
ACOS (_E)	1399
ACOS (P)	905
ACOSD (P)	917
ACOSD_M2	1571
ACOS_E_M2	1584
ACOS_M2	1571
ADD (_E)	1401
ADD_E_M2	1584
ADD_TIME (_E)	1458
ADD_TIME_E_M2	1584
ADRSET (P)	991
ANB	156
AND	148, 1247, 1249
AND (_E)	1424
AND< (_U)	212
AND<= (_U)	212
AND<> (_U)	212
AND= (_U)	212
AND> (_U)	212
AND>= (_U)	212
AND\$<	762
AND\$<=	762
AND\$<>	762
AND\$=	762
AND\$>	762
AND\$>=	762
ANDD< (_U)	214
ANDD<= (_U)	214
ANDD<> (_U)	214
ANDD= (_U)	214
ANDD> (_U)	214
ANDD>= (_U)	214
ANDD_EQ (_U)	214
ANDD_EQ_M2	1572
ANDD_GE (_U)	214
ANDD_GE_M2	1572
ANDD_GT (_U)	214
ANDD_GT_M2	1572
ANDD_LE (_U)	214
ANDD_LE_M2	1572
ANDD_LT (_U)	214
ANDD_LT_M2	1572
ANDD_NE (_U)	214
ANDD_NE_M2	1572
ANDDT<	1078
ANDDT<=	1078
ANDDT<>	1078
ANDDT=	1078

ANDDT>	1078
ANDDT>=	1078
ANDDT_EQ	1078
ANDDT_GE	1078
ANDDT_GT	1078
ANDDT_LE	1078
ANDDT_LT	1078
ANDDT_NE	1078
ANDE<	835
ANDE<=	835
ANDE<>	835
ANDE=	835
ANDE>	835
ANDE>=	835
ANDED<	838
ANDED<=	838
ANDED<>	838
ANDED=	838
ANDED>	838
ANDED>=	838
ANDED_EQ	838
ANDED_GE	838
ANDED_GT	838
ANDED_LE	838
ANDED_LT	838
ANDED_NE	838
ANDE_EQ	835
ANDE_GE	835
ANDE_GT	835
ANDE_LE	835
ANDE_LT	835
ANDE_M2	1584
ANDE_NE	835
ANDEQ (_U)	212
ANDEQ_M2	1571
ANDF.	150
ANDFI	153
AND_GE (_U)	212
AND_GE_M2	1571
AND_GT (_U)	212
AND_GT_M2	1571
AND_LE (_U)	212
AND_LE_M2	1571
AND_LT (_U)	212
AND_LT_M2	1571
AND_NE (_U)	212
AND_NE_M2	1571
ANDP.	150
ANDPI	153
ANDSTRING_EQ	762
ANDSTRING_GE	762
ANDSTRING_GT	762
ANDSTRING_LE	762
ANDSTRING_LT	762
ANDSTRING_NE	762
ANDTM<	1082
ANDTM<=	1082
ANDTM<>	1082
ANDTM=	1082
ANDTM>	1082
ANDTM>=	1082
ANDTM_EQ	1082

ANDTM_GE	1082
ANDTM_GT	1082
ANDTM_LE	1082
ANDTM_LT	1082
ANDTM_NE	1082
ANI	148, 1247, 1249
ASC2INT(P)	488
ASIN(_E)	1398
ASIN(P)	903
ASIND(P)	915
ASIND_M2	1572
ASIN_E_M2	1584
ASIN_M2	1572
ATAN(_E)	1400
ATAN(P)	907
ATAND(P)	919
ATAND_M2	1572
ATAN_E_M2	1584
ATAN_M2	1572

B

B-(P)	256, 257
B*(P)	265
B/(P)	267
B+(P)	253, 254
BACOS(P)	929
BAND(P) (_U)	685
BAND_M2	1572
BANDP_M2	1572
BASIN(P)	927
BATAN(P)	931
BCD(P)	405
BCDDA(P)	797
BCD_TO_DINT(_E)	1339
BCD_TO_DINT_E_M2	1584
BCD_TO_INT(_E)	1337
BCD_TO_INT_E_M2	1584
BCD_TO_STR_E_M2	1584
BCD_TO_STRING(_E)	1342
BCOS(P)	923
BDIVISION(P)	267
BDSQR_M2	1572
BDSQRP_M2	1572
BDSQRT(P)	955
BIN(P)	409
BINDA(P) (_U)	774
BINDA_M2	1572
BINDAP_M2	1572
BINHA(P)	783
BITARR_TO_DINT(_E)	1377
BITARR_TO_DINT_E_M2	1584
BITARR_TO_INT(_E)	1376
BITARR_TO_INT_E_M2	1584
BK-(P) (_U)	275
BK+(P) (_U)	273
BKAND(P)	299
BKBCD(P)	461
BKBIN(P)	463
BKCMPL(P) (_U)	224
BKCMPL=<(P) (_U)	224
BKCMPL<>(P) (_U)	224
BKCMPL=(P) (_U)	224
BKCMPL>(P) (_U)	224
BKCMPL>=(P) (_U)	224
BKCMPL_EQ(P) (_U)	224

BKCMPL_EQ_M2	1573
BKCMPL_EQP_M2	1573
BKCMPL_GE(P) (_U)	224
BKCMPL_GE_M2	1573
BKCMPL_GEP_M2	1573
BKCMPL_GT(P) (_U)	224
BKCMPL_GT_M2	1573
BKCMPL_GTP_M2	1573
BKCMPL_LE(P) (_U)	224
BKCMPL_LE_M2	1573
BKCMPL_LEP_M2	1573
BKCMPL_LT(P) (_U)	224
BKCMPL_LT_M2	1573
BKCMPL_LTP_M2	1573
BKCMPL_NE(P) (_U)	224
BKCMPL_NE_M2	1573
BKCMPL_NEP_M2	1573
BKMINUS(P) (_U)	275
BKMINUS_M2	1572, 1573
BKMINUSP_M2	1572, 1573
BKOR(P)	309
BKPLUS(P) (_U)	273
BKPLUS_M2	1572, 1573
BKPLUSP_M2	1572, 1573
BKRST(P)	339
BKRST_M2	1573
BKRSTP_M2	1573
BKXNR(P)	329
BKXOR(P)	319
BLKMOVB(P)	548
BMINUS(P)	257
BMOV(P)	525, 1256
BMOVL(P)	527
BMULTI(P)	265
BON(P)	713
BOOL_TO_DINT(_E)	1295
BOOL_TO_DINT_E_M2	1584
BOOL_TO_DWORD(_E)	1293
BOOL_TO_DWORD_E_M2	1584
BOOL_TO_INT(_E)	1294
BOOL_TO_INT_E_M2	1584
BOOL_TO_STR_E_M2	1584
BOOL_TO_STRING(_E)	1297
BOOL_TO_TIME(_E)	1296
BOOL_TO_TIME_E_M2	1584
BOOL_TO_WORD(_E)	1292
BOOL_TO_WORD_E_M2	1584
BPLUS(P)	254
BREAK(P)	572
BRSET	1271
BRST(P)	333
BSET(P)	331
BSFL(P)	347
BSFR(P)	345
BSIN(P)	921
BSQR_M2	1573
BSQRP_M2	1573
BSQRT(P)	953
BTAN(P)	925
BTOW(P)	510
BXCH(P)	541

C

CALL(P)	574
CCD(P)	753

DDIVISION(P) (_U)	251	DLIMITP_M2	1575
DDIVISION_M2	1574, 1575	DMAX(P) (_U)	719
DDIVISIONP_M2	1574, 1575	DMAX_M2	1575
DDSFL(P)	355	DMAXP_M2	1575
DDSFR(P)	353	DMEAN(P) (_U)	747
DEC(P) (_U)	285	DMEAN_M2	1575
DEC_M2	1575	DMEANP_M2	1575
DECO(P)	494	DMIN(P) (_U)	723
DECP_M2	1575	DMIN_M2	1575
DEG(P)	935	DMINP_M2	1575
DEGD(P)	939	DMINUS(P) (_U)	243
DELETE(_E)	1451	DMINUS_2_M2	1575
DELETE_E_M2	1584	DMINUS_M2	1573, 1575
DELTA(P)	190	DMINUSP_2_M2	1576
DFMOV(P)	533	DMINUSP_M2	1573, 1576
DFMOVL(P)	535	DMOV(P)	514, 1253
DFROM(P)	1109	DMULTI(P) (_U)	249
DFROMD(P)	1118	DMULTI_M2	1573, 1576
DGBIN(P) (_U)	459	DMULTIP_M2	1573, 1576
DGBIN_M2	1575	DNEG(P)	493
DGBINP_M2	1575	DOR(P)	305, 307
DGRY(P) (_U)	455	DPLUS(P) (_U)	239
DGRY_M2	1575	DPLUS_2_M2	1576
DGRYP_M2	1575	DPLUS_M2	1573, 1576
DHABIN(P)	473	DPLUSP_2_M2	1576
DHOURM	1101	DPLUSP_M2	1573, 1576
DI	557, 559	DRCL(P)	611
DI_1	559	DRCR(P)	609
DINC(P) (_U)	287	DROL(P)	611
DINC_M2	1575	DROL_M2	1576
DINCP_M2	1575	DROR(P)	609
DINT2DBL(P)	883	DROR_M2	1576
DINT2FLT(P)	875	DSCL(P) (_U)	696
DINT2INT(P)	441	DSCL2(P) (_U)	701
DINT2UDINT(P)	445	DSCL2_M2	1576
DINT2UINT(P)	443	DSCL2P_M2	1576
DINT_TO_BCD(_E)	1330	DSCL_M2	1576
DINT_TO_BCD_E_M2	1584	DSCLP_M2	1576
DINT_TO_BITARR(_E)	1379	DSERDATA(P)	705
DINT_TO_BITARR_E_M2	1584	DSERMM(P)	709
DINT_TO_BOOL(_E)	1325	DSFL(P)	351
DINT_TO_BOOL_E_M2	1584	DSFR(P)	349
DINT_TO_DWORD(_E)	1328	DSORTD(_U)	727
DINT_TO_DWORD_E_M2	1584	DSORT_M2	1576
DINT_TO_INT(_E)	1329	DSORTTBL2(_U)	737
DINT_TO_INT_E_M2	1584	DSQRT(P)	750
DINT_TO_LREAL(_E)	1333	DSTR(P) (_U)	794
DINT_TO_LREAL_E_M2	1584	DSTR_M2	1576
DINT_TO_REAL(_E)	1332	DSTRP_M2	1576
DINT_TO_REAL_E_M2	1584	DSUM(P)	712
DINT_TO_STR_E_M2	1584	DSWAP(P)	544
DINT_TO_STRING(_E)	1335	DTEST(P)	337
DINT_TO_TIME(_E)	1334	DTO(P)	1113
DINT_TO_TIME_E_M2	1584	DTOD(P)	1122
DINT_TO_WORD(_E)	1326	DTO_M2	1576
DINT_TO_WORD_E_M2	1584	DTOP_M2	1576
DIS(P)	500	DUTY	1095
DIV(_E)	1409	DVAL(P) (_U)	485
DIV_E_M2	1584	DVAL_M2	1576
DIVISION(P) (_U)	247	DVALP_M2	1576
DIVISION_M2	1571, 1575	DWORD_TO_BOOL(_E)	1305
DIVISIONP_M2	1571, 1575	DWORD_TO_BOOL_E_M2	1584
DIV_TIME(_E)	1464	DWORD_TO_DINT(_E)	1310
DIV_TIME_E_M2	1584	DWORD_TO_DINT_E_M2	1584
DLIMIT(P) (_U)	683	DWORD_TO_INT(_E)	1308
DLIMIT_M2	1575	DWORD_TO_INT_E_M2	1584

DWORD_TO_STR_E_M2	1584
DWORD_TO_STRING(_E)	1312
DWORD_TO_TIME(_E)	1311
DWORD_TO_TIME_E_M2	1584
DWORD_TO_WORD(_E)	1306
DWORD_TO_WORD_E_M2	1584
DWSFTL(P)	387
DWSFTR(P)	383
DWSUM(P)(_U)	743
DWSUM_M2	1576
DWSUMP_M2	1576
DXCH(P)	539
DXNR(P)	325, 327
DXOR(P)	315, 317
DZCP(P)(_U)	222
DZONE(P)(_U)	691
DZONE_M2	1576
DZONEP_M2	1576

EREXPP_M2	1576
ESFL(P)	359
ESFR(P)	357
ESFTL(P)	395
ESFTR(P)	391
ESQRT(P)	941
ESTR(P)	806
EVAL(P)	887
EXP(_E)	1394
EXP(P)	945
EXPD(P)	947
EXPD_M2	1577
EXP_E_M2	1584
EXP_M2	1577
EXPT(_E)	1413
EXPT_E_M2	1584
EZCP(P)	845

E

E-(P)	853, 855
E*(P)	865
E/(P)	867
E+(P)	849, 851
ECALL(P)	583
ECMP(P)	841
ECONTSW	1278
ED-(P)	861, 863
ED*(P)	869
ED/(P)	871
ED+(P)	857, 859
EDCMP(P)	843
EDDIVISION(P)	871
EDIVISION(P)	867
EDMAX(P)	967
EDMIN(P)	971
EDMINUS(P)	863
EDMOV(P)	896
EDMULTI(P)	869
EDNEG(P)	894
EDPLUS(P)	859
EDSFL(P)	363
EDSFR(P)	361
EDSFTL(P)	403
EDSFTR(P)	399
EDSQRT(P)	943
EDZCP(P)	847
EFCALL(P)	588
EGF	161
EGP	161
EI	557
EMAX(P)	965
EMIN(P)	969
EMINUS(P)	855
EMOD(P)	490
EMOV(P)	895
EMULTI(P)	865
ENCO(P)	496
END	205
ENEG(P)	893
EPLUS(P)	851
EQ(_E)	1437
EQ_E_M2	1584
EREXP(P)	891
EREXP_M2	1576

F

FCALL(P)	579
FDEL(P)	621
FEND	204
FF	188
FIFR(P)	613
FIFW(P)	617
FIND(_E)	1456
FIND_E_M2	1584
FINS(P)	619
FLT2DBL(P)	885
FLT2DINT(P)	417
FLT2INT(P)	413
FLT2UDINT(P)	419
FLT2UIINT(P)	415
FMOV(P)	529
FMOVL(P)	531
FOR	570
FPOP(P)	615
FROM(P)	1109
FROMD(P)	1118
F_TRIG(_E)	1474
F_TRIG_E_M2	1587
F_TRIG_M2	1587

G

G_BIDIN_M2	1572, 1577
G_BIDOUT_M2	1572, 1577
GBIN(P)(_U)	457
GBIN_M2	1577
GBINP_M2	1577
G_CCASET_M2	1577
G_CPRTCL_M2	1577
GE(_E)	1437
GE_E_M2	1584
GET_BIT_OF_INT(_E)	1381
GET_BIT_OF_INT_E_M2	1584
GET_BOOL_ADDR	1387
GET_INT_ADDR	1387
GET_WORD_ADDR	1387
G_GETE_M2	1577, 1578
G_INPUT_M2	1577, 1578
GOEND	556
G_OGLOAD_M2	1577, 1580
G_OGSTOR_M2	1577, 1580
G_ONDEMAND_M2	1577, 1580

G_OUTPUT_M2	1577, 1580
GP_BIDIN_M2	1572, 1578
GP_BIDOUT_M2	1572, 1578
GP_CCPASET_M2	1578
GP_CPRTCL_M2	1578
GP_ECPRTCL_M2	1578
GP_GETE_M2	1578
GP_OGLOAD_M2	1578, 1580
GP_OGSTOR_M2	1578, 1580
GP_ONDEMAND_M2	1578, 1580
GP_OUTPUT_M2	1578, 1580
GP_PRR_M2	1578, 1581
GP_PUTM2	1578, 1581
GP_RDMSG_M2	1578, 1581
GP_RECV_M2	1578, 1581
GP_REQ_M2	1578, 1581
GP_RIRCV_M2	1578, 1581
GP_RIRD_M2	1578, 1581
GP_RISEND_M2	1578, 1581
GP_RITO_M2	1578, 1581
GP_RIWT_M2	1578, 1581
GP_RLPASET_M2	1578, 1581
G_PRR_M2	1577, 1581
GP_SEND_M2	1578, 1581
GP_SPBUSY_M2	1578, 1581
G_PUTM2	1577, 1581
G_RDMSG_M2	1577, 1581
G_REQ_M2	1577, 1581
G_RIRCV_M2	1577, 1581
G_RIRD_M2	1577, 1581
G_RISEND_M2	1577, 1581
G_RITO_M2	1577, 1581
G_RIWT_M2	1577, 1581
G_RLPASET_M2	1577, 1581
GRY(P) (_U)	453
GRY_M2	1578
GRYP_M2	1578
G_SPBUSY_M2	1577, 1581
GT(_E)	1437
GT_E_M2	1584

H

HABIN(P)	471
HOURM	1099

I

IMASK	564
INC(P) (_U)	283
INC_M2	1578
INCP_M2	1578
INSERT(_E)	1449
INSERT_E_M2	1584
INSTR(P)	829
INT2ASC(P)	810
INT2DBL(P)	881
INT2DINT(P)	431
INT2FLT(P)	873
INT2UDINT(P)	433
INT2UINT(P)	429
INT_TO_BCD(_E)	1318
INT_TO_BCD_E_M2	1585
INT_TO_BITARR(_E)	1378
INT_TO_BITARR_E_M2	1585
INT_TO_BOOL(_E)	1313

INT_TO_BOOL_E_M2	1585
INT_TO_DINT(_E)	1317
INT_TO_DINT_E_M2	1585
INT_TO_DWORD(_E)	1315
INT_TO_DWORD_E_M2	1585
INT_TO_LREAL(_E)	1321
INT_TO_LREAL_E_M2	1585
INT_TO_REAL(_E)	1320
INT_TO_REAL_E_M2	1585
INT_TO_STR_E_M2	1585
INT_TO_STRING(_E)	1323
INT_TO_TIME(_E)	1322
INT_TO_TIME_E_M2	1585
INT_TO_WORD(_E)	1314
INT_TO_WORD_E_M2	1585
INV	159
IRET	568

J

JMP	553
---------------	-----

L

LD	148, 1247, 1249
LD<(_U)	212
LD<=(_U)	212
LD<>(_U)	212
LD=(_U)	212
LD>(_U)	212
LD>=(_U)	212
LD\$<.	762
LD\$<=.	762
LD\$<>.	762
LD\$=.	762
LD\$>.	762
LD\$>=.	762
LDD<(_U)	214
LDD<=(_U)	214
LDD<>(_U)	214
LDD=(_U)	214
LDD>(_U)	214
LDD>=(_U)	214
LDD_EQ(_U)	214
LDD_EQ_M2	1579
LDD_GE(_U)	214
LDD_GE_M2	1579
LDD_GT(_U)	214
LDD_GT_M2	1579
LDD_LE(_U)	214
LDD_LE_M2	1579
LDD_LT(_U)	214
LDD_LT_M2	1579
LDD_NE(_U)	214
LDD_NE_M2	1579
LDDT<.	1078
LDDT<=.	1078
LDDT<>.	1078
LDDT=.	1078
LDDT>.	1078
LDDT>=.	1078
LDDT_EQ	1078
LDDT_GE	1078
LDDT_GT	1078
LDDT_LE	1078
LDDT_LT	1078

DDT_NE	1078
LDE<	835
LDE<=	835
LDE<>	835
LDE=	835
LDE>	835
LDE>=	835
LDED<	838
LDED<=	838
LDED<>	838
LDED=	838
LDED>	838
LDED>=	838
LDED_EQ	838
LDED_GE	838
LDED_GT	838
LDED_LE	838
LDED_LT	838
LDED_NE	838
LDE_EQ	835
LDE_GE	835
LDE_GT	835
LDE_LE	835
LDE_LT	835
LDE_NE	835
LD_EQ(U)	212
LD_EQ_M2	1578
LDF	150
LDFI	153
LD_GE(U)	212
LD_GE_M2	1578
LD_GT(U)	212
LD_GT_M2	1578
LDI	148, 1247, 1249
LD_LE(U)	212
LD_LE_M2	1578
LD_LT(U)	212
LD_LT_M2	1578
LD_NE(U)	212
LD_NE_M2	1578
LDP	150
LDPI	153
LDSTRING_EQ	762
LDSTRING_GE	762
LDSTRING_GT	762
LDSTRING_LE	762
LDSTRING_LT	762
LDSTRING_NE	762
LDTM<	1082
LDTM<=	1082
LDTM<>	1082
LDTM=	1082
LDTM>	1082
LDTM>=	1082
LDTM_EQ	1082
LDTM_GE	1082
LDTM_GT	1082
LDTM_LE	1082
LDTM_LT	1082
LDTM_NE	1082
LE(E)	1437
LEDR	756
LE_E_M2	1585
LEFT(E)	1443
LEFT(P)	822
LEFT_E_M2	1585

LEFT_M2	1579
LEN(E)	1441
LEN(P)	818
LEN_E_M2	1585
LEN_M2	1579
LIMIT(E)	1432
LIMIT(P)(U)	681
LIMITATION_E_M2	1585
LIMIT_M2	1579
LIMITP_M2	1579
LN(E)	1391
LN_E_M2	1585
LOG(E)	1392
LOG(P)	949
LOG10(P)	961
LOG10D(P)	963
LOGD(P)	951
LOGD_M2	1579
LOG_E_M2	1585
LOG_M2	1579
LOGTRG	1140
LOGTRGR	1142
LREAL_TO_DINT(E)	1354
LREAL_TO_DINT_E_M2	1585
LREAL_TO_INT(E)	1353
LREAL_TO_INT_E_M2	1585
LREAL_TO_REAL(E)	1355
LREAL_TO_REAL_E_M2	1585
LT(E)	1437
LT_E_M2	1585

M

M(P). DDRD	1241
M(P). DDWR	1244
M(P)_ DDRD	1241
M(P)_ DDWR	1244
MAX(E)	1430
MAX(P)(U)	717
MAXIMUM_E_M2	1585
MAX_M2	1579
MAXP_M2	1579
MC	194
MCR	194
MEAN(P)(U)	745
MEAN_M2	1579
MEANP_M2	1579
MEF	160
MEP	160
MID(E)	1445
MID_E_M2	1585
MIDR(P)	824
MIDW(P)	826
MIN(E)	1430
MIN(P)(U)	721
MINIMUM_E_M2	1585
MIN_M2	1579
MINP_M2	1579
MINUS(P)(U)	235
MINUS_2_M2	1579
MINUS_M2	1571, 1579
MINUSP_2_M2	1579
MINUSP_M2	1571, 1579
MOD(E)	1411
MOD_E_M2	1585
MOV(P)	512, 1251

MOVB (P)	546
MOVE (_E)	1414
MOVE_E_M2	1585
MPP	157
MPS	157
MRD	157
M_REAL_TO_REAL_E_M2	1585
MTR	1014
MUL (_E)	1404
MUL_E_M2	1585
MULTI (P) (_U)	245
MULTI_M2	1571, 1579
MUL_TIME (_E)	1462
MUL_TIME_E_M2	1585
MULTIP_M2	1571, 1579
MUX (_E)	1435
MUX_E_M2	1585

N

NDIS (P)	504
NE (_E)	1439
NE_E_M2	1585
NEG (P)	492
NEXT	570
NOP	208
NOPLF	209
NOT (_E)	1427
NOT_E_M2	1585
NUNI (P)	506

O

OR	148, 1247, 1249
OR (_E)	1424
OR< (_U)	212
OR<= (_U)	212
OR<> (_U)	212
OR= (_U)	212
OR> (_U)	212
OR>= (_U)	212
OR\$<	762
OR\$<=	762
OR\$<>	762
OR\$=	762
OR\$>	762
OR\$>=	762
ORB	156
ORD< (_U)	214
ORD<= (_U)	214
ORD<> (_U)	214
ORD= (_U)	214
ORD> (_U)	214
ORD>= (_U)	214
ORD_EQ (_U)	214
ORD_EQ_M2	1580
ORD_GE (_U)	214
ORD_GE_M2	1580
ORD_GT (_U)	214
ORD_GT_M2	1580
ORD_LE (_U)	214
ORD_LE_M2	1580
ORD_LT (_U)	214
ORD_LT_M2	1580
ORD_NE (_U)	214
ORD_NE_M2	1580

ORDT<	1078
ORDT<=	1078
ORDT<>	1078
ORDT=	1078
ORDT>	1078
ORDT>=	1078
ORDT_EQ	1078
ORDT_GE	1078
ORDT_GT	1078
ORDT_LE	1078
ORDT_LT	1078
ORDT_NE	1078
ORE<	835
ORE<=	835
ORE<>	835
ORE=	835
ORE>	835
ORE>=	835
ORED<	838
ORED<=	838
ORED<>	838
ORED=	838
ORED>	838
ORED>=	838
ORED_EQ	838
ORED_GE	838
ORED_GT	838
ORED_LE	838
ORED_LT	838
ORED_NE	838
ORE_EQ	835
ORE_GE	835
ORE_GT	835
ORE_LE	835
ORE_LT	835
OR_E_M2	1585
ORE_NE	835
OR_EQ (_U)	212
OR_EQ_M2	1580
ORF	150
ORFI	153
OR_GE (_U)	212
OR_GE_M2	1580
OR_GT (_U)	212
OR_GT_M2	1580
ORI	148, 1247, 1249
OR_LE (_U)	212
OR_LE_M2	1580
OR_LT (_U)	212
OR_LT_M2	1580
OR_NE (_U)	212
OR_NE_M2	1580
ORP	150
ORPI	153
ORSTRING_EQ	762
ORSTRING_GE	762
ORSTRING_GT	762
ORSTRING_LE	762
ORSTRING_LT	762
ORSTRING_NE	762
ORTM<	1082
ORTM<=	1082
ORTM<>	1082
ORTM=	1082
ORTM>	1082
ORTM>=	1082

ORTM_EQ	1082
ORTM_GE	1082
ORTM_GT	1082
ORTM_LE	1082
ORTM_LT	1082
ORTM_NE	1082
OUT	163
OUT C	171
OUT F	175
OUT LC	173
OUT LST	168
OUT LT	168
OUT ST	165
OUT T	165
OUT_C	171
OUTH	165
OUTH ST	165
OUTH T	165
OUT_T	165

P

PABORT	760
PALERT (P)	758
PAUSE	1263
PHASE	199
PHASECHG	201
PHASEEND	203
PID	1203
PIDCONT (P)	1230
PIDINIT (P)	1228
PIDPRMW (P)	1235
PIDRUN (P)	1234
PIDSTOP (P)	1233
PLF	186
PLS	184
PLSY	1004
PLUS (P) (_U)	231
PLUS_2_M2	1580
PLUS_M2	1571, 1580
PLUSP_2_M2	1580
PLUSP_M2	1571, 1580
POFF (P)	599
POW (P)	957
POWD (P)	959
PSCAN (P)	601
PSTOP (P)	597
PWM	1006

Q

QDRSET (P)	985
----------------------	-----

R

RAD (P)	933
RADD (P)	937
RAMPQ	1011
RCL (P)	606
RCR (P)	603
REAL_TO_DINT (_E)	1346
REAL_TO_DINT_E_M2	1585
REAL_TO_INT (_E)	1344
REAL_TO_INT_E_M2	1585
REAL_TO_LREAL (_E)	1348
REAL_TO_LREAL_E_M2	1585

REAL_TO_M_REAL_E_M2	1585
REAL_TO_STR_E_M2	1585
REAL_TO_STRING (_E)	1350
REPLACE (_E)	1453
REPLACE_E_M2	1585
RET	578
RFS (P)	1103
RIGHT (_E)	1443
RIGHT (P)	820
RIGHT_E_M2	1585
RIGHT_M2	1581
RND (P)	973
ROL (_E)	1420
ROL (P)	606
ROL_E_M2	1585
ROL_M2	1581
ROR (_E)	1422
ROR (P)	603
ROR_E_M2	1585
ROR_M2	1581
ROTC	1008
RS (_E)	1470
RS_E_M2	1587
RSET (P)	983
RS_M2	1587
RST	178, 1261, 1269
RST F	182
RSTART	1265
R_TRIG (_E)	1472
R_TRIG_E_M2	1587
R_TRIG_M2	1587

S

S. SOCRCVS	1153
S (P). DATE-	1093
S (P). DATE+	1091
S (P). DATERD	1089
S (P). DEVL D	624
S (P). PIDCONT	1219
S (P). PIDINIT	1216
S (P). PIDPRMW	1224
S (P). PIDRUN	1223
S (P). PIDSTOP	1222
S (P). RTREAD	1136
S (P). RTWRITE	1138
S (P). SOCRDATA	1165
S (P). ZCOM	1107
S (P) _DATEMINUS	1093
S (P) _DATEPLUS	1091
S (P) _DATERD	1089
S (P) _DEVLD	624
S (P) _PIDCONT	1219
S (P) _PIDINIT	1216
S (P) _PIDPRMW	1224
S (P) _PIDRUN	1223
S (P) _PIDSTOP	1222
S (P) _RTREAD	1136
S (P) _RTWRITE	1138
S (P) _SOCRDATA	1165
S (P) _ZCOM	1107
SCJ	553
SCL (P) (_U)	693
SCL2 (P) (_U)	699
SCL2_M2	1581
SCL2P_M2	1581

SCL_M2	1581	SP. SOCSND	1155
SCLP_M2	1581	SP_CONTSW	1274
SEC2DATE(P) (_U)	1076	SPD	1002
SEC2TIME(P)	1072	SP_DEVST	626
SEG(P)	498	SP_ECPRTCL	1167
SEL(_E)	1428	SP_FCOPY	664
SEL_E_M2	1585	SP_FDELETE	661
SERDATA(P)	703	SP_FMOVE	669
SERMM(P)	707	SP_FREAD	628
SET	176, 1259, 1267	SP_FRENAME	674
SET F	180	SP_FSTATUS	677
SET_BIT_OF_INT(_E)	1383	SP_FTPGET	1186
SET_BIT_OF_INT_E_M2	1585	SP_FTPPUT	1181
SFL(P)	343	SP_FWRITE	645
SFR(P)	341	SP_SIDRD	1284
SFT(P)	192	SP_SLMPNSND	1174
SFTBL(P)	369	SP_SOCCINF	1158
SFTBR(P)	365	SP_SOCCLOSE	1148
SFTDWL(P)	385	SP_SOCCSET	1160
SFTDWR(P)	381	SP_SOCOPEN	1145
SFTEDL(P)	401	SP_SOCRVCV	1150
SFTEDR(P)	397	SP_SOCRMODE	1162
SFTEL(P)	393	SP_SOCSND	1155
SFTER(P)	389	SQRT(_E)	1390
SFTL(P)	371	SQRT(P)	749
SFTR(P)	367	SQRT_E_M2	1585
SFTWL(P)	377	SR(_E)	1468
SFTWR(P)	373	SR_E_M2	1587
SHL(_E)	1416	SR_M2	1587
SHL_E_M2	1585	SRND(P)	974
SHR(_E)	1418	S_SOCRCVS	1153
SHR_E_M2	1585	STMR.	1000
SIMASK	566	STOP.	207
SIN(_E)	1395	STR(P) (_U)	791
SIN(P)	897	STRDEL(P)	833
SIND(P)	909	STRINGMOV(P)	770
SIND_M2	1581	STRINGMOV(P)_WS.	772
SIN_E_M2	1585	STRINGPLUS(P)	768
SIN_M2	1581	STRING_TO_BCD(_E)	1370
SJIS2WS(P)	814	STRING_TO_BOOL(_E)	1363
SJIS2WSB(P)	816	STRING_TO_DINT(_E)	1368
SMOV(P)	520	STRING_TO_DWORD(_E)	1365
SORTD(_U)	725	STRING_TO_INT(_E)	1366
SORT_M2	1581	STRING_TO_REAL(_E)	1372
SORTTBL(_U)	729	STRING_TO_TIME(_E)	1375
SORTTBL2(_U)	733	STRING_TO_WORD(_E)	1364
SP. CONTSW	1274	STRINS(P)	831
SP. DEVST	626	STR_M2	1582
SP. ECPRTCL	1167	STRP_M2	1582
SP. FCOPY	664	STR_TO_BCD_E_M2.	1585
SP. FDELETE	661	STR_TO_BOOL_E_M2	1585
SP. FMOVE	669	STR_TO_DINT_E_M2	1585
SP. FREAD	628	STR_TO_DWORD_E_M2.	1585
SP. FRENAME	674	STR_TO_INT_E_M2.	1585
SP. FSTATUS	677	STR_TO_REAL_E_M2	1586
SP. FTPGET	1186	STR_TO_TIME_E_M2	1586
SP. FTPPUT	1181	STR_TO_WORD_E_M2	1586
SP. FWRITE	645	SUB(_E)	1406
SP. SIDRD	1284	SUB_E_M2	1586
SP. SLMPNSND	1174	SUB_TIME(_E)	1460
SP. SOCCINF	1158	SUB_TIME_E_M2	1586
SP. SOCCLOSE.	1148	SUM(P)	711
SP. SOCCSET	1160	SWAP(P)	543
SP. SOCOPEN	1145	SWAP_M2	1582
SP. SOCRVCV	1150	SWAPP_M2	1582
SP. SOCRMODE.	1162		

T

TAN(_E)	1397
TAN(P)	901
TAND(P)	913
TAND_M2	1582
TAN_E_M2	1586
TAN_M2	1582
TCMP(P)	1085
TEST(P)	335
TIMCHK	1097
TIME2SEC(P)	1070
TIMER_100_FB_M	1491
TIMER_10_FB_M	1491
TIMER_CONT_FB_M	1491
TIMER_CONTHFB_M	1491
TIMER_HIGH_FB_M	1491
TIMER_LOW_FB_M	1491
TIME_TO_BOOL(_E)	1356
TIME_TO_BOOL_E_M2	1586
TIME_TO_DINT(_E)	1360
TIME_TO_DINT_E_M2	1586
TIME_TO_DWORD(_E)	1358
TIME_TO_DWORD_E_M2	1586
TIME_TO_INT(_E)	1359
TIME_TO_INT_E_M2	1586
TIME_TO_STR_E_M2	1586
TIME_TO_STRING(_E)	1361
TIME_TO_WORD(_E)	1357
TIME_TO_WORD_E_M2	1586
TO(P)	1113
TOD(P)	1122
TOF(_E)	1489
TOF_HIGH_E_M2	1587
TOF_HIGH_M2	1587
TO_M2	1582
TON(_E)	1487
TON_HIGH_E_M2	1587
TON_HIGH_M2	1587
TOP_M2	1582
TP(_E)	1485
TP_HIGH_E_M2	1587
TP_HIGH_M2	1587
TRAN	1273
TTMR	998
TYPERD(P)	1127
TZCP(P)	1087

U

UDCNT1	993
UDCNT2	996
UDINT2DBL(P)	884
UDINT2DINT(P)	451
UDINT2FLT(P)	877
UDINT2INT(P)	447
UDINT2UINT(P)	449
UINT2DBL(P)	882
UINT2DINT(P)	437
UINT2FLT(P)	874
UINT2INT(P)	435
UINT2UDINT(P)	439
UNI(P)	502
UNIINFRD(P)	1131

V

VAL(P)(_U)	482
VAL_M2	1582
VALP_M2	1582

W

WAND(P)	291, 293
WDT(P)	569
WOR(P)	301, 303
WORD_TO_BOOL(_E)	1298
WORD_TO_BOOL_E_M2	1586
WORD_TO_DINT(_E)	1301
WORD_TO_DINT_E_M2	1586
WORD_TO_DWORD(_E)	1299
WORD_TO_DWORD_E_M2	1586
WORD_TO_INT(_E)	1300
WORD_TO_INT_E_M2	1586
WORD_TO_STR_E_M2	1586
WORD_TO_STRING(_E)	1304
WORD_TO_TIME(_E)	1303
WORD_TO_TIME_E_M2	1586
WS2SJIS(P)	812
WSFL(P)	379
WSFR(P)	375
WSUM(P)(_U)	741
WSUM_M2	1582
WSUMP_M2	1582
WTOB(P)	508
WXNR(P)	321, 323
WXOR(P)	311, 313

X

XCALL	592
XCH(P)	537
XOR(_E)	1424
XOR_E_M2	1586

Z

Z_ABRST1_M2	1571, 1582
Z_ABRST2_M2	1571, 1582
Z_ABRST3_M2	1571, 1582
Z_ABRST4_M2	1571, 1582
ZCP(P)(_U)	220
ZONE(P)(_U)	689
ZONE_M2	1582
ZONEP_M2	1582
ZP_BUFRCV_M2	1573, 1583
ZP_BUFSND_M2	1573, 1583
ZP_CLOSE_M2	1573, 1583
ZP_CSET_M2	1573, 1583
ZP_ERRCLR_M2	1576, 1583
ZP_ERRRD_M2	1576, 1583
ZPOP(P)	977, 981
ZPOP(P)_2	981
ZP_OPEN_M2	1580, 1583
ZP_PFWRT_M2	1580, 1583
ZP_PINIT_M2	1580, 1583
ZP_PSTR1_M2	1581, 1583
ZP_PSTR2_M2	1581, 1583
ZP_PSTR3_M2	1581, 1583
ZP_PSTR4_M2	1581, 1583
ZP_REMTO_M2	1581, 1583
ZP_TEACH1_M2	1582, 1583

ZP_TEACH2_M2	1582, 1583
ZP_TEACH3_M2	1582, 1583
ZP_TEACH4_M2	1582, 1583
ZP_UINI_M2	1582, 1583
ZPUSH(P)	975, 978
ZPUSH(P)_2	978
Z_RECVS_M2	1581, 1582
ZRRDB(P)	987
ZRWRB(P)	989
Z_UINI_M2	1582

修訂記錄

*本手冊編號在封底的左下角。

修訂日期	*手冊編號	修改內容
2014年8月	SH (NA) -081323CHT-A	第一版
2015年1月	SH (NA) -081323CHT-B	第二版 部分修改
2015年3月	SH (NA) -081323CHT-C	■第三版 部分修改
2015年7月	SH (NA) -081323CHT-D	■第四版 部分修改
2016年9月	SH (NA) -081323CHT-E	■第五版 部分修改
2016年11月	SH (NA) -081323CHT-F	■第六版 部分修改
2017年2月	SH (NA) -081323CHT-G	■第七版 部分修改
2017年5月	SH (NA) -081323CHT-H	■第八版 部分修改
2017年11月	SH (NA) -081323CHT-I	■第九版 部分修改
2018年6月	SH (NA) -081323CHT-J	• 程式控制指令移到MELSEC iQ-R 程式手冊(過程控制FB/指令篇) • 模組專用指令移到MELSEC iQ-R 程式手冊(模組專用指令篇) ■第十版 部分修改
2018年11月	SH (NA) -081323CHT-K	■第十一版 部分修改
2019年6月	SH (NA) -081323CHT-L	■第十二版 部分修改
2019年11月	SH (NA) -081323CHT-M	■第十三版 部分修改
2020年6月	SH (NA) -081323CHT-N	■第十四版 部分修改
2020年12月	SH (NA) -081323CHT-O	■第十五版 部分修改
2021年6月	SH (NA) -081323CHT-P	■第十六版 部分修改
2021年12月	SH (NA) -081323CHT-Q	■第十七版 部分修改
2022年4月	SH (NA) -081323CHT-R	■第十八版 部分修改
2023年12月	SH (NA) -081323CHT-S	■第十九版 部分修改

日語版手冊編號：SH-081226-AC

本手冊不授予工業產權或任何其它類型的權利，也不授予任何專利許可。三菱電機對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。

© 2014 MITSUBISHI ELECTRIC CORPORATION

保固

使用之前請確認以下產品保固的詳細說明。

1. 免費保固期限和免費保固範圍

在免費保固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。

【免費保固期限】

免費保固期限為自購買日或交貨的 36 個月內。

注意產品從三菱電機生產並出貨之後，最長分銷時間為 6 個月，生產後最長的免費保固期為 42 個月。維修零組件的免費保固期不得超過修理前的免費保固期。

【免費保固範圍】

- (1) 範圍局限於按照使用說明書、用戶手冊及產品上的警示標語規定的使用狀態，使用方法和環境正常使用的情況下。
- (2) 以下情況下，即使在免費保固期內，也要收取維修費用。
 - ① 因不適當存放或搬運、用戶過失或疏忽而引起的故障。因使用者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、風災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（包括備品）。

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. 機會損失、間接損失不在品質保證責任範圍

無論在保修期內的內和外，對於以下三菱將不承擔責任。

- (1) 非三菱責任原因所導致的損害。
- (2) 因三菱產品故障原因而引起客戶的機會損失，利潤的損失。
- (3) 無論三菱是否預測由特殊原因而導致的損失和間接損失、事故賠償、以及三菱產品以外的損失。
- (4) 對於用戶更換設備，重新調整了現場的機械設備，測試及其它作業等的補償。

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。

商標

Microsoft, Microsoft Access, Excel, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, Windows NT, Windows Server, Windows Vista, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as ‘™’ or ‘®’ are not specified in this manual.

SH(NA)-081323CHT-S(2312)STC

MODEL: R-P-MF-CHT

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

Specifications subject to change without notice.