MITSUBISHI ELECTRIC

Programmable Controller

MELSEC iQ-F series

MELSEC iQ-F
FX5 Programming Manual
(Instructions, Standard Functions/Function Blocks)

SAFETY PRECAUTIONS

(Read these precautions before use.)
Before using the MELSEC iQ-F series PLCs, please read the manual supplied with each product and the relevant manuals introduced in that manual carefully and pay full attention to safety to handle the product correctly.
Store this manual in a safe place so that it can be taken out and read whenever necessary. Always forward it to the end user.

INTRODUCTION

This manual describes the instructions and functions/function blocks required for programming MELSEC iQ-F series systems. This manual and the related manuals should be read and the functions and performance of the MELSEC iQ-F series PLC should be understood before attempting to use the unit.
However, before using a program example introduced in this manual to the actual system, always confirm that it poses no problem for control of the target system.

Regarding use of this product

- This product has been manufactured as a general-purpose part for general industries, and has not been designed or manufactured to be incorporated in a device or system used in purposes related to human life.
- Before using the product for special purposes such as nuclear power, electric power, aerospace, medicine or passenger movement vehicles, consult with Mitsubishi Electric.
- This product has been manufactured under strict quality control. However when installing the product where major accidents or losses could occur if the product fails, install appropriate backup or failsafe functions in the system.

Note

- If in doubt at any stage during the installation of the product, always consult a professional electrical engineer who is qualified and trained in the local and national standards. If in doubt about the operation or use, please consult the nearest Mitsubishi Electric representative.
- Since the examples indicated by this manual, technical bulletin, catalog, etc. are used as a reference, please use it after confirming the function and safety of the equipment and system. Mitsubishi Electric will accept no responsibility for actual use of the product based on these illustrative examples.
- This manual content, specification etc. may be changed without a notice for improvement.
- The information in this manual has been carefully checked and is believed to be accurate; however, if you have noticed a doubtful point, an error, etc., please contact the nearest Mitsubishi Electric representative. When doing so, please provide the manual number given at the end of this manual.

CONTENTS

SAFETY PRECAUTIONS 1
INTRODUCTION 1
RELEVANT MANUALS 16
TERMS 16
GENERIC TERMS AND ABBREVIATIONS 16
HOW TO READ THIS MANUAL 17

PART 1 OVERVIEW

CHAPTER 1 OVERVIEW 22

1.1 Instruction Configuration 22
1.2 Data Specification Method 23
Bit data 26
16-bit data (word data) 27
32-bit data (double word data) 29
Real number data (floating-point data) 32
Character string data 33
1.3 Execution Condition 35
1.4 Acceleration of Instruction Processing Time 36

CHAPTER 2 PRECAUTIONS ON PROGRAMMING 39

2.1 Errors Common to Instructions 39
2.2 Checking the Ranges of Instruction Runtime Devices and Labels 39
2.3 Operations Arising when the OUT, SET/RST, and PLS/PLF Instructions of the Same Device are Used . . . 40
2.4 Handling general flags 45
2.5 Standard Function/Function Block Return Values 47

PART 2 INSTRUCTION/FUNCTION LIST

CHAPTER 3 CPU MODULE INSTRUCTION 50

3.1 Sequence Instruction 50
3.2 Basic Instruction 54
3.3 Application Instruction 69
3.4 Step Ladder Instructions 90
3.5 PID Control Instruction 90
3.6 SFC Program Instructions 91

CHAPTER 4 MODULE SPECIFIC INSTRUCTION 93

4.1 Network Common Instruction 93
4.2 Ethernet Instruction 93
4.3 CC-Link IE TSN Instructions 95
4.4 CC-Link IE Field Network Instruction 95
4.5 High-speed Counter Instruction 96
4.6 External Device Communication Instruction 97
4.7 Positioning Instruction 98
4.8 BFM Device Read/ Write Instruction 99

CHAPTER 5 STANDARD FUNCTIONS/FUNCTION BLOCKS 100

5.1 **Standard Functions** 100
Type conversion functions 100
Standard functions of one numeric variable 105
Standard arithmetic functions 106
Standard bit shift functions 107
Standard bitwise boolean functions 107
Standard selection functions 108
Standard comparison functions 108
Standard character string functions 109
Time data functions 110

5.2 **Standard Function Blocks** 111
Bistable function blocks 111
Edge detection function blocks 111
Counter function blocks 111
Timer function blocks 112

PART 3 CPU MODULE INSTRUCTIONS

CHAPTER 6 SEQUENCE INSTRUCTIONS 114

6.1 **Contact Instructions** 114
Operation start, series connection, parallel connection 114
Pulse operation start, pulse series connection, pulse parallel connection 117
Pulse NOT operation start, pulse NOT series connection, pulse NOT parallel connection 121

6.2 **Association Instruction** 123
Ladder block series/parallel connection 123
Storing/reading/clearing the operation result 125
Inverting the operation result 128
Converting the operation result into a pulse 129

6.3 **Output Instructions** 130
Out (excluding the timer, counter and annunciator) 130
Timer 132
Counter 135
Long counter 137
Annunciator 139
Setting devices (excluding annunciator) 141
Resetting devices (excluding annunciator) 143
Setting annunciator 145
Resetting annunciator 147
Setting annunciator (with check time) 149
Resetting annunciator (smallest number reset) 151
Rising edge output 152
Falling edge output 154
Inverting the bit device output 156
Inverting the bit device output 157

6.4 **Shift Instructions** 159
Shifting bit devices 159
Shifting 16-bit data to the right by n bit(s) 161
Shifting 16-bit data to the left by n bit(s) 163

Shifting n-bit data to the right by 1 bit . . . 165
Shifting n-bit data to the left by 1 bit . . . 167
Shifting n-word data to the right by 1 word . . . 169
Shifting n-word data to the left by 1 word . . . 170
Shifting n-bit(s) data to the right by (n) bit(s) . . . 171
Shifting n-bit data to the left by n bit(s) . . . 173
Shifting n-word data to the right by n word(s) . . . 175
Shifting n-word data to the left by n word(s) . . . 177
6.5 Master Control Instruction . . . 179
Setting/resetting the master control . . . 179
6.6 Termination Instructions . . . 183
Ending the main routine program . . . 183
Ending the sequence program . . . 186
6.7 Stop Instruction . . . 188
Stopping the sequence program . . . 188

CHAPTER 7 BASIC INSTRUCTIONS 189

7.1 Comparison Operation Instructions . . . 189
Comparing 16-bit binary data . . . 189
Comparing 32-bit binary data . . . 192
Comparison output 16-bit binary data . . . 194
Comparison output 32-bit binary data . . . 196
Comparing 16-bit binary data band . . . 198
Comparing 32-bit binary data band . . . 200
Comparing 16-bit binary block data . . . 202
Comparing 32-bit binary block data . . . 205
7.2 Arithmetic Operation Instructions . . . 208
Adding 16-bit binary data . . . 208
Subtracting 16-bit binary data . . . 214
Adding 32-bit binary data . . . 220
Subtracting 32-bit binary data . . . 226
Multiplying 16-bit binary data . . . 232
Dividing 16-bit binary data . . . 236
Multiplying 32-bit binary data . . . 240
Dividing 32-bit binary data . . . 244
Adding BCD 4-digit data . . . 248
Subtracting BCD 4-digit data . . . 250
Adding BCD 8-digit data . . . 253
Subtracting BCD 8-digit data . . . 256
Multiplying BCD 4-digit data . . . 259
Dividing BCD 4-digit data . . . 261
Multiplying BCD 8-digit data . . . 263
Dividing BCD 8-digit data . . . 265
Adding 16-bit binary block data . . . 267
Subtracting 16-bit binary block data . . . 269
Adding 32-bit binary block data . . . 271
Subtracting 32-bit binary block data . . . 273
Incrementing 16-bit binary data . . . 275
Decrementing 16-bit binary data . . . 277
Incrementing 32-bit binary data . . . 278

Decrementing 32-bit binary data . . . 279
7.3 Logical Operation Instructions . . . 280
Performing an AND operation on 16-bit data . . . 280
Performing an AND operation on 32-bit data . . . 282
Performing an AND operation on 16-bit block data . . . 285
Performing an OR operation on 16-bit data . . . 287
Performing an OR operation on 32-bit data . . . 289
Performing an OR operation on 16-bit block data . . . 292
Performing an XOR operation on 16-bit data . . . 294
Performing an XOR operation on 32-bit data . . . 296
Performing an XOR operation on 16-bit block data . . . 299
Performing an XNOR operation on 16-bit data . . . 301
Performing an XNOR operation on 32-bit data . . . 303
Performing an XNOR operation on 16-bit block data . . . 306
7.4 Bit Processing Instructions . . . 308
Setting a bit in the word device . . . 308
Resetting a bit in the word device . . . 309
Performing a 16-bit test . . . 310
Performing a 32-bit test . . . 312
Batch-resetting bit devices . . . 314
Batch-resetting devices . . . 315
7.5 Data Conversion Instructions . . . 318
Converting binary data to BCD 4-digit data . . . 318
Converting binary data to BCD 8-digit data . . . 320
Converting BCD 4-digit data to binary data . . . 322
Converting BCD 8-digit data to binary data . . . 324
Converting single-precision real number to 16-bit signed binary data . . . 326
Converting single-precision real number to 16-bit unsigned binary data . . . 328
Converting single-precision real number to 32-bit signed binary data . . . 330
Converting single-precision real number to 32-bit unsigned binary data . . . 332
Converting 16-bit signed binary data to 16-bit unsigned binary data . . . 334
Converting 16-bit signed binary data to 32-bit signed binary data . . . 335
Converting 16-bit signed binary data to 32-bit unsigned binary data . . . 336
Converting 16-bit unsigned binary data to 16-bit signed binary data . . . 337
Converting 16-bit unsigned binary data to 32-bit signed binary data . . . 338
Converting 16-bit unsigned binary data to 32-bit unsigned binary data . . . 339
Converting 32-bit signed binary data to 16-bit signed binary data . . . 340
Converting 32-bit signed binary data to 16-bit unsigned binary data . . . 341
Converting 32-bit signed binary data to 32-bit unsigned binary data . . . 342
Converting 32-bit unsigned binary data to 16-bit signed binary data . . . 343
Converting 32-bit unsigned binary data to 16-bit unsigned binary data . . . 344
Converting 32-bit unsigned binary data to 32-bit signed binary data . . . 345
Converting 16-bit binary data to Gray code . . . 346
Converting 32-bit binary data to Gray code . . . 347
Converting Gray code to 16-bit binary data . . . 349
Converting Gray code to 32-bit binary data . . . 350
Converting decimal ASCII to 16-bit binary data . . . 352
Converting decimal ASCII to 32-bit binary data . . . 356
Converting ASCII to HEX . . . 360
Converting character string to 16-bit binary data . . . 364
Converting character string to 32-bit binary data . . . 367

Two's complement of 16-bit binary data (sign inversion) . . . 370
Two's complement of 32-bit binary data (sign inversion) . . . 372
Decoding from 8 to 256 bits . . . 373
Encoding from 256 to 8 bits . . . 375
Seven-segment decoding . . . 377
Seven Segment With Latch . . . 379
Separating 4 bits from 16-bit data . . . 382
Connecting 4 bits to 16-bit data . . . 384
Separating the specified number of bits . . . 386
Connecting the specified number of bits . . . 388
Separating data in byte units . . . 390
Connecting data in byte units . . . 392
7.6 Digital Switch . . . 395
7.7 Data Transfer Instructions . . . 397
Transferring 16-bit data . . . 397
Transferring 32-bit data . . . 399
Inverting and transferring 16-bit data . . . 401
Inverting and transferring 32-bit data . . . 403
Digit move . . . 404
Inverting and transferring 1-bit data . . . 406
Transferring 16-bit block data (65535 points maximum) . . . 407
Transferring identical 16-bit block data (65535 points maximum) . . . 409
Transferring identical 32-bit block data (65535 points maximum) . . . 411
Exchanging 16-bit data . . . 413
Exchanging 32-bit data . . . 415
Exchanging the upper and lower bytes of 16-bit data . . . 417
Exchanging the upper and lower bytes of 32-bit data . . . 418
Transferring 1-bit data . . . 419
Transferring octal bits (16-bit data) . . . 420
Transferring octal bits (32-bit data) . . . 422
Transferring n-bit data . . . 424

CHAPTER 8 APPLICATION INSTRUCTION 426

8.1 Rotation Instruction . . . 426
Rotating 16-bit data to the right . . . 426
Rotating 16-bit data to the left . . . 429
Rotating 32-bit data to the right . . . 432
Rotating 32-bit data to the left . . . 434
8.2 Program Branch Instruction . . . 436
Pointer branch . . . 436
Jump to END . . . 440
8.3 Program Execution Control Instruction . . . 441
Disabling/enabling interrupt programs . . . 441
Disabling the interrupt program with specified priority or lower . . . 443
Interrupt program mask . . . 447
Disabling/enabling the specified interrupt pointer . . . 449
Returning from the interrupt program . . . 451
Resetting the watchdog timer . . . 454
8.4 Structuring Instruction . . . 455
FOR to NEXT . . . 455

Forcibly terminating the FOR to NEXT instruction loop . . . 458
Calling a subroutine program . . . 460
Returning from the subroutine program . . . 465
Calling a subroutine program . . . 466
8.5 Data Table Operation Instruction . . . 468
Reading the oldest data from the data table . . . 468
Reading the newest data from the data table . . . 471
Writing data to the data table . . . 474
Inserting data to the data table . . . 476
Deleting data from the data table . . . 478
8.6 Reading/writing Data Instructions . . . 480
Reading data from the data memory . . . 481
Writing data to the data memory . . . 483
8.7 File Operation Instructions . . . 486
Reading data from the specified file . . . 486
Writing data to the specified file . . . 512
Deleting the specified file . . . 535
Copying the specified file . . . 543
Moving the specified file . . . 553
Renaming the specified file . . . 563
Acquiring the status of the specified file . . . 571
Error codes generated for file operation instructions . . . 579
8.8 Extended File Register Operation Instruction . . . 580
Reading extended file register . . . 580
Writing extended file register . . . 583
Batch initialization function of extended file register . . . 586
8.9 Character String Operation Instruction . . . 589
Comparing character strings . . . 589
Concatenating character strings . . . 592
Transferring character strings . . . 596
Transferring Unicode string data . . . 598
Converting 16-bit binary data to decimal ASCII . . . 600
Converting 32-bit binary data to decimal ASCII . . . 605
Converting HEX code data to ASCII . . . 611
Converting 16-bit binary data to character string . . . 615
Converting 32-bit binary data to character string . . . 618
Converting single-precision real number to character string . . . 621
Converting Unicode character string to Shift JIS character string . . . 628
Converting shift JIS character string to Unicode character string (without byte order mark) . . . 631
Converting shift JIS character string to Unicode (with byte order mark) . . . 634
Detecting a character string length . . . 637
Extracting character string data from the right . . . 639
Extracting character string data from the left . . . 642
Storing the specified number of character strings . . . 645
Replacing the specified number of character strings . . . 648
Searching character string . . . 652
Inserting character string . . . 655
Deleting character string . . . 657
8.10 Real Number Instruction . . . 659
Comparing single-precision real numbers . . . 659
Single-precision real number comparison . . . 661

Single-precision real number data band comparison . . . 663
Adding single-precision real numbers . . . 665
Subtracting single-precision real numbers . . . 669
Adding single-precision real numbers . . . 673
Subtracting single-precision real numbers . . . 675
Multiplying single-precision real numbers . . . 677
Dividing single-precision real numbers . . . 679
Multiplying single-precision real numbers . . . 681
Dividing single-precision real numbers . . . 683
Converting 16-bit signed binary data to single-precision real number . . . 685
Converting 16-bit unsigned binary data to single-precision real number . . . 686
Converting 32-bit signed binary data to single-precision real number . . . 687
Converting 32-bit unsigned binary data to single-precision real number . . . 688
Converting character string to single-precision real number . . . 689
Converting binary floating point to decimal floating point . . . 694
Converting decimal floating point to binary floating point . . . 696
Inverting the sign of single-precision real number . . . 698
Transferring single-precision real number data . . . 700
Calculating the sine of single-precision real number . . . 702
Calculating the cosine of single-precision real number . . . 704
Calculating the tangent of single-precision real number . . . 706
Calculating the arc sine of single-precision real number . . . 708
Calculating the arc cosine of single-precision real number . . . 711
Calculating the arc tangent of single-precision real number . . . 714
Converting single-precision real number angle to radian . . . 716
Converting single-precision real number radian to angle . . . 718
Calculating the square root of single-precision real number . . . 720
Calculating the exponent of single-precision real number . . . 722
Calculating the natural logarithm of single-precision real number . . . 724
Calculating the exponentiation of single-precision real number . . . 726
Calculating the common logarithm of single-precision real number . . . 728
Searching the maximum value of single-precision real number . . . 730
Searching the minimum value of single-precision real number . . . 732
8.11 Random Number Instruction . . . 734
Generating random number . . . 734
8.12 Index Register Operation Instruction . . . 736
Saving all data of the index register . . . 736
Returning all data of the index register . . . 739
Saving the selected data of the index register and long index register . . . 740
Returning the selected data of the index register and long index register . . . 742
8.13 Data Control Instruction . . . 743
Upper and lower limit control of 16-bit binary data . . . 743
Upper and lower limit control of 32-bit binary data . . . 745
Dead band control of 16-bit binary data . . . 747
Dead band control of 32-bit binary data . . . 749
Zone control of 16-bit binary data . . . 752
Zone control of 32-bit binary data . . . 754
Scaling 16-bit binary data (point coordinates) . . . 756
Scaling 32-bit binary data (point coordinates) . . . 759
Scaling 16-bit binary data (XY coordinates) . . . 762
Scaling 32-bit binary data (XY coordinates) . . . 766

8.14 Special Timer Instruction . . . 769
Teaching timer . . . 769
Special function timer . . . 772
8.15 Special Counter Instruction . . . 774
Signed 32-bit bi-directional counters . . . 774
8.16 Shortcut Control Instruction . . . 776
Rotary table shortest direction control . . . 776
8.17 Ramp Signal Instruction . . . 779
Ramp signal . . . 779
8.18 Pulse Related Instruction . . . 782
Measuring the density of 16 bit binary pulses . . . 782
Measuring the density of 32 bit binary pulses . . . 788
16 bit binary pulse output . . . 793
32 bit binary pulse output . . . 801
16 bit binary pulse width modulation . . . 809
32 bit binary pulse width modulation . . . 816
8.19 Input Matrix Instruction . . . 823
Input matrix . . . 823
8.20 Initial State . . . 827
Initial State . . . 827
8.21 Drum Sequence . . . 838
16-bit binary data absolute method . . . 838
32-bit binary data absolute method . . . 840
Relative method . . . 842
8.22 Check Code . . . 845
Check code . . . 845
8.23 Data Operation Instruction . . . 848
Searching 16-bit data . . . 848
Searching 32-bit data . . . 850
Bit check of 16-bit data . . . 852
Bit check of 32-bit data . . . 854
Bit judgment of 16-bit data . . . 855
Bit judgment of 32-bit data . . . 857
Searching the maximum value of 16-bit data . . . 859
Searching the maximum value of 32-bit data . . . 861
Searching the minimum value of 16-bit data . . . 863
Searching the minimum value of 32-bit data . . . 865
Sorting 16-bit data . . . 867
Sorting 16-bit data 2 . . . 870
Sorting 32-bit data 2 . . . 873
Adding 16-bit data . . . 876
Adding 32-bit data . . . 878
Calculating the mean value of 16-bit data . . . 880
Calculating the mean value of 32-bit data . . . 882
Calculating the square root of 16-bit data . . . 884
Calculating the square root of 32-bit data . . . 886
CRC calculation . . . 887
8.24 Indirect Address Read Instruction . . . 890
Reading the indirect address . . . 890
8.25 Clock Instruction . . . 892
Reading clock data . . . 892

Writing clock data . . . 894
Adding clock data . . . 897
Subtracting clock data . . . 900
Converting time data from hour/minute/second to seconds in 16 bits . . . 903
Converting time data from hour/minute/second to seconds in 32 bits . . . 905
Converting time data from seconds to hour/minute/second in 16 bits . . . 907
Converting time data from seconds to hour/minute/second in 32 bits . . . 909
Comparing date data . . . 911
Comparing time data . . . 914
Comparing clock data . . . 917
Comparing clock data zones . . . 920
8.26 Timing Check Instruction . . . 923
Generating timing pulses . . . 923
Hour meter . . . 926
8.27 Module Access Instruction . . . 930
I/O refresh . . . 930
Reading 1-word/2-word data from another module . . . 932
Writing 1-word/2-word data to another module . . . 936
Reading 1-word/2-word data from another module . . . 939
Writing 1-word/2-word data to another module (32-bit specification) . . . 942
8.28 Logging Instructions . . . 945
Setting trigger logging . . . 945
Resetting trigger logging . . . 946
8.29 Real-time Monitor Function Instruction . . . 947

CHAPTER 9 STEP LADDER INSTRUCTIONS 949

9.1 Starts/Ends Step Ladder . . . 949

CHAPTER 10 PID CONTROL INSTRUCTION 953

10.1 PID Control Loop . . . 953

CHAPTER 11 SFC PROGRAM INSTRUCTIONS 956

11.1 SFC Control Instructions . . . 956
Checking the status of a step . . . 956
Checking the status of a block . . . 958
Batch-reading the status of steps . . . 960
Starting a block . . . 970
Ending a block . . . 971
Pausing a block . . . 972
Restarting a block . . . 974
Activating a step . . . 976
Deactivating a step . . . 978
Activating/deactivating a step . . . 980
Batch-deactivating a step . . . 982
11.2 SFC Dedicated Instruction . . . 984
Creating a dummy transition condition . . . 984

PART 4 MODULE DEDICATED INSTRUCTION

CHAPTER 12 NETWORK COMMON INSTRUCTION 986

12.1 **Link Dedicated Instructions** **988**
Reading data from another station programmable controller 988
Reading data from another station programmable controller (with notification) 994
Writing data to another station programmable controller. 1000
Writing data to another station programmable controller (with notification). 1008
Sending data to another station programmable controller. 1016
Receiving data from another station programmable controller 1024

CHAPTER 13 ETHERNET INSTRUCTION 1030

13.1 **Built-in Ethernet Function Instruction** **1030**
Opening a connection 1030
Closing a connection 1033
13.2 **Socket Communications Function Instruction** **1035**
Reading receive data during the END processing. 1035
Sending data 1038
Reading connection information 1041
Reading socket communications receive data 1043
13.3 **Predefined Protocol Support Function Instruction** **1045**
Executing the registered protocols 1045
13.4 **SLMP Frame Send Instruction** **1049**
Sending the SLMP frame 1049
13.5 **File Transfer Function Instruction** **1054**
Sending FTP client files. 1054
Retrieving FTP client files 1059
13.6 **Ethernet Module** **1064**
Opening a connection 1064
Closing a connection 1067
Reading receive data. 1069
Sending data 1071

CHAPTER 14 CC-LINK IE TSN INSTRUCTION 1073

14.1 **Own Station Number/IP Address Setting** **1073**
14.2 **Sending an SLMP Frame** **1076**

CHAPTER 15 CC-LINK IE FIELD NETWORK INSTRUCTION 1085

15.1 **Setting parameters** **1085**
15.2 **Setting the station number to own station** **1088**

CHAPTER 16 HIGH-SPEED COUNTER INSTRUCTION 1091

16.1 **High-speed Processing Instruction** **1091**
Setting 32-bit data comparison 1091
Reset 32-bit data comparison 1094
Comparison of 32-bit data band 1097
Start/stop of the 16-bit data high-speed I/O function 1100
Start/stop of the 32-bit data high-speed I/O function 1104

16.2 High-speed Current Value Transfer Instruction 1108
High-speed current value transfer of 16-bit data 1108
High-speed current value transfer of 32-bit data 1110

CHAPTER 17 EXTERNAL DEVICE COMMUNICATION INSTRUCTION 1113

17.1 Serial Communication 2 1113
17.2 Inverter Communication Instruction 1115
Inverter operation monitoring (Status check) 1115
Inverter operations control (Drive) 1117
Inverter parameter read 1119
Inverter parameter write 1121
Inverter parameter block write 1123
Inverter multi command 1125
17.3 MODBUS Communication Instruction 1127
17.4 Predefined Protocol Support Function Instruction 1129

CHAPTER 18 POSITIONING INSTRUCTION 1133

18.1 Dedicated Instruction (Positioning Function) 1133
Zero return(OPR) with 16-bit data DOG search 1134
Zero return(OPR) with 32-bit data DOG search 1138
16-bit data interrupt positioning 1140
32-bit data interrupt positioning 1144
Positioning by one table operation 1148
Positioning by multiple table operation 1150
Multiple axes concurrent drive positioning 1152
32-bit data ABS current value read 1154
16-bit data variable speed pulse 1156
32-bit data variable speed pulse 1160
16-bit data relative positioning 1164
32-bit data relative positioning 1168
16-bit data absolute positioning 1172
32-bit data absolute positioning 1176
18.2 Positioning Module 1180
Restoring the absolute position 1180
Starting the positioning 1184
Teaching 1187
Backing up module data (writing data to the flash ROM) 1190
Initializing the Module 1193

CHAPTER 19 DIVIDED DATA READ/WRITE FROM/TO BFM INSTRUCTION 1196

19.1 Divided BFM Read 1196
19.2 Divided BFM Write 1200

PART 5 STANDARD FUNCTIONS

CHAPTER 20 TYPE CONVERSION FUNCTIONS 1204

20.1 Converting BOOL to WORD 1204
20.2 Converting BOOL to DWORD 1206
20.3 Converting BOOL to INT 1207

20.4 Converting BOOL to DINT 1208
20.5 Converting BOOL to TIME 1209
20.6 Converting BOOL to STRING 1210
20.7 Converting WORD to BOOL 1211
20.8 Converting WORD to DWORD 1212
20.9 Converting WORD to INT 1213
20.10 Converting WORD to DINT 1214
20.11 Converting WORD to TIME 1216
20.12 Converting DWORD to BOOL 1217
20.13 Converting DWORD to WORD 1218
20.14 Converting DWORD to INT 1220
20.15 Converting DWORD to DINT 1222
20.16 Converting DWORD to TIME 1223
20.17 Converting INT to BOOL 1224
20.18 Converting INT to WORD 1225
20.19 Converting INT to DWORD 1226
20.20 Converting INT to DINT 1228
20.21 Converting INT to BCD 1229
20.22 Converting INT to REAL 1231
20.23 Converting INT to TIME 1232
20.24 Converting INT to STRING 1233
20.25 Converting DINT to BOOL 1235
20.26 Converting DINT to WORD 1236
20.27 Converting DINT to DWORD 1238
20.28 Converting DINT to INT 1239
20.29 Converting DINT to BCD 1241
20.30 Converting DINT to REAL 1243
20.31 Converting DINT to TIME 1244
20.32 Converting DINT to STRING 1245
20.33 Converting BCD to INT 1247
20.34 Converting BCD to DINT 1249
20.35 Converting REAL to INT 1251
20.36 Converting REAL to DINT 1253
20.37 Converting REAL to STRING 1255
20.38 Converting TIME to BOOL 1258
20.39 Converting TIME to WORD 1259
20.40 Converting TIME to DWORD 1260
20.41 Converting TIME to INT 1261
20.42 Converting TIME to DINT 1262
20.43 Converting TIME to STRING 1263
20.44 Converting STRING to BOOL 1265
20.45 Converting STRING to INT 1266
20.46 Converting STRING to DINT 1268
20.47 Converting STRING to REAL 1270
20.48 Converting STRING to TIME 1273
20.49 Converting Bit Array to INT 1274
20.50 Converting Bit Array to DINT 1275
20.51 Converting INT to Bit Array 1276
20.52 Converting DINT to Bit Array 1277
20.53 Bit Array Copy 1278
20.54 Reading the Specified Bit of Word Label 1279

20.55 Writing the Specified Bit of Word Label 1280
20.56 Copying the Specified Bit of Word Label 1281
20.57 Unnecessary of Type Conversion 1282

CHAPTER 21 SINGLE NUMBER VARIABLE FUNCTIONS 1283

21.1 Absolute Value 1283
21.2 Square Root 1285
21.3 Natural Logarithm Operation 1286
21.4 Calculating the Common Logarithm 1287
21.5 Exponential Operation 1289
21.6 Sine Operation 1290
21.7 Cosine Operation 1291
21.8 Tangent Operation 1292
21.9 Arc Sine Operation 1294
21.10 Arc Cosine Operation 1296
21.11 Arc Tangent Operation 1298

CHAPTER 22 ARITHMETIC OPERATION FUNCTIONS 1300

22.1 Addition 1300
22.2 Multiplication 1302
22.3 Subtraction 1304
22.4 Division 1306
22.5 Remainder 1308
22.6 Exponentiation 1310
22.7 Move Operation 1312

CHAPTER 23 BIT SHIFT FUNCTIONS 1314

23.1 n-bit Left Shift 1314
23.2 n-bit Right Shift 1316
23.3 n-bit Left Rotation 1318
23.4 n-bit Right Rotation 1320

CHAPTER 24 STANDARD BITWISE BOOLEAN FUNCTIONS 1322

24.1 AND Operation, OR Operation, XOR Operation 1322
24.2 Logical Negation 1324

CHAPTER 25 SELECTION FUNCTIONS 1325

25.1 Selection 1325
25.2 Selecting Maximum/Minimum Value 1327
25.3 Limit Control 1329
25.4 Multiplexer 1331

CHAPTER 26 COMPARISON FUNCTIONS 1333

26.1 Compare 1333
26.2 Compare 1335

CHAPTER 27 CHARACTER STRING FUNCTIONS 1337

27.1 Character String Length Detection 1337
27.2 Extracting Character String Data from the Left/Right 1339
27.3 Extract Mid String 1341

27.4 Link Character Strings 1343
27.5 Inserting Character String 1345
27.6 Deleting Character String 1347
27.7 Replacing Character String 1349
27.8 Searching Character String 1352

CHAPTER 28 TIME DATA FUNCTIONS 1354

28.1 Addition 1354
28.2 Subtraction 1356
28.3 Multiplication 1358
28.4 Division 1360

PART 6 FUNCTION BLOCKS

CHAPTER 29 BISTABLE FUNCTION BLOCKS 1364

29.1 Bistable Function Blocks (Set Priority) 1364
29.2 Bistable Function Blocks (Reset Priority) 1366

CHAPTER 30 EDGE DETECTION FUNCTION BLOCKS 1368

30.1 Rising Edge Detector 1368
30.2 Falling Edge Detector 1370

CHAPTER 31 COUNTER FUNCTION BLOCKS 1372

31.1 Up Counter 1372
31.2 Down Counter 1374
31.3 Up-down Counter 1376
31.4 Counter Function Block 1379

CHAPTER 32 TIMER FUNCTION BLOCKS 1381

32.1 Pulse Timer 1381
32.2 On-delay Timer 1383
32.3 Off-delay Timer 1385
32.4 Timer Function Blocks 1387

APPENDIX 1390

Appendix 1 Instruction Processing Time 1390
Instruction processing time (High-speed instruction) 1390
Instruction processing time 1395
Appendix 2 Number of Instruction Steps 1431
Appendix 3 Added and Changed Functions 1452

INSTRUCTION INDEX 1455

REVISIONS 1463
WARRANTY 1465
TRADEMARKS 1466

RELEVANT MANUALS

Manual name <manual number>	Description
MELSEC iQ-F FX5 Programming Manual (Program Design) <JY997D55701>	Describes the specifications of ladder, ST, FBD/LD, and SFC programs, and labels.
MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/ Function Blocks) <JY997D55801> (This manual)	Describes the specifications of instructions and functions that can be used in programs.
GX Works3 Operating Manual <SH-081215ENG>	Describes the system configuration, parameter settings, and online operations of GX Works3.

TERMS

Unless otherwise specified, this manual uses the following terms.

For details of the FX3 devices that can be connected with the FX5, refer to the User's Manual (Hardware) of the CPU module to be used.

Terms	Description
Engineering tool	The product name of the software package for the MELSEC programmable controllers

GENERIC TERMS AND ABBREVIATIONS

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Generic term/abbreviation	Description
Battery	Different name for FX3U-32BL
Bus conversion module	Generic term for Bus conversion module (extension cable type) and Bus conversion module (extension connector type)
Extension power supply module	Generic term for FX5 extension power supply module and FX3 extension power supply module
FX3	Generic term for FX3S, FX3G, FX3GC, FX3U, and FX3UC PLCs
FX3 intelligent function module	Generic term for FX3U-4AD, FX3U-4DA, FX3U-4LC, FX3U-1PG, FX3U-2HC, FX3U-16CCL-M, FX3U-64CCL, FX3U-128ASL-M, and FX3U-32DP
FX5	Generic term for FX5S, FX5UJ, FX5U, and FX5UC PLCs
FX5 CPU module	Generic term for FX5S CPU module, FX5UJ CPU module, FX5U CPU module, and FX5UC CPU module
FX5S CPU module	Generic term for FX5S-30MR/ES, FX5S-30MT/ES, FX5S-30MT/ESS, FX5S-40MR/ES, FX5S-40MT/ES, FX5S-40MT/ESS, FX5S-60MR/ES, FX5S-60MT/ES, FX5S-60MT/ESS, FX5S-80MR/ES*1, FX5S-80MT/ES*1, and FX5S-80MT/ESS*1
FX5U CPU module	Generic term for FX5U-32MR/ES, FX5U-32MT/ES, FX5U-32MT/ESS, FX5U-64MR/ES, FX5U-64MT/ES, FX5U-64MT/ESS, FX5U-80MR/ES, FX5U-80MT/ES, FX5U-80MT/ESS, FX5U-32MR/DS, FX5U-32MT/DS, FX5U-32MT/DSS, FX5U-64MR/DS, FX5U-64MT/DS, FX5U-64MT/DSS, FX5U-80MR/DS, FX5U-80MT/DS, and FX5U-80MT/DSS
FX5UC CPU module	Generic term for FX5UC-32MT/D, FX5UC-32MT/DSS, FX5UC-64MT/D, FX5UC-64MT/DSS, FX5UC-96MT/D, FX5UC-96MT/DSS, FX5UC-32MT/DS-TS, FX5UC-32MT/DSS-TS, and FX5UC-32MR/DS-TS
FX5UJ CPU module	Generic term for FX5UJ-24MR/ES, FX5UJ-24MT/ES, FX5UJ-24MT/ESS, FX5UJ-40MR/ES, FX5UJ-40MT/ES, FX5UJ-40MT/ESS, FX5UJ-60MR/ES, FX5UJ-60MT/ES, and FX5UJ-60MT/ESS
GX Works3	The product name of the software package, SWnDND-GXW3, for the MELSEC programmable controllers (The 'n' represents a version.)
High-speed pulse input/output module	Generic term for FX5-16ET/ES-H and FX5-16ET/ESS-H
I/O module	Generic term for Input modules, Output modules, Input/output modules, Powered input/output modules, and High-speed pulse input/output modules
Input module	Generic term for Input modules (extension cable type) and Input modules (extension connector type)
Intelligent function module	Generic term for FX5 intelligent function modules and FX3 intelligent function modules
Intelligent module	The abbreviation for intelligent function modules
Output module	Generic term for Output modules (extension cable type) and Output modules (extension connector type)
Peripheral device	Generic term for engineering tools and GOTs
SD memory card	Generic term for NZ1MEM-2GBSD, NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD, L1MEM-2GBSD and L1MEM-4GBSD SD memory cards Abbreviation of Secure Digital Memory Card. Device that stores data using flash memory.

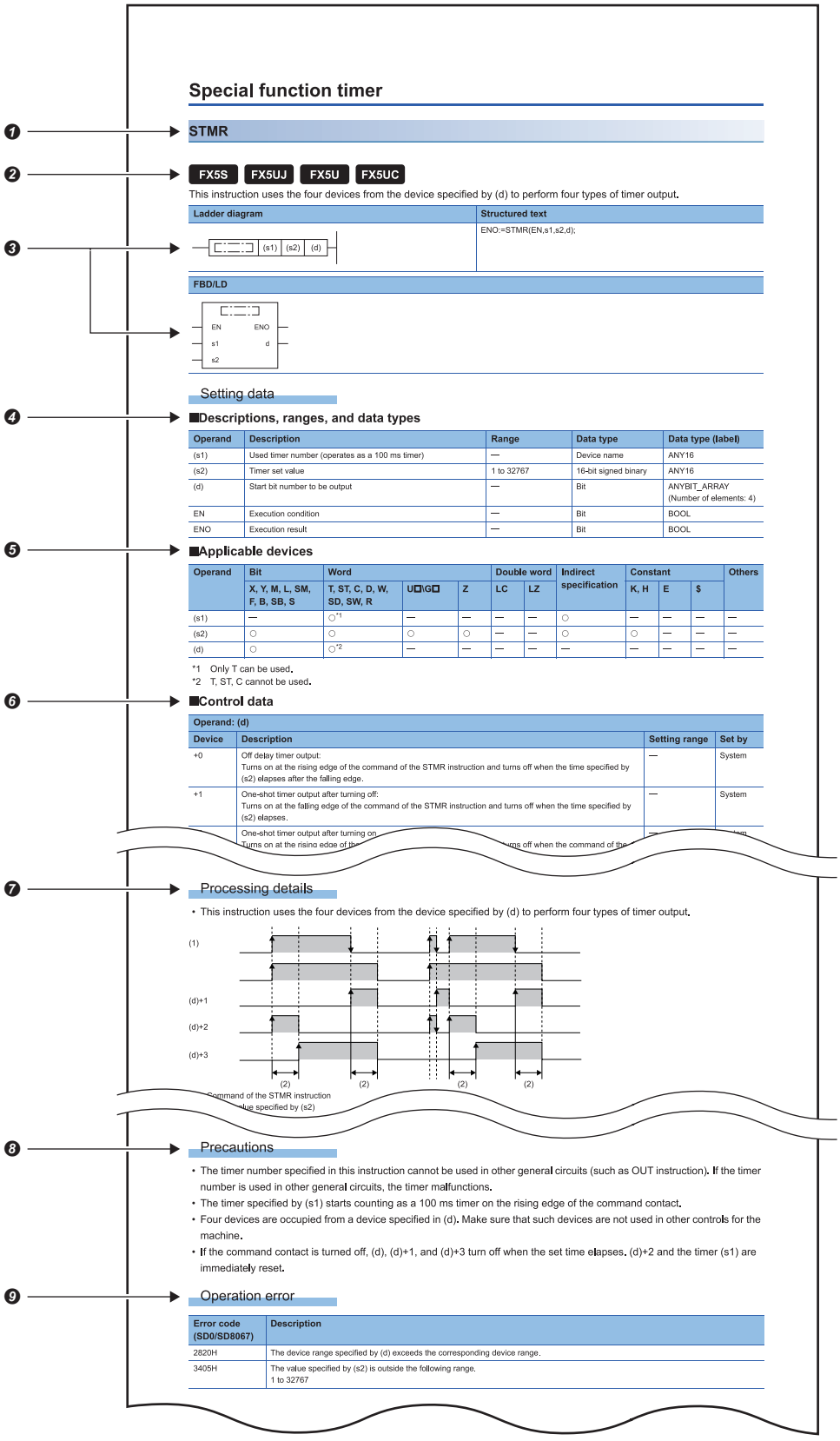
*1 Area-specific model

HOW TO READ THIS MANUAL

The following describes the page layout and symbols used in this manual.

How to read PART 3 and PART 4

The contents described in this section are provided only for explaining how to read this manual. Thus, the actual description may differ.



Special function timer

❶ **STMR**

❷ FX5S FX5UJ FX5U FX5UC

This instruction uses the four devices from the device specified by (d) to perform four types of timer output.

Ladder diagram	Structured text
(s1) (s2) (d)	ENO:=STMR(EN,s1,s2,d);

❸ FBD/LD

EN ENO
s1 d
s2

Setting data

❹ **■Descriptions, ranges, and data types**

Operand	Description	Range	Data type	Data type (label)
(s1)	Used timer number (operates as a 100 ms timer)	—	Device name	ANY16
(s2)	Timer set value	1 to 32767	16-bit signed binary	ANY16
(d)	Start bit number to be output	—	Bit	ANYBIT_ARRAY (Number of elements: 4)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

❺ **■Applicable devices**

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○*2	—	—	—	—	—	—	—	—	—

*1 Only T can be used.
*2 T, ST, C cannot be used.

❻ **■Control data**

Operand: (d)			
Device	Description	Setting range	Set by
+0	Off delay timer output: Turns on at the rising edge of the command of the STMR instruction and turns off when the time specified by (s2) elapses after the falling edge.	—	System
+1	One-shot timer output after turning off: Turns on at the falling edge of the command of the STMR instruction and turns off when the time specified by (s2) elapses.	—	System
	One-shot timer output after turning on Turns on at the rising edge of th... ...ns off when the command of the...	—	...tem

❼ Processing details

• This instruction uses the four devices from the device specified by (d) to perform four types of timer output.

❽ Precautions

• The timer number specified in this instruction cannot be used in other general circuits (such as OUT instruction). If the timer number is used in other general circuits, the timer malfunctions.
• The timer specified by (s1) starts counting as a 100 ms timer on the rising edge of the command contact.
• Four devices are occupied from a device specified in (d). Make sure that such devices are not used in other controls for the machine.
• If the command contact is turned off, (d), (d)+1, and (d)+3 turn off when the set time elapses. (d)+2 and the timer (s1) are immediately reset.

❾ Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
3405H	The value specified by (s2) is outside the following range. 1 to 32767

❶Indicates the instruction symbol.

- The instruction symbol with brackets means multiple instructions. For example, "GRY(P)(_U)" means the GRY, GRYP, GRY_U, and GRYP_U instructions.

Instruction symbol	Description of symbol
Instruction symbol with "(P)"	The instruction is executed on the rising edge.
Instruction symbol with "(_U)"	The instruction handles 16-bit or 32-bit unsigned binary data.

- The instruction symbol with "□" means multiple instructions. For example, "LDDT□" means the LDDT=, LDDT<>, LDDT>, LDDT<=, LDDT<, and LDDT>= instructions.

❷Indicates the availability of instructions for each CPU module. (The instruction cannot be used with CPU modules marked with an ×.)

❸Indicates the description format of the ladder diagram, FBD/LD language and ST (structured text) language

Instruction symbols are input in each corresponding place surrounded in a square in the ladder diagram.

❹Indicates the description, setting range, data type, and data type (label) of each operand.

- For the data type, refer to the following.

MELSEC iQ-F FX5 Programming Manual (Program Design)

❺Indicates the applicable devices for each operand. The following table describes the usage classification.

Operand	Bit	Word			Double word		Indirect specification	Constant			Others*5
	X*2, Y*2, M*2, L*2, SM*2, F*2, B*2, SB*2, S*2	T*3, ST*3, C*3, D*4,W*4, SD*4, SW*4, R*4	U□\G□*4	Z	LC*3	LZ		K, H	E	$	
Applicable devices*1	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ	@□ @□.□	K, H	E	$	P, I, U, DX, DY, N, BL, BL□\S□

*1 For the description of each device, refer to the following.
MELSEC iQ-F FX5 User's Manual (Application)

*2 "○" is described in positions where bit devices or digit specification of bit devices is available.

*3 When T, ST, C, and LC are used with an instruction other than the following instructions, they can be used only as word data. They cannot be used as bit data.
[Instruction which can be used as bit data]
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)

*4 "○" is described in positions where word device or bit specification of word device is available.

*5 Devices which can be set are described in the "Others" column.

❻Depending on the instruction, the control data to set the operation of the instruction exists. When the "Set by" column is "User", the value must be specified according to the setting range.

❼Indicates the function details of the instruction. When no details are described, the following programs correspond to "Interrupt program".

- Interrupt program using the interrupt pointer (I)
- Fixed scan execution type program
- Event execution type program which is triggered by an interrupt by the interrupt pointer (I)

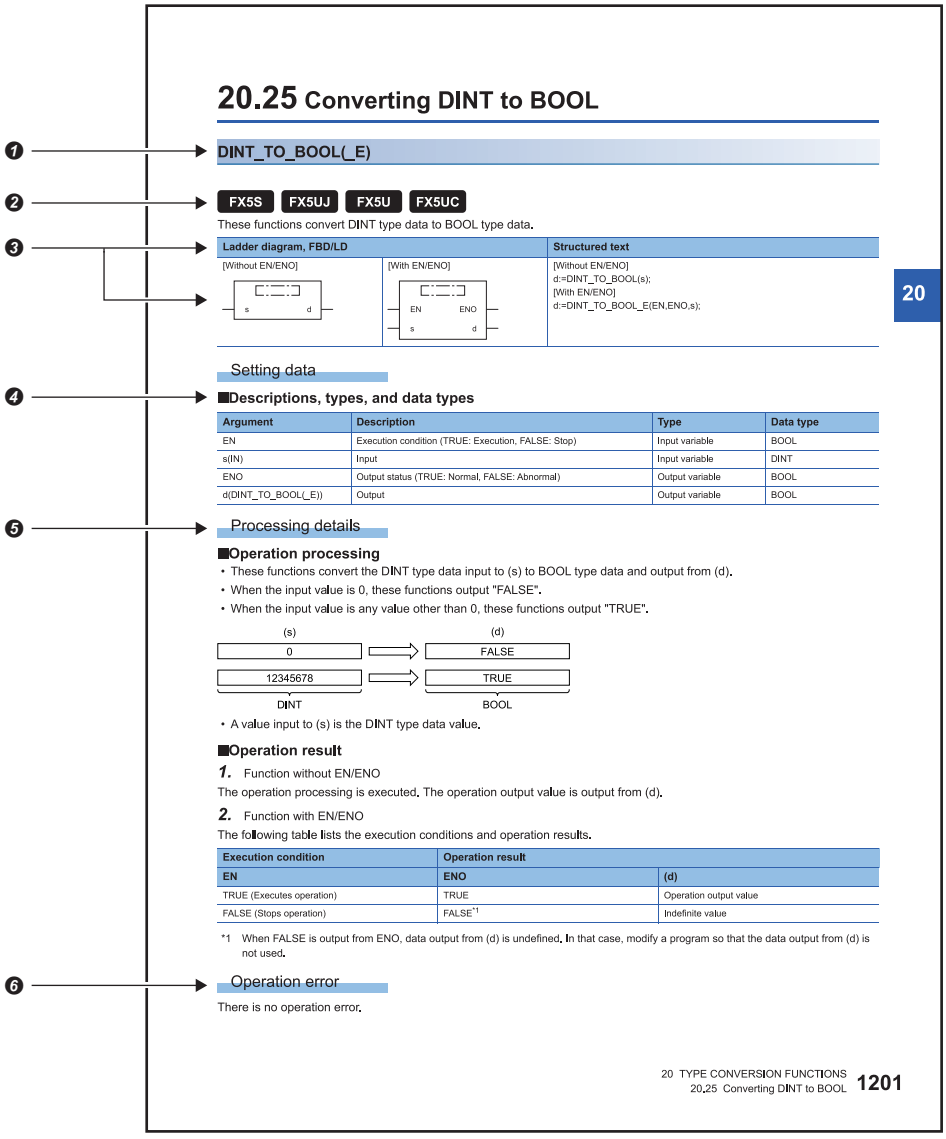
❽Indicates the cautions.

❾Indicates an error code (hexadecimal) which occurs at the execution and the error description when the instruction has a specific operation error.

- A device in which an error code is stored is described in the error code column. When an error code is stored in SD0/SD8067, the error flag (SM0, SM1, SM56, SM8067) turns on.

How to read PART 5 and PART 6

The contents described in this section are provided only for explaining how to read this manual. Thus, the actual description may differ.



20.25 Converting DINT to BOOL

❶ DINT_TO_BOOL(_E)

❷ FX5S FX5UJ FX5U FX5UC

These functions convert DINT type data to BOOL type data.

❸

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]	[With EN/ENO]	[Without EN/ENO]
		d:=DINT_TO_BOOL(s);
		[With EN/ENO]
		d:=DINT_TO_BOOL_E(EN,ENO,s);

20

Setting data

❹ ■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DINT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DINT_TO_BOOL(_E))	Output	Output variable	BOOL

❺ Processing details

■Operation processing

- These functions convert the DINT type data input to (s) to BOOL type data and output from (d).
- When the input value is 0, these functions output "FALSE".
- When the input value is any value other than 0, these functions output "TRUE".

- A value input to (s) is the DINT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

❻ Operation error

There is no operation error.

❶Indicates function symbols.

When character strings in brackets are added to the end of the function symbol for standard functions and function blocks, the function symbol indicates multiple functions. For example, "DINT_TO_INT(_E)" means "DINT_TO_INT" and "DINT_TO_INT_E".

Function symbol	Description of symbol
Function symbol to which "(_E)" is added.	Indicates that the description format with EN/ENO can be used in the standard function and function block.

❷Indicates the availability of standard functions or function blocks for each CPU module. (The standard function or function block cannot be used with CPU modules marked with an ×.)

❸Indicates the description format of the ladder diagram, FBD/LD language and ST (structured text) language.

In the square , either of the following symbol should be described.

- Standard function: Function symbol
- Standard function block: Instance name and function block symbol

The sign of return value of the standard function of FBD/LD is not displayed.

❹Indicates the description, type and data type of each argument.

❺Indicates the functions of each standard function or function block.

❻Indicates an error code which occurs at the execution and the error description when the standard function or the function block has a specific operation error.
A device in which an error code is stored is described in the error code column. When an error code is stored in SD0, the error flag SM0 turns on.

PART 1 OVERVIEW

Part 1 consists of the following chapter.

1 OVERVIEW

2 PRECAUTIONS ON PROGRAMMING

1 OVERVIEW

1.1 Instruction Configuration

Many instructions available for CPU module are each divided into the instruction part and device part.
The instruction part and device part are used as follows.

- Instruction part: Indicates the function of the relevant instruction.
- Device part: Indicates the data used for the instruction.

The device part is further classified to source data, destination data, and numerical data.

Source (s)

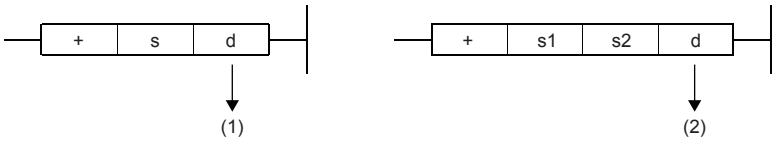
Source is the data used in the operation.
Depending on the label or device specified in each instruction, the source becomes as follows.

Type	Description
Constant	The constant specifies a numerical value used in the operation. It is set during program creation and cannot be changed during program execution.
Bit device Word device	The user specifies the device where the data to be used in the operation is stored. Necessary data must be thus stored in the specified device before operation execution. By changing the data to be stored in the specified device during program execution, the data to be used by the instruction can be changed.

Destination (d)

Data after operation is stored in the destination area.
However, some instructions require the data to be used in the operation to be stored before the operation.

Ex.
Binary 16-bit data addition instruction



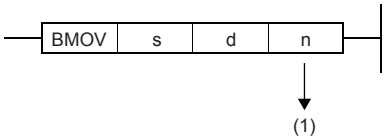
(1): The data required for operation is stored before the operation.
(2): Only the operation result is stored.

A label or device to store data must be set for the destination.

Numerical values (n)

In an instruction which uses multiple devices or an instruction which specifies the number of repetitions, data to be processed, and character strings, use numerical values to specify the number of devices, transfers, data, and character strings.

Ex.
Block transfer instruction



(1): The number of transfers executed by the BMOV instruction is specified.

A numerical value from 0 to 65535 or 0 to 4294967295 can be set for the size such as the number of devices, transfers, or characters.*1
Note, however, that when the size specification such as the number of devices, transfers, or characters is 0, the relevant instruction results in non-processing.

*1 The setting range varies depending on the instruction. For details, refer to the description of each instruction.

Point

Be careful when a large numerical value is used such as for the number of transfers. It delays the scan time.

1.2 Data Specification Method

The following table lists the types of data that can be used for instructions in CPU modules.

Data	Classification
Bit data	Bit data
16-bit data (word data)	16-bit signed binary data
	16-bit unsigned binary data
32-bit data (double-word data)	32-bit signed binary data
	32-bit unsigned binary data
Real number data (floating-point data)	Single-precision real number data
BCD data	BCD 4-digit data
	BCD 8-digit data
String data	String
	Unicode character string

Device data

Data type	Description	Specifiable device/constant*1
Bit	Bit data can be handled. ☞ Page 26 Bit data	• Bit device • Bit specification of word device
Word	Word data can be handled. ☞ Page 27 16-bit data (word data)	• Word device • Digit specification of bit devices (K1 to K4)*2 • Decimal constant • Hexadecimal constant
16-bit signed binary	16-bit data can be handled. The value range varies depending on whether the value is signed or unsigned. ☞ Page 27 16-bit data (word data)	
16-bit unsigned binary		
Double word	Double-word data can be handled. ☞ Page 29 32-bit data (double word data)	• Word device • Double-word device • Digit specification of bit devices (K1 to K8)*2 • Decimal constant • Hexadecimal constant
32-bit signed binary	Two consecutive sets of 32-bit data or 16-bit data can be handled. The value range varies depending on whether the value is signed or unsigned. ☞ Page 29 32-bit data (double word data)	
32-bit unsigned binary		
BCD 4-digit	BCD 4-digit data can be handled. 16-bit data is divided by 4 digits and each digit is specified in 0 to 9.	• Word device • Digit specification of bit devices (K1 to K4)*2 • Decimal constant • Hexadecimal constant
BCD 8-digit	BCD 8-digit data can be handled. 32-bit data is divided by 8 digits and each digit is specified in 0 to 9.	• Word device • Double-word device • Digit specification of bit devices (K1 to K8)*2 • Decimal constant • Hexadecimal constant
Single-precision real number	Single-precision real number data (single-precision floating-point data) can be handled. ☞ Page 32 Configuration of single-precision real number data	• Word device • Double-word device • Real constant
Character string	ASCII code and Shift JIS code character string data can be handled. ☞ Page 33 Character string data	• Word device • Character string constant
Unicode character string	Unicode character string data can be handled. ☞ Page 33 Character string data	• Word device • Character string constant
Device name	A device can be directly specified.	Name of the relevant applicable device

*1 A constant can be used in the data specified for the source (s) or numerical data (n) by an instruction.

*2 For the specification method, refer to the detail page of each data type.

Label data

■Primitive data type

Data type (label)	Specifiable label
Bit (BOOL)	• Bit type label • Bit-specified word [unsigned]/bit string [16 bits] type label • Bit-specified word [signed] type label • Timer/retentive timer type label contact/coil • Counter/ long counter type label contact/coil
Word [unsigned]/bit string [16 bits] (WORD)	• Word [unsigned]/bit string [16 bits] type label • Digit specified bit type label (K1 to K4) • Current value of timer/retentive timer type label • Current value of counter type label
Double word [unsigned]/bit string [32 bits] (DWORD)	• Double word [unsigned]/bit string [32 bits] type label • Digit specified bit type label (K1 to K8) • Current value of long counter type label
Word [signed] (INT)	• Word [signed] type label • Digit specified bit type label (K1 to K4) • Current value of timer/retentive timer type label • Current value of counter type label
Double word [signed] (DINT)	• Double word [signed] type label • Digit specified bit type label (K1 to K8) • Current value of long counter type label
Single-precision real number (REAL)	• Single-precision real data type label • Digit specified bit type label (K1 to K8)
Time (TIME)	• Time type label
Character string (STRING)	• Character string type label • Digit specified bit type label (K1 to K4)
Character string [Unicode] (WSTRING)	• Character string [Unicode] type label
Timer (TIMER)	• Timer type label
Retentive timer (RETENTIVETIMER)	• Retentive timer type label
Counter (COUNTER)	• Counter type label
Long counter (LCOUNTER)	• Long counter type label
Pointer (POINTER)	• Pointer type label

■Generic data type

Data type (label)	Specifiable label
ANY[*1]	Bit, word [signed], double word [signed], word [unsigned]/bit string [16 bits], double word [unsigned]/bit string[32 bits], single-precision real number, hour, character string, structure
ANY_BITADDR[*1]	Bit
ANY_BOOL	Bit
ANY_ELEMENTARY	Bit, word [signed], double word [signed], word [unsigned]/bit string [16 bits], double word [unsigned]/bit string[32 bits], single-precision real number, hour, character string
ANY_WORDADDR	Word [signed], double word [signed], word [unsigned]/bit string [16 bits], double word [unsigned]/bit string[32 bits], single-precision real number, hour, character string
Any 16-bit data (ANY16)	Word [signed], word [unsigned]/bit string [16 bits]
ANY16_S	Word [signed]
ANY16_U	Word [unsigned]/bit string [16 bits]
Any 32-bit data (ANY32)	Double word [signed], double word [unsigned]/bit string [32 bits], hour
ANY32_S	Double word [signed], hour
ANY32_U	Double word [unsigned]/bit string [32 bits]
ANY_REAL	Single-precision real number
ANYREAL_32	Single-precision real number
ANY_STRING	Character string
ANYSTRING_SINGLE	Character string
ANYSTRING_DOUBLE	Character string [Unicode]
ANY_STRUCT[*1]	Structures
ANY_DT	Word [signed], word [unsigned]/bit string [16 bits]
ANY_TM	Word [signed], word [unsigned]/bit string [16 bits]
STRUCT	Structures
ANY16_OR_STRING_SINGLE	Word [signed], word [unsigned]/bit string [16 bits], character string

*1 Can also be used as an array.

■Generic data type (array)

For the following generic data type, define the number of array elements.

Data type (label)	Specifiable label
ANYBIT_ARRAY	Bit
ANYWORD_ARRAY	Word [signed], double word [signed], word [unsigned]/bit string [16 bits], double word [unsigned]/bit string[32 bits], single-precision real number, hour, character string
ANY16_ARRAY	Word [signed], word [unsigned]/bit string [16 bits]
ANY16_S_ARRAY	Word [signed]
ANY16_U_ARRAY	Word [unsigned]/bit string [16 bits]
ANY32_ARRAY	Double word [signed], double word [unsigned]/bit string [32 bits]
ANY32_S_ARRAY	Double word [signed]
ANY32_U_ARRAY	Double word [unsigned]/bit string [32 bits]
ANY_REAL_ARRAY	Single-precision real number
ANY_REAL_32_ARRAY	Single-precision real number
ANY_STRING_ARRAY	Character string
ANY_STRING_SINGLE_ARRAY	Character string
ANY_STRING_DOUBLE_ARRAY	Character string [Unicode]
STRUCT_ARRAY	Structures

Bit data

Data size and data range

Bit data is handled in increments of bits such as contacts and coils.

Data name	Data size	Value range
Bit data	1 bit	0, 1

Handling bit data with bit devices and labels

Bit data of one point per point can be handled.

Handling bit data with bit word devices

By specifying a bit number for a word device, bit data of the specified bit number can be handled.

A bit in a word device can be specified by "Word device number.Bit number".

A bit number can be specified in hexadecimal in the range from 0 to F.

For example, bit 5 (b5) of D0 is specified as D0.5, and bit 10 (b10) of D0 is specified as D0.A.

The following word devices support bit specification.

Item	Device
Word devices which support bit specification	• Data register (D) • Link register (W) • Link special register (SW) • Special register (SD) • Module access device (U□\G) • File register (R)

Handling bit data with word type labels

By specifying a bit number for a word [unsigned]/bit string [16 bits] type label or word [signed] type label, bit data of the specified bit number can be handled.

A bit in a word type label can be specified by "Label name.Bit number".

16-bit data (word data)

Data size and data range

16-bit data includes signed and unsigned 16-bit data.

In signed 16-bit data, a negative number is represented in two's complement.

Data name	Data size	Value range	
		Decimal notation	Hexadecimal notation
Signed 16-bit data	16 bits (1 word)	-32768 to 32767	0000H to FFFFH
Unsigned 16-bit data		0 to 65535	

Handling 16-bit data with bit devices

A bit device can be handled as 16-bit data by performing digit specification.

Item	Notation	Example
Bit device	K□Bit device start number □: Number of digits (Specify the number within the range of 1 to 4.)	K4X10 K2M113

Handling 16-bit data with bit type array labels

A bit type array label can be handled as 16-bit data by performing digit specification.

The following table shows the notation for handling a bit type array label as 16-bit data by digit specification.

Item	Notation	Example
Bit type array label	K□Label name □: Number of digits (Specify the number within the range of 1 to 4.) Specify a label name without an array element.	K1L_BOOL

Digit specification range

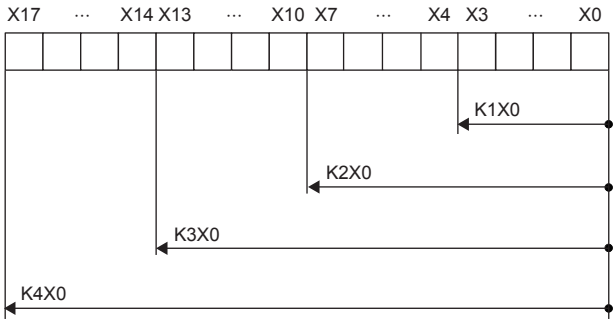
The following table lists the range of 16-bit data for each digit specification.

Digit specification	Decimal notation	Hexadecimal notation
K1	0 to 15	0H to FH
K2	0 to 255	00H to FFH
K3	0 to 4095	000H to FFFH
K4	Signed 16-bit data: -32768 to 32767 Unsigned 16-bit data: 0 to 65535	0000H to FFFFH

Ex.

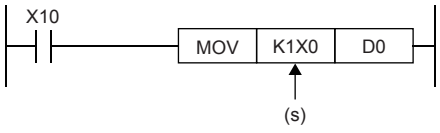
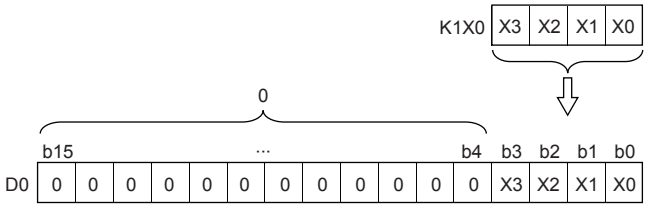
When digit specification is made for X0, the applicable number of points is as follows.

- K1X0→4 points from X0 to X3
- K2X0→8 points from X0 to X7
- K3X0→12 points from X0 to X13
- K4X0→16 points from X0 to X17



■Specifying a bit device with digit specification in the source (s)

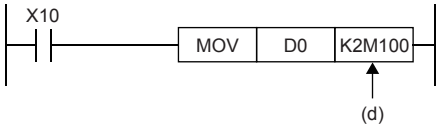
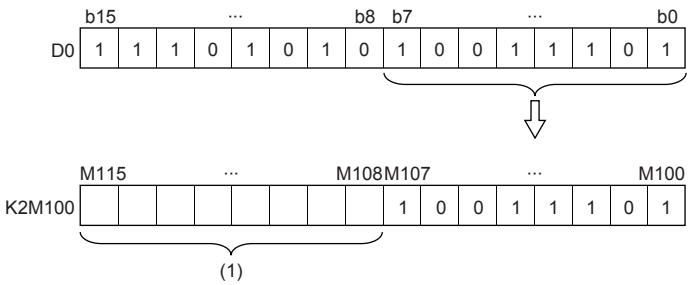
When a bit device with digit specification is specified in the source of an instruction, 0 is stored in the bits, which follow the bit for which digit specification is made in the source, in the word device of the destination.

Ladder example	Processing
• 16-bit data instruction 	

■Specifying a bit device with digit specification in the destination (d)

When a digit specification is made in the destination of an instruction, the number of points by the digit specification is applicable in the destination.

The bit devices after the number of points specified by nibble remain unchanged.

Ladder example	Processing
• When the source data is a word device 	 (1): The data remain the same.

Handling 16-bit data with word devices/labels

■Word device

One point of word device can handle 16-bit data.

■Word type label

One point of word type label can handle 16-bit data.

32-bit data (double word data)

Data size and data range

32-bit data includes signed and unsigned 32-bit data.

In signed 32-bit data, a negative number is represented in two's complement.

Data name	Data size	Value range	
		Decimal notation	Hexadecimal notation
Signed 32-bit data	32 bits (2 word)	-2147483648 to 2147483647	00000000H to FFFFFFFFH
Unsigned 32-bit data		0 to 4294967295	

Handling 32-bit data with bit devices

A bit device can be handled as 32-bit data by performing digit specification.

Item	Notation	Example
Bit device	K□Bit device start number □: Number of digits (Specify the number within the range from 1 to 8.)	K8X10 K6B018

Handling 32-bit data with bit type array labels

A bit type array label can be handled as 32-bit data by performing digit specification.

The following table shows the notation for handling a bit type array label as 32-bit data by digit specification.

Item	Notation	Example
Bit type array label	K□Label name □: Number of digits (Specify a number within the range of 1 to 8.) Specify a label name without an array element.	K8L_BOOL

Digit specification range

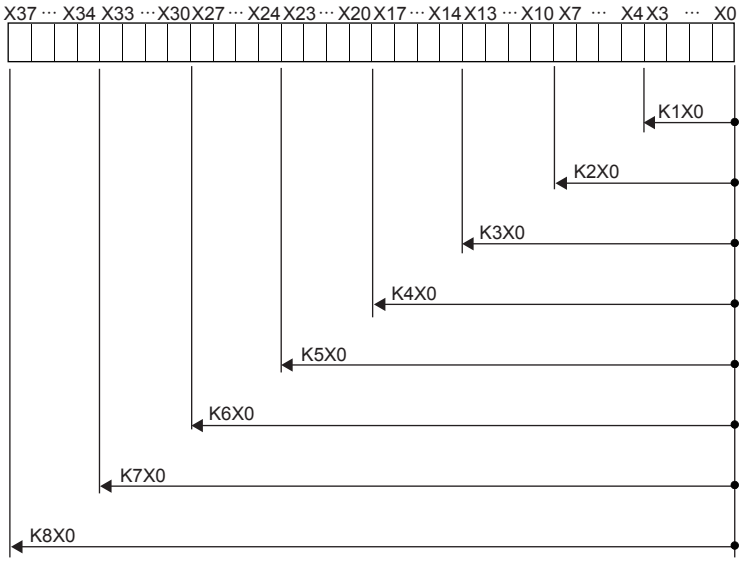
The following table lists the range of 32-bit data for each digit specification.

Digit specification	Decimal notation	Hexadecimal notation
K1	0 to 15	0H to FH
K2	0 to 255	00H to FFH
K3	0 to 4095	000H to FFFH
K4	0 to 65535	0000H to FFFFH
K5	0 to 1048575	00000H to FFFFFH
K6	0 to 16777215	000000H to FFFFFFH
K7	0 to 268435455	0000000H to FFFFFFFH
K8	Signed 32-bit data: -2147483648 to 2147483647 Unsigned 32-bit data: 0 to 4294967295	00000000H to FFFFFFFFH

Ex.

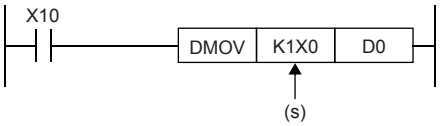
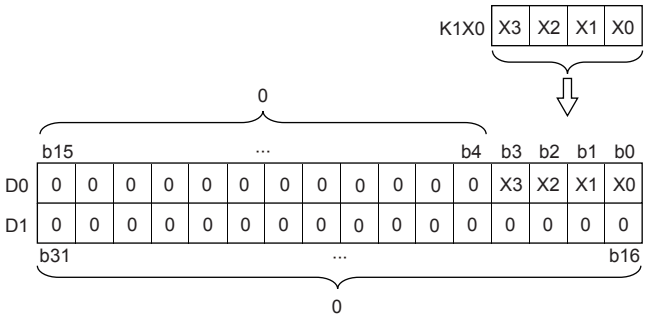
When digit specification is made for X0, the applicable number of points is as follows.

- K1X0→4 points from X0 to X3
- K2X0→8 points from X0 to X7
- K3X0→12 points from X0 to X13
- K4X0→16 points from X0 to X17
- K5X0→20 points from X0 to X23
- K6X0→24 points from X0 to X27
- K7X0→28 points from X0 to X33
- K8X0→32 points from X0 to X37



■Specifying a bit device with digit specification in the source (s)

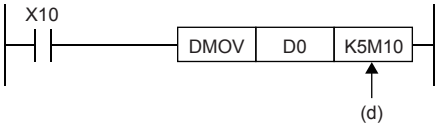
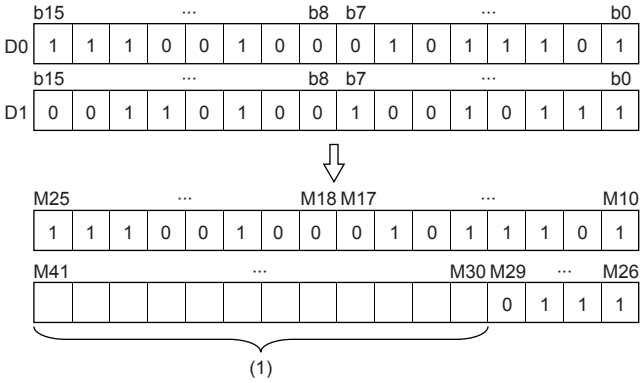
When a bit device with digit specification is specified in the source of an instruction, 0 is stored in the bits, which follow the bit for which digit specification is made in the source, in the word device of the destination.

Ladder example	Processing
• 32-bit data instruction 	

■Specifying a bit device with digit specification in the destination (d)

When a digit specification is made in the destination of an instruction, the number of points by the digit specification is applicable in the destination.

The bit devices after the number of points specified by nibble remain unchanged.

Ladder example	Processing
• When the source data is a word device 	 (1): The data remain the same.

Handling 32-bit data with word devices/labels

■Word device

Two points of word device can handle 32-bit data.

Note, however, that one point of the following devices can handle 32-bit data.

- Long counter (LC)
- Long index register (LZ)

■Double word type label

One point of double word device can handle 32-bit data.

Real number data (floating-point data)

Data size and data range

Real number data includes single-precision 32-bit real number data.

Real number data can be stored only in devices other than bit devices or in single-precision real data type labels.

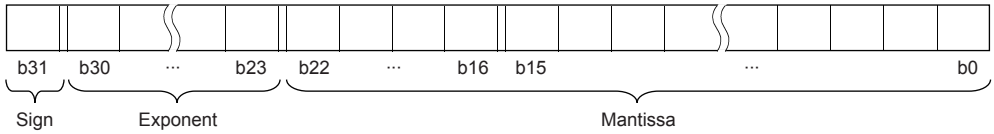
Data name		Data size	Value range
Single-precision real number data (single-precision floating-point data)	Positive number	32 bits (2 word)	$2^{-126} \leq$ real number $< 2^{128}$
	Zero		0
	Negative number		$-2^{128} <$ real number $\leq -2^{-126}$

Configuration of single-precision real number data

Single-precision real number data consists of a sign, mantissa, and exponent, and is expressed as shown below.

Sign 1. Mantissa × 2 Exponent

The following figure shows the bit configuration of the internal expression of single-precision real number data and the meaning of each part.



■Sign (1 bit)

This bit represents the positive or negative sign of a numerical value. "0" indicates a positive number or 0. "1" Indicates a negative number.

■Mantissa (23 bits)

A mantissa means XXXXX··· of 1.XXXXX···$\times 2^N$ representing a single-precision real number in binary.

■Exponent (8 bits)

An exponent means N of 1.XXXXX···$\times 2^N$ representing a single-precision real number in binary. The following table shows the relationships between the exponent value and N of a single-precision real number.

Exponent (b23 to b30)	FFH	FEH	FDH	···	81H	80H	7FH	7EH	···	02H	01H	00H
N	Not used	127	126	···	2	1	0	-1	···	-125	-126	Not used

Precautions

■When setting an input value of single-precision real number from the engineering tool

The number of significant digits is about 7 because the engineering tool processes single precision real number data in 32-bit single precision.

When the input value of single-precision real number data exceeds 7 digits, the 8th digit is rounded off.

Therefore, if the rounded-off value goes out of the range from -2147483648 to 2147483647, it will not be an intended value.

Ex.

When "2147483647" is set as an input value, it is handled as "2147484000" because 8th digit "6" is rounded off.

Ex.

When "E1.1754943562" is set as an input value, it is handled as "E1.175494" because 8th digit "3" is rounded off.

Point

The monitor function of the engineering tool can monitor real number data of CPU modules.
To represent "0" in real number data, set all numbers in each of the following range to 0.
• Single-precision real number data: b0 to b31
The setting range of real number data is as follows.
• Single precision real number data: -2^{128}<single precision real number data]≤-2^{-126}, 0, 2^{-126}≤[single precision real number data]<2^{128}
Do not specify "-0" (only the most significant bit is 1) in real number data. Performing a real number operation using -0 results in an operation error.

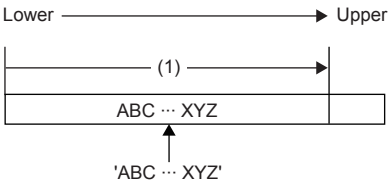
Character string data

Format of character string data

The following table lists the types of character string data, each of which ends with a NULL code to be handled as a character string.

Type	Character code	Last character
Character string	ASCII code	NULL(00H)
Unicode character string	Unicode (UTF-16 (little endian))	NULL(0000H)

Character string data is stored in devices or an array in ascending order of device numbers or array element numbers.



(1): Character code string

Method for expressing a character string

When a character string or a Unicode character string is specified in a program, an error may occur depending on the expressing method.
The expressing methods in the programming languages are shown below.

■Ladder program

Data type	Expressing method
String	Enclose character strings with single (') or double quotation marks (").
Character string [Unicode]	Enclose Unicode character strings with single (') or double quotation marks (").

■ST program

Data type	Expressing method
String	Enclose character strings with single quotation marks (').
Character string [Unicode]	Enclose Unicode character strings with double quotation marks (").

■FBD/LD program

Data type	Expressing method
String	Enclose character strings with single quotation marks (').
Character string [Unicode]	Enclose Unicode character strings with double quotation marks (").

Data range

The following table summarizes the ranges of character string data.

Type	Maximum number of character strings	Maximum number of character strings that can be handled in the program
Character string	255 single-byte characters (excluding the last NULL character)	16383 characters (excluding the last NULL character)
Unicode character string*1	255 characters (NULL at the end is not included.)	

*1 For Unicode strings, characters in the basic multilingual plane can be used.

Number of words required for storing data

Character string data can be stored in word devices.

The following table lists the numbers of words required for storing character string data.

Number of character string bytes	Number of words required for storing character strings	Number of words needed for storing Unicode character string
0 byte	1 [word]	1 [word]
Odd number of bytes	(Number of character string bytes+1) ÷ 2 [words]	— (Because one character is in an even byte.)
Even number of bytes	(Number of character string bytes÷2) +1 [words]	Number of characters + 1 [words]

Character string data storage location

An image of the character string data storage location is shown below.

■Character strings

In each character string storage image, "NULL" indicates a NULL code (00H).

Character string to be stored	Image of storing character string data from D0	Image of storing character string data from word type label array arrayA[0]
' ' (null character string)	D0 | NULL | NULL	arrayA[0] | NULL | NULL
'ABC'	D0 | B | A D1 | NULL | C	arrayA[0] | B | A arrayA[1] | NULL | C
'ABCD'	D0 | B | A D1 | D | C D2 | NULL | NULL	arrayA[0] | B | A arrayA[1] | D | C arrayA[2] | NULL | NULL

■Unicode character strings

In each Unicode character string storage image, "NULL" indicates the NULL code (0000H).

Character string to be stored	Image of storing character string data from D0	Image of storing character string data from word type label array arrayA[0]
" " (null character string)	D0 | NULL	arrayA[0] | NULL
"ABCD"	D0 | A D1 | B D2 | C D3 | D D4 | NULL	arrayA[0] | A arrayA[1] | B arrayA[2] | C arrayA[3] | D arrayA[4] | NULL

1.3 Execution Condition

Types of execution conditions

The following are the five types of execution conditions of the instructions and functions of CPU module.

Execution condition		Description
On		An instruction is executed during on. It is executed only while the precondition of the instruction is on. When the precondition is off, the instruction is not executed.
Rising edge		An instruction is executed one time when turned on. It is executed only once on the rising edge (off to on) of the precondition of the instruction and is no longer executed later even when the condition turns on.
Off		An instruction is executed during off. It is executed only while the precondition of the instruction is off. When the precondition is on, the instruction is not executed.
Falling edge		An instruction is executed one time when turned off. It is executed only once on the falling edge (on to off) of the precondition of the instruction and is no longer executed later even when the condition turns off.
Always	—	An instruction is always executed regardless of whether the precondition of the instruction is on or off. When the precondition is off, the instruction performs off processing.

Execution condition of each instruction

The execution condition varies depending on the instruction. The following table lists the execution conditions of individual instructions.

Execution condition	Applicable instruction
On	All instructions except for the following
Rising edge	• Instruction followed by symbol (P), (GP), and (SP) • PLS
Off	—
Falling edge	PLF
Always	LD, LDI, AND, ANI, OR, ORI, LDP, LDF, ANDP, ANDF, ORP, ORF, LDPI, LDFI, ANDPI, ANDFI, ORPI, ORFI, ANB, ORB, MPS, MRD, MPP, INV, MEP, MEF, OUT, OUT T, OUTH T, OUTHS T, OUT ST, OUTH ST, OUTHS ST, OUT C, OUT LC, MC, MCR, FEND, END, LD□, AND□, OR□, LD□_U, AND□_U, OR□_U, LDD□, ANDD□, ORD□, LDD□_U, ANDD□_U, ORD□_U, JMP, DI, EI, IMASK, SIMASK, IRET, FOR, NEXT, RET, LD$□, AND$□, OR$□, LDE□, ANDE□, ORE□, STMR, LDDT□, ANDDT□, ORDT□, LDTM□, ANDTM□, ORTM□

1.4 Acceleration of Instruction Processing Time

High-speed instruction

In some instructions, when the device and label specified in each operand satisfy the specific condition, the instruction processing time is accelerated.
Such accelerated instructions are called high-speed instructions.

The instructions capable of accelerating and the conditions of acceleration

The table below shows the instructions capable of accelerating and the conditions of acceleration (conditions of non-acceleration).

Classification	Instruction symbol	Conditions of non-acceleration
Contact instruction	LD, LDI, AND, ANI, OR, ORI, LDP, LDF, ANDP, ANDF, ORP, ORF, LDPI, LDFI, ANDPI, ANDFI, ORPI, ORFI	• When indexing is executed • When link register (W) is specified, the number of points of file register (R) + the number of link register (W) > 32767 • When link special register (SW) is specified, the number of points of file register (R) + the number of points of link register (W) + the number of link special register (SW) > 32767 • When module access device (Un\Gn) is specified • When direct access input (DX) is specified • When direct access output (DY) is specified • In OUT, SET, RST instructions, annunciator (F) or step relay (S) is specified in the operand • In OUT T instruction, routine timer is specified in the operand • In OUT T, OUT ST, OUTH T, OUTH ST, OUTHS T, OUTHS ST, OUT C instructions, a data other than constant is specified in the second operand • In OUT T, OUT ST instructions, K0 is specified in the second operand • In RST instruction, a data other than the bit type is specified in the operand
Association instruction	ANB, ORB, MPS, MRD, MPP, INV, MEP, MEF	
Output instruction	OUT, OUT T, OUTH T, OUTHS T, OUT ST, OUTH ST, OUTHS ST, OUT C, SET, RST, ALT	
Shift instruction	SFR, SFL	
Comparison operation instruction	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, LD>=, AND>=, OR>=, LD>=_U, AND>=_U, OR>=_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, LDD>=, ANDD>=, ORD>=, LDD>=_U, ANDD>=_U, ORD>=_U	• When indexing is executed • When link register (W) is specified, the number of points of file register (R) + the number of link register (W) > 32767 • When link special register (SW) is specified, the number of points of file register (R) + the number of points of link register (W) + the number of link special register (SW) > 32767 • When module access device (Un\Gn) is specified • When the indirect specification is used • When specifying the nibble of a device, the number of digits is other than K4 or K8 • When specifying the nibble of a device, the head device number is other than a multiple of 8 (Example) K4M8: Acceleration available, K4M9: Acceleration not available • When specifying the nibble of a label[*1]

Classification	Instruction symbol	Conditions of non-acceleration
Arithmetic operation instruction	+ (2 operands), +_U (2 operands), + (3 operands), +_U (3 operands), - (2 operands), -_U (2 operands), - (3 operands), -_U (3 operands), D+ (2 operands), D+_U (2 operands), D+ (3 operands), D+_U (3 operands), D- (2 operands), D-_U (2 operands), D- (3 operands), D-_U (3 operands), *, *_U, INC, INC_U, DEC, DEC_U, DINC, DINC_U, DDEC, DDEC_U	• When indexing is executed • When link register (W) is specified, the number of points of file register (R) + the number of link register (W) > 32767 • When link special register (SW) is specified, the number of points of file register (R) + the number of points of link register (W) + the number of link special register (SW) > 32767 • When module access device (Un\Gn) is specified • When the indirect specification is used • When specifying the nibble of a device, the number of digits is other than K4 or K8 • When specifying the nibble of a device, the head device number is other than a multiple of 8 (Example) K4M8: Acceleration available, K4M9: Acceleration not available • When specifying the nibble of a label*1 • When specifying timer (T), retentive timer (ST), counter (C), long counter (LC), timer type label, retentive timer type label, counter type label, and long counter type label in the operand with the data type of word/double word
Logical operation instruction	WAND (2 operands), WAND (3 operands), DAND (2 operands), DAND (3 operands), WOR (2 operands), WOR (3 operands), DOR (2 operands), DOR (3 operands), WXOR (2 operands), WXOR (3 operands), DXOR (2 operands), DXOR (3 operands), WXNR (2 operands), WXNR (3 operands), DXNR (2 operands), DXNR (3 operands)	
Bit processing instruction	BSET, BRST, TEST, DTEST	
Data conversion instruction	INT2UINT, INT2UDINT, INT2DINT, UINT2INT, UINT2DINT, UINT2UDINT, DINT2INT, DINT2UINT, DINT2UDINT, UDINT2INT, UDINT2DINT, UDINT2UINT	
Real number instruction	INT2FLT, DINT2FLT, UINT2FLT, UDINT2FLT, EMOV, DEMOV	

*1 When specifying the nibble of a label, some instructions are accelerated depending on the assignment position of the label. However, because the assignment position of label cannot be checked and changed, check the actual operation.

Execution time of high-speed instruction

For execution time of high-speed instruction, refer to the following.

☞ Page 1390 Instruction Processing Time

2 PRECAUTIONS ON PROGRAMMING

2.1 Errors Common to Instructions

The following table lists the conditions under which an error occurs when the instruction is executed.

Error content*1	Error code (SD0/SD8067)
An I/O number which corresponds to no module is specified.	2801H
• An I/O number which is out of range (0 to 1777 (Octal number)) is specified. • The device or label specified by the instruction exceeds the available range.	2820H
The range of the buffer memory of the module specified by the instruction is exceeded.	2823H

*1 For a contact instruction, an error is not detected but the operation result becomes no continuity.

2.2 Checking the Ranges of Instruction Runtime Devices and Labels

Checking the ranges of devices and labels

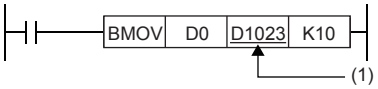
When a device or label is specified in an instruction, range check is performed. If a range exceeding that of the relevant device or label is specified, an error occurs.

The same applies when a label assigned to a device is specified in an instruction in the program.

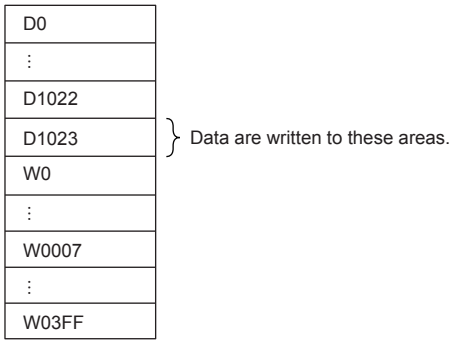
Create such a program that the operation result falls within the range of the relevant device or label.

Ex.

When a global device is specified



Device assignment image in the device/label memory



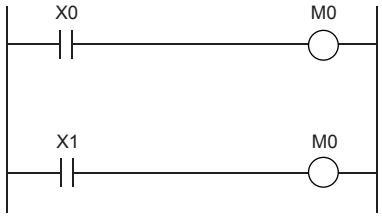
(1) The transfer destination is in the range corresponding to D1023 to D1032. Because D1024 to D1032 do not exist, the data are written only to D1023.

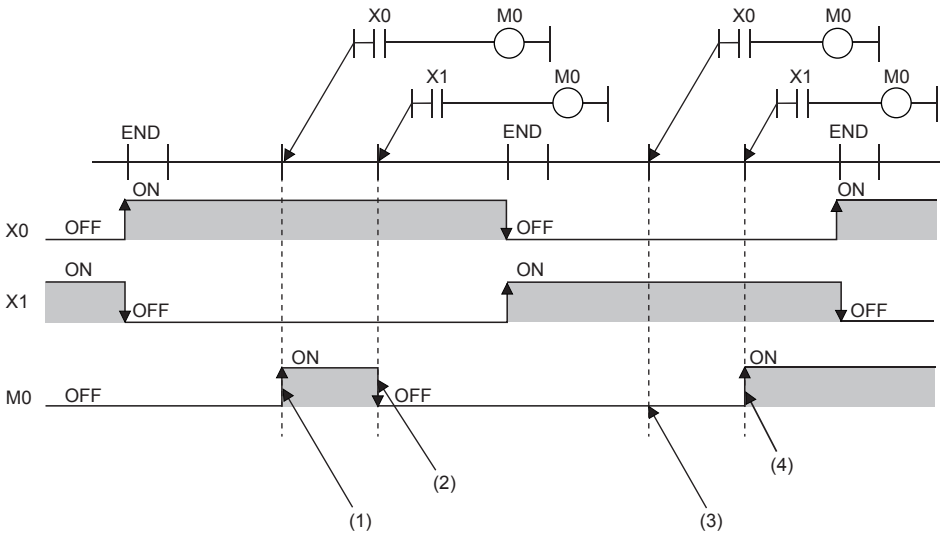
2.3 Operations Arising when the OUT, SET/RST, and PLS/PLF Instructions of the Same Device are Used

If two or more OUT, SET/RST, and PLS/PLF instructions are executed using the same device during one scan, they operate as described in this section.

For OUT instructions of the same device

More than one OUT instruction of the same device must not be issued during one scan.
Otherwise, the specified device turns on or off, depending on the operation result up to each OUT instruction while it is in execution.
In this case, the device may turn on/off during one scan because the on/off state of the specified device is determined during execution of each OUT instruction.
The following figure shows the behavior arising when a circuit turning on/off the same internal relay (M0) is created with input X0 and X1.





(1) Since X0 is on, M0 turns on.
(2) Since X1 is off, M0 turns off.
(3) Since X0 is off, M0 remains off.
(4) Since X1 is on, M0 turns on.

If output (Y) is specified using an OUT instruction, the on/off state of the last OUT instruction executed during the one scan will be output.

If SET/RST instructions of the same device are used

■For SET instructions

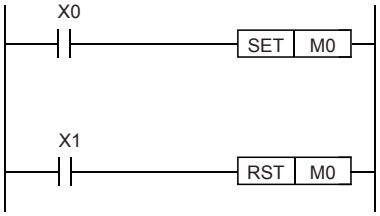
The SET instruction turns on the specified device if the execution command is on, and causes no operation if it is off.
Thus, if two or more SET instructions of the same device are executed during one scan, the specified device turns on even if one execution command is on.

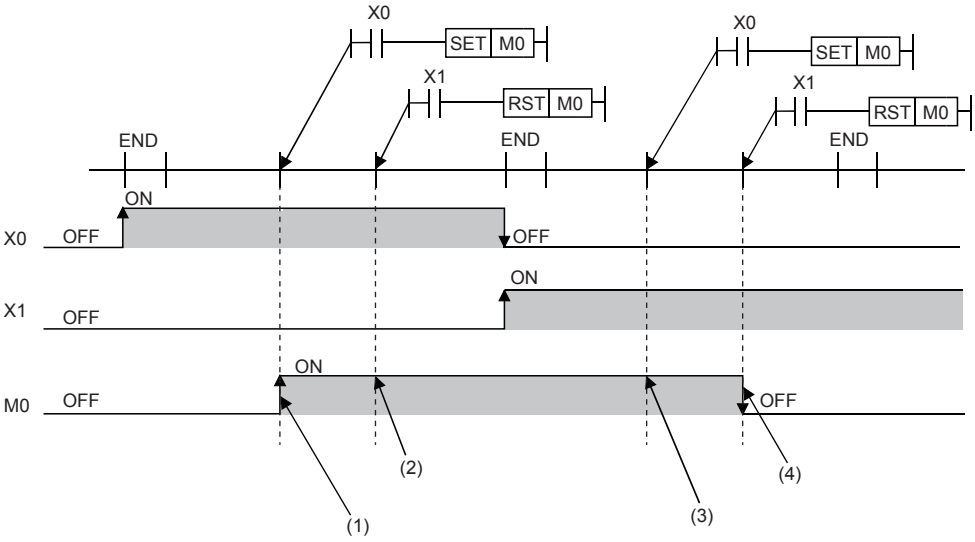
■For RST instructions

The RST instruction turns off the specified device if the execution command is on, and causes no operation if it is off.
Thus, if two or more RST instructions of the same device are executed during one scan, the specified device turns on even if one execution command is off.

■If the SET and RST instructions of the same device exist in one scan

If the SET and RST instructions of the same device exist in one scan, the SET instruction turns on the specified device if the execution command is on, and turns off the specified device if it is on.
If both the SET and RST instructions are off, the on/off state of the specified device will be unchanged.



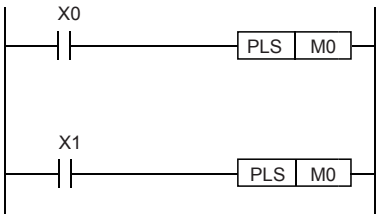


(1) Since X0 is on, M0 turns on.
(2) Since X1 is off, M0 remains on. (The RST instruction results in non-processing.)
(3) Since X0 is off, M0 remains on. (The SET instruction results in non-processing.)
(4) Since X1 is on, M0 turns off.

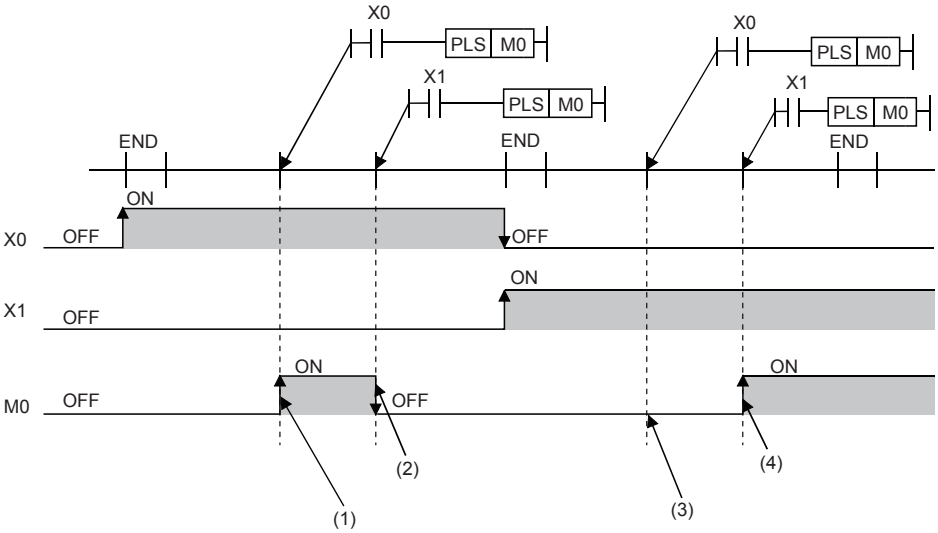
If output (Y) is specified using a SET/RST instruction, the on/off state of the last SET/RST instruction executed during the one scan will be output.

If PLS instructions of the same device are used

The PLS instruction turns on the specified device when the execution command specifies an off-to-on change. The specified device is turned off unless the execution command specifies an off-to-on change (i.e. off to off, on to on, on to off).
Thus, if two or more PLS instructions of the same device are issued during one scan, the specified device is turned on when the execution command of each PLS instruction specifies an off-to-on change. The specified device is turned off unless the execution command specifies an off-to-on change.
Thus, if two or more PLS instructions are issued during one scan, the device turned on by a PLS instruction may not turn on for one scan.

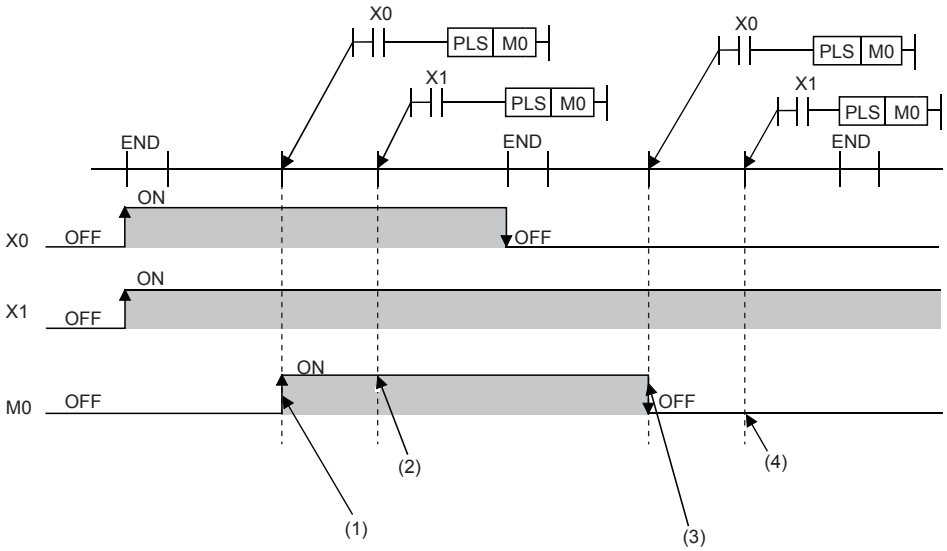


• If X0 and X1 differs in the on/off timing (i.e. the specified device does not turn on for one scan)



(1) Since X0 turns on, M0 turns on.
(2) Since X1 is other than turning on, M0 turns off.
(3) Since X0 is other than turning on, M0 remains off.
(4) Since X1 turns on, M0 turns on.

• If the off-to-on changes of X0 and X1 are at the same timing

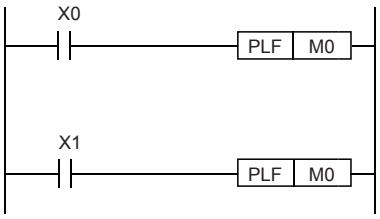


(1) Since X0 turns on, M0 turns on.
(2) Since X1 turns on, M0 remains on.
(3) Since X0 is other than turning on, M0 turns off.
(4) Since X1 is other than turning on, M0 remains off.

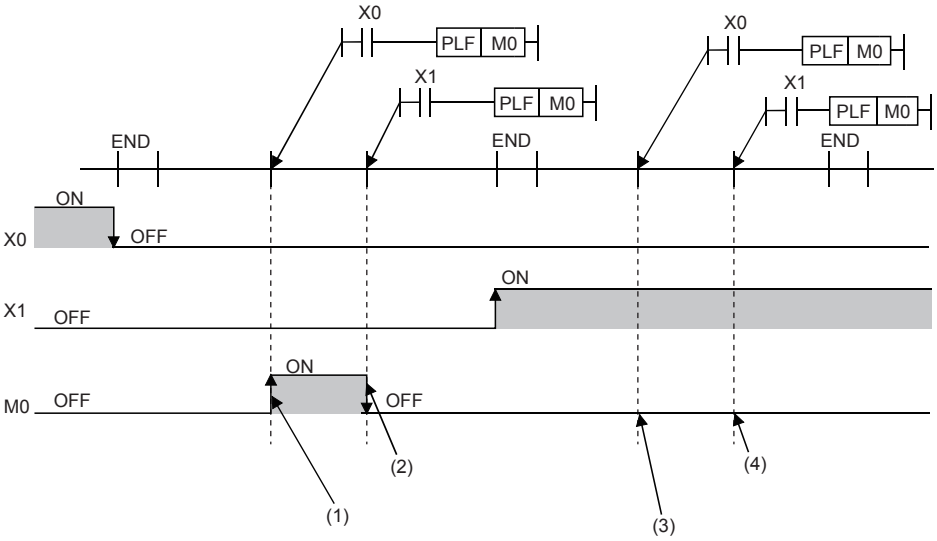
If output (Y) is specified using a PLS instruction, the on/off state of the last PLS instruction executed during the one scan will be output.

If PLF instructions of the same device are used

The PLF instruction turns on the specified device when the execution command specifies an on-to-off change. The specified device is turned off unless the execution command specifies an on-to-off change (i.e. off to off, off to on, on to on).
Thus, if two or more PLF instructions of the same device are issued during one scan, the specified device is turned on when the execution command of each PLF instruction specifies an on-to-off change. The specified device is turned off unless the execution command specifies an on-to-off change.
Thus, if two or more PLF instructions are issued during one scan, the device turned on by a PLF instruction may not turn on for one scan.

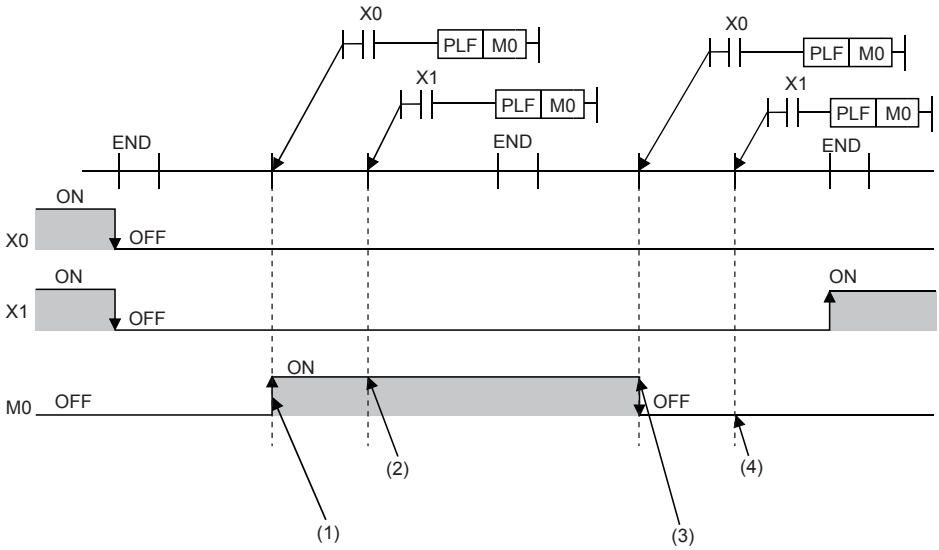


• If X0 and X1 differs in the on/off timing (i.e. the specified device does not turn on for one scan)



(1) Since X0 turns off, M0 turns on.
(2) Since X1 is other than turning off, M0 turns off.
(3) Since X0 is other than turning off, M0 remains off.
(4) Since X1 is other than turning off, M0 remains off.

• If the on-to-off changes of X0 and X1 are at the same timing



(1) Since X0 turns off, M0 turns on.
(2) Since X1 turns off, M0 remains on.
(3) Since X0 is other than turning off, M0 turns off.
(4) Since X1 is other than turning off, M0 remains off.

If output (Y) is specified using a PLF instruction, the on/off state of the last PLF instruction executed during the one scan will be output.

2.4 Handling general flags

In some types of instructions, the following flags operate:

- SM8020: Zero flag
- SM8021: Borrow flag
- SM8022: Carry flag
- SM8029: Instruction execution complete flag
- SM8090: Block comparison signal
- SM8304: Zero flag (MUL, DIV instructions only)
- SM8306: Carry flag (MUL, DIV instructions only)
- SM8328: Instruction non-execution flag
- SM8329: Instruction execution abnormal end flag

These general flags turn ON or OFF every time instructions turn ON, but do not change when various instructions turn OFF or when errors occur.

Because multiple instructions change the status of these flags, the ON/OFF status of flags change every time such instructions are executed.

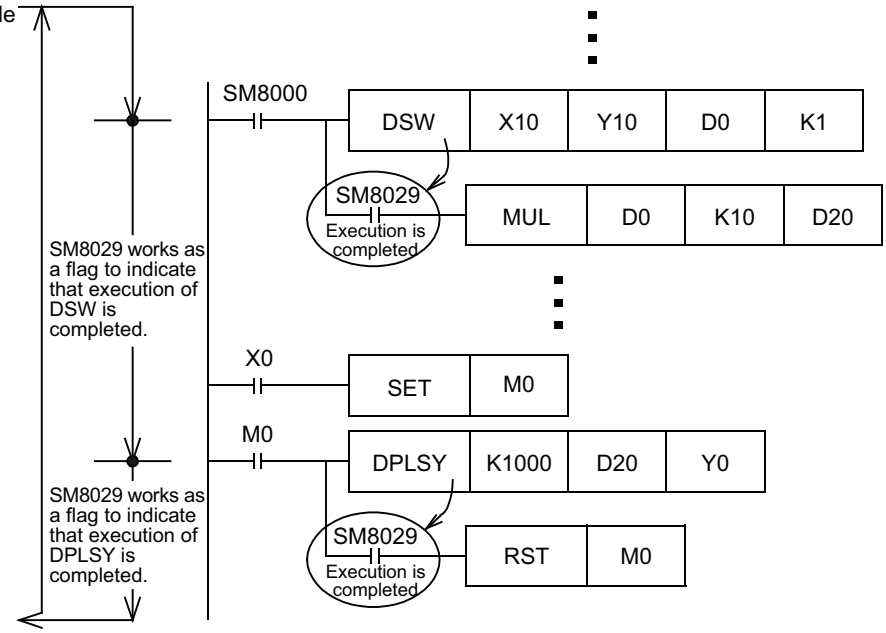
Program general flag contacts directly under each instruction. (☞ Page 46 Program containing many flags (example of SM8029 (Instruction execution complete flag)))

Program containing many flags (example of SM8029 (Instruction execution complete flag))

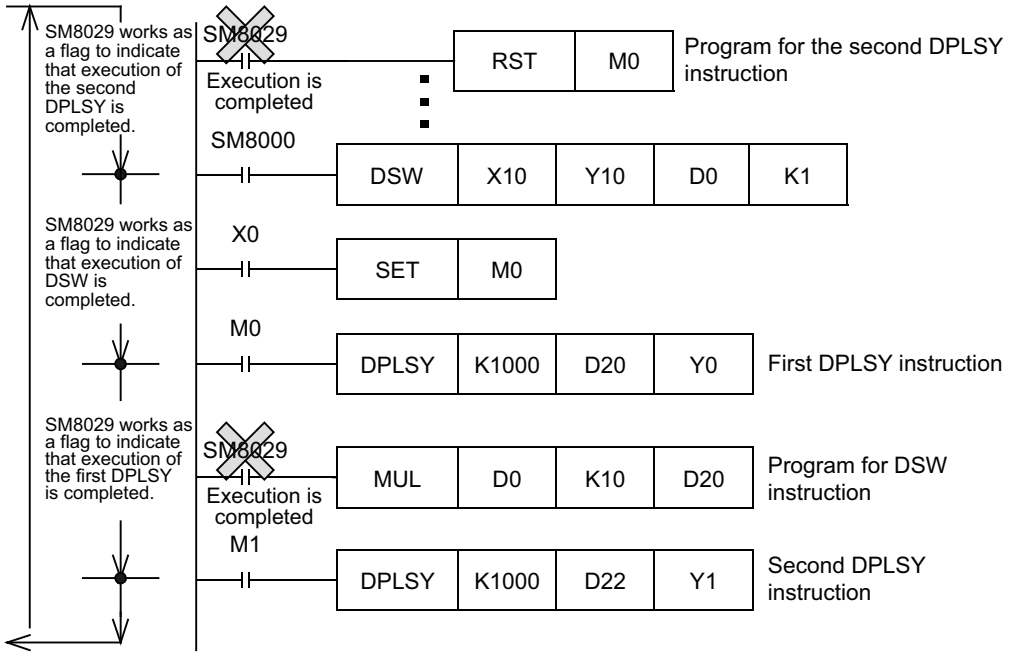
When two or more SM8029 (Instruction execution complete flag) are programmed together for instructions which operate the same flag, it is difficult to determine which instruction executes which flag. Also, the flag corresponding to each instruction cannot be read normally.

For using flags in any positions other than directly under instructions, refer to ☞ Page 47 Method for using flags in any positions other than directly under instructions (example of SM8029 (Instruction execution complete flag)).

Good example

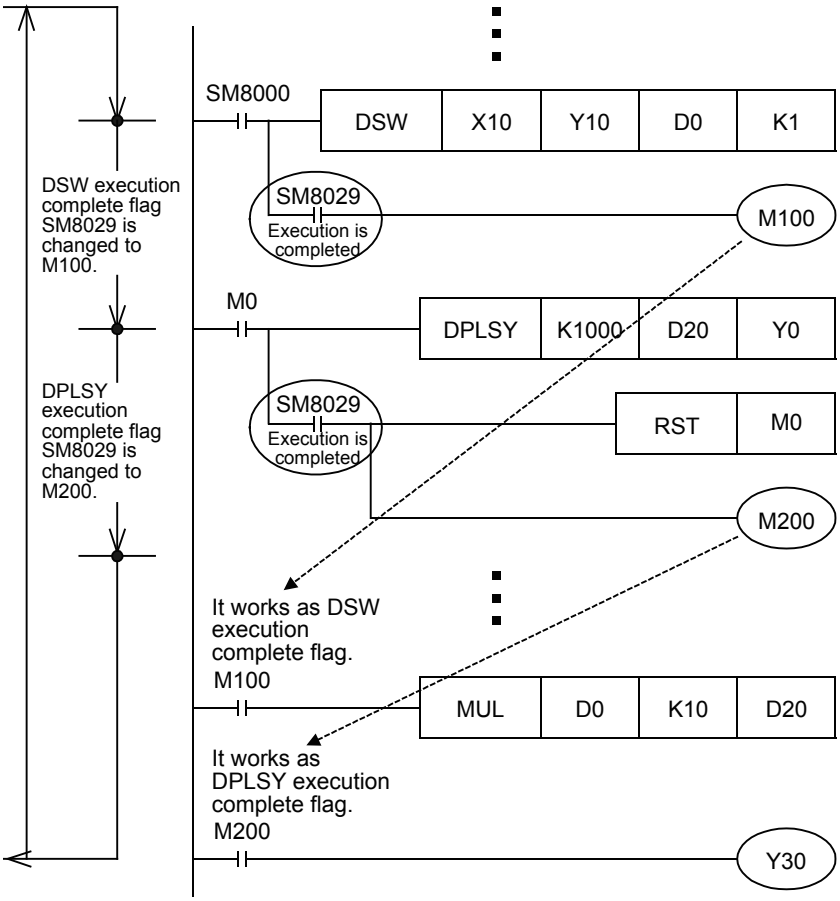


Bad example



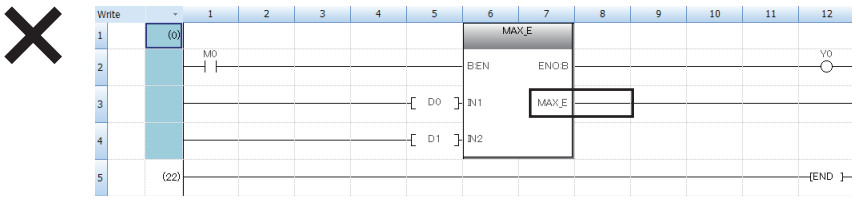
Method for using flags in any positions other than directly under instructions (example of SM8029 (Instruction execution complete flag))

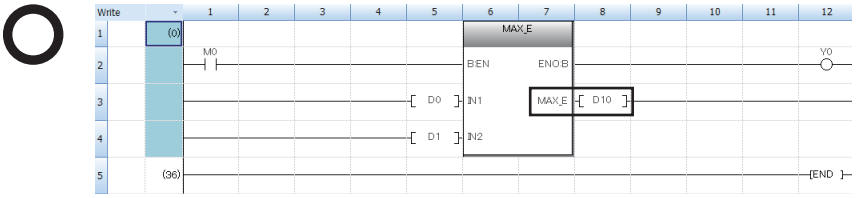
When two or more instructions are programmed, general flags turn ON or OFF when each instruction turns ON.
Accordingly, when using a flag in any position other than directly under an instruction, set another device to ON or OFF just under the instruction, and then use the contact of the device as the command input.



2.5 Standard Function/Function Block Return Values

When standard functions/function blocks are used, always set a device or label for the output variable, and make sure to receive the return value.
If the output variable is not set, an error may occur after writing to the PLC.





MEMO

PART 2 INSTRUCTION/ FUNCTION LIST

This part consists of the following chapters.

3 CPU MODULE INSTRUCTION

4 MODULE SPECIFIC INSTRUCTION

5 STANDARD FUNCTIONS/FUNCTION BLOCKS

3 CPU MODULE INSTRUCTION

3.1 Sequence Instruction

Contact instruction

■Operation start, series connection, parallel connection

Instruction symbol	Description	Reference
LD	Starts logical operation (Starts NO contact logical operation)	Page 114
LDI	Starts logical NOT operation (Starts NC contact logical operation)	
AND	Logical AND (NO contact series connection)	
ANI	Logical NAND (NC contact series connection)	
OR	Logical OR (NO contact parallel connection)	
ORI	Logical NOR (NC contact parallel connection)	

■Pulse operation start, pulse series connection, pulse parallel connection

Instruction symbol	Description	Reference
LDP	Starts rising edge pulse operation	Page 117
LDF	Starts falling edge pulse operation	
ANDP	Rising edge pulse series connection	
ANDF	Falling edge pulse series connection	
ORP	Rising edge pulse parallel connection	
ORF	Falling edge pulse parallel connection	

■Pulse NOT operation start, pulse NOT series connection, pulse NOT parallel connection

Instruction symbol	Description	Reference
LDPI	Starts rising edge pulse NOT operation	Page 121
LDFI	Starts falling edge pulse NOT operation	
ANDPI	Rising edge pulse NOT series connection	
ANDFI	Falling edge pulse NOT series connection	
ORPI	Rising edge pulse NOT parallel connection	
ORFI	Falling edge pulse NOT parallel connection	

Association instruction

■Ladder block series/parallel connection

Instruction symbol	Description	Reference
ANB	AND between logical blocks (series connection between logical blocks)	Page 123
ORB	OR between logical blocks (parallel connection between logical blocks)	

■Storing/reading/clearing the operation result

Instruction symbol	Description	Reference
MPS	Stores the operation result	Page 125
MRD	Reads the operation result stored by MPS	
MPP	Reads and resets of the operation result stored by MPS	

■Inverting the operation result

Instruction symbol	Description	Reference
INV	Inversion of the operation result	Page 128

■Converting the operation result into a pulse

Instruction symbol	Description	Reference
MEP	Conversion of operation result to rising edge pulse	Page 129
MEF	Conversion of operation result to falling edge pulse	

Output instruction

■Out (excluding the timer, counter and annunciator)

Instruction symbol	Description	Reference
OUT	Device output	Page 130

■Timer (low-speed, high-speed, low-speed retentive, high-speed retentive)

Instruction symbol	Description	Reference
OUT T	Low-speed timer	Page 132
OUTH T	Timer	
OUTHS T	High-speed timer	
OUT ST	Low-speed retentive timer	
OUTH ST	Retentive timer	
OUTHS ST	High-speed retentive timer	

■Counter, long counter

Instruction symbol	Description	Reference
OUT C	Counter	Page 135
OUT LC	Long counter	Page 137

■Annunciator

Instruction symbol	Description	Reference
OUT F	Annunciator	Page 139

■Setting devices (excluding annunciator)

Instruction symbol	Description	Reference
SET	Sets devices	Page 141

■Resetting devices (excluding annunciator)

Instruction symbol	Description	Reference
RST	Resets devices	Page 143

■Setting/resetting annunciator

Instruction symbol	Description	Reference
SET F	Sets annunciator	Page 145
RST F	Resets annunciator	Page 147
ANS	Sets annunciator (with evaluation time)	Page 149
ANR	Resets annunciator (smallest number reset)	Page 151
ANRP		

■Rising/falling edge output

Instruction symbol	Description	Reference
PLS	Generates a pulse for 1 cycle of a program at the rising edge of the input signal.	Page 152
PLF	Generates a pulse for 1 cycle of a program at the falling edge of the input signal.	Page 154

■Inverting the bit device output

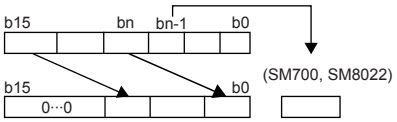
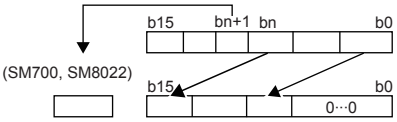
Instruction symbol	Description	Reference
FF	Inversion of device output	Page 156
ALT		Page 157
ALTP		

Shift instruction

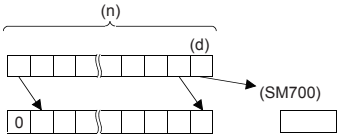
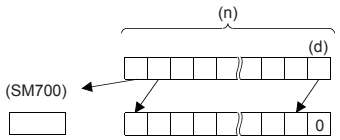
■Shifting bit devices

Instruction symbol	Description	Reference
SFT	1 bit shift of the device	Page 159
SFTP		

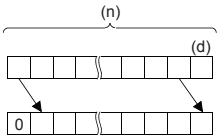
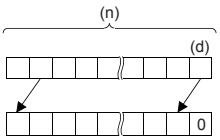
■Shifting 16-bit data to the right/left by n bit (s)

Instruction symbol	Description	Reference
SFR		Page 161
SFRP		
SFL		Page 163
SFLP		

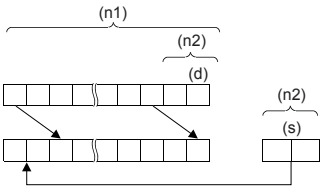
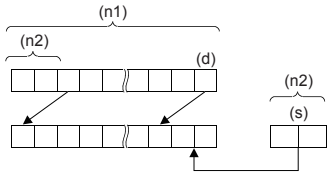
■Shifting n-bit data to the right/left by 1 bit

Instruction symbol	Description	Reference
BSFR		Page 165
BSFRP		
BSFL		Page 167
BSFLP		

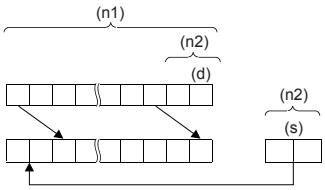
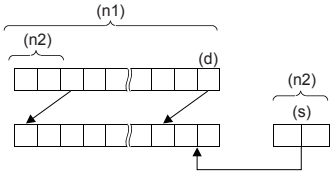
■Shifting n-word data to the right/left by 1 word

Instruction symbol	Description	Reference
DSFR		Page 169
DSFRP		
DSFL		Page 170
DSFLP		

■Shifting n-bit data to the right/left by n bit (s)

Instruction symbol	Description	Reference
SFTR SFTRP		Page 171
SFTL SFTLP		Page 173

■Shifting n-word data to the right/left by n word (s)

Instruction symbol	Description	Reference
WSFR WSFRP		Page 175
WSFL WSFLP		Page 177

Master control instruction

■Setting/resetting the master control

Instruction symbol	Description	Reference
MC	Starts master control	Page 179
MCR	Releases master control	

Termination instruction

■Ending the main routine program

Instruction symbol	Description	Reference
FEND	Ends the main routine program	Page 183

■Ending the sequence program

Instruction symbol	Description	Reference
END	Ends the sequence program	Page 186

Stop instruction

■Stopping the sequence program

Instruction symbol	Description	Reference
STOP	Stops the sequence operation after input conditions are met. Executes the sequence program, upon setting the RUN/STOP/RESET switch to RUN again.	Page 188

3.2 Basic Instruction

Comparison operation instruction

■Comparing 16-bit binary data

Instruction symbol	Description	Reference
LD=, AND=, OR=	(s1)=(s2): Conductive (s1)≠(s2): Non-Conductive	Page 189
LD=_U, AND=_U, OR=_U		
LD<>, AND<>, OR<>	(s1)≠(s2): Conductive (s1)=(s2): Non-Conductive	
LD<>_U, AND<>_U, OR<>_U		
LD>, AND>, OR>	(s1)>(s2): Conductive (s1)≤(s2): Non-Conductive	
LD>_U, AND>_U, OR>_U		
LD<=, AND<=, OR<=	(s1)≤(s2): Conductive (s1)>(s2): Non-Conductive	
LD<=_U, AND<=_U, OR<=_U		
LD<, AND<, OR<	(s1)<(s2): Conductive (s1)≥(s2): Non-Conductive	
LD<_U, AND<_U, OR<_U		
LD>=, AND>=, OR>=	(s1)≥(s2): Conductive (s1)<(s2): Non-Conductive	
LD>=_U, AND>=_U, OR>=_U		

■Comparing 32-bit binary data

Instruction symbol	Description	Reference
LDD=, ANDD=, ORD=	[(s1)+1, (s1)] = [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] ≠ [(s2)+1, (s2)]: Non-Conductive	Page 192
LDD=_U, ANDD=_U, ORD=_U		
LDD<>, ANDD<>, ORD<>	[(s1)+1, (s1)] ≠ [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] = [(s2)+1, (s2)]: Non-Conductive	
LDD<>_U, ANDD<>_U, ORD<>_U		
LDD>, ANDD>, ORD>	[(s1)+1, (s1)] > [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] ≤ [(s2)+1, (s2)]: Non-Conductive	
LDD>_U, ANDD>_U, ORD>_U		
LDD<=, ANDD<=, ORD<=	[(s1)+1, (s1)] ≤ [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] > [(s2)+1, (s2)]: Non-Conductive	
LDD<=_U, ANDD<=_U, ORD<=_U		
LDD<, ANDD<, ORD<	[(s1)+1, (s1)] < [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] ≥ [(s2)+1, (s2)]: Non-Conductive	
LDD<_U, ANDD<_U, ORD<_U		
LDD>=, ANDD>=, ORD>=	[(s1)+1, (s1)] ≥ [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] < [(s2)+1, (s2)]: Non-Conductive	
LDD>=_U, ANDD>=_U, ORD>=_U		

■Comparison output 16-bit binary data

Instruction symbol	Description	Reference
CMP	(s1)>(s2): (d) is on (s1)=(s2): (d) +1 is on (s1)<(s2): (d) +2 is on	Page 194
CMPP		
CMP_U		
CMPP_U		

■Comparison output 32-bit binary data

Instruction symbol	Description	Reference
DCMP	[(s1)+1, (s1)] > [(s2)+1, (s2)]: (d) is on [(s1)+1, (s1)] = [(s2)+1, (s2)]: (d) + 1 is on [(s1)+1, (s1)] < [(s2)+1, (s2)]: (d) + 2 is on	Page 196
DCMPP		
DCMP_U		
DCMPP_U		

■Comparing 16-bit binary data band

Instruction symbol	Description	Reference
ZCP	(s1)>(s3): (d) is on (s1)≤(s3)≤(s2): (d) + 1 is on (s3)>(s2): (d) + 2 is on	Page 198
ZCPP		
ZCP_U		
ZCPP_U		

■Comparing 32-bit binary data band

Instruction symbol	Description	Reference
DZCP	[(s1)+1, (s1)] > [(s3)+1, (s3)]: (d) is on [(s1)+1, (s1)]≤[(s3)+1, (s3)]≤[(s2)+1, (s2)]: (d) + 1 is on [(s3)+1, (s3)]>[(s2)+1, (s2)]: (d) + 2 is on	Page 200
DZCPP		
DZCP_U		
DZCPP_U		

■Comparing 16-bit binary block data

Instruction symbol	Description	Reference
BKCMP=, BKCMP<>, BKCMP>, BKCMP<=, BKCMP<, BKCMP>=	Compares the 16-bit binary data in the device area ((n) points) from (s1) with the 16-bit binary data in the device area ((n) points) from (s2), and stores the result in the device area ((n) points) from (d).	Page 202
BKCMP=P, BKCMP<>P, BKCMP>P, BKCMP<=P, BKCMP<P, BKCMP>=P		
BKCMP=_U, BKCMP<>_U, BKCMP>_U, BKCMP<=_U, BKCMP<_U, BKCMP>=_U		
BKCMP=P_U, BKCMP<>P_U, BKCMP>P_U, BKCMP<=P_U, BKCMP<P_U, BKCMP>=P_U		

■Comparing 32-bit binary block data

Instruction symbol	Description	Reference
DBKCMP=, DBKCMP<>, DBKCMP>, DBKCMP<=, DBKCMP<, DBKCMP>=	Compares the 32-bit binary data in the device area ((n) points) from (s1) with the 32-bit binary data in the device area ((n) points) from (s2), and stores the result in the device area ((n) points) from (d).	Page 205
DBKCMP=P, DBKCMP<>P, DBKCMP>P, DBKCMP<=P, DBKCMP<P, DBKCMP>=P		
DBKCMP=_U, DBKCMP<>_U, DBKCMP>_U, DBKCMP<=_U, DBKCMP<_U, DBKCMP>=_U		
DBKCMP=P_U, DBKCMP<>P_U, DBKCMP>P_U, DBKCMP<=P_U, DBKCMP<P_U, DBKCMP>=P_U		

Arithmetic operation instruction

■Adding/subtracting 16-bit binary data

Instruction symbol	Description	Reference
+	(d)+(s) → (d)	Page 208
+P		
+_U		
+P_U		
+	(s1)+(s2) → (d)	Page 210
+P		
+_U		
+P_U		
ADD	(s1)+(s2) → (d)	Page 212
ADDP		
ADD_U		
ADDP_U		
-	(d)-(s) → (d)	Page 214
-P		
-_U		
-P_U		
-	(s1)-(s2) → (d)	Page 216
-P		
-_U		
-P_U		
SUB	(s1)-(s2) → (d)	Page 218
SUBP		
SUB_U		
SUBP_U		

■Adding/subtracting 32-bit binary data

Instruction symbol	Description	Reference
D+	[(d)+1, (d)] + [(s)+1, (s)] → [(d)+1, (d)]	Page 220
D+P		
D+_U		
D+P_U		
D+	[(s1)+1, (s1)] + [(s2)+1, (s2)] → [(d)+1, (d)]	Page 222
D+P		
D+_U		
D+P_U		
DADD	[(s1)+1, (s1)] + [(s2)+1, (s2)] → [(d)+1, (d)]	Page 224
DADDP		
DADD_U		
DADDP_U		
D-	[(d)+1, (d)] - [(s)+1, (s)] → [(d)+1, (d)]	Page 226
D-P		
D-_U		
D-P_U		
D-	[(s1)+1, (s1)] - [(s2)+1, (s2)] → [(d)+1, (d)]	Page 228
D-P		
D-_U		
D-P_U		
DSUB	[(s1)+1, (s1)] - [(s2)+1, (s2)] → [(d)+1, (d)]	Page 230
DSUBP		
DSUB_U		
DSUBP_U		

■Multiplying/dividing 16-bit binary data

Instruction symbol	Description	Reference
*	(s1) × (s2) → [(d)+1, (d)]	Page 232
*P		
*_U		
*P_U		
MUL	(s1) × (s2) → [(d)+1, (d)]	Page 234
MULP		
MUL_U		
MULP_U		
/	(s1) ÷ (s2) → quotient (d), remainder (d)+1	Page 236
/P		
/_U		
/P_U		
DIV	(s1) ÷ (s2) → quotient (d), remainder (d)+1	Page 238
DIVP		
DIV_U		
DIVP_U		

■Multiplying/dividing 32-bit binary data

Instruction symbol	Description	Reference
D*	[(s1)+1, (s1)] × [(s2)+1, (s2)] → [(d)+3, (d)+2, (d)+1, (d)]	Page 240
D*P		
D*_U		
D*P_U		
DMUL	[(s1)+1, (s1)] × [(s2)+1, (s2)] → [(d)+3, (d)+2, (d)+1, (d)]	Page 242
DMULP		
DMUL_U		
DMULP_U		
D/	[(s1)+1, (s1)] ÷ [(s2)+1, (s2)] → quotient [(d)+1, (d)], remainder [(d)+3, (d)+2]	Page 244
D/P		
D/_U		
D/P_U		
DDIV	[(s1)+1, (s1)] ÷ [(s2)+1, (s2)] → quotient [(d)+1, (d)], remainder [(d)+3, (d)+2]	Page 246
DDIVP		
DDIV_U		
DDIVP_U		

■Adding/subtracting BCD 4-digit data

Instruction symbol	Description	Reference
B+	(d) + (s) → (d)	Page 248
B+P		
B+	(s1) + (s2) → (d)	Page 249
B+P		
B-	(d) - (s) → (d)	Page 250
B-P		
B-	(s1) - (s2) → (d)	Page 251
B-P		

■Adding/subtracting BCD 8-digit data

Instruction symbol	Description	Reference
DB+	[(d)+1, (d)] + [(s)+1, (s)] → [(d)+1, (d)]	Page 253
DB+P		
DB+	[(s1)+1, (s1)] + [(s2)+1, (s2)] → [(d)+1, (d)]	Page 254
DB+P		
DB-	[(d)+1, (d)] - [(s)+1, (s)] → [(d)+1, (d)]	Page 256
DB-P		
DB-	[(s1)+1, (s1)] - [(s2)+1, (s2)] → [(d)+1, (d)]	Page 257
DB-P		

■Multiplying/dividing BCD 4-digit data

Instruction symbol	Description	Reference
B*	(s1) × (s2) → [(d)+1, (d)]	Page 259
B*P		
B/	(s1) ÷ (s2) → quotient (d), remainder (d)+1	Page 261
B/P		

■Multiplying/dividing BCD 8-digit data

Instruction symbol	Description	Reference
DB*	[(s1)+1, (s1)] × [(s2)+1, (s2)] → [(d)+3, (d)+2, (d)+1, (d)]	Page 263
DB*P		
DB/	[(s1)+1, (s1)] ÷ [(s2)+1, (s2)] → quotient [(d)+1, (d)], remainder [(d)+3, (d)+2]	Page 265
DB/P		

■Adding/subtracting 16-bit binary block data

Instruction symbol	Description	Reference
BK+	Adds the 16-bit binary bit data in the device area ((n) points) from (s1) and the data or constants in the device area ((n) points) from (s2) at once, and stores the result in the device area ((n) points) from (d).	Page 267
BK+P		
BK+_U		
BK+P_U		
BK-	Subtracts the 16-bit binary bit data in the device area ((n) points) from (s1) and the data or constants in the device area ((n) points) from (s2) at once, and stores the result in the device area ((n) points) from (d).	Page 269
BK-P		
BK-_U		
BK-P_U		

■Adding/subtracting 32-bit binary block data

Instruction symbol	Description	Reference
DBK+	Adds the 32-bit binary bit data in the device area ((n) points) from (s1) and the 32-bit data or constants in the device area ((n) points) from (s2), and stores the result in the device area specified by (d) and later.	Page 271
DBK+P		
DBK+_U		
DBK+P_U		
DBK-	Subtracts the 32-bit binary bit data in the device area ((n) points) from (s1) and the 32-bit data or constants in the device area ((n) points) from (s2) and later, and stores the result in the device area specified by (d) and later.	Page 273
DBK-P		
DBK-_U		
DBK-P_U		

■Incrementing/decrementing 16-bit binary data

Instruction symbol	Description	Reference
INC	(d) + 1 → (d)	Page 275
INCP		
INC_U		
INCP_U		
DEC	(d) - 1 → (d)	Page 277
DECP		
DEC_U		
DECP_U		

■Incrementing/decrementing 32-bit binary data

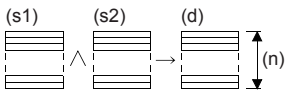
Instruction symbol	Description	Reference
DINC	[(d)+1, (d)] + 1 → [(d)+1, (d)]	Page 278
DINCP		
DINC_U		
DINCP_U		
DDEC	[(d)+1, (d)] -1 → [(d)+1, (d)]	Page 279
DDECP		
DDEC_U		
DDECP_U		

Logical operation instruction

■Performing an AND operation on 16-bit/32-bit data

Instruction symbol	Description	Reference
WAND	(d)∧(s) → (d)	Page 280
WANDP		
WAND	(s1)∧ (s2) → (d)	Page 281
WANDP		
DAND	[(d)+1, (d)] ∧[(s)+1, (s)] → [(d)+1, (d)]	Page 282
DANDP		
DAND	[(s1)+1, (s1)]∧ [(s2)+1, (s2)] → [(d)+1, (d)]	Page 283
DANDP		

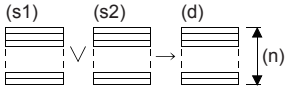
■Performing an AND operation on 16-bit block data

Instruction symbol	Description	Reference
BKAND		Page 285
BKANDP		

■Performing an OR operation on 16-bit/32-bit data

Instruction symbol	Description	Reference
WOR	(d)∨(s) → (d)	Page 287
WORP		
WOR	(s1) ∨(s2) → (d)	Page 288
WORP		
DOR	[(d)+1, (d)] ∨ [(s)+1, (s)] → [(d)+1, (d)]	Page 289
DORP		
DOR	[(s1)+1, (s1)] ∨ [(s2)+1, (s2)] → [(d)+1, (d)]	Page 290
DORP		

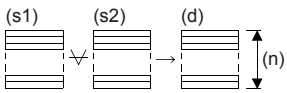
■Performing an OR operation on 16-bit block data

Instruction symbol	Description	Reference
BKOR		Page 292
BKORP		

■Performing an XOR operation on 16-bit/32-bit data

Instruction symbol	Description	Reference
WXOR	(d) ∀ (s) → (d)	Page 294
WXORP		
WXOR	(s1) ∀ (s2) → (d)	Page 295
WXORP		
DXOR	[(d)+1, (d)] ∀ [(s)+1, (s)] → [(d)+1, (d)]	Page 296
DXORP		
DXOR	[(s1)+1, (s1)] ∀ [(s2)+1, (s2)] → [(d)+1, (d)]	Page 297
DXORP		

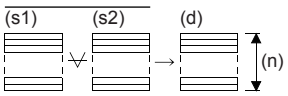
■Performing an XOR operation on 16-bit block data

Instruction symbol	Description	Reference
BKXOR		Page 299
BKXORP		

■Performing an XNOR operation on 16-bit/32-bit data

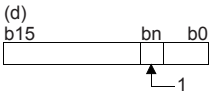
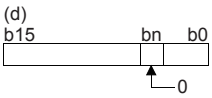
Instruction symbol	Description	Reference
WXNR WXNRP	$\overline{(d) \veebar (s)} \rightarrow (d)$	Page 301
WXNR WXNRP	$\overline{(s1) \veebar (s2)} \rightarrow (d)$	Page 302
DXNR DXNRP	$\overline{[(d)+1, (d)] \veebar [(s)+1, (s)]} \rightarrow [(d)+1, (d)]$	Page 303
DXNR DXNRP	$\overline{[(s1)+1, (s1)] \veebar [(s2)+1, (s2)]} \rightarrow [(d)+1, (d)]$	Page 304

■Performing an XNOR operation on 16-bit block data

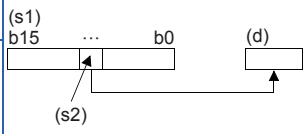
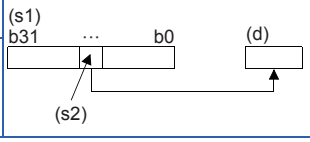
Instruction symbol	Description	Reference
BKXNR BKXNRP		Page 306

Bit processing instruction

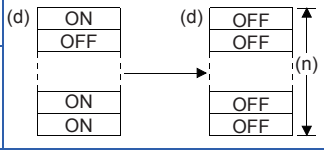
■Setting/resetting a bit in the word device

Instruction symbol	Description	Reference
BSET BSETP		Page 308
BRST BRSTP		Page 309

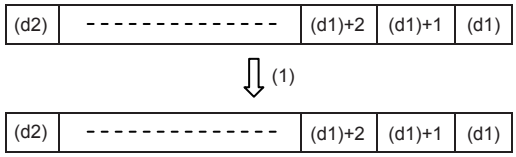
■Performing a bit test

Instruction symbol	Description	Reference
TEST TESTP	Extracts the 'n'th bit in the specified word device. 	Page 310
DTEST DTESTP	Extracts the 'n'th bit in the specified word device. 	Page 312

■Batch-resetting bit devices

Instruction symbol	Description	Reference
BKRST BKRSTP	Resets the (n) points of bit devices starting from the bit device specified. 	Page 314

■Batch-resetting devices

Instruction symbol	Description	Reference
ZRST ZRSTP	 (1): (d1), (d2) are bit devices: Writes off (reset) from (d1) to (d2) (d1), (d2) are word devices: Writes K0 from (d1) to (d2)	Page 315

Data conversion instruction

■Converting binary data to BCD 4-digit/8-digit data

Instruction symbol	Description	Reference
BCD BCDP	Converts the specified 16-bit binary data (0 to 9999) to BCD 4-digit data. (s) → (d) BIN: (s) BCD: (d)	Page 318
DBCD DBCDP	Converts the specified 32-bit binary data (0 to 99999999) to BCD 8-digit data. (s+1, s) → (d+1, d) BIN: (s+1, s) BCD: (d+1, d)	Page 320

■Converting BCD 4-digit/8-digit data to binary data

Instruction symbol	Description	Reference
BIN BINP	Converts the specified BCD 4-digit data (0 to 9999) to 16-bit binary data. (s) → (d) BCD: (s) BIN: (d)	Page 322
DBIN DBINP	Converts the specified BCD 8-digit data (0 to 99999999) to 32-bit binary data. (s+1, s) → (d+1, d) BCD: (s+1, s) BIN: (d+1, d)	Page 324

■Converting single-precision real number to 16-bit/32-bit signed binary data

Instruction symbol	Description	Reference
FLT2INT FLT2INTP	Converts the specified single-precision real number (-32768 to 32767) to 16-bit signed binary data. (s+1, s) → (d) (1): (s+1, s) BIN: (d) (1): Real number	Page 326
FLT2DINT FLT2DINTP	Converts the specified single-precision real number (-2147483648 to 2147483647) to 32-bit signed binary data. (s+1, s) → (d+1, d) (1): (s+1, s) BIN: (d+1, d) (1): Real number	Page 330

■Converting single-precision real number to 16-bit/32-bit unsigned binary data

Instruction symbol	Description	Reference
FLT2UINT FLT2UINTP	Converts the specified single-precision real number (0 to 65535) to 16-bit unsigned binary data. (s+1, s) → (d) (1): (s+1, s) BIN: (d) (1): Real number	Page 328
FLT2UDINT FLT2UDINTP	Converts the specified single-precision real number (0 to 4294967295) to 16-bit unsigned binary data. (s+1, s) → (d+1, d) (1): (s+1, s) BIN: (d+1, d) (1): Real number	Page 332

■Converting 16-bit signed binary data to 16-bit/32-bit unsigned binary data

Instruction symbol	Description	Reference
INT2UINT	Converts 16-bit signed data in the device specified by (s) to 16-bit unsigned data, and stores the converted data in the device specified by (d).	Page 334
INT2UINTP		
INT2UDINT	Converts 16-bit signed data in the device specified by (s) to 32-bit unsigned data, and stores the converted data in the device specified by (d).	Page 336
INT2UDINTP		

■Converting 16-bit signed binary data to 32-bit signed binary data

Instruction symbol	Description	Reference
INT2DINT	Converts 16-bit signed data in the device specified by (s) to 32-bit signed data, and stores the converted data in the device specified by (d).	Page 335
INT2DINTP		

■Converting 16-bit unsigned binary data to 16-bit/32-bit signed binary data

Instruction symbol	Description	Reference
UINT2INT	Converts 16-bit unsigned data in the device specified by (s) to 16-bit signed data, and stores the converted data in the device specified by (d).	Page 337
UINT2INTP		
UINT2DINT	Converts 16-bit unsigned data in the device specified by (s) to 32-bit signed data, and stores the converted data in the device specified by (d).	Page 338
UINT2DINTP		

■Converting 16-bit unsigned binary data to 32-bit unsigned binary data

Instruction symbol	Description	Reference
UINT2UDINT	Converts 16-bit unsigned data in the device specified by (s) to 32-bit unsigned data, and stores the converted data in the device specified by (d).	Page 339
UINT2UDINTP		

■Converting 32-bit signed binary data to 16-bit signed binary data

Instruction symbol	Description	Reference
DINT2INT	Converts 32-bit signed data in the device specified by (s) to 16-bit signed data, and stores the converted data in the device specified by (d).	Page 340
DINT2INTP		

■Converting 32-bit signed binary data to 16-bit/32-bit unsigned binary data

Instruction symbol	Description	Reference
DINT2UINT	Converts 32-bit signed data in the device specified by (s) to 16-bit unsigned data, and stores the converted data in the device specified by (d).	Page 341
DINT2UINTP		
DINT2UDINT	Converts 32-bit signed data in the device specified by (s) to 32-bit unsigned data, and stores the converted data in the device specified by (d).	Page 342
DINT2UDINTP		

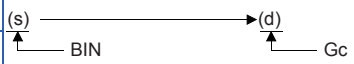
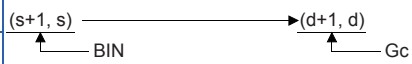
■Converting 32-bit unsigned binary data to 16-bit/32-bit signed binary data

Instruction symbol	Description	Reference
UDINT2INT	Converts 32-bit unsigned data in the device specified by (s) to 16-bit signed data, and stores the converted data in the device specified by (d).	Page 343
UDINT2INTP		
UDINT2DINT	Converts 32-bit unsigned data in the device specified by (s) to 32-bit signed data, and stores the converted data in the device specified by (d).	Page 345
UDINT2DINTP		

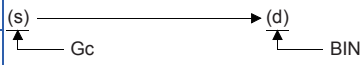
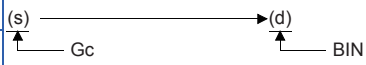
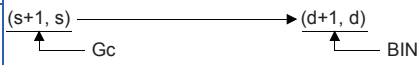
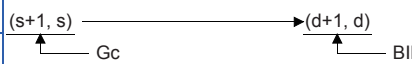
■Converting 32-bit unsigned binary data to 16-bit unsigned binary data

Instruction symbol	Description	Reference
UDINT2UINT	Converts 32-bit unsigned data in the device specified by (s) to 16-bit unsigned data, and stores the converted data in the device specified by (d).	Page 344
UDINT2UINTP		

■Converting 16-bit/32-bit binary data to Gray code

Instruction symbol	Description	Reference
GRY	Converts the specified 16-bit binary data (-32768 to 32767) to 16-bit binary Gray code data. Gc: Gray code	Page 346
GRYP		
GRY_U	Converts the specified 16-bit binary data (0 to 65535) to 16-bit binary Gray code data.  Gc: Gray code	
GRYP_U		
DGRY	Converts the specified 32-bit binary data (-2147483648 to 2147483647) to 32-bit binary Gray code data. Gc: Gray code	Page 347
DGRYP		
DGRY_U	Converts the specified 32-bit binary data (0 to 4294967295) to 32-bit binary Gray code data.  Gc: Gray code	
DGRYP_U		

■Converting Gray code to 16-bit/32-bit binary data

Instruction symbol	Description	Reference
GBIN	Converts the specified 16-bit binary Gray code data (-32768 to 32767) to 16-bit binary data.  Gc: Gray code	Page 349
GBINP		
GBIN_U	Converts the specified 16-bit binary Gray code data (0 to 65535) to 16-bit binary data.  Gc: Gray code	
GBINP_U		
DGBIN	Converts the specified 32-bit binary Gray code data (-2147483648 to 2147483647) to 32-bit binary data.  Gc: Gray code	Page 350
DGBINP		
DGBIN_U	Converts the specified 32-bit binary Gray code data (0 to 4294967295) to 32-bit binary data.  Gc: Gray code	
DGBINP_U		

■Converting decimal ASCII to 16-bit/32-bit binary data

Instruction symbol	Description	Reference
DABIN	Converts a 5-digit decimal ASCII value in the device specified by (s) to a 1 word binary value, and stores the converted data in the word device number specified by (d).	Page 352
DABINP		
DABIN_U		
DABINP_U		
DDABIN	Converts a 10-digit decimal ASCII value in the device specified by (s) to a 2 word binary value, and stores the converted data in the word device number specified by (d).	Page 356
DDABINP		
DDABIN_U		
DDABINP_U		

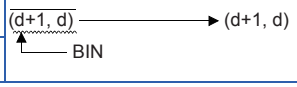
■Converting ASCII to HEX

Instruction symbol	Description	Reference
HEXA	Converts the ASCII data stored in the number of characters specified by (n) starting from device specified in (s), and stores the converted data in the device specified by (d) onwards.	Page 360
HEXAP		

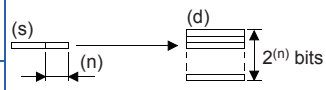
■Converting character string to 16-bit/32-bit binary data

Instruction symbol	Description	Reference
VAL VALP VAL_U VALP_U	Converts a character string including decimal point in the device specified by (s) to a 1 word binary value and number of decimal fraction digits, and stores the converted data in the devices specified by (d1) and (d2).	Page 364
DVAL DVALP DVAL_U DVALP_U	Converts a character string including decimal point in the device specified by (s) to a 2 words binary value and number of decimal fraction digits, and stores the converted data in the devices specified by (d1) and (d2).	Page 367

■Two's complement of 16-bit/32-bit binary data (sign inversion)

Instruction symbol	Description	Reference
NEG NEGP		Page 370
DNEG DNEGP		Page 372

■Decoding from 8 to 256 bits

Instruction symbol	Description	Reference
DECO DECOP		Page 373

■Encoding from 256 to 8 bits

Instruction symbol	Description	Reference
ENCO ENCOP		Page 375

■Seven-segment decoding

Instruction symbol	Description	Reference
SEGD SEGDP	Decoded to data for the seven-segment display unit in the device specified by (s),and stores in the device specified by (d).	Page 377

■Seven-segment with latch

Instruction symbol	Description	Reference
SEGL	The 4-digit numeric value stored in (s) is converted into BCD data, and each digit is output to the seven-segment display unit with the BCD decoder by the time division method.	Page 379

■Separating 4 bits from 16-bit data

Instruction symbol	Description	Reference
DIS DISP	Separates the 16-bit data specified by (s) into 4-bit units and stores in the lower 4 bits of (n) points from (d). (n < 4)	Page 382

■Connecting 4 bits to 16-bit data

Instruction symbol	Description	Reference
UNI UNIP	Connects the lower 4 bits of (n) points from the device specified by (s), and stores the result in the device specified by (d). (n < 4)	Page 384

■Separating/connecting the specified number of bits

Instruction symbol	Description	Reference
NDIS	Separates the data in the devices starting from the one specified by (s1) into bits specified by the devices from (s2), and stores them to the devices starting from the one specified by (d).	Page 386
NDISP		
NUNI	Connects the data in the devices starting from the one specified by (s1) with bits specified by the devices from (s2), and stores them to the devices starting from the one specified by (d).	Page 388
NUNIP		

■Separating/connecting data in byte units

Instruction symbol	Description	Reference
WTOB	Breaks (n) points of 16 bit data from the device specified by (s) into 8-bit units, and stores in the devices starting from the one specified by (d).	Page 390
WTOBP		
BTOW	Connects the lower 8 bits of 16-bit data of (n) points from the device specified by (s) into 16-bit units, and stores in the devices starting from the one specified by (d).	Page 392
BTOWP		

Digital Switch

Instruction symbol	Description	Reference
DSW	Stores the value of the (n) sets of digital switches connected to (s) to (d2).	Page 395

Data transfer instruction

■Transferring 16-bit/32-bit data

Instruction symbol	Description	Reference
MOV MOVP	(s) ⟶ (d)	Page 397
DMOV DMOVP	(s+1, s) ⟶ (d+1, d)	Page 399

■Inverting and transferring 16-bit/32-bit data

Instruction symbol	Description	Reference
CML CMLP	$\overline{(s)}$ ⟶ (d)	Page 401
DCML DCMLP	$\overline{(s+1, s)}$ ⟶ (d+1, d)	Page 403

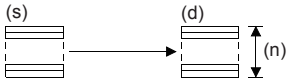
■Shift move

Instruction symbol	Description	Reference
SMOV SMOVP	Shifts the specified no. of digits from the word device specified by (s), and store in (d).	Page 404

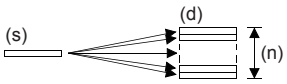
■Inverting and transferring 1-bit data

Instruction symbol	Description	Reference
CMLB CMLBP	Inverts the bit data specified by (s), and store in (d).	Page 406

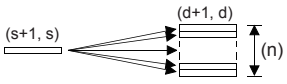
■Transferring 16-bit block data

Instruction symbol	Description	Reference
BMOV BMOVP	 (n) = 1 to 65535	Page 407

■Transferring identical 16-bit block data

Instruction symbol	Description	Reference
FMOV FMOVP	 (n) = 1 to 65535	Page 409

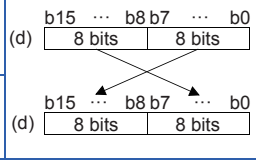
■Transferring identical 32-bit block data

Instruction symbol	Description	Reference
DFMOV DFMOVP	 (n) = 1 to 65535	Page 411

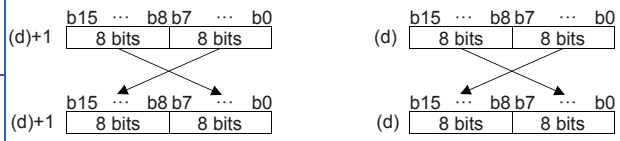
■Exchanging 16-bit/32-bit data

Instruction symbol	Description	Reference
XCH XCHP	(d1) ⟷ (d2)	Page 413
DXCH DXCHP	(d1+1, d1) ⟷ (d2+1, d2)	Page 415

■Exchanging the upper and lower bytes of 16-bit data

Instruction symbol	Description	Reference
SWAP SWAPP		Page 417

■Exchanging the upper and lower bytes of 32-bit data

Instruction symbol	Description	Reference
DSWAP DSWAPP		Page 418

■Transferring 1-bit data

Instruction symbol	Description	Reference
MOVB MOVBP	Stores the bit data specified by (s) in (d).	Page 419

■Parallel run (octal mode) (16-bit data)

Instruction symbol	Description	Reference
PRUN PRUNP	Handles device number specified by (s) in digit specification and (d) as octal, and stores into (d) from (s).	Page 420

■Parallel run (octal mode) (32-bit data)

Instruction symbol	Description	Reference
DPRUN DPRUNP	Handles device number specified by (s) in digit specification and (d) as octal, and stores into (d) from (s).	Page 422

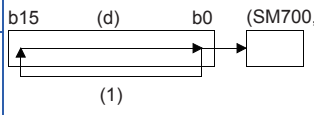
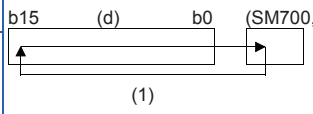
■Transferring n-bit data

Instruction symbol	Description	Reference
BLKMOVB BLKMOVBP	Block transfers bit data for (n) points from (s) to bit data for (n) points from (d).	Page 424

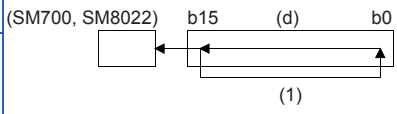
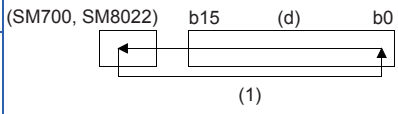
3.3 Application Instruction

Rotation instruction

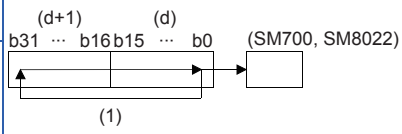
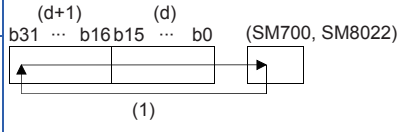
■Rotating 16-bit data to the right

Instruction symbol	Description	Reference
ROR RORP	Rotates the 16-bit binary data to the right by (n) bit(s) (not including the carry flag).  (1): (n) bit right rotation	Page 426
RCR RCRP	Rotates the 16-bit binary data to the right by (n) bit(s) (including the carry flag).  (1): (n) bit right rotation	

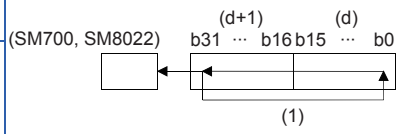
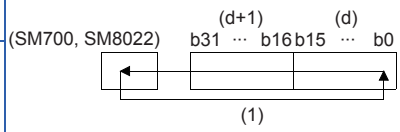
■Rotating 16-bit data to the left

Instruction symbol	Description	Reference
ROL ROLP	Rotates the 16-bit binary data to the left by (n) bit(s) (not including the carry flag).  (1): (n) bit left rotation	Page 429
RCL RCLP	Rotates the 16-bit binary data to the left by (n) bit(s) (including the carry flag).  (1): (n) bit left rotation	

■Rotating 32-bit data to the right

Instruction symbol	Description	Reference
DROR DRORP	Rotates the 32-bit binary data to the right by (n) bit(s) (not including the carry flag).  (1): (n) bit right rotation	Page 432
DRCR DRCRP	Rotates the 32-bit binary data to the right by (n) bit(s) (including the carry flag).  (1): (n) bit right rotation	

■Rotating 32-bit data to the left

Instruction symbol	Description	Reference
DROL DROLP	Rotates the 32-bit binary data to the left by (n) bit(s) (not including the carry flag).  (1): (n) bit left rotation	Page 434
DRCL DRCLP	Rotates the 32-bit binary data to the left by (n) bit(s) (including the carry flag).  (1): (n) bit left rotation	

Program branch instruction

■Pointer branch

Instruction symbol	Description	Reference
CJ CJP	When the input condition is met, jump to pointer (P)	Page 436

■Jumping to END

Instruction symbol	Description	Reference
GOEND	When the input condition is met, jump to END instruction	Page 440

Program execution control instruction

■Disabling/enabling interrupt programs

Instruction symbol	Description	Reference
DI	Disables the execution of interrupt programs.	Page 441
EI	Releases the execution disabled state of interrupt program.	

■Disabling the interrupt program with specified priority or lower

Instruction symbol	Description	Reference
DI	Disables the execution of the interrupt program with a priority specified by (s) or lower until the EI instruction is executed.	Page 443

■Interrupt program mask

Instruction symbol	Description	Reference
IMASK	Interrupt disable/enable settings	Page 447

■Disabling/enabling the specified interrupt pointer

Instruction symbol	Description	Reference
SIMASK	Disables/enables the interrupt pointer specified by (I)	Page 449

■Returning from the interrupt program

Instruction symbol	Description	Reference
IRET	Returns from the interrupt program to the sequence program	Page 451

■Resetting the watchdog timer

Instruction symbol	Description	Reference
WDT WDTP	Resets the watchdog timer (WDT) in the program	Page 454

Structuring instruction

■Performing the FOR to NEXT instruction loop

Instruction symbol	Description	Reference
FOR	Execute the instructions between FOR instruction and NEXT instruction (n) times	Page 455
NEXT		

■Forcibly terminating the FOR to NEXT instruction loop

Instruction symbol	Description	Reference
BREAK	Forcibly end execution between FOR instruction and NEXT instruction, and jump to pointer (P)	Page 458
BREAKP		

■Calling a subroutine program

Instruction symbol	Description	Reference
CALL	Executes a subroutine program specified by (P) when the input condition is met.	Page 460
CALLP		

■Returning from the subroutine program

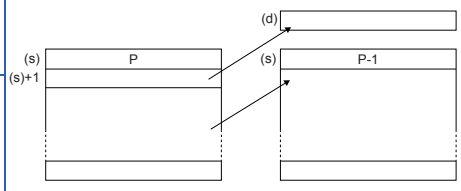
Instruction symbol	Description	Reference
RET	Returns from the subroutine program.	Page 465
SRET		

■Calling a subroutine program

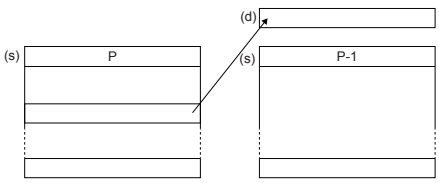
Instruction symbol	Description	Reference
XCALL	Executes a subroutine program specified by (P) when the input condition is met. Carry out non-execution processing for the subroutine program (P), when input conditions are not met.	Page 466

Data table operation instruction

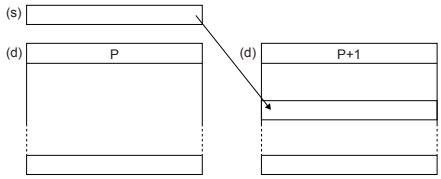
■Reading the oldest data from the data table

Instruction symbol	Description	Reference
SFRD	 P: Pointer	Page 468
SFRDP		

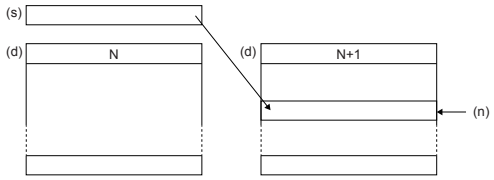
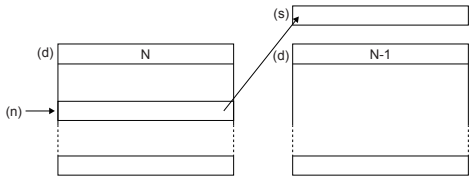
■Reading the newest data from the data table

Instruction symbol	Description	Reference
POP	 P: Pointer	Page 471
POPP		

■Writing data to the data table

Instruction symbol	Description	Reference
SFWR SFWRP	 P: Pointer	Page 474

■Deleting/inserting data from/to the data table

Instruction symbol	Description	Reference
FINS FINSP	 N: Number of stored data	Page 476
FDEL FDELP	 N: Number of stored data	Page 478

Reading/writing data instructions

■Reading data from the data memory

Instruction symbol	Description	Reference
S.DEVLD SP.DEVLD	Reads data from the device data storage file in data memory.	Page 481

■Writing data to the data memory

Instruction symbol	Description	Reference
SP.DEVST	Writes the specified number of points of data to the device data storage file in data memory.	Page 483

File operation instructions

■Reading data from the specified file

Instruction symbol	Processing details	Reference
SP.FREAD	Reads data from the specified file.	Page 486

■Writing data to the specified file

Instruction symbol	Processing details	Reference
SP.FWRITE	Writes data to the specified file.	Page 512

■Deleting the specified file

Instruction symbol	Processing details	Reference
SP.FDELETE	Deletes the specified file or folder.	Page 535

■Copying the specified file

Instruction symbol	Processing details	Reference
SP.FCOPY	Copies the specified file or folder.	Page 543

■Moving the specified file

Instruction symbol	Processing details	Reference
SP.FMOVE	Moves the specified file or folder.	Page 553

■Renaming the specified file

Instruction symbol	Processing details	Reference
SP.FRENAME	Renames the specified file or folder.	Page 563

■Acquiring the status of the specified file

Instruction symbol	Processing details	Reference
SP.FSTATUS	Acquires the status of the specified file or folder.	Page 571

Extended file register operation instruction

■Reading extended file register

Instruction symbol	Description	Reference
ERREAD	Reads the current value of the extended file register (ER) to the file register (R) in the CPU built-in memory.	Page 580

■Writing extended file register

Instruction symbol	Description	Reference
ERWRITE	Writes the current value of the file register (R) in the CPU built-in memory to the extended file register (ER).	Page 583

■Batch initialization function of extended file register

Instruction symbol	Description	Reference
ERINIT	Initialize all the points of the extended file register (ER) in a batch.	Page 586

Character string operation instruction

■Comparing character strings

Instruction symbol	Description	Reference
LD$=, AND$=, OR$=	Compares the character string (s1) with the character string (s2) one character at a time.*1 [Character string (s1)] = [Character string (s2)]: Conductive state [Character string (s1)] ≠ [Character string (s2)]: Non-Conductive state	Page 589
LD$<>, AND$<>, OR$<>	Compares the character string (s1) with the character string (s2) one character at a time.*1 [Character string (s1)] ≠ [Character string (s2)]: Conductive state [Character string (s1)] = [Character string (s2)]: Non-Conductive state	
LD$>, AND$>, OR$>	Compares the character string (s1) with the character string (s2) one character at a time.*1 [Character string (s1)] > [Character string (s2)]: Conductive state [Character string (s1)] ≤ [Character string (s2)]: Non-Conductive state	
LD$<=, AND$<=, OR$<=	Compares the character string (s1) with the character string (s2) one character at a time.*1 [Character string (s1)] ≤ [Character string (s2)]: Conductive state [Character string (s1)] > [Character string (s2)]: Non-Conductive state	
LD$<, AND$<, OR$<	Compares the character string (s1) with the character string (s2) one character at a time.*1 [Character string (s1)] < [Character string (s2)]: Conductive state [Character string (s1)] ≥ [Character string (s2)]: Non-Conductive state	
LD$>=, AND$>=, OR$>	Compares the character string (s1) with the character string (s2) one character at a time.*1 [Character string (s1)] ≥ [Character string (s2)]: Conductive state [Character string (s1)] < [Character string (s2)]: Non-Conductive state	

*1 The following shows comparison conditions for comparing character strings.
- Match: All characters in the strings must match
- Larger string: In case of different character strings, character string with the larger character code
(If character string lengths are different, the longer character string)
- Smaller string: In case of different character strings, character string with the smaller character code
(If character string lengths are different, the shorter character string)

■Concatenating character strings

Instruction symbol	Description	Reference
$+ $+P	• In case of 2 operands Connect the character string specified by (s) to the end of the character string specified by (d), and store in (d).	Page 592
$+ $+P	• In case of 3 operands Connect the character string specified by (s2) to the end of the character string specified by (s1), and store in (d).	Page 594

■Transferring character strings

Instruction symbol	Description	Reference
$MOV $MOVP	Transfer the character strings specified by (s) to the devices specified by (d) onwards.	Page 596
$MOV_WS $MOVP_WS	Transfers the Unicode character strings in the device specified by (s) to the device specified by (d) and later.	Page 598

■Converting 16-bit/32-bit binary data to decimal ASCII

Instruction symbol	Description	Reference
BINDA BINDAP BINDA_U BINDAP_U	Converts the 1 word binary value specified by (s) to 5 digits decimal ASCII value, and stores in the word device specified by (d).	Page 600
DBINDA DBINDAP DBINDA_U DBINDAP_U	Converts the 2 word binary value specified by (s) to 10 digits decimal ASCII value, and stores in the word device area specified by (d) onwards.	Page 605

■Converting HEX code data to ASCII

Instruction symbol	Description	Reference
ASCI	Converts the (n) characters within the HEX code data specified by (s) to ASCII, and stores in the device area specified by (d) onwards.	Page 611
ASCIP		

■Converting 16-bit/32-bit binary data to character string

Instruction symbol	Description	Reference
STR	Converts the 1 word binary value specified by (s2) to the decimal character string with total number of digits and the number of digits in the decimal fraction part as specified in (s1), and stores this in the device specified by (d).	Page 615
STRP		
STR_U		
STRP_U		
DSTR	Convert the 2 word binary value specified by (s2) to the decimal character string with total number of digits and the number of digits in the decimal fraction part as specified in (s1), and stores this in the device specified by (d).	Page 618
DSTRP		
DSTR_U		
DSTRP_U		

■Converting single-precision real number to character string

Instruction symbol	Description	Reference
ESTR	Converts the single-precision real number data specified by (s1) to a character string, and store this in the device specified by (d).	Page 621
ESTRP		
DESTR		
DESTRP		

■Converting Unicode character string to Shift JIS character string

Instruction symbol	Processing details	Reference
WS2SJIS	Converts the Unicode character string in the device specified by (s) to the shift JIS character string, and stores the converted data in the device specified by (d).	Page 628
WS2SJISP		

■Converting shift JIS character string to Unicode character string (without byte order mark)

Instruction symbol	Processing details	Reference
SJIS2WS	Converts the shift JIS character string in the device specified by (s) to a Unicode character string, and stores the converted data in the device specified by (d).	Page 631
SJIS2WSP		

■Converting shift JIS to Unicode (with byte order mark)

Instruction symbol	Processing details	Reference
SJIS2WSB	Converts the shift JIS character string in the device specified by (s) to the Unicode character string, add a byte order mark to the head of the converted data, and stores it in the device specified by (d).	Page 634
SJIS2WSBP		

■Detecting a character string length

Instruction symbol	Description	Reference
LEN	Stores the length of the character string data stored in the device specified by (s) in the device specified by (d).	Page 637
LENP		

■Extracting character string data from the right/left

Instruction symbol	Description	Reference
RIGHT	Stores the (n) characters from the last character of the character string specified by (s) in the device specified by (d).	Page 639
RIGHTP		
LEFT	Stores the (n) characters from the first character of the character string specified by (s) in the device specified by (d).	Page 642
LEFTP		

■Storing/replacing the specified number of character strings

Instruction symbol	Description	Reference
MIDR	Stores the specified number of characters from the position specified by (s2) of the character string (s1) into the device specified by (d).	Page 645
MIDRP		
MIDW	Stores the specified number of characters from the character string (s1) into the location specified by (s2) of the character string (d).	Page 648
MIDWP		

■Searching character string

Instruction symbol	Description	Reference
INSTR	Searches the character string in the device specified by (s2), starting from the (s3)th character, for the character string in the device specified by (s1), and stores the matching location in the device specified by (d).	Page 652
INSTRP		

■Inserting character string

Instruction symbol	Description	Reference
STRINS	Inserts the character string data specified in (s1) at the position (s2)(Insert position) from the beginning of the character string data specified by (d).	Page 655
STRINSP		

■Deleting character string

Instruction symbol	Description	Reference
STRDEL	From the head of the character string data specified in (d), delete (n2) characters from the location specified as the character number (n1) (deletion start location).	Page 657
STRDELP		

Real number instruction

■Comparing single-precision real numbers

Instruction symbol	Description	Reference
LDE=, ANDE=, ORE=	[(s1)+1, (s1)] = [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] ≠ [(s2)+1, (s2)]: Non-Conductive	Page 659
LDE<>, ANDE<>, ORE<>	[(s1)+1, (s1)] ≠ [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] = [(s2)+1, (s2)]: Non-Conductive	
LDE>, ANDE>, ORE>	[(s1)+1, (s1)] > [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] ≤ [(s2)+1, (s2)]: Non-Conductive	
LDE<=, ANDE<=, ORE<=	[(s1)+1, (s1)] ≤ [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] > [(s2)+1, (s2)]: Non-Conductive	
LDE<, ANDE<, ORE<	[(s1)+1, (s1)] < [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] ≥ [(s2)+1, (s2)]: Non-Conductive	
LDE>=, ANDE>=, ORE>	[(s1)+1, (s1)] ≥ [(s2)+1, (s2)]: Conductive [(s1)+1, (s1)] < [(s2)+1, (s2)]: Non-Conductive	
DECMP	This instruction compares two data values (single-precision real numbers), and outputs the result (larger, smaller or equal) to three bit devices.	Page 661
DECMPP		
DEZCP	This instruction compares two data values (single-precision real numbers), and outputs the result (larger, smaller or data band) to three bit devices.	Page 663
DEZCPP		

■Adding/subtracting single-precision real numbers

Instruction symbol	Description	Reference
E+	• In case of 2 operands [(d)+1, (d)] + [(s)+1, (s)] → [(d)+1, (d)]	Page 665
E+P		
E+	• In case of 3 operands [(s1)+1, (s1)] + [(s2)+1, (s2)] → [(d)+1, (d)]	Page 667
E+P		
DEADD		Page 673
DEADDP		
E-	• In case of 2 operands [(d)+1, (d)] - [(s)+1, (s)] → [(d)+1, (d)]	Page 669
E-P		
E-	• In case of 3 operands [(s1)+1, (s1)] - [(s2)+1, (s2)] → [(d)+1, (d)]	Page 671
E-P		
DESUB		Page 675
DESUBP		

■Multiplying/dividing single-precision real numbers

Instruction symbol	Description	Reference
E*	[(s1)+1, (s1)] × [(s2)+1, (s2)] → [(d)+1, (d)]	Page 677
E*P		
DEMUL		Page 681
DEMULP		
E/	[(s1)+1, (s1)] ÷ [(s2)+1, (s2)] → quotient [(d)+1, (d)]	Page 679
E/P		
DEDIV		Page 683
DEDIVP		

■Converting 16-bit/32-bit signed binary data to single-precision real number

Instruction symbol	Description	Reference
INT2FLT	Converts the 16-bit signed binary data in the device specified by (s) to single-precision real number, and stores the converted data in the device specified by (d).	Page 685
INT2FLTP		
DINT2FLT	Converts the 32-bit signed binary data in the device specified by (s) to single-precision real number, and stores the converted data in the device specified by (d).	Page 687
DINT2FLTP		

■Converting 16-bit/32-bit unsigned binary data to single-precision real number

Instruction symbol	Description	Reference
UINT2FLT	Converts the 16-bit unsigned binary data in the device specified by (s) to single-precision real number, and stores the converted data in (d).	Page 686
UINT2FLTP		
UDINT2FLT	Converts the 32-bit unsigned binary data in the device specified by (s) to single-precision real number, and stores the converted data in (d).	Page 688
UDINT2FLTP		

■Converting character string to single-precision real number

Instruction symbol	Description	Reference
EVAL	Converts the character string specified by (s) to a single-precision real number, and stores the converted data in (d).	Page 689
EVALP		
DEVAL		
DEVALP		

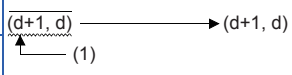
■Converting binary floating point to decimal floating point

Instruction symbol	Description	Reference
DEBCD	Converts the binary floating point specified by (s) into decimal floating point, and stores in (d).	Page 694
DEBCDP		

■Converting decimal floating point to binary floating point

Instruction symbol	Description	Reference
DEBIN	Converts the decimal floating point specified by (s) into binary floating point, and stores in (d).	Page 696
DEBINP		

■Inverting the sign of single-precision real number

Instruction symbol	Description	Reference
ENEG	 (1): Real number	Page 698
ENEGP		
DENEG		
DENEGP		

■Transferring single-precision real number data

Instruction symbol	Description	Reference
EMOV	(1): Real number	Page 700
EMOVP		
DEMOV		
DEMOVP		

■Calculating the sine of single-precision real number

Instruction symbol	Description	Reference
SIN	Sin [(s)+1, (s)] → [(d)+1, (d)]	Page 702
SINP		
DSIN		
DSINP		

■Calculating the cosine of single-precision real number

Instruction symbol	Description	Reference
COS	Cos [(s)+1, (s)] → [(d)+1, (d)]	Page 704
COSP		
DCOS		
DCOSP		

■Calculating the tangent of single-precision real number

Instruction symbol	Description	Reference
TAN	Tan [(s)+1, (s)] → [(d)+1, (d)]	Page 706
TANP		
DTAN		
DTANP		

■Calculating the arc sine of single-precision real number

Instruction symbol	Description	Reference
ASIN	Sin^{-1} [(s)+1, (s)] → [(d)+1, (d)]	Page 708
ASINP		
DASIN		
DASINP		

■Calculating the arc cosine of single-precision real number

Instruction symbol	Description	Reference
ACOS	Cos^{-1} [(s)+1, (s)] → [(d)+1, (d)]	Page 711
ACOSP		
DACOS		
DACOSP		

■Calculating the arc tangent of single-precision real number

Instruction symbol	Description	Reference
ATAN	Tan^{-1} [(s)+1, (s)] → [(d)+1, (d)]	Page 714
ATANP		
DATAN		
DATANP		

■Converting single-precision real number angle to radian

Instruction symbol	Description	Reference
RAD	(s+1, s) ⟶ (d+1, d) Converts from degrees to radians	Page 716
RADP		
DRAD		
DRADP		

■Converting single-precision real number radian to angle

Instruction symbol	Description	Reference
DEG	(s+1, s) ⟶ (d+1, d) Converts from radians to degrees	Page 718
DEGP		
DDEG		
DDEGP		

■Calculating the square root of single-precision real number

Instruction symbol	Description	Reference
DESQR	$\sqrt{(s+1, s)}$ ⟶ (d+1, d)	Page 720
DESQRP		
ESQRT		
ESQRTP		

■Calculating the exponent of single-precision real number

Instruction symbol	Description	Reference
EXP	$e^{[(s)+1, (s)]}$ → [(d)+1, (d)]	Page 722
EXPP		
DEXP		
DEXPP		

■Calculating the natural logarithm of single-precision real number

Instruction symbol	Description	Reference
LOG	Log_e[(s)+1, (s)] → [(d)+1, (d)]	Page 724
LOGP		
DLOGE		
DLOGEP		

■Calculating the exponentiation of single-precision real number

Instruction symbol	Description	Reference
POW	[(s1)+1, (s1)]$^{[(s2)+1, (s2)]}$ → [(d)+1, (d)]	Page 726
POWP		

■Calculating the common logarithm of single-precision real number

Instruction symbol	Description	Reference
LOG10	log_{10}[(s)+1, (s)] → [(d)+1, (d)]	Page 728
LOG10P		
DLOG10		
DLOG10P		

■Searching the maximum value of single-precision real number

Instruction symbol	Description	Reference
EMAX	These instructions search for the maximum value in the (n) points of single-precision real number block data specified by the device starting from the one specified by (s), and store the maximum value in the device area specified by (d).	Page 730
EMAXP		

■Searching the minimum value of single-precision real number

Instruction symbol	Description	Reference
EMIN	These instructions search for the minimum value in the (n) points of single-precision real number block data specified by the device starting from the one specified by (s), and store the minimum value in the device areas specified by (d).	Page 732
EMINP		

Random number instruction

■Generating random number

Instruction symbol	Description	Reference
RND	Generates a random number from 0 to 32767, and stores this in the device specified by (d).	Page 734
RNDP		

Index register operation instruction

■Saving/returning all data of the index register

Instruction symbol	Description	Reference
ZPUSH	Saves the contents of index registers to the devices specified by (d) onwards.	Page 736
ZPUSHP		
ZPOP	Reads the data in devices specified by (d) onwards to the index registers.	Page 739
ZPOPP		

■Saving/returning the selected data of the index register and long index register

Instruction symbol	Description	Reference
ZPUSH	Saves the contents of the index registers and long index registers in the range specified by (s) to devices specified by (d) onwards.	Page 740
ZPUSHP		
ZPOP	Reads data in the devices specified by (d) onwards to the index registers and long index registers.	Page 742
ZPOPP		

Data control instruction

■Upper and lower limit control of 16-bit/32-bit binary data

Instruction symbol	Description	Reference
LIMIT	(s3) < (s1): The (s1) value is stored in (d) (s1) ≤ (s3) ≤ (s2): The (s3) value is stored in (d) (s2) < (s3): The (s2) value is stored in (d)	Page 743
LIMITP		
LIMIT_U		
LIMITP_U		
DLIMIT	[(s3)+1, (s3)] < [(s1)+1, (s1)]: The [(s1)+1, (s1)] value is stored in [(d)+1, (d)] [(s1)+1, (s1)] ≤ [(s3)+1, (s3)] ≤ [(s2)+1, (s2)]: The [(s3)+1, (s3)] value is stored in [(d)+1, (d)] [(s2)+1, (s2)] < [(s3)+1, (s3)]: The [(s2)+1, (s2)] value is stored in [(d)+1, (d)]	Page 745
DLIMITP		
DLIMIT_U		
DLIMITP_U		

■Dead band control of 16-bit/32-bit binary data

Instruction symbol	Description	Reference
BAND BANDP BAND_U BANDP_U	When (s1) ≤ (s3) ≤ (s2): 0 → (d) When (s3) < (s1): (s3) - (s1) → (d) When (s2) < (s3): (s3) - (s2) → (d)	Page 747
DBAND DBANDP DBAND_U DBANDP_U	When [(s1)+1, (s1)] ≤ [(s3)+1, (s3)] ≤ [(s2)+1, (s2)]: 0 → (d+1, d) When [(s3)+1, (s3)] < [(s1)+1, (s1)]: [(s3)+1, (s3)] - [(s1)+1, (s1)] → [(d)+1, (d)] When [(s2)+1, (s2)] < [(s3)+1, (s3)]: [(s3)+1, (s3)] - [(s2)+1, (s2)] → [(d)+1, (d)]	Page 749

■Zone control of 16-bit/32-bit binary data

Instruction symbol	Description	Reference
ZONE ZONEP ZONE_U ZONEP_U	When (s3) = 0: 0 → (d) When (s3) > 0: (s3) + (s2) → (d) When (s3) < 0: (s3) + (s1) → (d)	Page 752
DZONE DZONEP DZONE_U DZONEP_U	When [(s3)+1, (s3)] = 0: 0 → [(d)+1, (d)] When [(s3)+1, (s3)] > 0: [(s3)+1, (s3)] + [(s2)+1, (s2)] → [(d)+1, (d)] When [(s3)+1, (s3)] < 0: [(s3)+1, (s3)] + [(s1)+1, (s1)] → [(d)+1, (d)]	Page 754

■Scaling 16-bit/32-bit binary data (point coordinates)

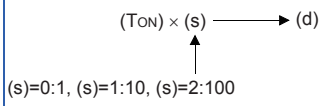
Instruction symbol	Description	Reference
SCL SCLP SCL_U SCLP_U	Executes scaling using the scaling conversion data (16-bit data units) specified by (s2) for the input value specified by (s1), and then stores the result in the device specified by (d). The scaling conversion is executed based on the scaling conversion data stored in the device specified by (s2) onwards.	Page 756
DSCL DSCLP DSCL_U DSCLP_U	Executes scaling using the scaling conversion data (32-bit data units) specified by (s2) for the input value specified by (s1), and then stores the result in the device specified by (d). The scaling conversion is executed based on the scaling conversion data stored in the device specified by (s2) onwards.	Page 759

■Scaling 16-bit/32-bit binary data (XY coordinates)

Instruction symbol	Description	Reference
SCL2 SCL2P SCL2_U SCL2P_U	Executes scaling using the scaling conversion data (16-bit data units) specified by (s2) for the input value specified by (s1), and then stores the result in the device specified by (d). The scaling conversion is executed based on the scaling conversion data stored in the device specified by (s2) onwards.	Page 762
DSCL2 DSCL2P DSCL2_U DSCL2P_U	Executes scaling using the scaling conversion data (32-bit data units) specified by (s2) for the input value specified by (s1), and then stores the result in the device specified by (d). The scaling conversion is executed based on the scaling conversion data stored in the device specified by (s2) onwards.	Page 766

Special timer instruction

■Teaching timer

Instruction symbol	Description	Reference
TTMR	 (s)=0:1, (s)=1:10, (s)=2:100 T_{ON}: On time of TTMR	Page 769

■Special function timer

Instruction symbol	Description	Reference
STMR	The 4 points from the bit device specified by (d) operate as shown below, depending on the ON/OFF status of the input conditions for the STMR instruction: (d)+0: Off delay timer output (d)+1: One shot after off timer output (d)+2: One shot after on timer output (d)+3: On delay and off delay timer output	Page 772

Special counter instruction

■Signed 32-bit bi-directional counters

Instruction symbol	Description	Reference
UDCNTF	This instruction increments the current value of the counter specified by (d) by 1 when the operation result up to UDCNTF instruction changes from OFF to ON, and when the counter reaches the end of its count, NO contact becomes turns ON and NC contact becomes turns OFF. When the long counter specified by (d) is a high-speed counter, up-counting and down-counting are enabled.	Page 774

Shortcut control instruction

■Rotary table shortest direction control

Instruction symbol	Description	Reference
ROTC	Rotates a rotary table with (n1) divisions from the stop position to the position specified by (s)+1 in the shortest direction.	Page 776

Ramp signal instruction

■Ramp signal

Instruction symbol	Description	Reference
RAMPF	Shifts the value from the one specified by (s1) to the one specified by (s2) in (n) scans. The current value is stored in the device specified by (d1)+0.	Page 779

Pulse related instruction

■Measuring the density of 16 bit binary/32 bit binary pulses

Instruction symbol	Description	Reference
SPD	Counts the pulse input from the device specified by (s1) for the duration of time specified by (s2), and stores the count in the device specified by (d).	Page 782
DSPD		Page 788

■16 bit binary/32 bit binary pulse output

Instruction symbol	Description	Reference
PLSY	• When an FX3 compatible operand is specified This instruction outputs a pulse at a frequency specified by (s) for the number of times specified by (n) from the output number (Y) specified by (d). • When an FX5 compatible operand is specified This instruction outputs a pulse at a frequency specified by (s) for the number of times specified by (n), from the output number (axis number) specified by (d).	Page 793
DPLSY		Page 801

■16 bit binary/32 bit binary pulse width modulation

Instruction symbol	Description	Reference
PWM	Outputs the pulse of the cycle specified by (s2), for the ON time on specified by (s1), to the output number specified by (d).	Page 809
DPWM		Page 816

Input matrix instruction

■Input matrix

Instruction symbol	Description	Reference
MTR	Reads matrix input as 8-point input × "n"-point output (transistor) in the time division method.	Page 823

Initial State

■Initial State

Instruction symbol	Description	Reference
IST	Automatically controls the initial state and special relays in a step ladder program.	Page 827

Drum sequence

■16-bit binary data absolute method

Instruction symbol	Description	Reference
ABSD	Creates many output patterns corresponding to the current value of a counter.	Page 838

■32-bit binary data absolute method

Instruction symbol	Description	Reference
DABSD	Creates many output patterns corresponding to the current value of a counter.	Page 840

■Relative method

Instruction symbol	Description	Reference
INCD	This instruction compares the current value of a counter with the data table having (n) lines starting from (s1) (which occupies (n) lines × 1 device). If the counter value is equivalent to the table data, the current output is reset, and the ON/OFF status of the specified sequential outputs is controlled.	Page 842

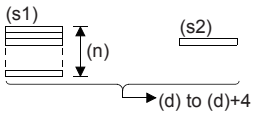
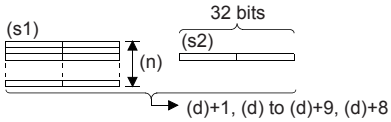
Check code

■Check code

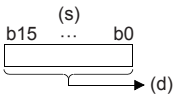
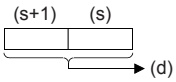
Instruction symbol	Description	Reference
CCD	This instruction calculates the sum data and horizontal parity value of data stored in (s) to (s)+(n)-1. The sum data is stored in (d), and the horizontal parity value is stored in (d)+1.	Page 845
CCDP		

Data operation instruction

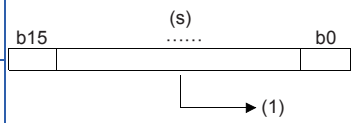
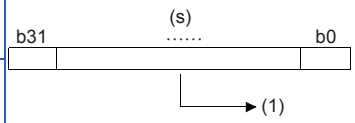
■Searching 16-bit/32-bit data

Instruction symbol	Description	Reference
SERMM SERMMP	Searches for data same as (s2) in (s1).  (d) to (d)+4: Search result	Page 848
DSERMM DSERMMP	Searches for data same as (s2) in (s1).  (d)+1, (d) to (d)+9, (d)+8: Search result	Page 850

■Bit check of 16-bit/32-bit data

Instruction symbol	Description	Reference
SUM SUMP	 (d): Total number of 1s	Page 852
DSUM DSUMP	 (d): Total number of 1s	Page 854

■Bit judgment of 16-bit data/32-bit data

Instruction symbol	Description	Reference
BON BONP	 (1): b(n) ON → (d)=On b(n) OFF → (d)=Off	Page 855
DBON DBONP	 (1): b(n) ON → (d)=On b(n) OFF → (d)=Off	Page 857

■Searching the maximum value of 16-bit/32-bit data

Instruction symbol	Description	Reference
MAX MAXP MAX_U MAXP_U	This instruction searches the data of (n) points from the device specified by (s) in 16-bit units, and stores the maximum value in the device specified by (d).	Page 859
DMAX DMAXP DMAX_U DMAXP_U	This instruction searches the data of (n) points from the device specified by (s) in 32-bit units, and stores the maximum value in the device specified by (d).	Page 861

■Searching the minimum value of 16-bit/32-bit data

Instruction symbol	Description	Reference
MIN	This instruction searches the data of (n) points from the device specified by (s) in 16-bit units, and stores the minimum value in the device specified by (d).	Page 863
MINP		
MIN_U		
MINP_U		
DMIN	This instruction searches the data of (n) points from the device specified by (s) in 32-bit units, and stores the minimum value in the device specified by (d).	Page 865
DMINP		
DMIN_U		
DMINP_U		

■Sorting 16-bit data

Instruction symbol	Description	Reference
SORTTBL	In the data table (sorting source) having ((n1)×(n2)) points specified by (s), sorts the data lines in the ascending order based on the group data in the column number (n3), and stores the result in the data table (sorting result) having ((n1)×(n2)) points specified by (d).	Page 867
SORTTBL_U		

■Sorting 16-bit/32-bit data 2

Instruction symbol	Description	Reference
SORTTBL2	In the data table (sorting source) of 16-bit binary data having (n1×n2) points specified by (s), sorts the data lines in the ascending order based on the group data in the column number (n3), and stores the result in the data table (sorting result) of 16-bit binary data having ((n1)×(n2)) points specified by (d).	Page 870
SORTTBL2_U		
DSORTTBL2	In the data table (sorting source) of 32-bit binary data having (n1×n2) points specified by (s), sorts the data lines in the ascending order based on the group data in the column number (n3), and stores the result in the data table (sorting result) of 32-bit binary data having ((n1)×(n2)) points specified by (d).	Page 873
DSORTTBL2_U		

■Adding 16-bit data

Instruction symbol	Description	Reference
WSUM	These instructions add the (n) points of 16-bit binary data in the device starting from the one specified by (s), and store the result in the device specified by (d).	Page 876
WSUM_U		
WSUMP		
WSUMP_U		

■Adding 32-bit data

Instruction symbol	Description	Reference
DWSUM	These instructions add the (n) points of 32-bit binary data in the device starting from the one specified by (s), and store the result in the device specified by (d).	Page 878
DWSUM_U		
DWSUMP		
DWSUMP_U		

■Calculating the mean value of 16-bit/32-bit data

Instruction symbol	Description	Reference
MEAN	These instructions calculate the mean value of (n) points (16-bit binary data) in the devices starting from the one specified by (s), and store the result in the device specified by (d).	Page 880
MEANP		
MEAN_U		
MEANP_U		
DMEAN	These instructions calculate the mean value of (n) points (32-bit binary data) in the devices starting from the one specified by (s), and store the result in the device specified by (d).	Page 882
DMEANP		
DMEAN_U		
DMEANP_U		

■Calculating the square root of 16-bit/32-bit data

Instruction symbol	Description	Reference
SQRT SQRTP	$\sqrt{(s)} \rightarrow (d)$	Page 884
DSQRT DSQRTP	$\sqrt{(s)+1,(s)} \rightarrow (d)+1,(d)$	Page 886

■CRC calculation

Instruction symbol	Description	Reference
CRC CRCP	This instruction generates a CRC value for (n) 8-bit data (unit: byte) starting from the device specified by (s), and stores the CRC value to (d).	Page 887

Indirect address read instruction

■Reading the indirect address

Instruction symbol	Description	Reference
ADRSET ADRSETP	(s) → (d) (1) (2) (1): Indirect address of the specified device (2): Device name	Page 890

Clock instruction

■Reading clock data

Instruction symbol	Description		Reference
TRD TRDP	(1) → (d)+0 to (d)+6	(d)+0: Year (d)+1: Month (d)+2: Day (d)+3: Hour (d)+4: Minute (d)+5: Seconds (d)+6: Day of week (1): Clock element	Page 892

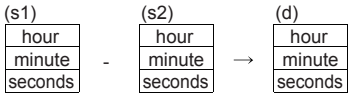
■Writing clock data

Instruction symbol	Description		Reference
TWR TWRP	(d)+0 to (d)+6 → (1)	(d)+0: Year (d)+1: Month (d)+2: Day (d)+3: Hour (d)+4: Minute (d)+5: Seconds (d)+6: Day of week (1): (Clock element)	Page 894

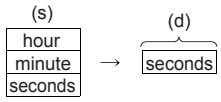
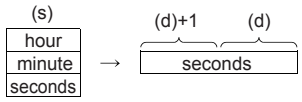
■Adding clock data

Instruction symbol	Description	Reference
TADD TADDP	(s1) hour / minute / seconds + (s2) hour / minute / seconds → (d) hour / minute / seconds	Page 897

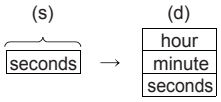
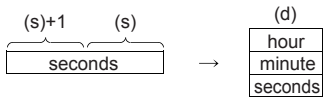
■Subtracting clock data

Instruction symbol	Description	Reference
TSUB TSUBP		Page 900

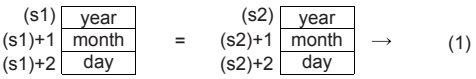
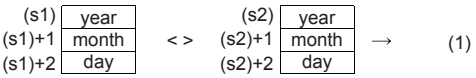
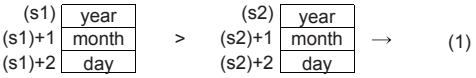
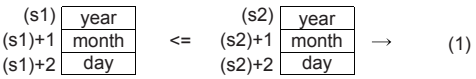
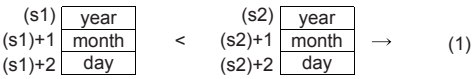
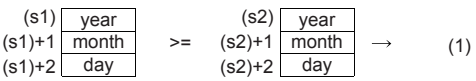
■Converting time data from hour/minute/second to seconds in 16 bits/32 bits

Instruction symbol	Description	Reference
HTOS HTOSP		Page 903
DHTOS DHTOSP		Page 905

■Converting time data from seconds to hour/minute/second in 16 bits/32 bits

Instruction symbol	Description	Reference
STOH STOHP		Page 907
DSTOH DSTOHP		Page 909

■Comparing date data

Instruction symbol	Description	Reference
LDDT=, ANDDT=, ORDT=	 (1): Result	Page 911
LDDT<>, ANDDT<>, ORDT<>	 (1): Result	
LDDT>, ANDDT>, ORDT>	 (1): Result	
LDDT<=, ANDDT<=, ORDT<=	 (1): Result	
LDDT<, ANDDT<, ORDT<	 (1): Result	
LDDT>=, ANDDT>=, ORDT>=	 (1): Result	

■Comparing time data

Instruction symbol	Description	Reference
LDTM=, ANDTM=, ORTM=	(s1) hour / (s1)+1 minute / (s1)+2 seconds = (s2) hour / (s2)+1 minute / (s2)+2 seconds → (1) (1): Result	Page 914
LDTM<>, ANDTM<>, ORTM<>	(s1) hour / (s1)+1 minute / (s1)+2 seconds < > (s2) hour / (s2)+1 minute / (s2)+2 seconds → (1) (1): Result	
LDTM>, ANDTM>, ORTM>	(s1) hour / (s1)+1 minute / (s1)+2 seconds > (s2) hour / (s2)+1 minute / (s2)+2 seconds → (1) (1): Result	
LDTM<=, ANDTM<=, ORTM<=	(s1) hour / (s1)+1 minute / (s1)+2 seconds <= (s2) hour / (s2)+1 minute / (s2)+2 seconds → (1) (1): Result	
LDTM<, ANDTM<, ORTM<	(s1) hour / (s1)+1 minute / (s1)+2 seconds < (s2) hour / (s2)+1 minute / (s2)+2 seconds → (1) (1): Result	
LDTM>=, ANDTM>=, ORTM>=	(s1) hour / (s1)+1 minute / (s1)+2 seconds >= (s2) hour / (s2)+1 minute / (s2)+2 seconds → (1) (1): Result	

■Comparing clock data

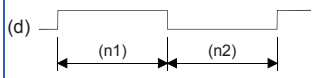
Instruction symbol	Description	Reference
TCMP TCMPP	(s1) hour / (s2) minute / (s3) seconds > (s4) hour / (s4)+1 minute / (s4)+2 seconds ⇨ (d) = ON (s1) hour / (s2) minute / (s3) seconds = (s4) hour / (s4)+1 minute / (s4)+2 seconds ⇨ (d)+1 = ON (s1) hour / (s2) minute / (s3) seconds < (s4) hour / (s4)+1 minute / (s4)+2 seconds ⇨ (d)+2 = ON	Page 917

■Comparing clock data zones

Instruction symbol	Description	Reference
TZCP	(s1) hour / (s1)+1 minute / (s1)+2 seconds > (s3) hour / (s3)+1 minute / (s3)+2 seconds ⇨ (d) = ON	Page 920
TZCPP	(s1) hour / (s1)+1 minute / (s1)+2 seconds ≤ (s3) hour / (s3)+1 minute / (s3)+2 seconds ≤ (s2) hour / (s2)+1 minute / (s2)+2 seconds ⇨ (d)+1 = ON (s3) hour / (s3)+1 minute / (s3)+2 seconds > (s2) hour / (s2)+1 minute / (s2)+2 seconds ⇨ (d)+2 = ON	

Timing check instruction

■Generating timing pulses

Instruction symbol	Description	Reference
DUTY	 (n1): (n1) scans (n2): (n2) scans (d): SM420 to SM424, SM2330 to SM2334	Page 923

■Hour meter

Instruction symbol	Description	Reference
HOURM	This instruction adds the time during which the input contact is ON in units of 1 hour, turns ON the device specified by (d2) when the total ON time exceeds the time specified by (s) (16-bit binary data), and stores the current value in units of 1 hour (16-bit binary data) to (d1), and the current value that is less than one hour (16-bit binary data) to (d1)+1 in units of seconds.	Page 926
DHOURM	This instruction adds the time during which the input contact is ON in units of 1 hour, turns ON the device specified by (d2) when the total ON time exceeds the time specified by (s) (32-bit binary data), and stores the current value in units of 1 hour (32-bit binary data) to (d1)+1 and (d1), and the current value that is less than one hour (16-bit binary data) to (d1)+2 in units of seconds.	Page 928

Module access instruction

■Performing I/O refresh

Instruction symbol	Description	Reference
REF	This instruction refreshes the relevant I/O area during a scan.	Page 930
REFP		
RFS		
RFSP		

■Reading 1-word/2-word data from another module (16-bit specification)

Instruction symbol	Description	Reference
FROM	These instructions read the (n) word data from the buffer memory of the intelligent function module.	Page 932
FROMP		
DFROM	These instructions read the (n)×2 word data from the buffer memory of the intelligent function module.	
DFROMP		

■Writing 1-word/2-word data to another module (16-bit specification)

Instruction symbol	Description	Reference
TO	These instructions write the (n) word data to the buffer memory of the intelligent function module.	Page 936
TOP		
DTO	These instructions write the (n)×2 word data to the buffer memory of the intelligent function module.	
DTOP		

■Reading 1-word/2-word data from another module (32-bit specification)

Instruction symbol	Description	Reference
FROMD	These instructions read the (n) word data from the buffer memory of the intelligent function module.	Page 939
FROMDP		
DFROMD	These instructions read the (n)×2 word data from the buffer memory of the intelligent function module.	
DFROMDP		

■Writing 1-word/2-word data to another module (32-bit specification)

Instruction symbol	Description	Reference
TOD	These instructions write the (n) word data to the buffer memory of the intelligent function module.	Page 942
TODP		
DTOD	These instructions write the (n)×2 word data to the buffer memory of the intelligent function module.	
DTODP		

Logging instructions

■Setting/resetting trigger logging

Instruction symbol	Description	Reference
LOGTRG	Generates a trigger for trigger logging. Data sampled for the number of records (specified in the trigger logging setting parameter using the engineering tool) are stored in the data logging file.	Page 945
LOGTRGR	Resets the trigger condition.	Page 946

Real-time monitor function instruction

Instruction symbol	Description	Reference
RTM	Monitors the data when the instruction is executed by using the real-time monitor function.	Page 947

3.4 Step Ladder Instructions

Starts/Ends step ladder

Instruction symbol	Description	Reference
STL	Start of step ladder	Page 949
RETSTL	End of step ladder	

3.5 PID Control Instruction

PID control loop

Instruction symbol	Description	Reference
PID	This instruction executes PID control which changes the output value according to the input variation.	Page 953

3.6 SFC Program Instructions

SFC control instructions

■Checking the status of a step

Instruction symbol	Description	Reference
LD [S□/BL□\S□]	Outputs the status (active or inactive) of the specified step as the operation result. (Normally open contact instruction)	Page 956
LDI [S□/BL□\S□]	Outputs the status (active or inactive) of the specified step as the operation result. (Normally closed contact instruction)	
AND [S□/BL□\S□]	Performs an AND operation between the status (active or inactive) of the specified step and the previous operation result(s), and output the operation result. (Normally open contact series connection instruction)	
ANI [S□/BL□\S□]	Performs an AND operation between the status (active or inactive) of the specified step and the previous operation result(s), and output the operation result. (Normally closed contact series connection instruction)	
OR [S□/BL□\S□]	Performs an OR operation between the status (active or inactive) of the specified step and the previous operation result(s), and output the operation result. (Single normally open contact parallel connection instruction)	
ORI [S□/BL□\S□]	Performs an OR operation between the status (active or inactive) of the specified step and the previous operation result(s), and output the operation result. (Single normally closed contact parallel connection instruction)	

■Checking the status of a block

Instruction symbol	Description	Reference
LD [BL□]	Outputs the status (active or inactive) of the specified block as the operation result. (Normally open contact instruction)	Page 958
LDI [BL□]	Outputs the status (active or inactive) of the specified block as the operation result. (Normally closed contact instruction)	
AND [BL□]	Performs an AND operation between the status (active or inactive) of the specified block and the previous operation result(s), and output the operation result. (Normally open contact series connection instruction)	
ANI [BL□]	Performs an AND operation between the status (active or inactive) of the specified block and the previous operation result(s), and output the operation result. (Normally closed contact series connection instruction)	
OR [BL□]	Performs an OR operation between the status (active or inactive) of the specified block and the previous operation result(s), and output the operation result. (Single normally open contact parallel connection instruction)	
ORI [BL□]	Performs an OR operation between the status (active or inactive) of the specified block and the previous operation result(s), and output the operation result. (Single normally closed contact parallel connection instruction)	

■Batch-reading the status of steps

Instruction symbol	Description	Reference
MOV [KnS□/BL□\KnS□]	Batch-reads (in units of 16-bit binary data) the status (active or inactive) of steps in the specified block, and stores the read data in the specified device. (Kn: K1 to K4)	Page 960
MOVP [KnS□/BL□\KnS□]		
DMOV [KnS□/BL□\KnS□]	Batch-reads (in units of 32-bit binary data) the status (active or inactive) of steps in the specified block, and stores the read data in the specified device. (Kn: K1 to K8)	Page 963
DMOVP [KnS□/BL□\KnS□]		
BMOV [KnS□/BL□\KnS□]	Batch-reads (in units of the specified number of words starting from the specified step) the status (active or inactive) of steps in the specified block. (Kn: K1 to K4)	Page 966
BMOVP [KnS□/BL□\KnS□]		

■Starting a block

Instruction symbol	Description	Reference
SET [BL□]	Activates the specified block, and executes a step sequence starting from an initial step.	Page 970

■Ending a block

Instruction symbol	Description	Reference
RST [BL□]	Deactivates the specified block.	Page 971

■Pausing a block

Instruction symbol	Description	Reference
PAUSE [BL□]	Temporarily stops the step sequence in the specified block.	Page 972

■Restarting a block

Instruction symbol	Description	Reference
RSTART [BL□]	Releases the temporary stop, and restarts the sequence from the step where the sequence was stopped in the specified block.	Page 974

■Activating a step

Instruction symbol	Description	Reference
SET [S□/BL□\S□]	Activates the specified step.	Page 976

■Deactivating a step

Instruction symbol	Description	Reference
RST [S□/BL□\S□]	Deactivates the specified step.	Page 978

■Activating/deactivating a step

Instruction symbol	Description	Reference
OUT [S□/BL□\S□]	This instruction activates or deactivates the specified step.	Page 980

■Batch-deactivating a step

Instruction symbol	Description	Reference
ZRST [S□/BL□\S□]	Batch-deactivates the steps in the specified range.	Page 982
ZRSTP [S□/BL□\S□]		

SFC dedicated instruction

■Creating a dummy transition condition

Instruction symbol	Description	Reference
TRAN	A dummy output which satisfies a transition condition.	Page 984

4 MODULE SPECIFIC INSTRUCTION

4.1 Network Common Instruction

Link dedicated instruction

■Reading data from another station programmable controller

Instruction symbol	Description	Reference
GP.READ	Reads data from another station PLC device in units of words.	Page 988

■Reading data from another station programmable controller (with notification)

Instruction symbol	Description	Reference
GP.SREAD	Reads data from another station PLC device in units of words. When reading data has finished, a device of the other station is turned on. (This makes it possible for the other station to recognize that data has been read by the GP.SREAD instruction.)	Page 994

■Writing data to another station programmable controller

Instruction symbol	Description	Reference
GP.WRITE	Writes data to another station PLC device in units of words.	Page 1000

■Writing data to another station programmable controller (with notification)

Instruction symbol	Description	Reference
GP.SWRITE	Writes data to another station PLC device in units of words. When writing data has finished, a device in the other station is turned on. (This makes it possible for the other station to recognize that data has been written by the GP.SWRITE instruction.)	Page 1008

■Sending data to another station programmable controller

Instruction symbol	Description	Reference
GP.SEND	Sends data to another station PLC.	Page 1016

■Receiving data from another station programmable controller

Instruction symbol	Description	Reference
GP.RECV	Reads data received from another station PLC. (For the main routine program)	Page 1024

4.2 Ethernet Instruction

Built-in Ethernet function instruction

■Opening a connection

Instruction symbol	Description	Reference
SP.SOCOPEN	This instruction opens the connection specified by (s1).	Page 1030

■Closing a connection

Instruction symbol	Description	Reference
SP.SOCCLOSE	This instruction closes the connection specified by (s1). (Closing a connection)	Page 1033

Socket communication function instruction

■Reading receive data during the END processing

Instruction symbol	Description	Reference
SP.SOCRCV	This instruction reads the received data of the connection specified by (s1) from the socket communication receive data area, during the END processing.	Page 1035

■Sending data

Instruction symbol	Description	Reference
SP.SOCSND	This instruction sends the data set in (s3) to the target device of the connection specified by (s1).	Page 1038

■Reading connection information

Instruction symbol	Description	Reference
SP.SOCCINF	This instruction reads the connection information of the connection specified by (s1).	Page 1041

■Reading socket communication receive data

Instruction symbol	Description	Reference
S.SOCRDATA	This instruction reads the data of the number of words specified in (n) from the socket communication receive data area of the connection specified by (s1), and stores it to the device specified by (d) onwards.	Page 1043
SP.SOCRDATA		

Predefined Protocol Support Function Instruction

■Executing the protocols registered for the predefined protocol support function

Instruction symbol	Description	Reference
SP.ECPRTCL	Executes the protocol specified by the communication protocol support tool of the engineering tool.	Page 1045

SLMP frame send instruction

■Sending the SLMP frame

Instruction symbol	Description	Reference
SP.SLMPSND	Sends SLMP messages to an SLMP-compatible device.	Page 1049

File transfer function instruction

■Sending FTP client files

Instruction symbol	Description	Reference
SP.FTPPUT	Sends the CPU module file specified in (s2) to the FTP server folder specified in (s3).	Page 1054

■Retrieving FTP client files

Instruction symbol	Processing details	Reference
SP.FTPGET	Retrieves files on the FTP server, which are specified by (s2), to the folder path of the CPU module, which is specified by (s3).	Page 1059

Ethernet module

■Opening a connection

Instruction symbol	Description	Reference
GP.OPEN	This instruction establishes (opens) a connection with an external device for data communication.	Page 1064

■Closing a connection

Instruction symbol	Description	Reference
GP.CLOSE	This instruction disconnects (closes) the connection from the external device during data communication.	Page 1067

■Reading receive data

Instruction symbol	Description	Reference
GP.SOCRCV	This instruction reads receive data from the external device through socket communications.	Page 1069

■Sending data

Instruction symbol	Description	Reference
GP.SOCSND	This instruction sends data to the external device through socket communications	Page 1071

4.3 CC-Link IE TSN Instructions

Own station number/IP address setting

Instruction symbol	Description	Reference
G.UINI	Sets the station number and IP address for FX5-CCLGN-MS (own station).	Page 1073
GP.UINI		

Sending an SLMP Frame

Instruction symbol	Description	Reference
G.SLMPSND	Sends an SLMP message to the SLMP-compatible device on the same network.	Page 1076
GP.SLMPSND		

4.4 CC-Link IE Field Network Instruction

Setting parameters

Instruction symbol	Description	Reference
G.CCPASET	Sets parameters in the FX5-CCLIEF (own station). Use the G(P).CCPASET instruction to set parameters in the following cases. • If you want to change parameters without resetting the CPU module • If you want to mount more intelligent device station modules than the number of modules set by the engineering tool	Page 1085
GP.CCPASET		

Setting the station number to own station

Instruction symbol	Description	Reference
G.UINI	Sets the station number in the FX5-CCLIEF (own station).	Page 1088
GP.UINI		

4.5 High-speed Counter Instruction

High-speed processing instruction

■Setting 32-bit data comparison

Instruction symbol	Description	Reference
DHSCS	Turns ON the bit device of (d) when the current value of the high-speed counter of CH specified by (s2) is changed to the value specified by (s1).	Page 1091

■Reset 32-bit data comparison

Instruction symbol	Description	Reference
DHSCR	Turns OFF the bit device of (d) when the current value of the high-speed counter of CH specified by (s2) is changed to the value specified by (s1).	Page 1094

■Comparison of 32-bit data band

Instruction symbol	Description	Reference
DHSZ	Compares whether the current value of the high-speed counter is within or outside the value range specified by (s1) or (s2).	Page 1097

■Start/stop of the 16-bit/32-bit data high-speed I/O function

Instruction symbol	Description	Reference
HIOEN	Start or stop high-speed I/O for the specified CH.	Page 1100
HIOENP		
DHIOEN		Page 1104
DHIOENP		

High-speed current value transfer instruction

■High-speed current value transfer of 16-bit/32-bit data

Instruction symbol	Description	Reference
HCMOV	Transfers the current value of the high-speed I/O.	Page 1108
HCMOVP		
DHCMOV		Page 1110
DHCMOVP		

4.6 External Device Communication Instruction

Serial communication 2

Instruction symbol	Description	Reference
RS2	Sends/receives data by non-protocol communication.	Page 1113

Inverter Communication Instruction

■Inverter operation monitoring (Status check)

Instruction symbol	Description	Reference
IVCK	Reads the contents of the corresponding instruction code from the specified inverter station number.	Page 1115

■Inverter operations control (Drive)

Instruction symbol	Description	Reference
IVDR	Writes the contents of the corresponding instruction code to the specified inverter station number.	Page 1117

■Inverter parameter read

Instruction symbol	Description	Reference
IVRD	Reads a parameter from the specified inverter station number.	Page 1119

■Inverter parameter write

Instruction symbol	Description	Reference
IVWR	Writes a parameter to the specified inverter station number.	Page 1121

■Inverter parameter block write

Instruction symbol	Description	Reference
IVBWR	Writes the range of the specified data tables to the specified inverter station number in batch.	Page 1123

■Inverter multi command

Instruction symbol	Description	Reference
IVMC	Sends/receives data corresponding to the send/receive data type to/from the specified inverter station number.	Page 1125

MODBUS Communication Instruction

Instruction symbol	Description	Reference
ADPRW	Sends the function code from the master to the slave of the MODBUS serial communication and reads or writes the data.	Page 1127

Predefined Protocol Support Function Instruction

Instruction symbol	Description	Reference
S.CPRTCL	Executes the protocol specified by the communication protocol support tool of the engineering tool.	Page 1129
SP.CPRTCL		

4.7 Positioning Instruction

Positioning instruction

■Zero return(OPR) with 16-bit/32- bit data DOG search

Instruction symbol	Description	Reference
DSZR	• When FX3 compatible operand is specified Specifies the proximity dog signal, zero signal and device (Y). Outputs a pulse with the specified device (Y) to perform the zero return operation. • When FX5 operand is specified Specifies the original position return speed, creep speed and axis number. Outputs a pulse with the specified axis to perform the zero return operation.	Page 1134
DDSZR		Page 1138

■16-bit/32-bit data interrupt positioning

Instruction symbol	Description	Reference
DVIT	• When FX3 compatible operand is specified Performs interrupt positioning with the specified travel distance, speed, and device (Y). • When FX5 operand is specified Performs interrupt positioning with the specified travel distance, speed, and axis number.	Page 1140
DDVIT		Page 1144

■Positioning by one table operation

Instruction symbol	Description	Reference
TBL	• When FX3 compatible operand is specified Outputs 1 table operation from the table set by the parameter as pulse with specified device (Y). • When FX5 operand is specified Outputs 1 table operation from the table set by the parameter as pulse with specified axis number.	Page 1148

■Positioning by multiple table operation

Instruction symbol	Description	Reference
DRVTBL	Outputs continuous multiple table operations from the table set by the parameter as pulse with specified axis number.	Page 1150

■Multiple axes concurrent drive positioning

Instruction symbol	Description	Reference
DRVMUL	Outputs the table set by the parameter as pulse with specified multiple axes.	Page 1152

■32-bit data ABS current value read

Instruction symbol	Description	Reference
DABS	Reads the absolute position data of the servo amplifier.	Page 1154

■16-bit/32-bit data variable speed pulse

Instruction symbol	Description	Reference
PLSV	• When FX3 compatible operand is specified Specifies the command speed and output device (Y) and uses the specified device (Y) to perform pulse output. • When FX5 operand is specified Specifies the command speed and performs pulse output with the specified axis number.	Page 1156
DPLSV		Page 1160

■16-bit/32-bit data relative positioning

Instruction symbol	Description	Reference
DRVI	• When FX3 compatible operand is specified Specifies the travel distance from the current position, speed and performs pulse output with the specified device (Y). • When FX5 operand is specified Specifies the travel distance from the current position, speed and performs pulse output with the specified axis number.	Page 1164
DDRVI		Page 1168

■16-bit/32-bit data absolute positioning

Instruction symbol	Description	Reference
DRVA	• When FX3 compatible operand is specified Specifies the travel distance from the reference position, speed and performs pulse output with the specified device (Y). • When FX5 operand is specified Specifies the travel distance from the reference position, speed and performs pulse output with the specified axis number.	Page 1172
DDRVA		Page 1176

Positioning module

■Restoring the absolute position

Instruction symbol	Description	Reference
G.ABRST1	Restores the absolute position of specified axis.	Page 1180
G.ABRST2		

■Starting the positioning

Instruction symbol	Description	Reference
GP.PSTRT1	Starts positioning of the specified axis.	Page 1184
GP.PSTRT2		

■Teaching

Instruction symbol	Description	Reference
GP.TEACH1	Performs teaching for the specified axis.	Page 1187
GP.TEACH2		

■Backing up module data (writing data to the flash ROM)

Instruction symbol	Description	Reference
GP.PFWRT	Writes the positioning data and block start data in the buffer memory to the flash ROM.	Page 1190

■Initializing the module

Instruction symbol	Description	Reference
GP.PINIT	Initializes the setting data in the buffer memory and flash ROM.	Page 1193

4.8 BFM Device Read/ Write Instruction

Divided BFM Read

Instruction symbol	Description	Reference
RBFM	Divides and reads data from the continuous buffer memory in the intelligent module. (This instruction cannot be used with the FX5 intelligent module.)	Page 1196

Divided BFM Write

Instruction symbol	Description	Reference
WBFM	Divides and writes data to the continuous buffer memory in the intelligent module. (This instruction cannot be used with the FX5 intelligent module.)	Page 1200

5 STANDARD FUNCTIONS/FUNCTION BLOCKS

5.1 Standard Functions

Type conversion functions

Converting BOOL to WORD/DWORD

Function symbol	Description	Reference
BOOL_TO_WORD	Converts BOOL type data to WORD type data.	Page 1204
BOOL_TO_WORD_E		
BOOL_TO_DWORD	Converts BOOL type data to DWORD type data.	Page 1206
BOOL_TO_DWORD_E		

Converting BOOL to INT/DINT

Function symbol	Description	Reference
BOOL_TO_INT	Converts BOOL type data to INT type data.	Page 1207
BOOL_TO_INT_E		
BOOL_TO_DINT	Converts BOOL type data to DINT type data.	Page 1208
BOOL_TO_DINT_E		

Converting BOOL to TIME

Function symbol	Description	Reference
BOOL_TO_TIME	Converts BOOL type data to TIME type data.	Page 1209
BOOL_TO_TIME_E		

Converting BOOL to STRING

Function symbol	Description	Reference
BOOL_TO_STRING	Converts BOOL type data to STRING type data.	Page 1210
BOOL_TO_STRING_E		

Converting WORD to BOOL

Function symbol	Description	Reference
WORD_TO_BOOL	Converts WORD type data to BOOL type data.	Page 1211
WORD_TO_BOOL_E		

Converting WORD to DWORD

Function symbol	Description	Reference
WORD_TO_DWORD	Converts WORD type data to DWORD type data.	Page 1212
WORD_TO_DWORD_E		

Converting WORD to INT/DINT

Function symbol	Description	Reference
WORD_TO_INT	Converts WORD type data to INT type data.	Page 1213
WORD_TO_INT_E		
WORD_TO_DINT	Converts WORD type data to DINT type data.	Page 1214
WORD_TO_DINT_E		

Converting WORD to TIME

Function symbol	Description	Reference
WORD_TO_TIME	Converts WORD type data to TIME type data.	Page 1216
WORD_TO_TIME_E		

Converting DWORD to BOOL

Function symbol	Description	Reference
DWORD_TO_BOOL	Converts DWORD type data to BOOL type data.	Page 1217
DWORD_TO_BOOL_E		

Converting DWORD to WORD

Function symbol	Description	Reference
DWORD_TO_WORD	Converts DWORD type data to WORD type data.	Page 1218
DWORD_TO_WORD_E		

Converting DWORD to INT/DINT

Function symbol	Description	Reference
DWORD_TO_INT	Converts DWORD type data to INT type data.	Page 1220
DWORD_TO_INT_E		
DWORD_TO_DINT	Converts DWORD type data to DINT type data.	Page 1222
DWORD_TO_DINT_E		

Converting DWORD to TIME

Function symbol	Description	Reference
DWORD_TO_TIME	Converts DWORD type data to TIME type data.	Page 1223
DWORD_TO_TIME_E		

Converting INT to BOOL

Function symbol	Description	Reference
INT_TO_BOOL	Converts INT type data to BOOL type data.	Page 1224
INT_TO_BOOL_E		

Converting INT to WORD/DWORD

Function symbol	Description	Reference
INT_TO_WORD	Converts INT type data to WORD type data.	Page 1225
INT_TO_WORD_E		
INT_TO_DWORD	Converts INT type data to DWORD type data.	Page 1226
INT_TO_DWORD_E		

Converting INT to DINT

Function symbol	Description	Reference
INT_TO_DINT	Converts INT type data to DINT type data.	Page 1228
INT_TO_DINT_E		

Converting INT to BCD

Function symbol	Description	Reference
INT_TO_BCD	Converts INT type data to BCD type data.	Page 1229
INT_TO_BCD_E		

Converting INT to REAL

Function symbol	Description	Reference
INT_TO_REAL	Converts INT type data to REAL type data.	Page 1231
INT_TO_REAL_E		

Converting INT to TIME

Function symbol	Description	Reference
INT_TO_TIME	Converts INT type data to TIME type data.	Page 1232
INT_TO_TIME_E		

Converting INT to STRING

Function symbol	Description	Reference
INT_TO_STRING	Converts INT type data to STRING type data.	Page 1233
INT_TO_STRING_E		

Converting DINT to BOOL

Function symbol	Description	Reference
DINT_TO_BOOL	Converts DINT type data to BOOL type data.	Page 1235
DINT_TO_BOOL_E		

Converting DINT to WORD/DWORD

Function symbol	Description	Reference
DINT_TO_WORD	Converts DINT type data to WORD type data.	Page 1236
DINT_TO_WORD_E		
DINT_TO_DWORD	Converts DINT type data to DWORD type data.	Page 1238
DINT_TO_DWORD_E		

Converting DINT to INT

Function symbol	Description	Reference
DINT_TO_INT	Converts DINT type data to INT type data.	Page 1239
DINT_TO_INT_E		

Converting DINT to BCD

Function symbol	Description	Reference
DINT_TO_BCD	Converts DINT type data to BCD type data.	Page 1241
DINT_TO_BCD_E		

Converting DINT to REAL

Function symbol	Description	Reference
DINT_TO_REAL	Converts DINT type data to REAL type data.	Page 1243
DINT_TO_REAL_E		

Converting DINT to TIME

Function symbol	Description	Reference
DINT_TO_TIME	Converts DINT type data to TIME type data.	Page 1244
DINT_TO_TIME_E		

Converting DINT to STRING

Function symbol	Description	Reference
DINT_TO_STRING	Converts DINT type data to STRING type data.	Page 1245
DINT_TO_STRING_E		

Converting BCD to INT/DINT

Function symbol	Description	Reference
BCD_TO_INT	Converts BCD type data to INT type data.	Page 1247
BCD_TO_INT_E		
BCD_TO_DINT	Converts BCD type data to DINT type data.	Page 1249
BCD_TO_DINT_E		

Converting REAL to INT/DINT

Function symbol	Description	Reference
REAL_TO_INT	Converts REAL type data to INT type data.	Page 1251
REAL_TO_INT_E		
REAL_TO_DINT	Converts REAL type data to DINT type data.	Page 1253
REAL_TO_DINT_E		

Converting REAL to STRING

Function symbol	Description	Reference
REAL_TO_STRING	Converts REAL type data to STRING type data (exponent format).	Page 1255
REAL_TO_STRING_E		

Converting TIME to BOOL

Function symbol	Description	Reference
TIME_TO_BOOL	Converts TIME type data to BOOL type data.	Page 1258
TIME_TO_BOOL_E		

Converting TIME to WORD/DWORD

Function symbol	Description	Reference
TIME_TO_WORD	Converts TIME type data to WORD type data.	Page 1259
TIME_TO_WORD_E		
TIME_TO_DWORD	Converts TIME type data to DWORD type data.	Page 1260
TIME_TO_DWORD_E		

Converting TIME to INT/DINT

Function symbol	Description	Reference
TIME_TO_INT	Converts TIME type data to INT type data.	Page 1261
TIME_TO_INT_E		
TIME_TO_DINT	Converts TIME type data to DINT type data.	Page 1262
TIME_TO_DINT_E		

Converting TIME to STRING

Function symbol	Description	Reference
TIME_TO_STRING	Converts TIME type data to STRING type data.	Page 1263
TIME_TO_STRING_E		

Converting STRING to BOOL

Function symbol	Description	Reference
STRING_TO_ BOOL	Converts STRING type data to BOOL type data.	Page 1265
STRING_TO_ BOOL_E		

Converting STRING to INT/DINT

Function symbol	Description	Reference
STRING_TO_INT	Converts STRING type data to INT type data.	Page 1266
STRING_TO_INT_E		
STRING_TO_DINT	Converts STRING type data to DINT type data.	Page 1268
STRING_TO_DINT_E		

Converting STRING to REAL

Function symbol	Description	Reference
STRING_TO_REAL	Converts STRING type data to REAL type data.	Page 1270
STRING_TO_REAL_E		

Converting STRING to TIME

Function symbol	Description	Reference
STRING_TO_TIME	Converts STRING type data to TIME type data.	Page 1273
STRING_TO_TIME_E		

Converting bit array to INT/DINT

Function symbol	Description	Reference
BITARR_TO_INT	Converts a bit array to INT type data for a specified number of bits.	Page 1274
BITARR_TO_INT_E		
BITARR_TO_DINT	Converts a bit array to DINT type data for a specified number of bits.	Page 1275
BITARR_TO_DINT_E		

Converting INT/DINT to bit array

Function symbol	Description	Reference
INT_TO_BITARR	Outputs low-order (n) bits of INT type data to a bit array.	Page 1276
INT_TO_BITARR_E		
DINT_TO_BITARR	Outputs low-order (n) bits of DINT type data to a bit array.	Page 1277
DINT_TO_BITARR_E		

Bit array copy

Function symbol	Description	Reference
CPY_BITARR	Copies specified number of bits of a bit array.	Page 1278
CPY_BITARR_E		

Reading the specified bit of word label

Function symbol	Description	Reference
GET_BIT_OF_INT	Reads a value of a specified bit of INT type data.	Page 1279
GET_BIT_OF_INT_E		

Writing the specified bit of word label

Function symbol	Description	Reference
SET_BIT_OF_INT	Writes a value to a specified bit of INT type data.	Page 1280
SET_BIT_OF_INT_E		

Copying the specified bit of word label

Function symbol	Description	Reference
CPY_BIT_OF_INT	Copies a specified bit of INT type data to a specified bit of another INT type data.	Page 1281
CPY_BIT_OF_INT_E		

Unnecessary of type conversion

Function symbol	Description	Reference
GET_BOOL_ADDR	Converts a data type to the BOOL type.	Page 1282
GET_INT_ADDR	Converts a data type to the INT type.	
GET_WORD_ADDR	Converts a data type to the WORD type.	

Standard functions of one numeric variable

Absolute value

Function symbol	Description	Reference
ABS	Outputs the absolute value of an input value.	Page 1283
ABS_E		

Square root

Function symbol	Description	Reference
SQRT	Outputs the square root of an input value.	Page 1285
SQRT_E		

Natural logarithm operation

Function symbol	Description	Reference
LN	Outputs the natural logarithm operation result of an input value.	Page 1286
LN_E		

Calculating the common logarithm

Function symbol	Description	Reference
LOG	Outputs the operation result of the common logarithm (the logarithm whose base is 10) of an input value.	Page 1287
LOG_E		

Exponential operation

Function symbol	Description	Reference
EXP	Outputs the exponential operation result of an input value.	Page 1289
EXP_E		

Sine operation

Function symbol	Description	Reference
SIN	Outputs the sine of the angle of an input value.	Page 1290
SIN_E		

Cosine operation

Function symbol	Description	Reference
COS	Outputs the cosine of the angle of an input value.	Page 1291
COS_E		

Tangent operation

Function symbol	Description	Reference
TAN	Outputs the tangent of the angle value of an input value.	Page 1292
TAN_E		

Arc sine operation

Function symbol	Description	Reference
ASIN	Outputs the arc sine value of an input value.	Page 1294
ASIN_E		

Arc cosine operation

Function symbol	Description	Reference
ACOS	Outputs the arc cosine value of an input value.	Page 1296
ACOS_E		

Arc tangent operation

Function symbol	Description	Reference
ATAN	Outputs the arc tangent value of an input value.	Page 1298
ATAN_E		

Standard arithmetic functions

Addition

Function symbol	Description	Reference
ADD	Outputs the sum of input values ((s1) + (s2) + ... + (s28)).	Page 1300
ADD_E		

Multiplication

Function symbol	Description	Reference
MUL	Outputs the product of input values ((s1) × (s2) × ... × (s28)).	Page 1302
MUL_E		

Subtraction

Function symbol	Description	Reference
SUB	Outputs the difference of input values ((s1) - (s2)).	Page 1304
SUB_E		

Division

Function symbol	Description	Reference
DIV	Outputs the quotient of input values ((s1) ÷ (s2)).	Page 1306
DIV_E		

Remainder

Function symbol	Description	Reference
MOD	Outputs the remainder of input values ((s1) ÷ (s2)).	Page 1308
MOD_E		

Exponentiation

Function symbol	Description	Reference
EXPT	Outputs the exponentiation of an input value.	Page 1310
EXPT_E		

Move operation

Function symbol	Description	Reference
MOVE	Assigns an input value to (d).	Page 1312
MOVE_E		

Standard bit shift functions

Shifting n-bit data to left/right

Function symbol	Description	Reference
SHL	Shifts an input value leftward by (n) bits and outputs the result.	Page 1314
SHL_E		
SHR	Shifts an input value rightward by (n) bits and outputs the result.	Page 1316
SHR_E		

Rotating n-bit data to left/right

Function symbol	Description	Reference
ROL	Rotates an input value leftward by (n) bits and outputs the result.	Page 1318
ROL_E		
ROR	Rotates an input value rightward by (n) bits and outputs the result.	Page 1320
ROR_E		

Standard bitwise boolean functions

AND operation, OR operation, XOR operation, NOT operation

Function symbol	Description	Reference
AND	Outputs the logical product of input values.	Page 1322
AND_E		
OR	Outputs the logical sum of input values.	
OR_E		
XOR	Outputs the exclusive logical sum of input values.	
XOR_E		
NOT	Outputs the logical negation of input values.	Page 1324
NOT_E		

Standard selection functions

Selection

Function symbol	Description	Reference
SEL	Outputs a selected input value.	Page 1325
SEL_E		

Selecting Maximum/Minimum Value

Function symbol	Description	Reference
MAX	Outputs the maximum value of an input value.	Page 1327
MAX_E		
MIN	Outputs the minimum value of an input value.	
MIN_E		

Limit Control

Function symbol	Description	Reference
LIMIT	Outputs an input value controlled with the upper and lower limits.	Page 1329
LIMIT_E		

Multiplexer

Function symbol	Description	Reference
MUX	Outputs one of multiple input values.	Page 1331
MUX_E		

Standard comparison functions

Compare

Function symbol	Description	Reference
GT	Outputs the data comparison result of input values.	Page 1333
GT_E		
GE		
GE_E		
EQ		
EQ_E		
LE		
LE_E		
LT		
LT_E		
NE		Page 1335
NE_E		

Standard character string functions

Character string length detection

Function symbol	Description	Reference
LEN	Detects the length of an input character string and outputs the result.	Page 1337
LEN_E		

Extracting character string data from the left/right

Function symbol	Description	Reference
LEFT	Outputs specified number of characters from the left of input character string data.	Page 1339
LEFT_E		
RIGHT	Outputs specified number of characters from the right of input character string data.	
RIGHT_E		

Extract mid string

Function symbol	Description	Reference
MID	Outputs specified number of characters from an arbitrary position of an input character string.	Page 1341
MID_E		

String concatenation

Function symbol	Description	Reference
CONCAT	Concatenates character strings and output the result.	Page 1343
CONCAT_E		

Inserting character string

Function symbol	Description	Reference
INSERT	Inserts a character string into another character string and output the result.	Page 1345
INSERT_E		

Deleting character string

Function symbol	Description	Reference
DELETE	Deletes an arbitrary range of a character string and outputs the result.	Page 1347
DELETE_E		

Replacing character string

Function symbol	Description	Reference
REPLACE	Replaces an arbitrary range of a character string and outputs the result.	Page 1349
REPLACE_E		

Searching character string

Function symbol	Description	Reference
FIND	Searches for a character string and outputs the result.	Page 1352
FIND_E		

Time data functions

Addition

Function symbol	Description	Reference
ADD_TIME	Outputs the sum of input values (time data) ((s1) + (s2)).	Page 1354
ADD_TIME_E		

Subtraction

Function symbol	Description	Reference
SUB_TIME	Outputs the difference of input values (time data) ((s1) - (s2)).	Page 1356
SUB_TIME_E		

Multiplication

Function symbol	Description	Reference
MUL_TIME	Outputs the product of input values (time data) ((s1) × (s2)).	Page 1358
MUL_TIME_E		

Division

Function symbol	Description	Reference
DIV_TIME	Outputs the quotient of input values (time data) ((s1) ÷ (s2)).	Page 1360
DIV_TIME_E		

5.2 Standard Function Blocks

Bistable function blocks

Bistable function blocks (set priority)

Function block symbol	Description	Reference
SR	Judges two input values and outputs 1 (TRUE) or 0 (FALSE). (Set priority)	Page 1364
SR_E		

Bistable function blocks (reset priority)

Function block symbol	Description	Reference
RS	Judges two input values and outputs 1 (TRUE) or 0 (FALSE). (Reset priority)	Page 1366
RS_E		

Edge detection function blocks

Rising edge detector

Function block symbol	Description	Reference
R_TRIG	Detects the rising edge of a signal, and outputs a pulse signal.	Page 1368
R_TRIG_E		

Falling edge detector

Function block symbol	Description	Reference
F_TRIG	Detects the falling edge of a signal, and outputs a pulse signal.	Page 1370
F_TRIG_E		

Counter function blocks

Up counter

Function block symbol	Description	Reference
CTU	Counts up the number of times of rising of a signal.	Page 1372
CTU_E		

Down counter

Function block symbol	Description	Reference
CTD	Counts down the number of times of rising of a signal.	Page 1374
CTD_E		

Up/Down counter

Function block symbol	Description	Reference
CTUD	Counts up/down the number of times of rising of a signal.	Page 1376
CTUD_E		

Counter function block

Function block symbol	Description	Reference
COUNTER_FB_M	Counts up the number of times of rising of a signal from (s3) to (s2).	Page 1379

Timer function blocks

Pulse timer

Function block symbol	Description	Reference
TP	Keeps ON a signal for specified duration.	Page 1381
TP_E		
TP_10		
TP_10_E		

On-delay timer

Function block symbol	Description	Reference
TON	Turns ON a signal after a specified time.	Page 1383
TON_E		
TON_10		
TON_10_E		

Off-delay timer

Function block symbol	Description	Reference
TOF	Turns OFF a signal after a specified time.	Page 1385
TOF_E		
TOF_10		
TOF_10_E		

Timer function blocks

Function block symbol	Description	Reference
TIMER_1_FB_M	When the execution condition is established, these function blocks start the timer count to the set time.	Page 1387
TIMER_10_FB_M		
TIMER_100_FB_M		
TIMER_CONT_FB_M		
TIMER_CONTHS_FB_M		

PART 3 CPU MODULE INSTRUCTIONS

This part consists of the following chapters.

6 SEQUENCE INSTRUCTIONS

7 BASIC INSTRUCTIONS

8 APPLICATION INSTRUCTION

9 STEP LADDER INSTRUCTIONS

10 PID CONTROL INSTRUCTION

11 SFC PROGRAM INSTRUCTIONS

6 SEQUENCE INSTRUCTIONS

6.1 Contact Instructions

Operation start, series connection, parallel connection

LD, LDI, AND, ANI, OR, ORI

FX5S FX5UJ FX5U FX5UC

• LD: NO contact operation start instruction/LDI: NC contact operation start instruction

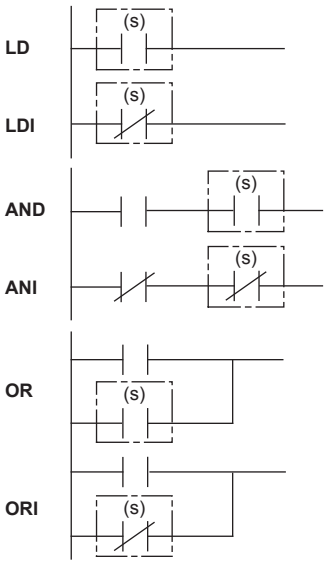
These instructions capture the ON/OFF information of the device specified by (s), and use that as the operation result.

• AND: NO contact series connection instruction/ANI: NC contact series connection instruction

These instructions capture the ON/OFF information of the device specified by (s), AND with the operation result so far, and use the result as the operation result.

• OR: NO contact parallel connection instruction/ORI: NC contact parallel instruction

These instructions capture the ON/OFF information of the device specified by (s), OR with the operation result so far, and use the result as the operation result.

Ladder diagram	Structured text
	This becomes a assignment statement, operator, control syntax, etc. In the ST language, there are sometimes no instructions (symbols) that directly correspond to contacts such as LD, AND, and OR. When programming using assignment statements, express as shown in the following example. Example Y1:=(X0 OR X1) AND X2 AND NOT X3; Y2:=NOT X4 OR NOT X5;

FBD/LD
In the FBD/LD language, contact is used as well as the Ladder diagram.

Setting data

■Descriptions, ranges, and data types

Operand	Remarks	Range	Data type	Data type (label)
(s)	Device used as a contact	—	Bit	ANY_BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DX)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	○	—	—	—	—	—	○

Processing details

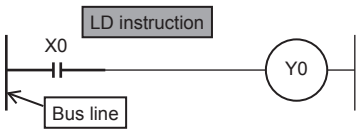
■LD, LDI

- The LD instruction is the NO contact operation start instruction, and the LDI instruction is the NC contact operation start instruction. These instructions capture the ON/OFF information*1 of the specified device, and use the result as the operation result.

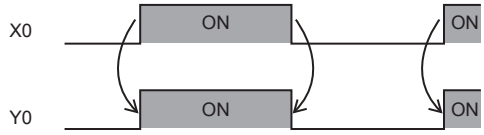
*1 When bits of word devices are specified, devices turn ON/OFF by the 1/0 status of the specified bit.

Ex.

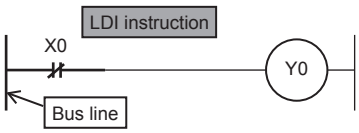
• LD instruction (NO contact operation start)



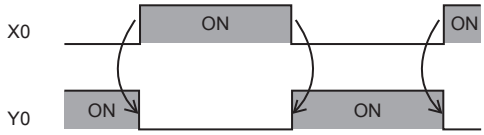
Timing chart



• LDI instruction (NC contact operation start)



Timing chart



■AND, ANI

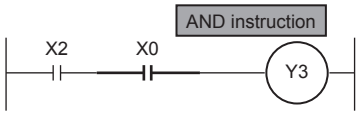
- The AND instruction is NO contact series connection instruction and the ANI instruction is NC contact series connection instruction. These instructions capture the ON/OFF information*1 of the specified bit device, AND with the operation result so far, and use the result as the operation result.

*1 When bits of word devices are specified, devices turn ON/OFF by the 1/0 status of the specified bit.

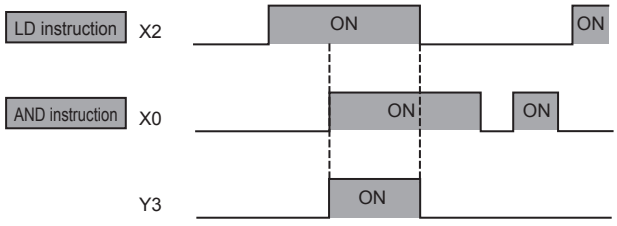
- There is no limitation to the number of series contacts. Any number of contacts can use this instructions consecutively.
- Output to other coils through contacts after the OUT instruction is called cascade output, and these outputs can be repeated any number of times as long as their order is correct.

Ex.

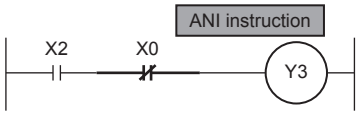
• AND instruction
(series connection of NO contacts)



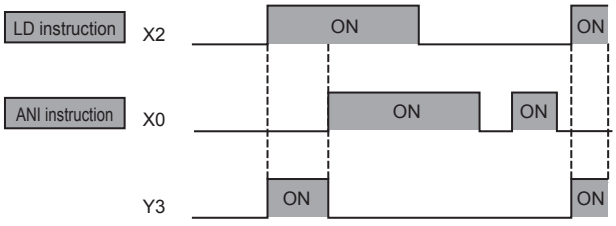
Timing chart



• ANI instruction
(series connection of NC contacts)



Timing chart



6

■OR, ORI

- The OR instruction is NO contact parallel connection and the ORI instruction is NC contact parallel connection. These instructions capture the ON/OFF information*1 of the specified device, OR with the operation result so far, and use the result as the operation result.

*1 When bits of word devices are specified, devices turn ON/OFF by the 1/0 status of the specified bit.

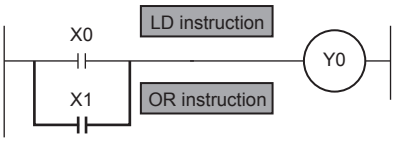
- These instructions are connected in parallel from the step with this instruction to the previous step with the LD and LDI instruction.
- There is no limitation in the number of parallel connections.

Point

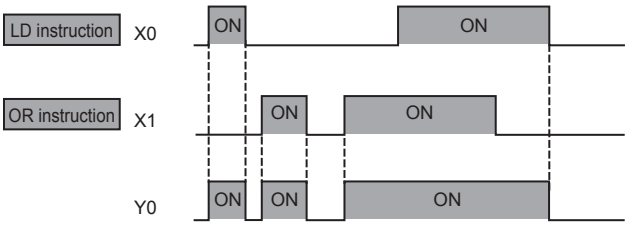
- When bits of word devices are specified, bits are specified in hexadecimal. (For example, b11 of D0 is specified by "D0.B".)

Ex.

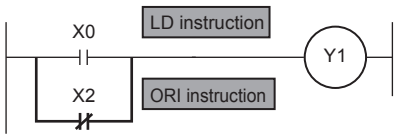
- OR instruction
  (parallel connection of NO contacts)



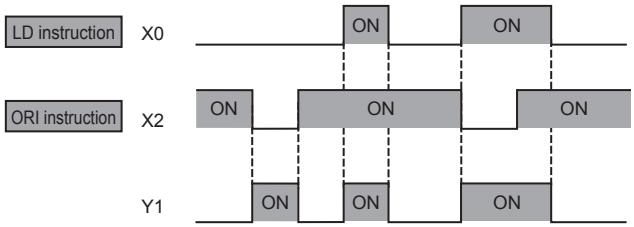
Timing chart



- ORI instruction
  (parallel connection of NC contacts)



Timing chart



Operation error

There is no operation error.

Pulse operation start, pulse series connection, pulse parallel connection

LDP, LDF, ANDP, ANDF, ORP, ORF

FX5S FX5UJ FX5U FX5UC

• LDP: Rising edge pulse operation start instruction

This becomes conductive (ON) only at the rising edge (OFF to ON) of the bit device specified by (s).

• LDF: Falling edge pulse operation start instruction

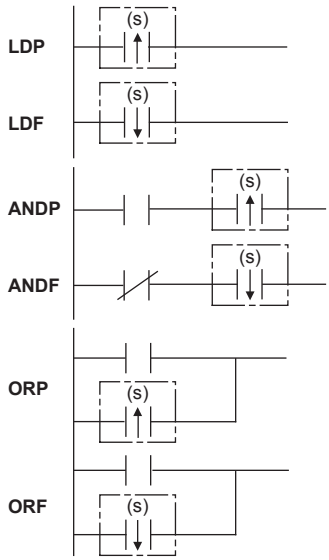
This becomes conductive (ON) only at the falling edge (ON to OFF) of the bit device specified by (s).

• ANDP: Rising edge pulse series connection instruction/ANDF: Falling edge pulse series connection instruction

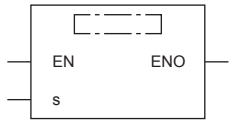
This instruction ANDs the bit device specified by (s) with the operation result so far, and uses the result as the operation result.

• ORP: Rising edge pulse parallel connection/ORF: Falling edge pulse parallel connection

This instruction ORs the bit device specified by (s) with the operation result so far, and uses the result as the operation result.

Ladder diagram	Structured text
	ENO:=LDP(EN,s); ENO:=LDF(EN,s); ENO:=ANDP(EN,s); ENO:=ANDF(EN,s); ENO:=ORP(EN,s); ENO:=ORF(EN,s);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Remarks	Range	Data type	Data type (label)
(s)	Device used as a contact	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

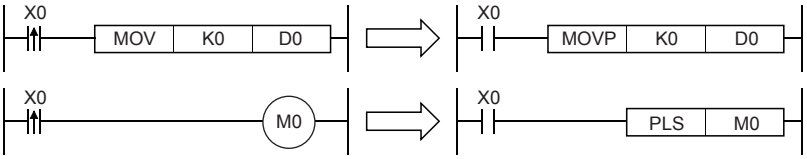
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DX)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	○	—	—	—	—	—	○

Processing details

■LDP, LDF

- The LDP instruction is the rising edge pulse operation start instruction, and becomes conductive (ON) only at the rising edge (OFF to ON) of the specified bit device. When word devices are specified by bits, this instruction becomes conductive (ON) only when the status of the specified bit changes to 0→1. When only the LDP instruction is programmed, operation is the same as the conversion of the instruction under execution to pulse instruction (□P).

The following figure shows an example when a ladder using the LDP instruction is replaced with a ladder not using the LDP instruction.



- The LDF instruction is the falling edge pulse operation start instruction, and becomes conductive (ON) at the falling edge (ON to OFF) of the specified bit device. When word devices are specified by bits, this instruction becomes conductive only when the status of the specified bit changes to 1→0.

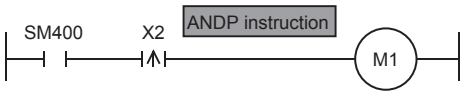
■ANDP, ANDF

- The ANDP instruction is the rising edge pulse series connection instruction, and the ANDF instruction is the falling edge pulse series connection. These instructions AND with the operation result so far, and uses the result as the operation result. The table below shows the ON/OFF information used by these instructions.

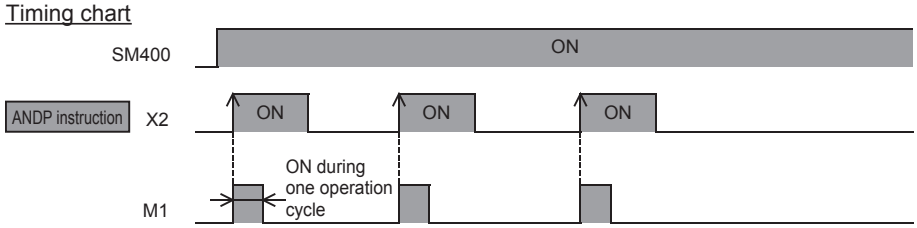
Device specified by ANDP, ANDF		ANDP status	ANDF status
Bit device	Bit specification of word device		
OFF to ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON to OFF	1→0	OFF	ON

Ex.

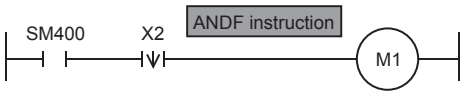
- ANDP instruction (series connection of rising edge pulse)



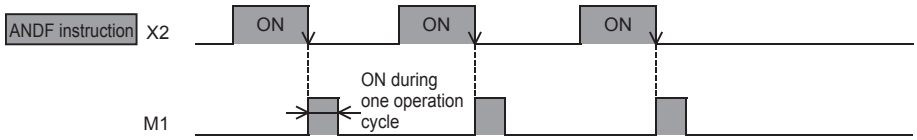
Timing chart



- ANDF instruction (series connection of falling edge pulse)



Timing chart



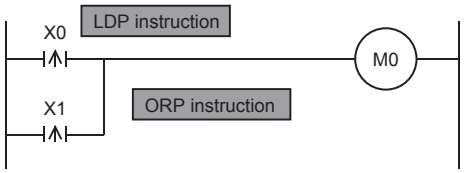
■ORP, ORF

- The ORP instruction is the rising edge pulse parallel connection instruction, and the ORF instruction is the falling edge pulse parallel connection instruction. These instructions OR with the operation result so far, and use the result as the operation result. The table below shows the ON/OFF information used by these instructions.

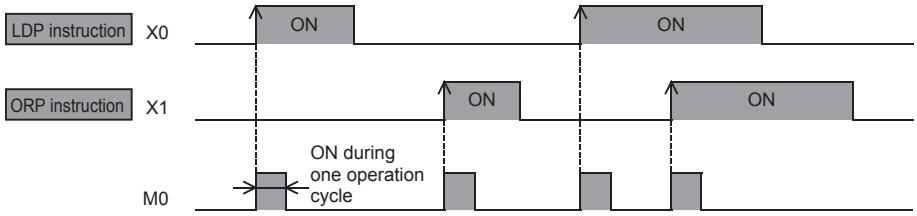
Device specified by ORP, ORF		ORP status	ORF status
Bit device	Bit specification of word device		
OFF to ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON to OFF	1→0	OFF	ON

Program example

- LDP and ORP instructions (initial logical operation of rising edge pulse, and parallel connection of rising edge pulse)

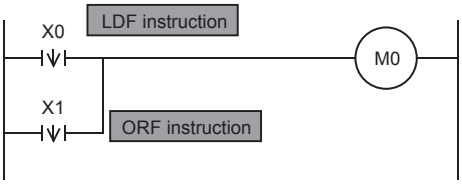


Timing chart

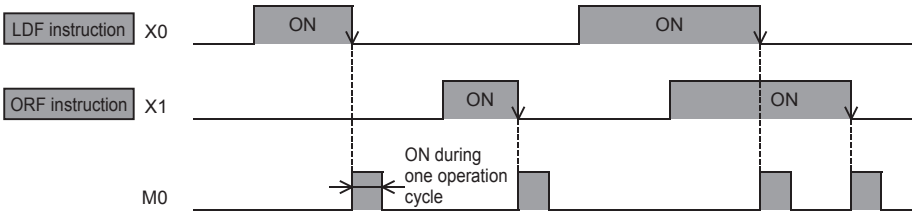


In the example shown above, M0 is ON during only one operation cycle when X0 or X1 turn from OFF to ON.

• LDF and ORF instructions (initial logical operation of falling edge pulse, and parallel connection of falling edge pulse)



Timing chart



In the example shown above, M0 is ON during only one operation cycle when X0 or X1 turn from ON to OFF.

Operation error

There is no operation error.

Pulse NOT operation start, pulse NOT series connection, pulse NOT parallel connection

LDPI, LDFI, ANDPI, ANDFI, ORPI, ORFI

FX5S FX5UJ FX5U FX5UC

• LDPI: Rising edge pulse NOT operation start instruction

This instruction becomes conductive (ON) at OFF, ON and the falling edge (ON to OFF) of the bit device specified by (s).

• LDFI: Falling edge pulse NOT operation start instruction

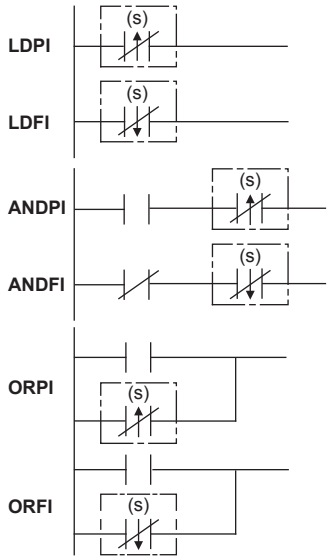
This instruction becomes conductive (ON) at the rising edge (OFF to ON), OFF and ON of the bit device specified by (s).

• ANDPI: Rising edge pulse NOT series connection instruction/ANDFI: Falling edge pulse NOT series connection instruction

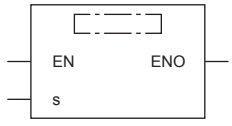
This instruction ANDs the bit devices specified by (s) with the operation result so far, and uses the result as the operation result.

• ORPI: Rising edge pulse NOT parallel connection instruction/ORFI: Falling edge pulse NOT parallel connection instruction

This instruction ORs the bit devices specified by (s) with the operation result so far, and uses the result as the operation result.

Ladder diagram	Structured text
	ENO:=LDPI(EN,s); ENO:=LDFI(EN,s); ENO:=ANDPI(EN,s); ENO:=ANDFI(EN,s); ENO:=ORPI(EN,s); ENO:=ORFI(EN,s);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Device used as a contact	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DX)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	○	—	—	—	—	—	○

Processing details

■LDPI, LDFI

- The LDPI instruction is the rising edge pulse NOT operation start instruction, and becomes conductive (ON) at OFF, ON and the falling edge (ON to OFF) of the specified bit device. When word devices are specified by bits, this instruction becomes conductive when the status of the specified bit is 0, 1, and when it changes 1→0.
- The LDFI instruction is the falling edge pulse NOT operation start instruction, and becomes conductive (ON) at the rising edge (OFF to ON), OFF and ON of the specified bit device. When word devices are specified by bits, this instruction becomes conductive (ON) when the status of the specified bit is 0, 1, and when it changes 0→1. The table below shows the ON/OFF information used by these instructions.

Device specified by LDPI, LDFI		LDPI status	LDFI status
Bit device	Bit specification of word device		
OFF to ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON to OFF	1→0	ON	OFF

■ANDPI, ANDFI

- The ANDPI instruction is the rising edge pulse NOT series connection instruction, and the ANDFI instruction is the falling edge pulse NOT series connection instruction. These instructions AND with the operation result so far, and use the result as the operation result. The table below shows the ON/OFF information used by these instructions.

Device specified by ANDPI, ANDFI		ANDPI status	ANDFI status
Bit device	Bit specification of word device		
OFF to ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON to OFF	1→0	ON	OFF

■ORPI, ORFI

- The ORPI instruction is the rising edge pulse NOT parallel connection instruction, and the ORFI instruction is the falling edge pulse NOT parallel connection instruction. These instructions OR with the operation result so far, and use the result as the operation result. The table below shows the ON/OFF information used by these instructions.

Device specified by ORPI, ORFI		ORPI status	ORFI status
Bit device	Bit specification of word device		
OFF to ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON to OFF	1→0	ON	OFF

Operation error

There is no operation error.

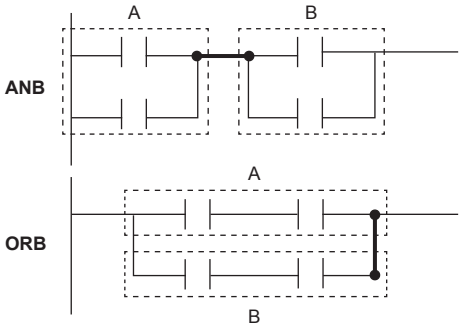
6.2 Association Instruction

Ladder block series/parallel connection

ANB, ORB

FX5S FX5UJ FX5U FX5UC

These instructions AND or OR the A and B blocks, and use the result as the operation result.

Ladder diagram	Structured text
 A: A block B: B block	Not supported.

FBD/LD
Not supported.

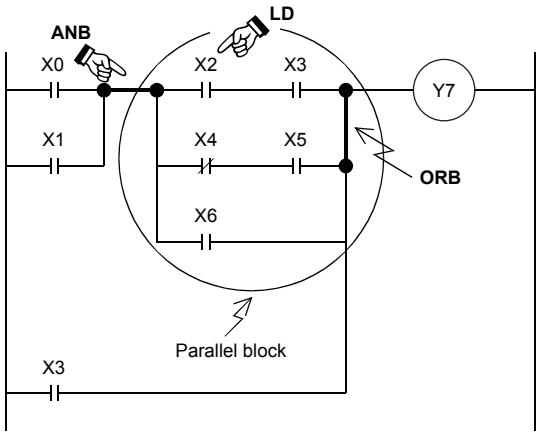
Processing details

■ANB

- This instruction ANDs the A and B blocks, and uses the result as the operation result.
- The symbol of this instruction is not NO contact symbol but a connection symbol.

Ex.

ANB instruction (series connection of ladder block)



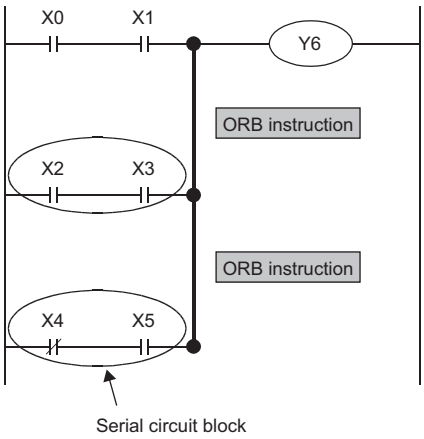
■ORB

- This instruction ORs the A and B blocks, and uses the result as the operation result.
- This instruction connects the ladder blocks of two contacts or more in parallel. For parallel connection of only one contact, use the OR and ORI instructions; there is no need to use this instruction.
- The symbol of this instruction is not NO contact symbol but a connection symbol.

Point

- The number of parallel circuits that can be connected by the ORB instruction is not limited. (when the recommended program is used)
- The ORB instruction can be used by one operation. However, note that the LD instruction and the LDI instruction can repeatedly be used 8 times or less. (when the recommended program is not used)

Circuit program



List program

Appropriate program

```
0  LD   X0
1  AND  X1
2  LD   X2
3  AND  X3
4  ORB        <-
5  LDI  X4
6  AND  X5
7  ORB        <-
8  OUT  Y6
```

Inappropriate program

```
0  LD   X0
1  AND  X1
2  LD   X2
3  AND  X3
4  LDI  X4
5  AND  X5
6  ORB        <-
7  ORB        <-
8  OUT  Y6
```

Operation error

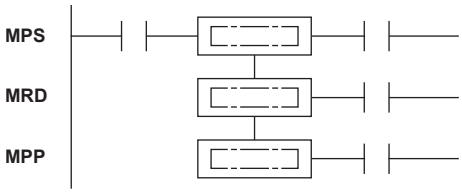
There is no operation error.

Storing/reading/clearing the operation result

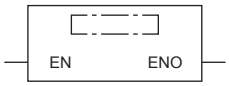
MPS, MRD, MPP

FX5S FX5UJ FX5U FX5UC

- MPS: This instruction stores the preceding operation result (ON/OFF) to memory.
- MRD, MPP: These instructions read the operation result stored by the MPS instruction, and executes operations from the next step using that operation result.

Ladder diagram	Structured text
	ENO:=MPS(EN); ENO:=MRD(EN); ENO:=MPP(EN);

FBD/LD



Processing details

■MPS

- This instruction stores the preceding operation result (ON/OFF) to memory.
- This instruction can be used up to 16 times consecutively. When MPP instruction is used in between, the number of uses of MPS instruction is decremented by 1.

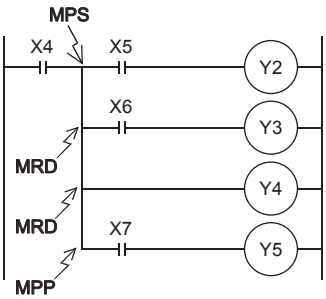
■MRD

- This instruction reads the operation result stored by the MPS instruction to memory, and executes operations from the next step using that operation result.

■MPP

- This instruction reads the operation result stored by the MPS instruction to memory, and executes operations from the next step using that operation result.
- This instruction clears the operation result stored by the MPS instruction from memory.
- The number of uses of MPS instruction is decremented by 1.

Ex.



Use MPS instruction to store the intermediate result of operation, and then drive the output Y2.

Use MRD instruction to read the stored data, and then drive the output Y3.

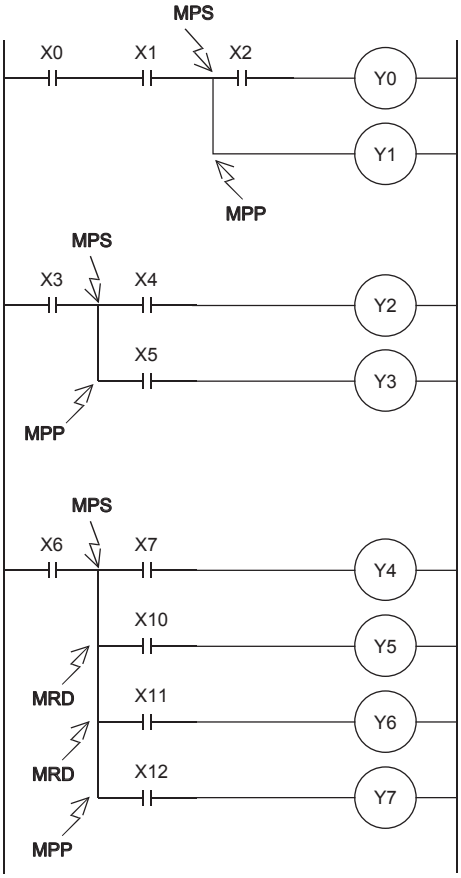
MRD instruction can be programmed as many times as necessary.

In the final output circuit, use MPP instruction instead of MRD instruction.
MPP instruction reads the stored data described above, and then resets it.

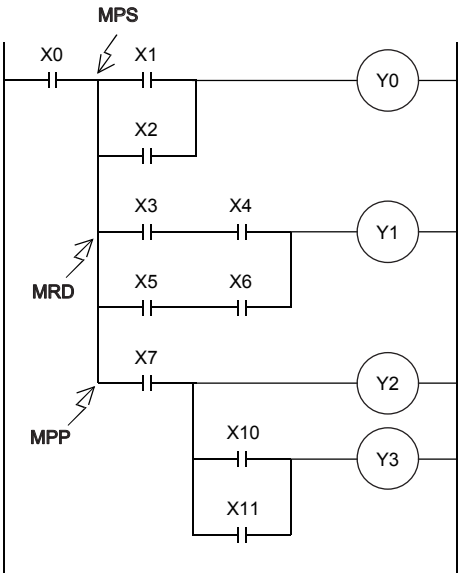
Program example

• Continuous use of MPS instruction - once

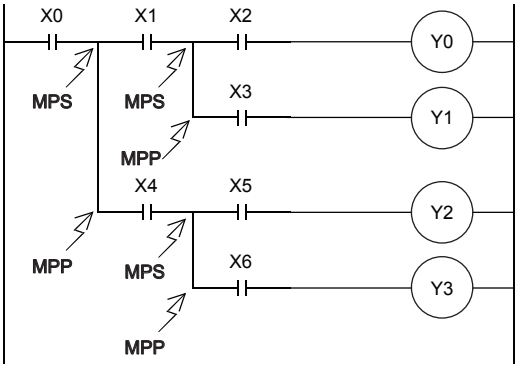
Only one stack is used in this example.



• Continuous use of MPS instruction - once (both ANB and ORB used)

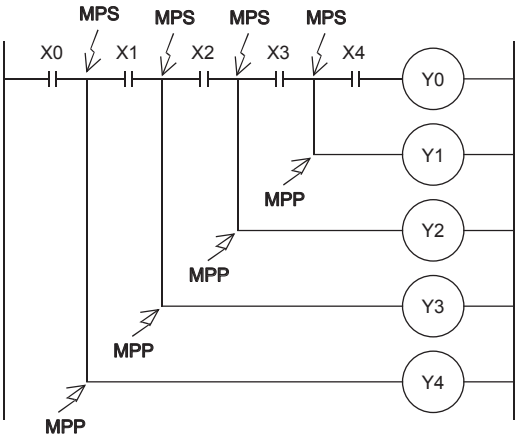


• Continuous use of MPS instruction - twice

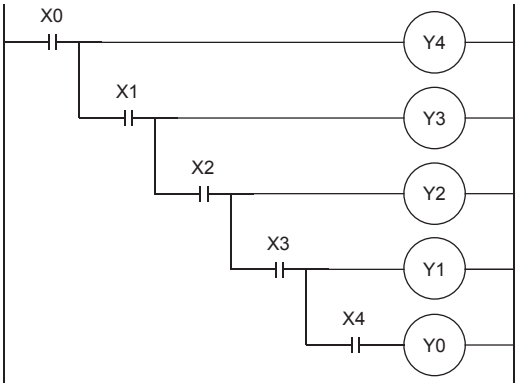


• Continuous use of MPS instruction - four times

MPS instruction is used three times in programming.



By changing the circuit as follows, the same contents can be programmed easily without MPS instruction.



Operation error

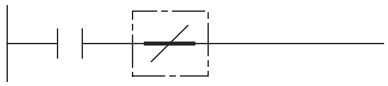
There is no operation error.

Inverting the operation result

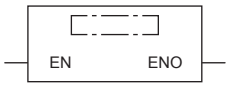
INV

FX5S FX5UJ FX5U FX5UC

This instruction inverts the operation result up to this instruction.

Ladder diagram	Structured text
	ENO:=INV(EN);

FBD/LD



Processing details

- This instruction inverts the operation result up to this instruction.

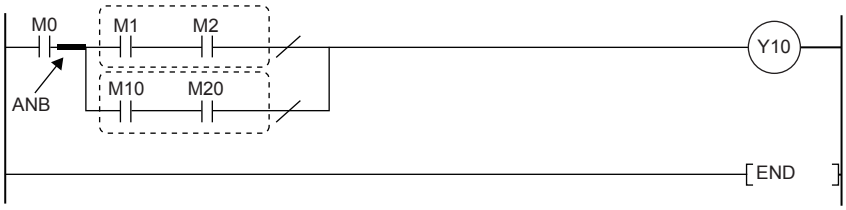
Operation result up to the INV instruction	Operation result after execution of INV instruction
OFF	ON
ON	OFF

Operation error

There is no operation error.

Point

- This instruction operates using the operation result so far. Hence, use it at the same position as the AND instruction. This instruction cannot be used at positions where the LD and OR instructions are programmed.
- If a ladder block is used, the operation result is inverted within the range of the ladder block. When operating a ladder with this instruction and the ANB instruction, pay attention to the inversion range.



: Inversion range

For details ANB instruction, refer to the following.

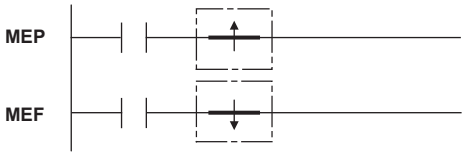
☞ Page 123 ANB, ORB

Converting the operation result into a pulse

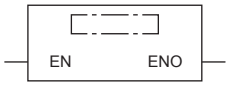
MEP, MEF

FX5S FX5UJ FX5U FX5UC

- MEP: This instruction turns ON at the rising edge of the operation result up to the MEP instruction and turns OFF in other instances.
- MEF: This instruction turns ON at the falling edge of the operation result up to the MEF instruction and turns OFF in other instances.

Ladder diagram	Structured text
	ENO:=MEP(EN); ENO:=MEF(EN);

FBD/LD



Processing details

■MEP

- This instruction turns ON (conductive state) at the rising edge (OFF to ON) of the operation result up to this instruction. This instruction turns OFF (non-conductive state) in instances other than the rising edge of the operation result up to this instruction.
- Use of this instruction makes conversion to pulse easier when multiple contacts are connected in series.

■MEF

- This instruction turns ON (conductive state) at the falling edge (ON to OFF) of the operation result up to this instruction. This instruction turns OFF (non-conductive state) in instances other than the falling edge of the operation result up to this instruction.
- Use of this instruction makes conversion to pulse easier when multiple contacts are connected in series.

Operation error

There is no operation error.

Point

- If an indexed contact is converted to pulse by the subroutine program and the FOR to NEXT instructions, etc., these instructions may not function properly.
- These instructions operate using the operation result so far. Hence, use them at the same position as the AND instruction. These instructions cannot be used at positions where the LD and OR instructions are programmed.

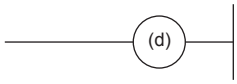
6.3 Output Instructions

Out (excluding the timer, counter and annunciator)

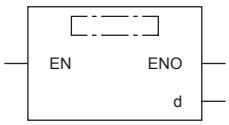
OUT

FX5S FX5UJ FX5U FX5UC

This instruction outputs the operation result up to this instruction to the specified device.

Ladder diagram	Structured text
	ENO:=OUT(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Number of the device that turns ON/OFF	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DY)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1	○*2	○*3	—	—*4	—	—	—	—	—	○

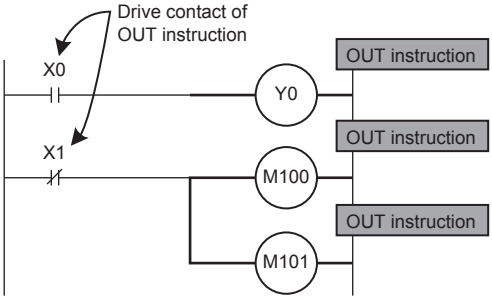
*1 When using F, refer to ☞ Page 139 OUT F.
*2 When using T, ST, refer to ☞ Page 132 OUT T, OUTH T, OUTHS T, OUT ST, OUTH ST, OUTHS ST.
When using C, refer to ☞ Page 135 OUT C.
*3 Only the FX5 intelligent function module can be specified.
*4 When using LC, refer to ☞ Page 137 OUT LC.

Processing details

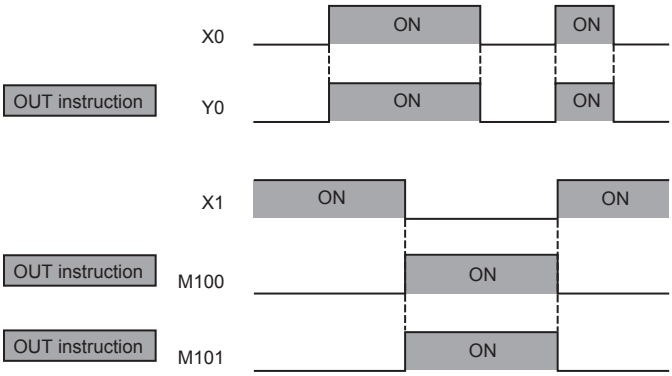
- This instruction outputs the operation result up to this instruction to the specified device.

Condition	Operation result	Coil/specified bit
When bit device is used	OFF	OFF
	ON	ON
When bit of word device is specified	OFF	0
	ON	1

Program example



Timing chart



Operation error

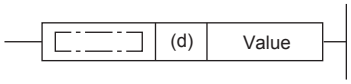
There is no operation error.

Timer

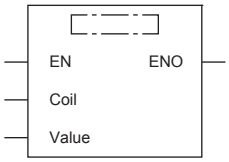
OUT T, OUTH T, OUTHS T, OUT ST, OUTH ST, OUTHS ST

FX5S FX5UJ FX5U FX5UC

The timer counts up to the set value when the operation result up to the OUT instruction is ON and the coil of the timer/retentive timer specified by (d) turns ON. When the timer times up, NO contact becomes conductive and NC contact becomes non conductive.

Ladder diagram	Structured text
	ENO:=OUT_T(EN,Coil,Value); ENO:=OUTH(EN,Coil,Value); ENO:=OUTHS(EN,Coil,Value);

FBD/LD



("OUT_T", "OUTH", "OUTHS" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)*1	Timer Number	—	Timer/retentive timer	ANY*3
(Set value)*2	Timer set value	0 to 32767	16-bit unsigned binary*4	ANY16*4
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, d displays as Coil.
*2 In the case of the ST language and the FBD/LD language, Set value displays as Value.
*3 Only timer type/retentive timer type label can be used.
*4 In the case of the OUT_T instruction of the ST language and the FBD/LD language , data type is ANY_INT.
Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○*1	—	—	—	—	—	—	—	—	—
(Set value)	—	○*2	○	—	—	—	—	○*3	—	—	—

*1 Only T and ST can be used.
*2 T, ST, and C cannot be used.
*3 Only decimal constant (K) can be used.

Processing details

- These instructions count up to the set value when the operation result up to the OUT instruction is ON and the coil of the timer/retentive timer specified by (d) turns ON. When the timer reaches the end of its count (current value ≥ set value), NO contact becomes conductive and NC contact becomes non-conductive.
- Operation is as follows when the operation result up to the OUT instruction changes from ON to OFF.

Timer type	Timer coil	Current timer value	Before time-out		After time-out	
			NO contact	NC contact	NO contact	NC contact
Timer	off	0	Non-Conductive state	Conductive state	Non-Conductive state	Conductive state
Retentive timer	off	Holds current value	Non-Conductive state	Conductive state	Conductive state	Non-Conductive state

- After the timer times up, clear the current value of the retentive timer and turn the contact off by the RST instruction.
- The following processing is executed when the OUT instruction is executed:
  - The coil in the OUT T, OUTH T, OUTHS T, OUT ST, OUTH ST and OUTHS ST instructions turns ON/OFF
  - The contact in the OUT T, OUTH T, OUTHS T, OUT ST, OUTH ST and OUTHS ST instructions turns ON/OFF
  - The current value in the OUT T, OUTH T, OUTHS T, OUT ST, OUTH ST and OUTHS ST instructions changes
- When the OUT T instruction is skipped using the CJ instruction, etc. while the OUT T and OUT ST instructions are ON, these instructions do not update the current value or turn ON/OFF the contacts.
- When the same OUT T and OUT ST instructions are executed in the same scan twice or more, these instructions update the current value for the same number of times of execution.

Point

Values used for timers can be set in the range 1 to 32767. Actual timer constants are as follows since the OUT, OUTH, and OUTHS instructions operate as 100 ms, 10 ms, and 1 ms timers, respectively.

- OUT instruction: 0.1 to 3276.7 seconds
- OUTH instruction: 0.01 to 327.67 seconds
- OUTHS instruction: 0.001 to 32.767 seconds

For the counting method, refer to the following.

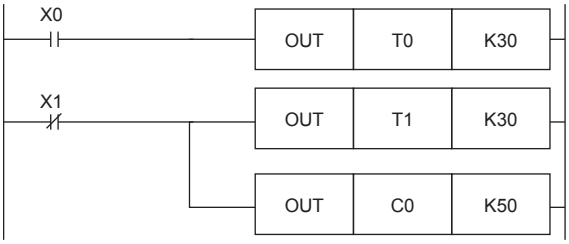
MELSEC iQ-F FX5 User's Manual (Application)

Program example

The set value is required after OUT instruction for the coil of a timer or counter.

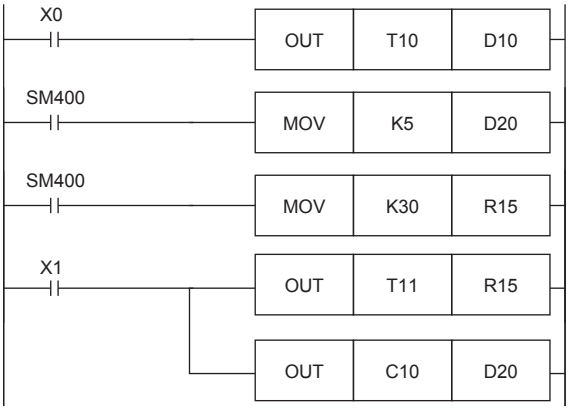
The set value can be specified directly by a decimal number (K) or indirectly using a data register (D) or file register (R).

- Direct specification



The set value of a timer or counter can be specified directly by a decimal number (K).

- Indirect specification



The set value of a timer or counter can be set by a data register (D) or file register (R). At this time, the current value of the data register (D) or file register (R) is regarded as the set value of the timer or counter.

It is necessary to write the set value to a data register (D) or file register (R) used for the set value by MOV instruction, DSW instruction or indicator before driving the timer or counter.

- Indexing

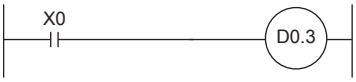
Devices used in OUT instruction can be indexed with index registers (Z and LZ).

(State relays (S), long counter (LC), and "D□.b" cannot be indexed.)

24 words in total as the number of points of Z and LZ are available for indexing.
When a used device is an input (X) or output (Y), the value of an index register (Z and LZ) is converted into an octal number, and then added.
Example: When the value of Z0 is 20, Y24 turns ON or OFF.

- Bit specification of word devices

A bit in data register (D) can be specified as a device used in OUT instruction.



When specifying a bit in data register, input "." after a data register (D) number, and then input a bit number (0 to F) consecutively.
Only 16-bit data registers are available.
Specify a bit number as "0, 1, 2, ... 9, A, B, ... F" from the least significant bit.
Example: In the example shown on the left, the bit 3 of D0 turns ON or OFF when X0 turns ON or OFF.

Operation error

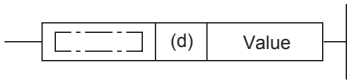
Error code (SD0/SD8067)	Description
3405H	A negative value is specified for the timer value.

Counter

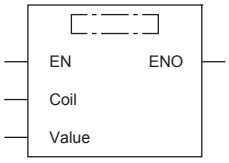
OUT C

FX5S FX5UJ FX5U FX5UC

This instruction increments the current value of the counter specified by (d) by 1 when the operation result up to OUT instruction changes from OFF to ON, and when the counter reaches the end of its count, NO contact becomes conductive and NC contact becomes non-conductive.

Ladder diagram	Structured text
 Value: Set value	ENO:=OUT_C(EN,Coil,Value);

FBD/LD



("OUT_C" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)*1	Counter Number	—	Counter	ANY*3
(Set value)*2	Counter set value	0 to 32767	16-bit unsigned binary	ANY_INT*4
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, d displays as Coil.
*2 In the case of the ST language and the FBD/LD language, Set value displays as Value.
*3 Only counter type label can be used.
*4 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○*1	—	—	—	—	—	—	—	—	—
(Set value)	—	○*2	○	—	—	—	—	○*3	—	—	—

*1 Only C can be used.
*2 T, ST, and C cannot be used.
*3 Only decimal constant (K) can be used.

Processing details

- This instruction increments the current value of the counter specified by (d) by 1 when the operation result up to OUT instruction changes from OFF to ON, and when the counter reaches the end of its count (current value ≥ set value), NO contact becomes conductive and NC contact becomes non-conductive.
- The counter does not count while the operation result remains on. (Count input does not need to be converted to pulses.)
- After a count up, the count value and contact status do not change until the RST instruction is executed.
- When the set value is 0, the same processing as for set value 1 is performed.

Operation error

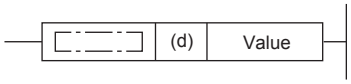
Error code (SD0/SD8067)	Description
3405H	A negative value is specified for the set value.

Long counter

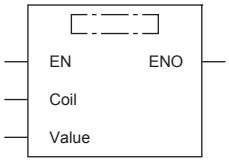
OUT LC

FX5S FX5UJ FX5U FX5UC

This instruction increments the current value of the long counter specified by (d) by 1 when the operation result up to the OUT instruction changes from OFF to ON, and when the counter reaches the end of its count, NO contact becomes conductive and NC contact becomes non-conductive.

Ladder diagram	Structured text
 Value: Set value	ENO:=OUT_C(EN,Coil,Value);

FBD/LD



("OUT_C" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)*1	Long counter number	—	Long counter	ANY*3
(Set value)*2	Long counter set value	0 to 4294967295	32-bit unsigned binary	ANY_INT*4
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, d displays as Coil.
*2 In the case of the ST language and the FBD/LD language, Set value displays as Value.
*3 Only long counter type label can be used.
*4 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	—	—	—	○	—	—	—	—	—	—
(Set value)	—	○*1	○	—	—	—	—	○*2	—	—	—

*1 T, ST, and C cannot be used.
*2 Only decimal constant (K) can be used.

Processing details

- This instruction increments the current value of the long counter specified by (d) by 1 when the operation result up to the OUT instruction changes from OFF to ON, and when the counter reaches the end of its count (current value ≥ set value), NO contact becomes conductive and NC contact becomes non-conductive.
- The counter does not count while the operation result remains on. (Count input does not need to be converted to pulses.)
- After a count up, the count value and contact status do not change until the RST or ZRST instruction is executed.
- When the set value is 0, the same processing as for set value 1 is performed.
- When signed (-2147483648 to + 2147483647) or high speed counter is assigned to the LC, the UDCNTF instruction is used. For the UDCNTF instruction, refer to ☞ Page 774 UDCNTF.

Operation error

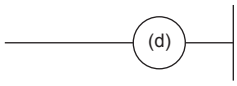
Error code (SD0/SD8067)	Description
2821H	When the high speed counter is assigned to the specification long counter.

Annunciator

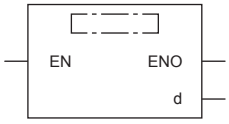
OUT F

FX5S FX5UJ FX5U FX5UC

This instruction outputs the operation result up to the OUT F instruction to the specified annunciator.

Ladder diagram	Structured text
	ENO:=OUT(EN,d);

FBD/LD



("OUT" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Annunciator number that turns ON	—	Bit	—*1 (ANY_BOOL)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1	—	—	—	—	—	—	—	—	—	—

*1 Only F can be used.

Processing details

- This instruction outputs the operation result up to the OUT F instruction to the specified annunciator.
- Operation is as follows when annunciator (F) is turned ON by the OUT F instruction.
  - The annunciator number (F number) that turns ON is stored in special registers (SD64 to SD79).
  - The content of SD63 is incremented by 1.
- When the content of SD63 is 16 (16 annunciators are already on), the annunciator number that turns ON is not stored in SD64 to SD79 even if a new annunciator turns ON.
- Operation is as follows when annunciator (F) is turned OFF by the OUT F instruction:
  - The coil turns OFF, but the contents of SD64 to SD79 do not change.
  - To delete an annunciator that has turned OFF by the OUT F instruction from SD64 to SD79, use the RST F instruction.

■Related devices

Device	Name	Description
SD62	Annunciator (F) Detection No.	This register stores the earliest detected annunciator (F) No.
SD63	Annunciator (F) Detection Number	This register stores the number of annunciator (F) detections.
SD64 to SD79	Annunciator (F) Detection No. table	This register stores the annunciator (F) detection No.

Operation error

There is no operation error.

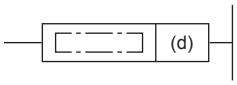
Setting devices (excluding annunciator)

SET

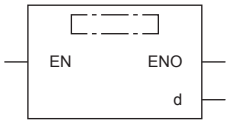
FX5S FX5UJ FX5U FX5UC

The status of the device specified by (d) changes as follows when the execution command turns ON.

- Bit device: Turns the coils and contacts ON.
- Bit specification of word device: Set the specified bit to 1.

Ladder diagram	Structured text
	ENO:=SET(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Bit device number/ Bit specification of word device to be set (turns ON)	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DY)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1	○*2	○*3	—	—	—	—	—	—	—	○

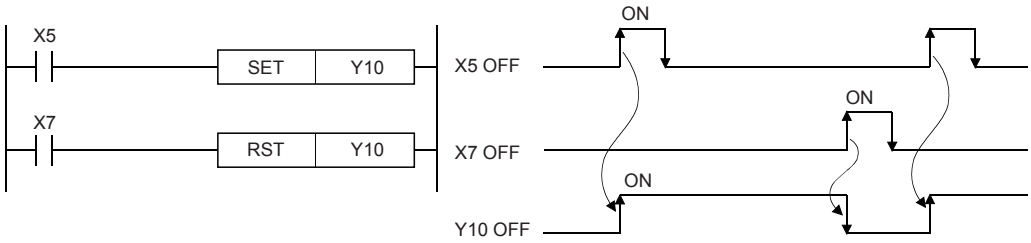
*1 When using F, refer to ☞ Page 145.
*2 T, ST, and C cannot be used.
*3 Only the FX5 intelligent function module can be used.

Processing details

- The status of the specified device changes as follows when the execution command turns ON.

Device	Device status
Bit devices	Turns coils and contacts ON.
Bit specification of word device	Sets the specified bit to 1.

- A device that is turned ON is held on even if the execution command turns OFF. Devices that are turned ON by the SET instruction can be turned OFF by the RST instruction.



- When the execution command is OFF, the device status does not change.

Precautions

When the SET and RST instructions are executed on the same output relay (Y), the result of the instruction nearer the END instruction (end of program) is output.

Operation error

There is no operation error.

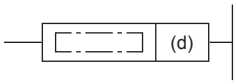
Resetting devices (excluding annunciator)

RST

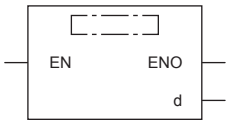
FX5S FX5UJ FX5U FX5UC

The status of the device specified by (d) changes as follows when the RST input turns ON.

- Bit devices: Turns the coils and contacts OFF.
- Timers, counters: Sets the current value to 0, and turns contacts OFF.
- Bit specification of word device: Sets the specified bit to 0.
- Word devices, module access devices, index registers: Sets content to 0.

Ladder diagram	Structured text
	ENO:=RST(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Bit device number/ bit specification of word device to be reset, or word device number to be reset	—	Bit/word/double word	ANY_ELEMENTARY*1
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DY)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1	○	○*2	○	○	○	○	—	—	—	○

*1 When using F, refer to ☞ Page 147.
Digit specified bit device cannot be used.
*2 Only the FX5 intelligent function module can use digit specification of bit devices.

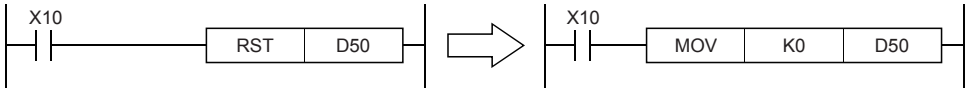
Processing details

- The status of the specified device changes as follows when the execution command turns ON.

Device	Device status
Bit devices	Turns coils and contacts OFF.
Timers, counters	Sets the current value to 0, and turns contacts OFF.
Bit specification of word device	Set the specified bit to 0.
Word devices, module access device, index registers	Sets content to 0.

- When the execution command is OFF, the device status does not change.

• Function when a word device is specified by the RST instruction is the same as the following circuit.



Precautions

When the RST instruction for a timer or counter is executed by a program containing a jump or by a subroutine program or interrupt program, the timer or counter is held in a reset state, and the timer or counter may not work normally.

Operation error

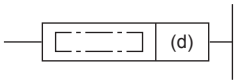
There is no operation error.

Setting annunciator

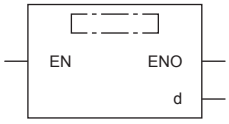
SET F

FX5S FX5UJ FX5U FX5UC

This instruction turns ON the specified annunciator.

Ladder diagram	Structured text
	ENO:=SET(EN,d);

FBD/LD



("SET" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Annunciator number (F number) that is set	—	Bit	—*1 (ANY_BOOL)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

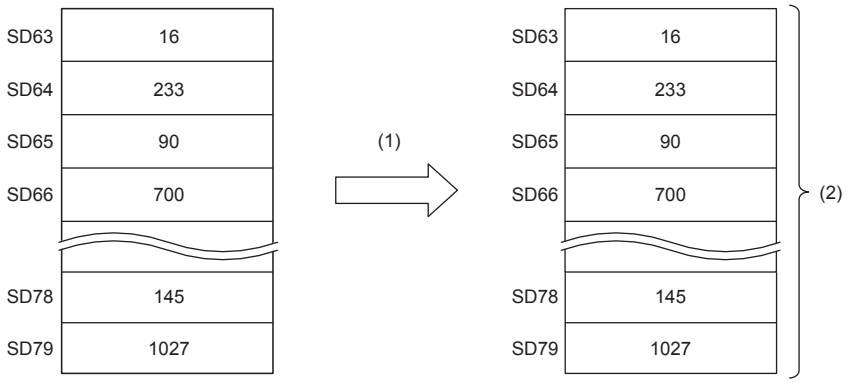
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1	—	—	—	—	—	—	—	—	—	—

*1 Only F can be used.

Processing details

- This instruction turns ON the annunciator specified by (d) when the execution command turns ON.
- Operation is as follows when annunciator (F) is turned ON.
  - The annunciator number (F number) that turns ON is stored in special registers (SD64 to SD79).
  - The content of SD63 is incremented by 1.
- When the content of SD63 is 16 (16 annunciators are already ON), the annunciator number that turns ON is not stored in SD64 to SD79 even if a new annunciator turns ON.



(1): F30 is turned ON.
(2): Does not change.

■Related devices

Device	Name	Description
SD62	Annunciator (F) Detection No.	This register stores the earliest detected annunciator (F) No.
SD63	Annunciator (F) Detection Number	This register stores the number of annunciator (F) detections.
SD64 to SD79	Annunciator (F) Detection No. table	This register stores the annunciator (F) detection No.

Operation error

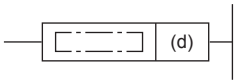
There is no operation error.

Resetting annunciator

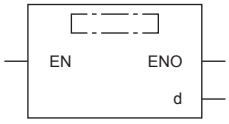
RST F

FX5S FX5UJ FX5U FX5UC

This instruction turns OFF the specified annunciator.

Ladder diagram	Structured text
	ENO:=RST(EN,d);

FBD/LD



("RST" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Annunciator number (F number) that is reset	—	Bit	ANY_ELEMENTARY
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

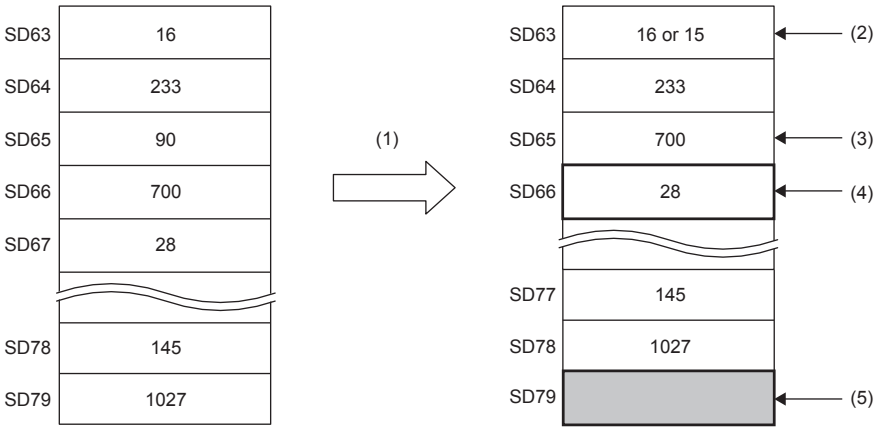
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1	—	—	—	—	—	—	—	—	—	—

*1 Only F can be used.

Processing details

- This instruction turns OFF the annunciator specified by (d) when the execution command turns ON.
- An annunciator number (F number) that turns OFF is deleted from special registers (SD64 to SD79) and the content of SD63 is decremented by 1.
- When the content of SD63 is 16, annunciator numbers are deleted from SD64 to SD79 by the RST instruction. Also, if an annunciator not registered in SD64 to SD79 turns ON, its number is registered. When there are two or more unregistered numbers, this instruction adds the numbers starting from the smallest annunciator number. SD63 is not decremented by 1 when the numbers not registered in SD64 to SD79 are turned OFF.



(1): F90 is reset.
(2): When F number that is not registered in SD79 is stored, this remains as 16. When SD79 is 0, the number is decremented by -1 to become 15.
(3): The F number in SD66 is shifted to this area.
(4): F number of SD67 is stored.
(5): Not registered F number or 0 is stored.

■Related devices

Device	Name	Description
SD62	Annunciator (F) Detection No.	This register stores the earliest detected annunciator (F) No.
SD63	Annunciator (F) Detection Number	This register stores the number of annunciator (F) detections.
SD64 to SD79	Annunciator (F) Detection No. table	This register stores the annunciator (F) detection No.

Operation error

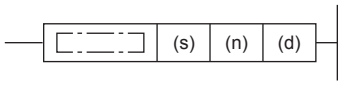
There is no operation error.

Setting annunciator (with check time)

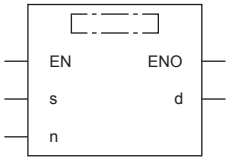
ANS

FX5S FX5UJ FX5U FX5UC

This instruction sets the annunciator (F device).

Ladder diagram	Structured text
	ENO:=ANS(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Timer number for evaluation time	—	16-bit signed binary	ANY16
(n)	Evaluation time data	1 to 32767	16-bit unsigned binary	ANY16_U
(d)	Annunciator device to be set	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

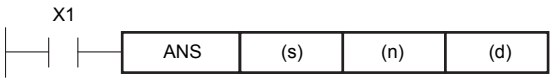
■Applicable devices

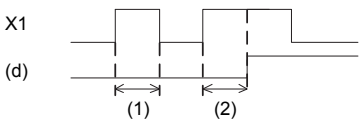
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—
(d)	○*2	—	—	—	—	—	—	—	—	—	—

*1 Only T can be used.
*2 Only F can be used.

Processing details

- This instruction sets (d) when the command input remains ON continuously for the evaluation time [(n)×100 ms, (s)] or more. This instruction resets the current value of (s) evaluation timer and does not set (d) when the command time is less than the evaluation time [(n)×100 ms]. Also, this instruction resets the evaluation timer when the command input turns OFF.





(1): Less than the evaluation time ((n) × less than 100 ms)
(2): Equivalent to or longer than the evaluation time ((n) × 100 ms or more)

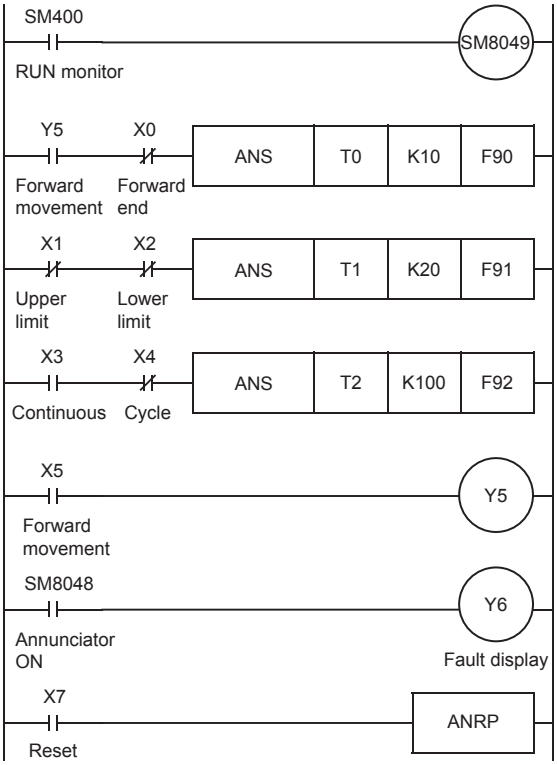
■Related devices

Device	Name	Description
SM8049	ON status annunciator smallest number enabled	When SM8049 is turned ON, SM8048 and SD8049 are enabled.
SM8048	Annunciator operation	When one of the F devices is operating, SM8048 turns ON.
SD8049	ON status annunciator smallest number	The smallest number of the F devices that are operating is stored.

Program example

• Displaying a fault number using an annunciator

When the program for external fault diagnosis shown below is created and the content of SD8049 (smallest state relay number in the ON status) is monitored, the smallest state relay number in the ON status from F90 to F92 is displayed.
If two or more faults are present at the same time, the next smallest fault number is displayed after the fault of the smallest fault number is cleared.



When SM8049 turns ON, monitoring becomes valid.

If the forward end detection input X0 does not turn ON within 1 second after the forward movement output Y5 is driven, F90 turns ON.

If both the upper limit input X1 and the lower limit input X2 are OFF for 2 seconds or more due to a dog error, F91 turns ON.

The switch X4 is set to ON in one operation cycle of the machine. If the switch X4 is not set to ON while the continuous operation mode input X3 is ON in the machine whose tact time is less than 10 seconds, F92 turns ON.

When one among F90 to F92 turns ON, SM8048 turns ON and the fault display output Y6 turns ON.

A state relay which was set to ON by the external fault diagnosis program is set to OFF by the reset button X7. Every time X7 is set to ON, an operation state relay in the ON status with the smallest device number is reset (set to OFF) in turn.

Operation error

There is no operation error.

Resetting annunciator (smallest number reset)

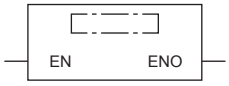
ANR(P)

FX5S FX5UJ FX5U FX5UC

This instruction resets the lowest number annunciator (F device) in the ON status.

Ladder diagram	Structured text
	ENO:=ANR(EN); ENO:=ANRP(EN);

FBD/LD



Processing details

• Annunciator (F device) that is operating (in ON status) is reset when the command input turns ON.

This instruction resets the annunciator with the smallest number when multiple annunciators are ON. If the command input is turned ON again, this instruction resets the annunciator with the next smallest number among annunciators (F devices) that are operating.

■Related devices

Device	Name	Description
SM8049	On status annunciator smallest number enabled	When SM8049 is turned ON, SM8048 and SD8049 are enabled.
SM8048	Annunciator operation	When one of the F devices is operating, SM8048 turns ON.
SD8049	On status annunciator smallest number	The smallest number of the F devices that are operating is stored.

Precautions

• When ANR instruction is used, annunciators in the ON status are reset in turn in each operation cycle.
• This is executed for only 1 operation cycle (only once) when the ANRP instruction is used.

Program example

For a program example, refer to ☞ Page 149 Setting annunciator (with check time).

Operation error

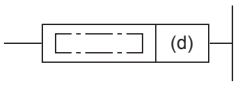
There is no error.

Rising edge output

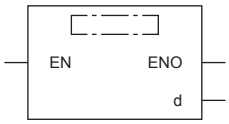
PLS

FX5S FX5UJ FX5U FX5UC

This instruction turns ON the device specified by (d) for one scan when the PLS command turns from OFF to ON, and turns OFF in other cases.

Ladder diagram	Structured text
	ENO:=PLS(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Device to be converted to pulse	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

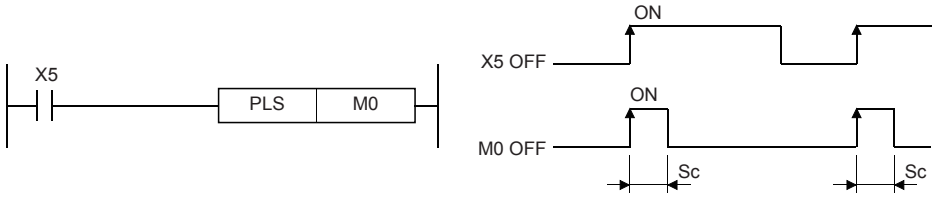
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DY)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○*1	○*2	—	—	—	—	—	—	—	○

*1 T, ST, and C cannot be used.
*2 Only the FX5 intelligent function module can be used.

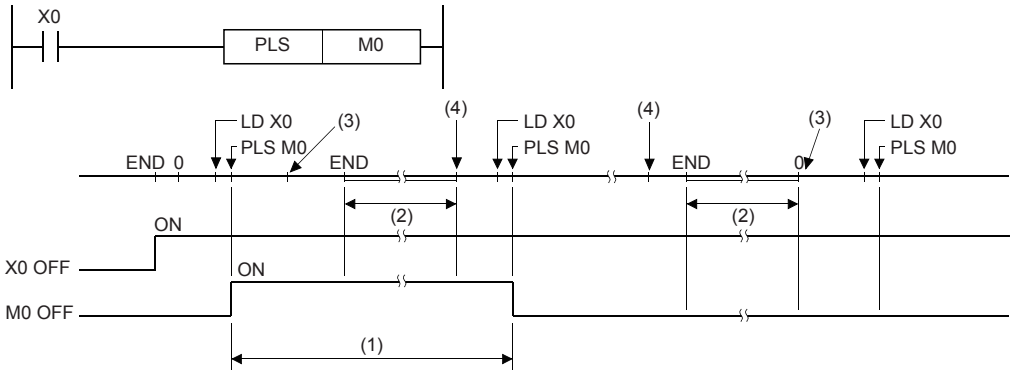
Processing details

- This instruction turns ON the specified device for one scan when the PLS command turns from OFF to ON, and turns OFF in other cases. When there is one PLS instruction programmed for the device specified by (d) during a scan, the specified device turns ON for one scan.



Sc: 1 scan

- If the RUN/STOP/RESET switch is changed from RUN to STOP after execution of the PLS instruction, the PLS instruction will not be executed even if the switch is set to RUN again.

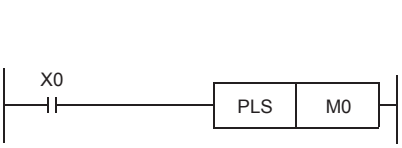


(1) 1 scan of PLS M0
(2) CPU module operation stop time
(3) Set the RUN/STOP/RESET switch on the CPU module to RUN→STOP.
(4) Set the RUN/STOP/RESET switch on the CPU module to STOP→RUN.

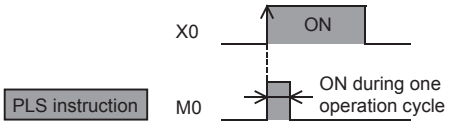
Precautions

- When write during RUN is completed for a circuit including a rising edge instruction (LDP/ANDP/ORP instruction), the instruction is not executed regardless of the ON/OFF status of the target device of the rising edge instruction. Also, in the case of a rising edge instruction (PLS instruction), the instruction is not executed regardless of the ON/OFF status of the device that is set as the operation condition. The instruction is executed when the target device and the device in the operation conditions is set from OFF to ON again.
- Note that the device specified by (d) sometimes turns ON for one scan or more when the PLS instruction is made to jump by the CJ instruction or the executed subroutine program was not called by the CALL(P) instruction.

Program example



Timing chart



In the figure above, M0 is ON during only one operation cycle when X0 changes from OFF to ON.

Operation error

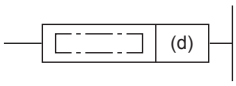
There is no operation error.

Falling edge output

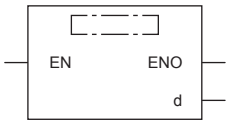
PLF

FX5S FX5UJ FX5U FX5UC

This instruction turns ON the device specified by (d) for one scan when the PLF command turns from ON to OFF, and turns OFF in other cases.

Ladder diagram	Structured text
	ENO:=PLF(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Device to be converted to pulse	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

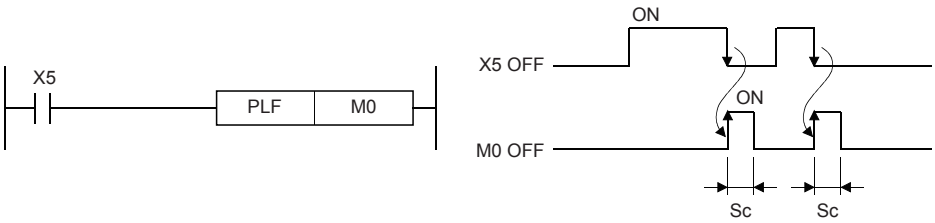
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DY)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○*1	○*2	—	—	—	—	—	—	—	○

*1 T, ST, and C cannot be used.
*2 Only the FX5 intelligent function module can be used.

Processing details

- This instruction turns ON the specified device for one scan when the PLF command turns OFF from ON, and turns OFF in other cases. When there is one PLF instruction programmed for the device specified by (d) during a scan, the specified device turns ON for one scan.



Sc: 1 scan

- If the RUN/STOP/RESET switch is changed from RUN to STOP after execution of the PLF instruction, the PLF instruction will not be executed even if the switch is set to RUN again.

Precautions

- When write during RUN is completed for a circuit including a falling edge instruction (LDF/ANDF/ORF instruction), the instruction is not executed regardless of the ON/OFF status of the target device of the falling edge instruction. Also, in the case of a falling edge instruction (PLF instruction), the instruction is not executed regardless of the ON/OFF status of the device that is set as the operation condition. The instruction is executed when the target device and the device in the operation conditions is set from ON to OFF again.
- Note that the device specified by (d) sometimes turns ON for one scan or more when the PLF instruction is made to jump by the CJ instruction or the executed subroutine program was not called by the CALL(P) instruction.

Operation error

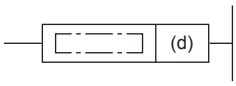
There is no operation error.

Inverting the bit device output

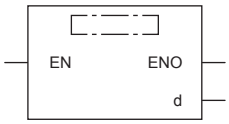
FF

FX5S FX5UJ FX5U FX5UC

This instruction reverses the output status of the device specified by (d) when the execution command changes from OFF to ON.

Ladder diagram	Structured text
	ENO:=FF(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Device number to be reversed	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DY)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○*1	○*2	—	—	—	—	—	—	—	○

*1 T, ST, and C cannot be used.
*2 Only the FX5 intelligent function module can be used.

Processing details

- This instruction reverses the state of the device specified by (d) when the execution command changes from OFF to ON.

Device	Device status	
	Before execution of FF instruction	After execution of FF instruction
Bit devices	OFF	ON
	ON	OFF
Bit specification of word device	0	1
	1	0

Operation error

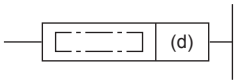
There is no operation error.

Inverting the bit device output

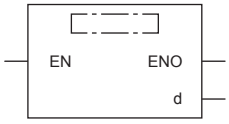
ALT(P)

FX5S FX5UJ FX5U FX5UC

These instructions reverse (ON ↔ OFF) bit devices when input turns ON.

Ladder diagram	Structured text
	ENO:=ALT(EN,d); ENO:=ALTP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Bit device number whose output is alternated	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

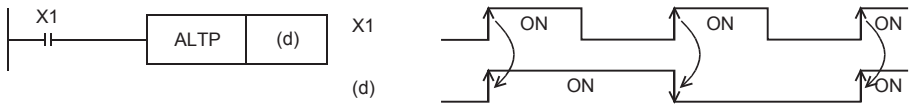
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○*1	○*2	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.
*2 Only the FX5 intelligent function module can be used.

Processing details

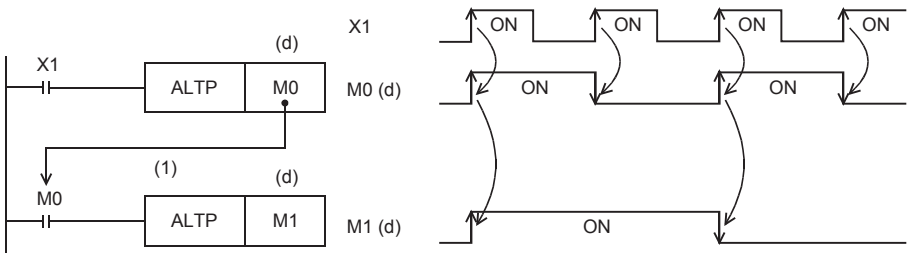
■Alternating output (1-step)

The bit device specified by (d) is reversed ON ↔ OFF each time the command input changes from OFF to ON.



■Division output (according to alternating output (2-step))

The ALTP instruction can be used in multiple combinations to perform division output.



(1): Specify the same device

Precautions

When the CPU module is programmed with the ALT instruction, reversal operation is performed at every operation cycle. To perform reversal operation by command ON/OFF, either use the ALTP instruction (pulse execution type) or set a command contact as LDP etc. (pulse execution type).

Operation error

There is no operation error.

6.4 Shift Instructions

Shifting bit devices

SFT(P)

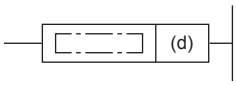
FX5S FX5UJ FX5U FX5UC

• In case of bit device:

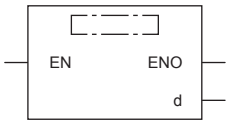
These instructions shift the ON/OFF status of the device before the device specified by (d) to the device specified by (d).

• When bit of word device is specified:

These instructions shift the 1/0 status of the bit before the bit specified by (d) to the bit specified by (d).

Ladder diagram	Structured text
	ENO:=SFT(EN,d); ENO:=SFTP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number to be shifted	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (DY)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○*1	—	—	—	—	—	—	—	—	○

*1 T, ST, and C cannot be used.

Processing details

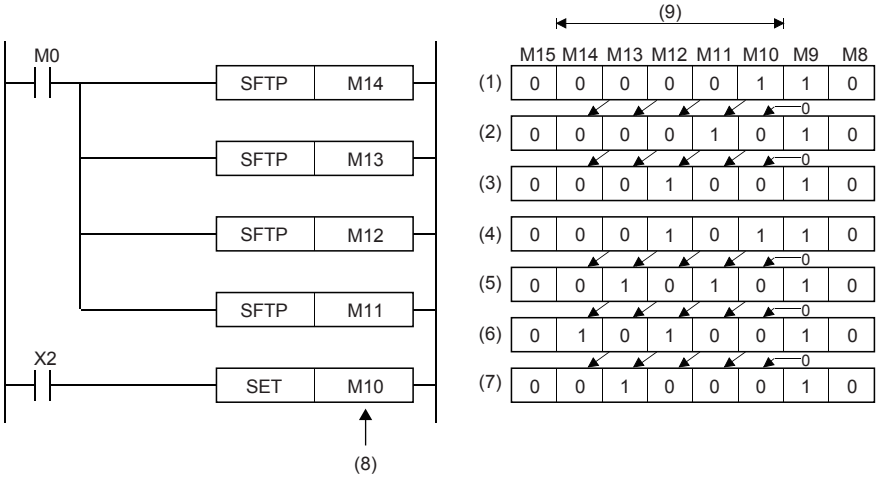
■In case of bit device

- This instruction shifts the ON/OFF status of the device before the device specified by (d) to the device specified by (d). The device before the device specified by (d) turns OFF.

Ex.

When M11 is specified by the SFTP instruction and the SFTP instruction is executed, the ON/OFF status of M10 is shifted to M11 and M10 is turned OFF.

- Turn ON the first device to be shifted by the SET instruction.
- When the SFT(P) instruction is used consecutively, create the program to start from the device with the largest number.



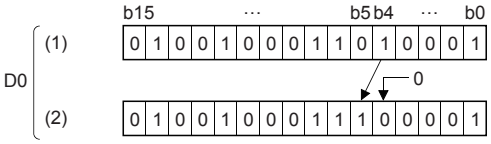
(1): X2 ON
(2): After the 1st shift input
(3): After the 2nd shift input
(4): X2 ON
(5): After the 3rd shift input
(6): After the 4th shift input
(7): After the 5th shift input
(8): First device of shift
(9): Shift range

■When bit of word device is specified:

- This instruction shifts the 1/0 status of the bit before the bit specified by (d) to the bit specified by (d). The bit before the bit specified by (d) becomes 0.

Ex.

When D0.5 (bit 5 (b5) of D0) is specified by the SFT(P) instruction and the SFT(P) instruction is executed, the 1/0 status of b4 of the D0 is shifted to b5 and b4 is set to 0.



(1): Before shift execution
(2): After shift execution

Operation error

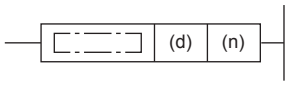
Error code (SD0/SD8067)	Description
2820H	The device specified by (d) exceeds the corresponding device range.

Shifting 16-bit data to the right by n bit(s)

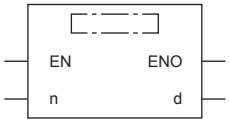
SFR(P)

FX5S FX5UJ FX5U FX5UC

These instructions shift the 16-bit data in the device specified by (d) to the right by (n) bit(s).

Ladder diagram	Structured text
	ENO:=SFR(EN,n,d); ENO:=SFRP(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number where the shift-target data is stored	—	16-bit signed binary	ANY16
(n)	Number of shifts	0 to 15	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

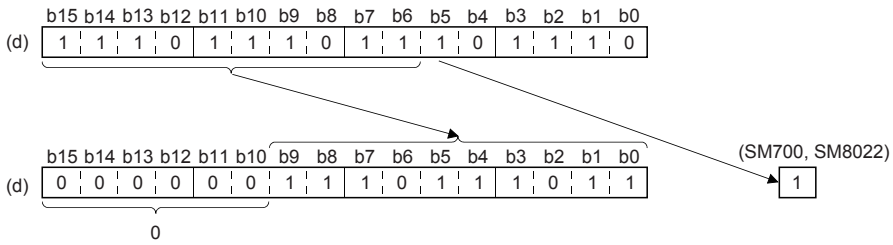
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

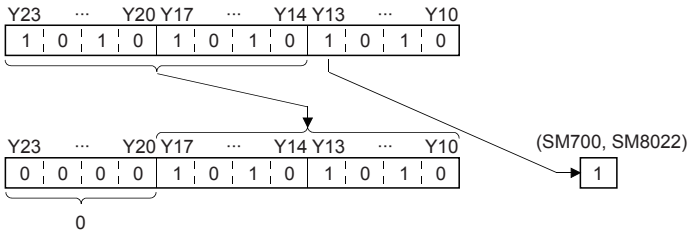
- This instruction shifts the 16-bit data in the device specified by (d) to the right by (n) bit(s) from the most significant bit. The (n) bit(s) from the most significant bit is/are filled with 0(s).

When (n)=6



• When (d) is a bit device, bits are shifted to the right within the device range specified by digit specification.

When (n)=4



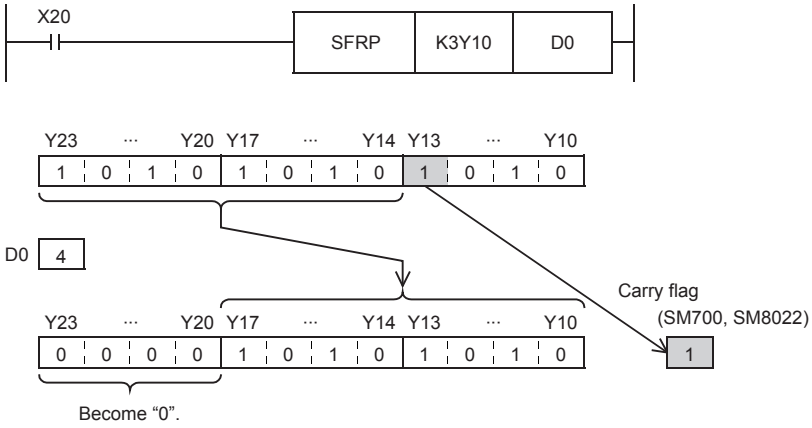
• Specify any value between 0 and 15 for (n). If a value 16 or larger is specified for (n), bits are shifted to the right by the remainder value of (n)÷16. For example, when (n) is 18, data is shifted by 2 bits to the right because 18 divided by 16 equals 1 with a remainder of 2.

■Related devices

Device	Name	Description
SM700	Carry	ON/OFF according to the status (1/0) of the (n-1)th bit.
SM8022		

Program example

In the program example shown below, the contents of Y10 to Y23 are shifted rightward by the number of bits specified by D0 when X20 turns ON.



Operation error

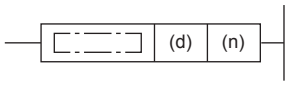
There is no operation error.

Shifting 16-bit data to the left by n bit(s)

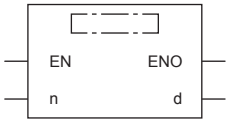
SFL(P)

FX5S FX5UJ FX5U FX5UC

These instructions shift the 16-bit data in the device specified by (d) to the left by (n) bit(s).

Ladder diagram	Structured text
	ENO:=SFL(EN,n,d); ENO:=SFLP(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number where the shift-target data is stored	—	16-bit signed binary	ANY16
(n)	Number of shifts	0 to 15	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

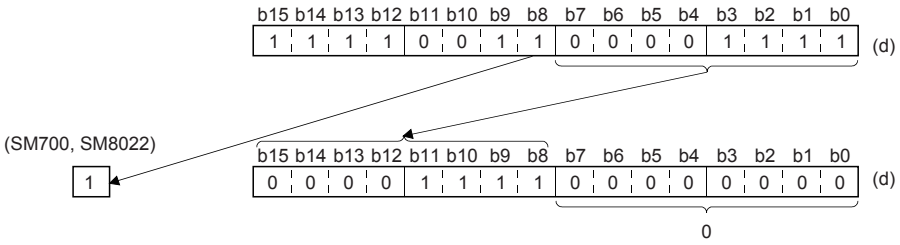
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

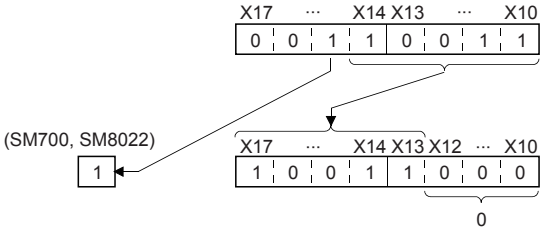
- These instructions shift the 16-bit data in the device specified by (d) to the left by (n) bit(s) from the least significant bit. (n) bits from the least significant bit are filled with "0".

When (n)=8



• When (d) is a bit device, bit(s) are shifted to the left within the device range specified by digit specification.

When (n)=3



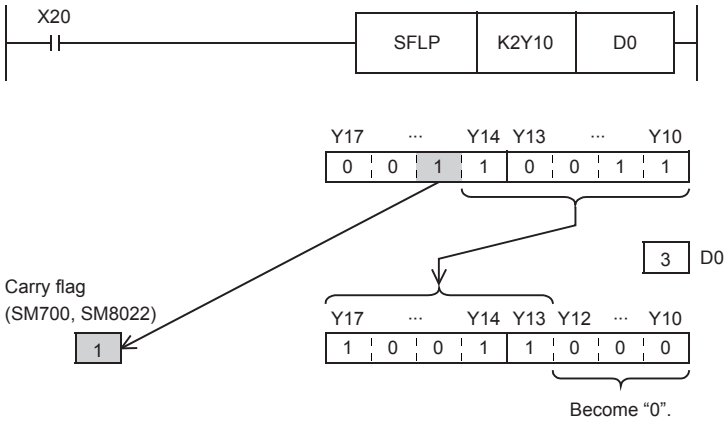
• Specify any value between 0 and 15 for (n). If a value 16 or larger is specified for (n), bit(s) are shifted to the left by the remainder value of (n)÷16. For example, when (n) is 18, data is shifted by 2 bits to the left because 18 divided by 16 equals 1 with a remainder of 2.

■Related devices

Device	Name	Description
SM700	Carry	ON/OFF according to the status (1/0) of the (n-1)th bit.
SM8022		

Program example

In the program example shown below, the contents of Y10 to Y17 are shifted leftward by the number of bits specified by D0 when X20 turns ON.



Operation error

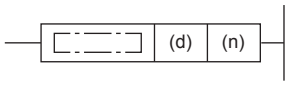
There is no operation error.

Shifting n-bit data to the right by 1 bit

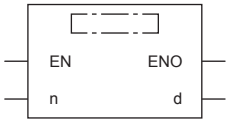
BSFR(P)

FX5S FX5UJ FX5U FX5UC

These instructions shift (n) point(s) of data to the right by 1 bit from the device specified by (d).

Ladder diagram	Structured text
	ENO:=BSFR(EN,n,d); ENO:=BSFRP(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number to be shifted	—	Bit	ANY_BOOL
(n)	Number of devices to be shifted	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

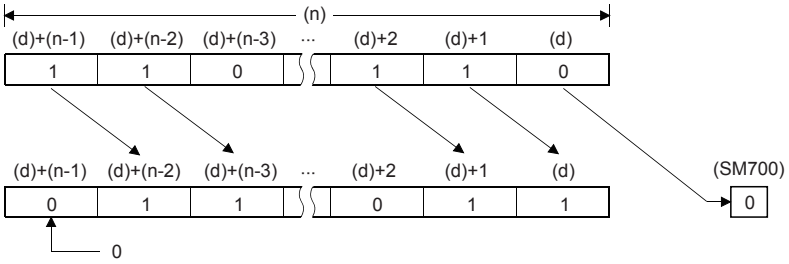
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○*1	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions shift (n) point(s) of data to the right by 1 bit from the device specified by (d).



- The value of the device specified by (d) + (n-1) becomes 0.

■Related devices

Device	Name	Description
SM700	Carry	ON/OFF according to the status (1/0) of the (d) bit.

Operation error

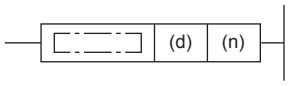
Error code (SD0/SD8067)	Description
2820H	The (n) points of data starting from the device specified by (d) exceed in the corresponding device.

Shifting n-bit data to the left by 1 bit

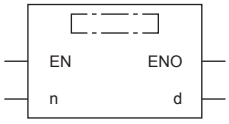
BSFL(P)

FX5S FX5UJ FX5U FX5UC

These instructions shift (n) point(s) of data to the left by 1 bit from the device specified by (d).

Ladder diagram	Structured text
	ENO:=BSFL(EN,n,d); ENO:=BSFLP(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number to be shifted	—	Bit	ANY_BOOL
(n)	Number of devices to be shifted	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

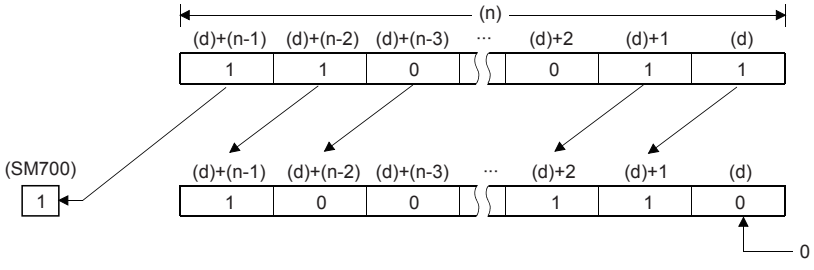
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○*1	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions shift (n) point(s) of data to the left by 1 bit from the device specified by (d).



- The value of the device specified by (d) becomes 0.

■Related devices

Device	Name	Description
SM700	Carry	ON/OFF according to the status (1/0) of the (d) bit.

Operation error

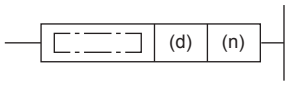
Error code (SD0/SD8067)	Description
2820H	The (n) points of data starting from the device specified by (d) exceed in the corresponding device.

Shifting n-word data to the right by 1 word

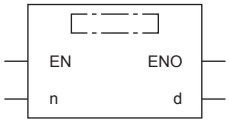
DSFR(P)

FX5S FX5UJ FX5U FX5UC

These instructions shift (n) point(s) of data to the right by 1 word from the device specified by (d).

Ladder diagram	Structured text
	ENO:=DSFR(EN,n,d); ENO:=DSFRP(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

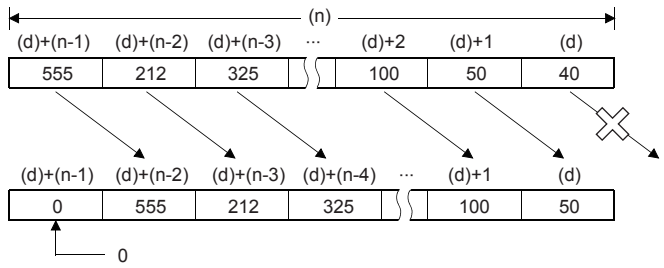
Operand	Description	Range	Data type	Data type (label)
(d)	Head device number to be shifted	—	Word	ANY16
(n)	Number of devices to be shifted	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions shift (n) point(s) of data to the right by 1 word from the device specified by (d).



- The value of the device specified by (d) + (n-1) becomes 0.

Operation error

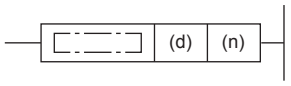
Error code (SD0/SD8067)	Description
2820H	The (n) points of data starting from the device specified by (d) exceed in the corresponding device.

Shifting n-word data to the left by 1 word

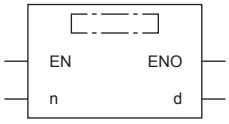
DSFL(P)

FX5S FX5UJ FX5U FX5UC

These instructions shift (n) point(s) of data to the left by 1 word from the device specified by (d).

Ladder diagram	Structured text
	ENO:=DSFL(EN,n,d); ENO:=DSFLP(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

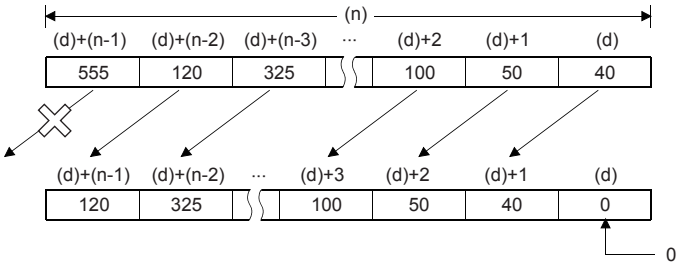
Operand	Description	Range	Data type	Data type (label)
(d)	Head device number to be shifted	—	Word	ANY16
(n)	Number of devices to be shifted	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions shift (n) point(s) of data to the left by 1 word from the device specified by (d).



- The value of the device specified by (d) becomes 0.

Operation error

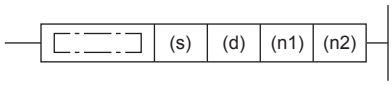
Error code (SD0/SD8067)	Description
2820H	The (n) points of data starting from the device specified by (d) exceed in the corresponding device.

Shifting n-bit(s) data to the right by (n) bit(s)

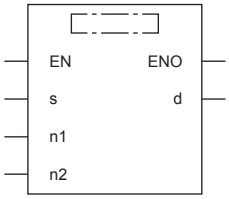
SFTR(P)

FX5S FX5UJ FX5U FX5UC

These instructions shift (n1) bits of data to the right by (n2) bit(s) from the device specified by (d).

Ladder diagram	Structured text
	ENO:=SFTR(EN,s,n1,n2,d); ENO:=SFTRP(EN,s,n1,n2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number stored to the shift data after the shift	—	Bit	ANY_BOOL
(d)	Head device number to be shifted	—	Bit	ANY_BOOL
(n1)*1	Data length of shift data	0 to 65535	16-bit unsigned binary	ANY16_U
(n2)*1	Number of shifts	0 to 65535	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Set so that n2≤n1.

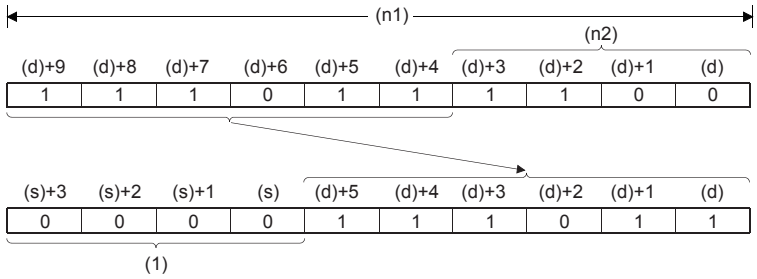
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○*1	—	—	—	—	—	○*2	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.
*2 Only 0 or 1 can be used.

Processing details

- These instructions shift (n1) bits of data to the right by (n2) bit(s) from the device specified by (d). After the shift, (n2) points from (s) are set into (n2) points from (d)+(n1-n2).
- When K0 is specified for (s), set 0s for (n2) points of bits from (d)+(n1-n2) after the shift.
- When K1 is specified for (s), set 1s for (n2) points of bits from (d)+(n1-n2) after the shift.



(1): When (s)=K0, it is 0.

Operation error

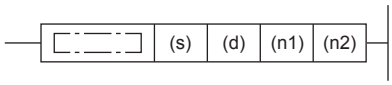
Error code (SD0/SD8067)	Description
2820H	The (n2) points of data starting from the device specified by (s) exceed in the corresponding device.
	The (n1) points of data starting from the device specified by (d) exceed in the corresponding device.
2821H	The transfer source data (s) overlaps with shift device (d).
3405H	A constant other than K0 or K1 is specified when the constant (s) is specified.
	The values specified in (n1) and (n2) are such that (n1)<(n2).

Shifting n-bit data to the left by n bit(s)

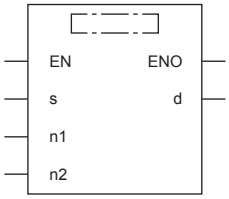
SFTL(P)

FX5S FX5UJ FX5U FX5UC

These instructions shift (n1) bits of data to the left by (n2) bit(s) from the device specified by (d).

Ladder diagram	Structured text
	ENO:=SFTL(EN,s,n1,n2,d); ENO:=SFTLP(EN,s,n1,n2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number stored to the shift data after the shift	—	Bit	ANY_BOOL
(d)	Head device number to be shifted	—	Bit	ANY_BOOL
(n1)*1	Data length of shift data	0 to 65535	16-bit unsigned binary	ANY16_U
(n2)*1	Number of shifts	0 to 65535	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Set so that n2≤n1.

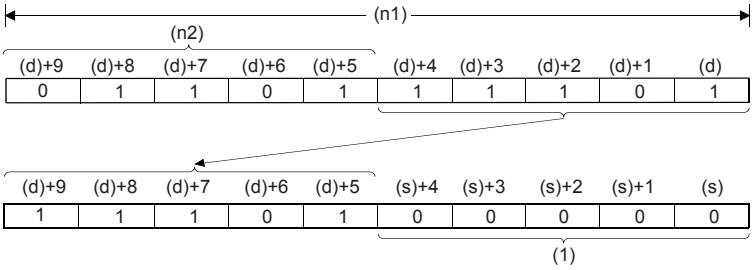
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○*1	—	—	—	—	—	○*2	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.
*2 Only 0 or 1 can be used.

Processing details

- These instructions shift (n1) bits of data to the left by (n2) bit(s) from the device specified by (d). After the shift, (n2) points from (s) are set into (n2) points from (d).
- When K0 is specified for (s), set 0s for (n2) points of bits from (d) after the shift.
- When K1 is specified for (s), set 1s for (n2) points of bits from (d) after the shift.



(1): When (s)=K0, it is 0.

Operation error

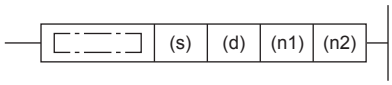
Error code (SD0/SD8067)	Description
2820H	The (n2) points of data starting from the device specified by (s) exceed in the corresponding device.
	The (n1) points of data starting from the device specified by (d) exceed in the corresponding device.
2821H	The transfer source data (s) overlaps with shift device (d).
3405H	A constant other than K0 or K1 is specified when the constant (s) is specified.
	The values specified in (n1) and (n2) are such that (n1)<(n2).

Shifting n-word data to the right by n word(s)

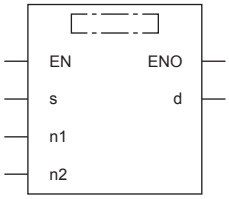
WSFR(P)

FX5S FX5UJ FX5U FX5UC

This instruction shifts (n1) words of data to the right by (n2) word(s) from the device specified by (d).

Ladder diagram	Structured text
	ENO:=WSFR(EN,s,n1,n2,d); ENO:=WSFRP(EN,s,n1,n2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number stored to the shift data after the shift	—	Word	ANY16
(d)	Head device number to be shifted	—	Word	ANY16
(n1)*1	Data length of shift data	0 to 65535	16-bit unsigned binary	ANY16_U
(n2)*1	Number of shifts	0 to 65535	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

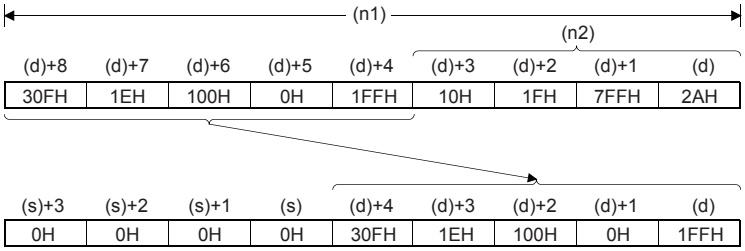
*1 Set so that n2≤n1.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—

Processing details

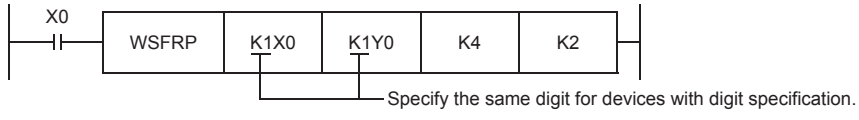
- This instruction shifts (n1) words of data to the right by (n2) word(s) from the device specified by (d). After the shift, (n2) points from (s) are set into (n2) points from (d)+(n1-n2).
- This instruction sets the specified value for (n2) points of devices from (d) + (n1-n2) after the shift when K is specified for (s).

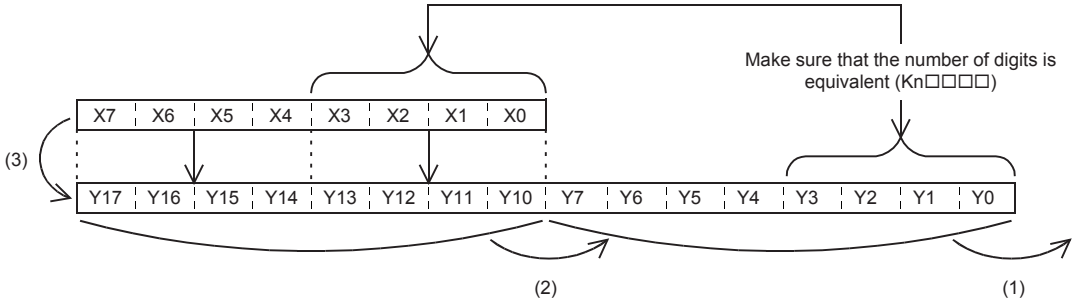


- When the value specified for (n1) or (n2) is 0, the processing is not performed.

Program example

- Shifting devices with digit specification





Operation error

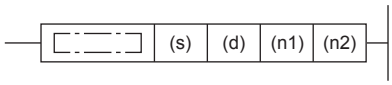
Error code (SD0/SD8067)	Description
2820H	The (n2) points of data starting from the device specified by (s) exceed in the corresponding device.
	The (n1) points of data starting from the device specified by (d) exceed in the corresponding device.
2821H	The transfer source data (s) overlaps with shift device (d).
3405H	A constant other than K0 or K1 is specified when the constant (s) is specified.
	The values specified in (n1) and (n2) are such that (n1)<(n2).

Shifting n-word data to the left by n word(s)

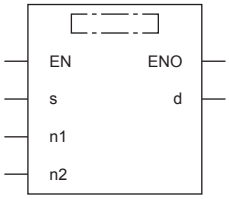
WSFL(P)

FX5S FX5UJ FX5U FX5UC

This instruction shifts (n1) words of data to the left by (n2) word(s) from the device specified by (d).

Ladder diagram	Structured text
	ENO:=WSFL(EN,s,n1,n2,d); ENO:=WSFLP(EN,s,n1,n2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number stored to the shift data after the shift	—	Word	ANY16
(d)	Head device number to be shifted	—	Word	ANY16
(n1)*1	Data length of shift data	0 to 65535	16-bit unsigned binary	ANY16_U
(n2)*1	Number of shifts	0 to 65535	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

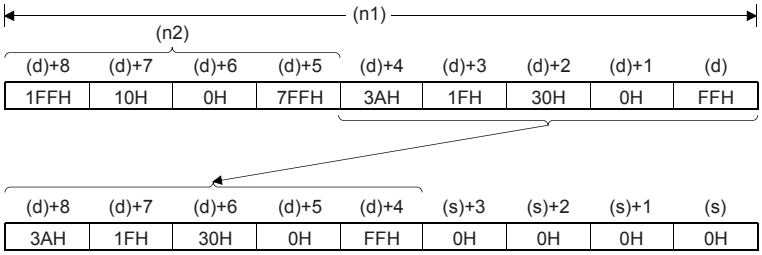
*1 Set so that n2≤n1.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—

Processing details

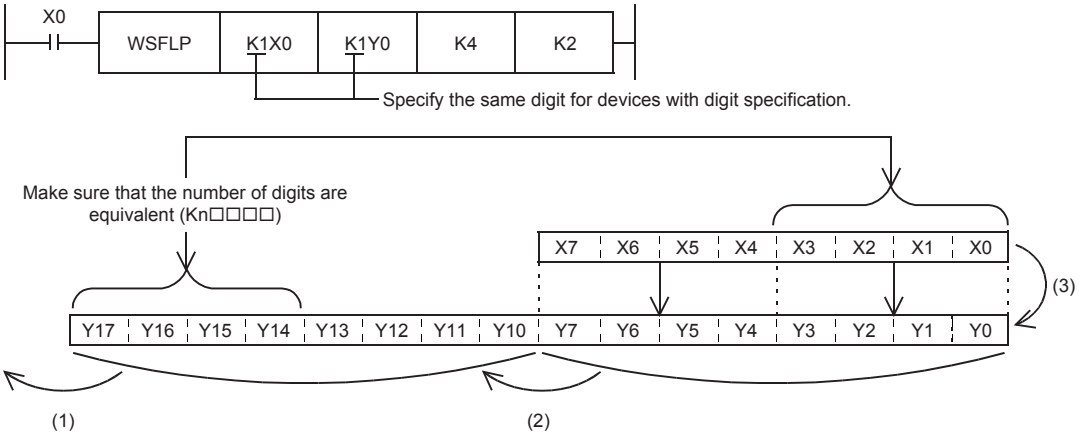
- This instruction shifts (n1) words of data to the left by (n2) word(s) from the device specified by (d). After the shift, (n2) points from (s) are set into (n2) points from (d).
- This instruction sets the specified value for (n2) points of devices from (d) + (n1-n2) after the shift when K is specified for (s).



- When the value specified for (n1) or (n2) is 0, the processing is not performed.

Program example

- Shifting devices with digit specification



Operation error

Error code (SD0/SD8067)	Description
2820H	The (n2) points of data starting from the device specified by (s) exceed in the corresponding device.
	The (n1) points of data starting from the device specified by (d) exceed in the corresponding device.
2821H	The transfer source data (s) overlaps with shift device (d).
3405H	A constant other than K0 or K1 is specified when the constant (s) is specified.
	The values specified in (n1) and (n2) are such that (n1)<(n2).

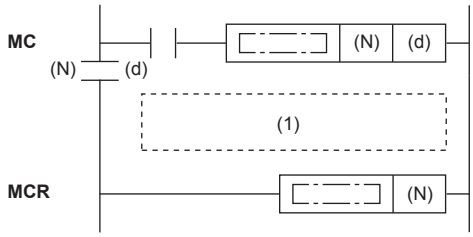
6.5 Master Control Instruction

Setting/resetting the master control

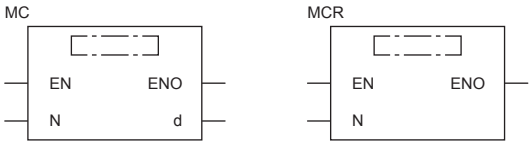
MC, MCR

FX5S FX5UJ FX5U FX5UC

- MC: This instruction starts master control.
- MCR: This instruction ends master control.

Ladder diagram	Structured text
 (1): Master control ladder	ENO:=MC(EN,n,d); ENO:=MCR(EN,n);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(N)*1	Nesting	0 to 14	Device name	ANY16_S
(d)	Number of device to be turned ON	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, N displays as n.

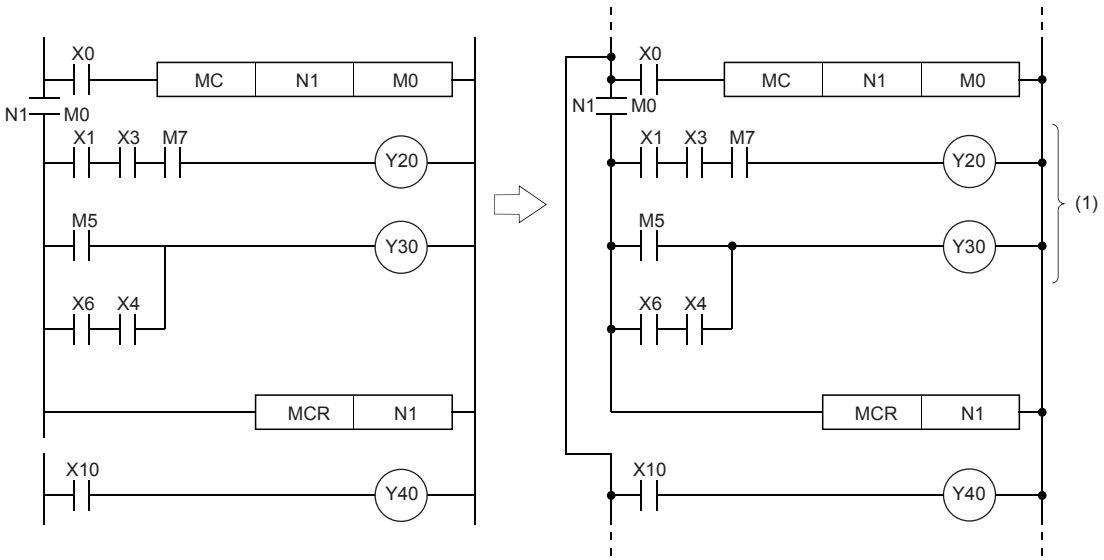
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others	
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	N	DY
(N)	—	—	—	—	—	—	—	—	—	—	○	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—	○

*1 T, ST, and C cannot be used.

Processing details

These instructions create program with efficient ladder switching by opening/closing common buses in ladders.
Ladder using master control is illustrated below.
(Left: Display on the engineering tool, Right: Actual operation)



(1) Executed only when X0 is on

■MC

- When the execution command of the MC instruction turns ON at the start of master control, the operation result between the MC and MCR instructions is as per the instructions (according to ladder). When the execution command of MC instruction turns OFF, the operation result between the MC and MCR instructions becomes as follows.

Device	Device status
Timer	The count value becomes 0, and both coils and contacts turn OFF.
Counters, retentive timers	Coils turn OFF but the current status of both count values and contacts is maintained.
Devices in OUT instruction	Forcibly turned OFF.
Devices in SET and RST instructions Devices in SFT(P) instruction Devices in basic instructions and applied instructions	Current status is maintained.

Point

When an instruction (e.g. FOR to NEXT instructions etc.) not requiring NO contact instruction is programmed in a ladder using master control, the CPU module executes that instruction regardless of the execution command of this instruction.

- With this instruction, the same nesting (N) number can be used as many times as necessary by changing the device specified by (d).
- When this instruction is ON, the coil of the device specified by (d) turns ON. Also, the coil becomes a double coil when the same device is used by the OUT instruction, for example. So, do not use the device specified by (d) in other instructions.

■MCR

- This instruction indicates the end of the master control range by the master control release instruction.
- Do not prefix this instruction with NO contact instruction.
- Use these (MC and MCR) instructions with same nesting number as a pair. Note, however, that when this instruction is nested at a single location, all master controls can be ended by just one (N) number, the smallest number. (Refer to Caution.)

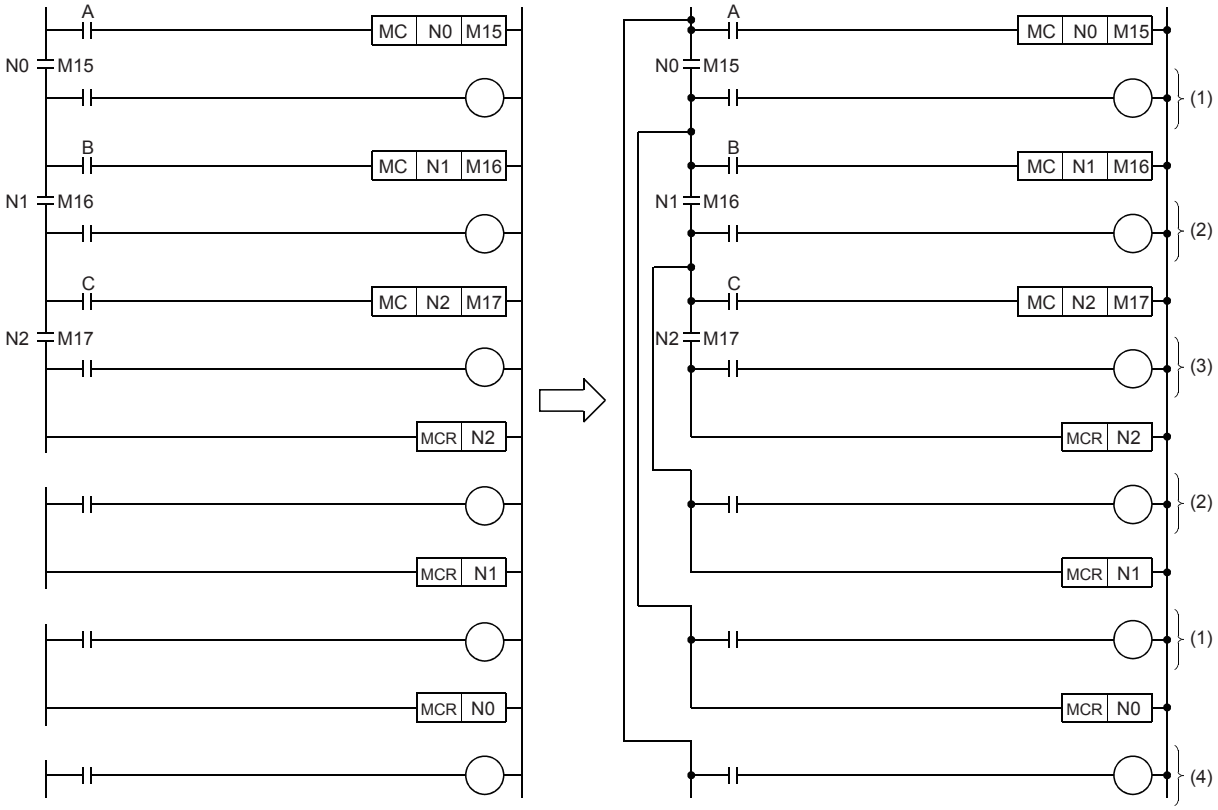
Point

Master control instructions can be used in a nested fashion. Each master control section is distinguished by nesting (N). Nesting is available within the range N0 to N14.

A nested structure allows you to create a ladder for successively restricting program execution conditions.

A nested structure ladder is illustrated as follows:

(Left: Display on the engineering tool, Right: Actual operation)



(1) Executed when A is ON

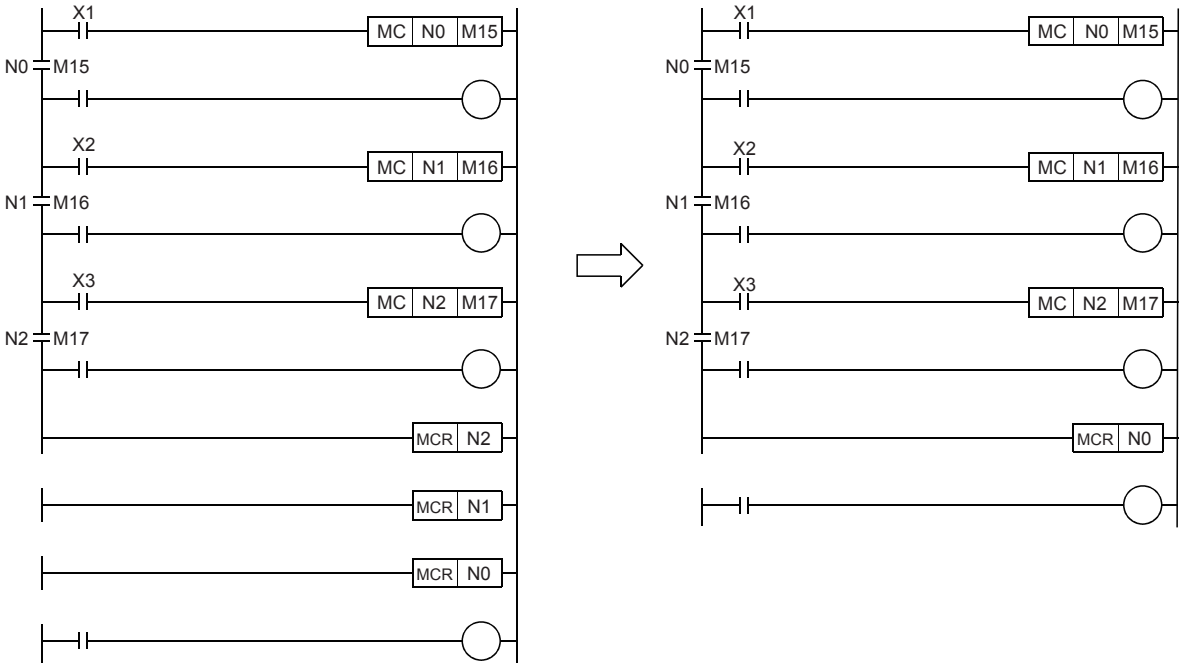
(2) Executed when A and B are ON

(3) Executed when A, B, and C are ON

(4) Regardless of A, B, and C

Precautions

- If an instruction (e.g. LD, LDI) to be connected to the bus is not programmed following the MC instruction, a program structure error (error code: 33E0) occurs.
- These instructions cannot be used in FOR to NEXT, STL to RETSTL, P to RET (SRET), and I to IRET. Also, do not block by I, IRET, FEND, END, RET (SRET), RETSTL, etc. Addition by write during RUN mode results in an error.
- Nesting up to 15 levels (N0 to N14) is possible. When nesting instructions, the MC instruction is used starting from the smallest (N) number and the MCR instruction is started starting from the biggest number. Programming in reverse order does not produce a nested structure and hence the CPU module cannot execute operations properly.
- When the MCR instruction is nested at a single location, all master controls can be ended by just one nesting (N) number, the smallest number.



Operation error

There is no operation error.

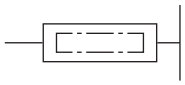
6.6 Termination Instructions

Ending the main routine program

FEND

FX5S FX5UJ FX5U FX5UC

This instruction is used to branch operation of the sequence program by the CJ instruction or to divide the main routine program into a subroutine program or an interrupt program.

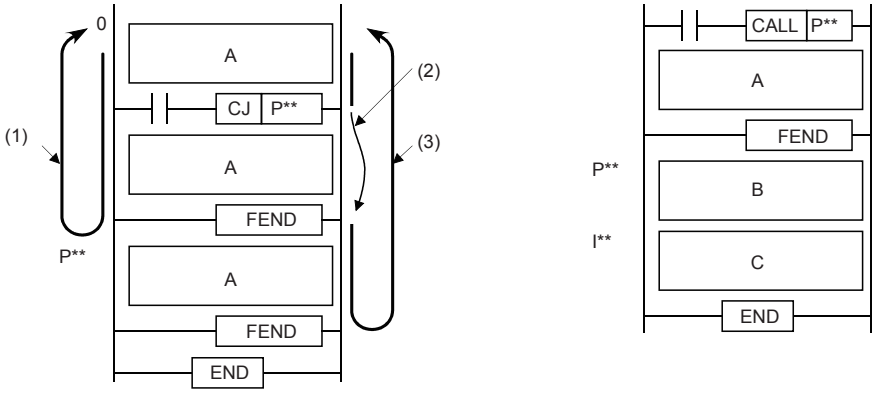
Ladder diagram	Structured text
	Not supported.

FBD/LD
Not supported.

Processing details

- This instruction branches operation of the sequence program by the CJ instruction or dividing the main routine program into subroutine programs and interrupt programs.
- When this instruction is executed, program execution returns to the program at step 0 after output processing, input processing and refreshing of the watchdog timer.
- The sequence program from this instruction onwards can also be displayed as ladder by the engineering tool.

(Left: When the CJ instruction is used, Right: When there are subroutine programs and interrupt programs)

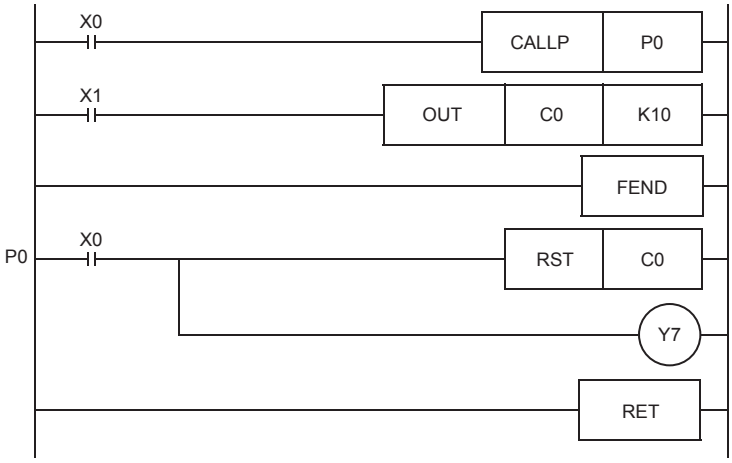


A: Main routine program
B: Subroutine program
C: Interrupt Program
(1): Operation when the CJ instruction is not executed
(2): Jump by the CJ instruction
(3): Operation when the CJ instruction has been executed

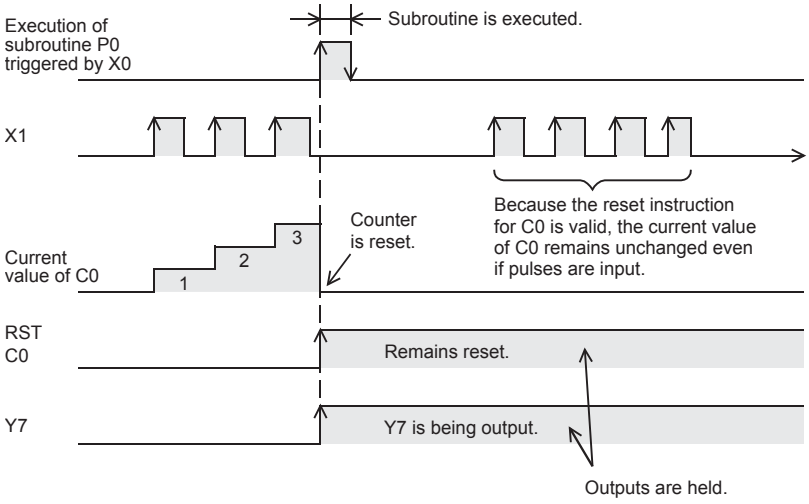
Program example

• Example in which outputs in the subroutine are latched

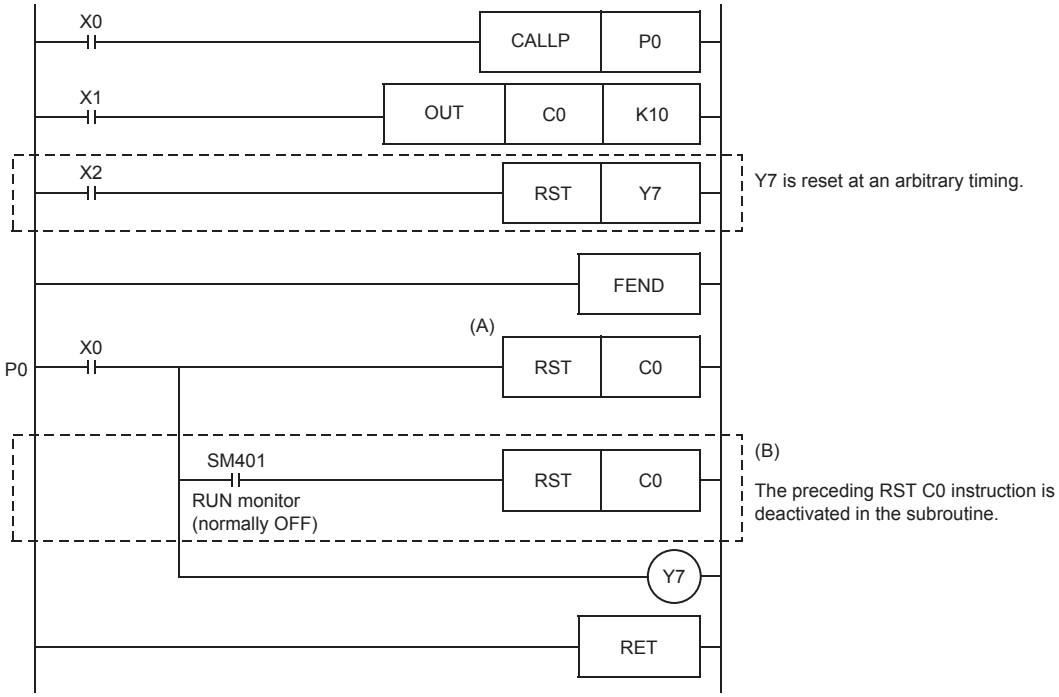
In the following program example, the counter C0 is provided to count X1. When X0 is input, the subroutine P0 is executed only in one scan, and then the counter is reset and Y7 is output.



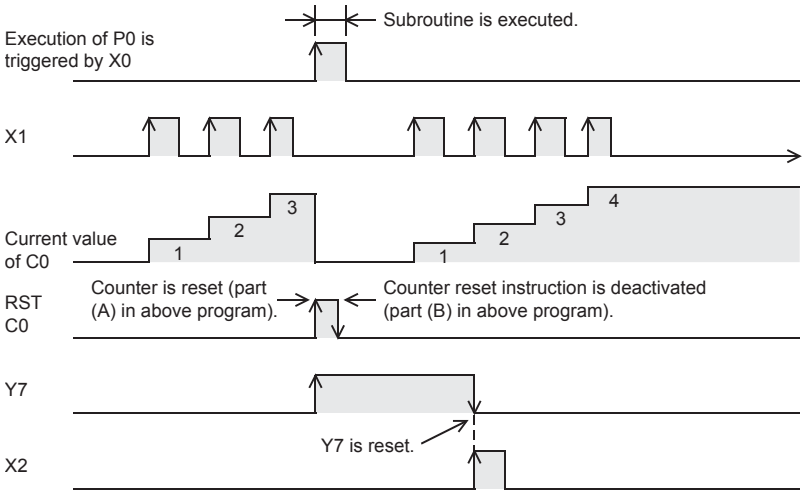
Timing chart



• Example for resetting outputs in the subroutine



Timing chart



Operation error

Error code (SD0/SD8067)	Description
3340H	The FEND instruction is executed before the NEXT instruction after the FOR instruction is executed.
3381H	The FEND instruction is executed before the RET instruction after the CALL(P) instruction is executed.
33E3H	The FEND instruction is programmed between FOR-NEXT.
33E4H	The FEND instruction is programmed between MC-MCR.
33E5H	The FEND instruction is programmed between STL-RETSTL.
33E7H	The FEND instruction is programmed between I-IRET.
3100H	The FEND instruction is programmed in standby type program.
	The FEND instruction is programmed in FB file.

Ending the sequence program

END

FX5S FX5UJ FX5U FX5UC

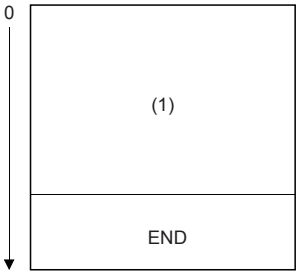
This instruction indicates the end of a program.

Ladder diagram	Structured text
	Not supported.

FBD/LD
Not supported.

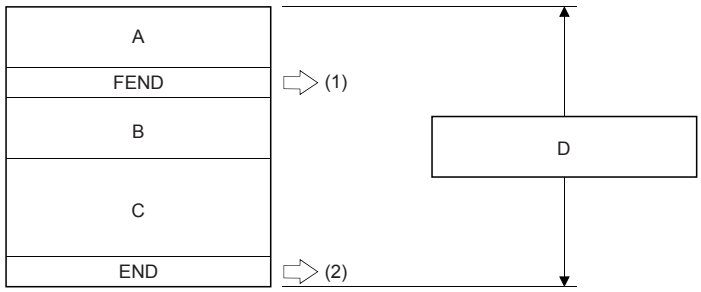
Processing details

- This instruction indicates the end of all programs including the main routine program, subroutine program, and interrupt program. When this instruction is executed, the CPU module ends execution of the currently executing program.



(1): Sequence program

- The first time the RUN is started, execution begins from this instruction.
- This instruction cannot be programmed midway during the main sequence program. When this processing is required midway during the program, use the FEND instruction.
- When programming is performed using the engineering tool in ladder edit mode, the END instruction is automatically input and cannot be edited.
- The following illustrates how the END and FEND instructions are used properly when a program contains a main routine program, subroutine program, and interrupt program.



A: Main routine program
B: Subroutine program
C: Interrupt Program
D: Main sequence program area
(1): FEND instruction is required.
(2): END instruction is required.

Point

The END instruction executed while a program is divided into multiple program blocks indicates the end of a program block.

The END instruction executed for END processing is executed at the end of the last executed program registered in the program settings.

Operation error

Error code (SD0/SD8067)	Description
3340H	The END instruction is executed before the NEXT instruction after the FOR instruction is executed.
3381H	The END instruction is executed before the RET instruction after the CALL(P) instruction is executed.
33E3H	The END instruction is programmed between FOR-NEXT.
33E4H	The END instruction is programmed between MC-MCR.
33E5H	The END instruction is programmed between STL-RETSTL.
33E7H	The END instruction is programmed between I-IRET.

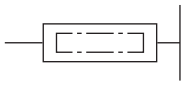
6.7 Stop Instruction

Stopping the sequence program

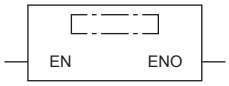
STOP

FX5S FX5UJ FX5U FX5UC

This instruction resets outputs (Y) and stops operation of the CPU module when the execution command turns ON. (This operation is the same as setting the switch to STOP.)

Ladder diagram	Structured text
	ENO:=STOP(EN);

FBD/LD



Processing details

- This instruction resets outputs (Y) and stops operation of the CPU module when the execution command turns ON. (This operation is the same as setting the switch to STOP.)
- To restart operation of the CPU module after this instruction is executed, return the switch from RUN→STOP and set it to RUN again.

Operation error

Error code (SD0/SD8067)	Description
3340H	The STOP instruction is executed before the NEXT instruction is executed after the FOR instruction is executed.
3381H	The STOP instruction is executed before the RET instruction is executed after the CALL(P) or XCALL(P) instruction is executed.
3582H	The STOP instruction is executed before the IRET instruction is executed in the interruption program.

7 BASIC INSTRUCTIONS

7.1 Comparison Operation Instructions

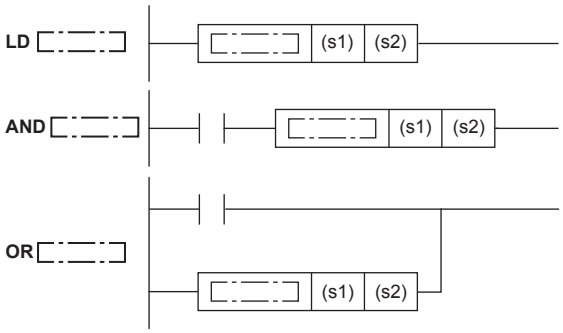
Comparing 16-bit binary data

LD□(_U), AND□(_U), OR□(_U)

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation between the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2). (Devices are used as NO contacts.)

Ladder diagram



("=(_U)", "<>(_U)", ">(_U)", "<=(_U)", "<(_U)", ">=(_U)" enters □.)

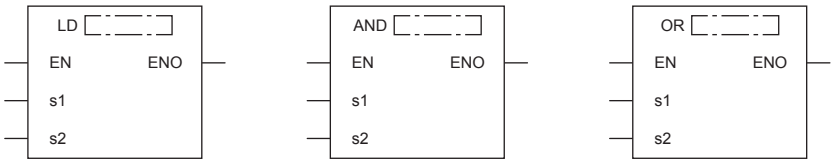
Structured text[*1]

ENO:=LD_□(EN,s1,s2);
ENO:=AND_□(EN,s1,s2);
ENO:=OR_□(EN,s1,s2);

ENO:=LD_□_U(EN,s1,s2);
ENO:=AND_□_U(EN,s1,s2);
ENO:=OR_□_U(EN,s1,s2);

("EQ", "NE", "GT", "LE", "LT", "GE" enters □.)[*2]

FBD/LD



("_EQ(_U)", "_NE(_U)", "_GT(_U)", "_LE(_U)", "_LT(_U)", "_GE(_U)" enters □.)[*2]

*1 Supported by engineering tool version "1.035M" and later.
*2 EQ is =, NE is <>, GT is >, LE is <=, LT is <, and GE is >=.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	LD□, AND□, OR□	Comparison data or device where the comparison data is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	LD□_U, AND□_U, OR□_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	LD□, AND□, OR□	Comparison data or device where the comparison data is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	LD□_U, AND□_U, OR□_U		0 to 65535	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions perform a comparison operation between the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2). (Devices are used as NO contacts.)
- The following table lists the comparison operation results of each instruction.

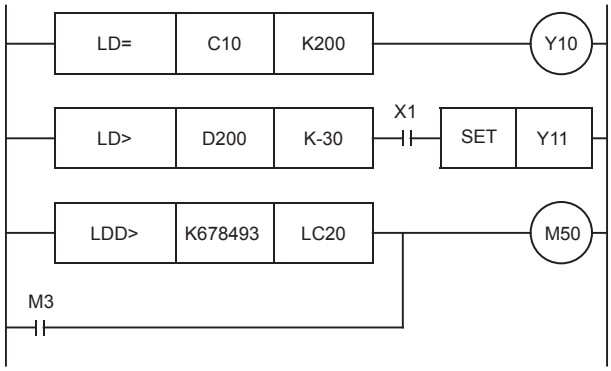
Instruction symbol	Condition	Result
=(_U)	(s1)=(s2)	Conductive state
<>(_U)	(s1)≠(s2)	
>(_U)	(s1)>(s2)	
<=(_U)	(s1)≤(s2)	
<(_U)	(s1)<(s2)	
>=(_U)	(s1)≥(s2)	
=(_U)	(s1)≠(s2)	Non-conductive state
<>(_U)	(s1)=(s2)	
>(_U)	(s1)≤(s2)	
<=(_U)	(s1)>(s2)	
<(_U)	(s1)≥(s2)	
>=(_U)	(s1)<(s2)	

Precautions

- When the most significant bit is "1" in the data stored in (s1) or (s2), it is regarded as a negative binary value for comparison. (Excluding unsigned operation)

Program example

- LD□(_U)

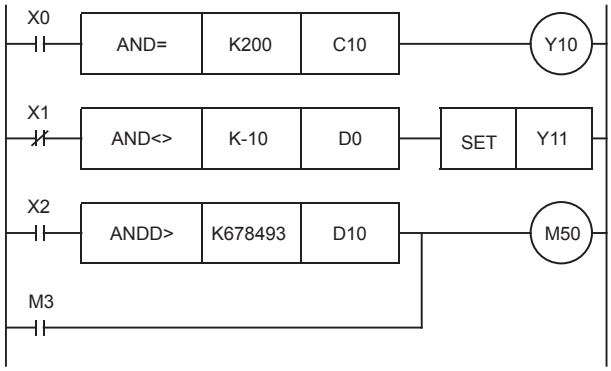


When the current value of C10 is "200", Y10 turns ON.

When the content value of D200 is "-29" or higher and X1 is ON, Y11 is set.

When the content value of LC20 is less than "K678493" or when M3 turns ON, M50 turns ON.

- AND□(_U)

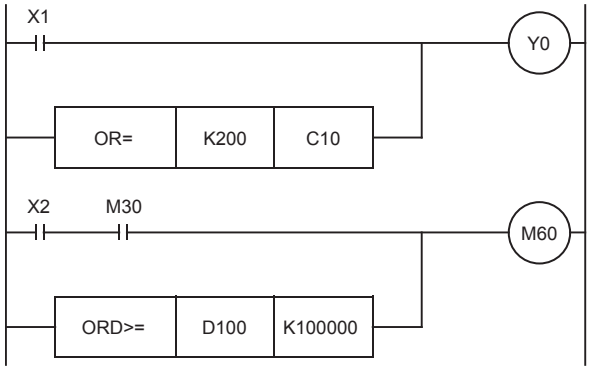


When X0 is ON and the current value of C10 is "200", Y10 turns ON.

When X1 is OFF and the content value of D0 is not "-10", Y11 is set.

When X2 is ON, and the contents value of D11 and D10 are less than "K678493", or when M3 turns ON, M50 turns ON.

- OR□(_U)



When X1 turns ON or when the current value of C10 is "200", Y0 turns ON.

When X2 and M30 turn ON or when the contents value of D101 and D100 are more than "K100000", M60 turns ON.

Operation error

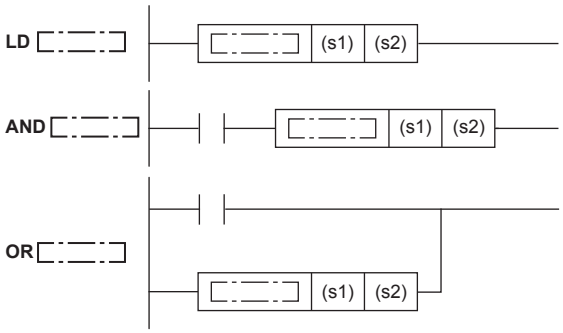
There is no operation error.

Comparing 32-bit binary data

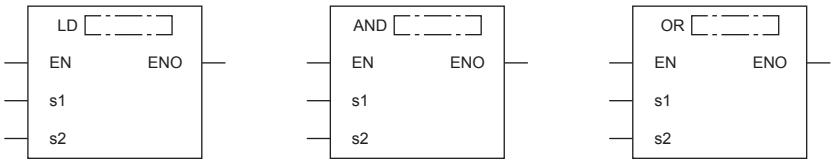
LDD□(_U), ANDD□(_U), ORD□(_U)

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation between the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2). (Devices are used as NO contacts)

Ladder diagram	Structured text[*1]	
	ENO:=LDD_□(EN,s1,s2); ENO:=ANDD_□(EN,s1,s2); ENO:=ORD_□(EN,s1,s2);	ENO:=LDD_□_U(EN,s1,s2); ENO:=ANDD_□_U(EN,s1,s2); ENO:=ORD_□_U(EN,s1,s2);
("D=(_U)", "D<>(_U)", "D>(_U)", "D<=(_U)", "D<(_U)", "D>=(_U)" enters □.)	("EQ", "NE", "GT", "LE", "LT", "GE" enters □.)[*2]	

FBD/LD



("D_EQ(_U)", "D_NE(_U)", "D_GT(_U)", "D_LE(_U)", "D_LT(_U)", "D_GE(_U)" enters □.)[*2]

*1 Supported by engineering tool version "1.035M" and later.

*2 EQ is =, NE is <>, GT is >, LE is <=, LT is <, and GE is >=.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	LDD□, ANDD□, ORD□	Comparison data or head device where the comparison data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	LDD□_U, ANDD□_U, ORD□_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	LDD□, ANDD□, ORD□	Comparison data or head device where the comparison data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	LDD□_U, ANDD□_U, ORD□_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—

Processing details

- These instructions perform a comparison operation between the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2). (Devices are used as NO contacts)
- The following table lists the comparison operation results of each instruction.

Instruction symbol	Condition	Result
D=(_U)	(s1)=(s2)	Conductive state
D<>(_U)	(s1)≠(s2)	
D>(_U)	(s1)>(s2)	
D<=(_U)	(s1)≤(s2)	
D<(_U)	(s1)<(s2)	
D>=(_U)	(s1)≥(s2)	
D=(_U)	(s1)≠(s2)	Non-conductive state
D<>(_U)	(s1)=(s2)	
D>(_U)	(s1)≤(s2)	
D<=(_U)	(s1)>(s2)	
D<(_U)	(s1)≥(s2)	
D>=(_U)	(s1)<(s2)	

Precautions

- When the most significant bit is "1" in the data stored in (s1) or (s2), it is regarded as a negative binary value for comparison. (Excluding unsigned operation)
- For comparison of 32-bit counter (LC), specify an instruction (LDD=, etc.) that handles 32-bit data. If an instruction (LD=, etc.) that handles 16-bit data is specified, a program error or operation error occurs. (Same applies for index device (LZ) as well.)

Program example

For a program example, refer to ☞ Page 189 Comparing 16-bit binary data.

Operation error

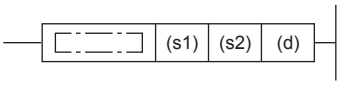
There is no operation error.

Comparison output 16-bit binary data

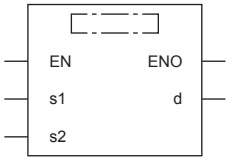
CMP(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation between the 16-bit binary data in the devices specified by (s1) and (s2).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	CMP(P)	Comparison value data or the device where the comparison value data is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	CMP(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	CMP(P)	Comparison source data or the device where the comparison source data is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	CMP(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)		The starting bit device to which the comparison result is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

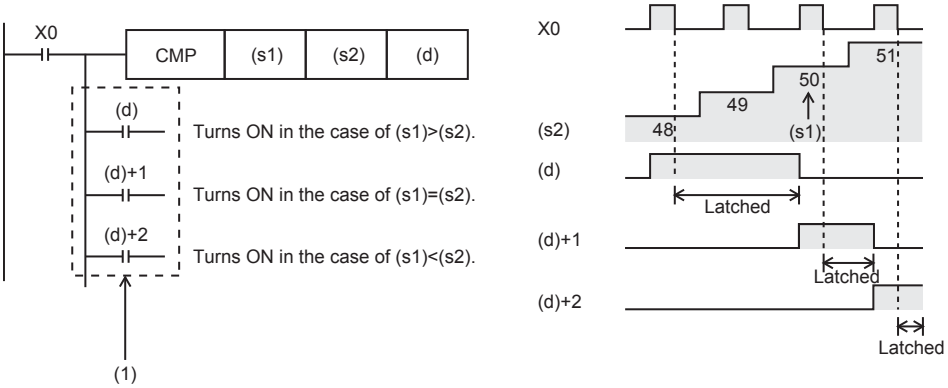
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions perform a comparison operation between the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2) and according to the result (small, equal, large), (d), (d) + 1, or (d) + 2 is turned ON.
- (s1) and (s2) are handled as binary values within the range of above data setting.
- Large and small comparison is executed algebraically.
  - With sign…-10 (FFF6H) < 2 (0002H)
  - Without sign…32767 (7FFFH) < 65280 (FF00H)



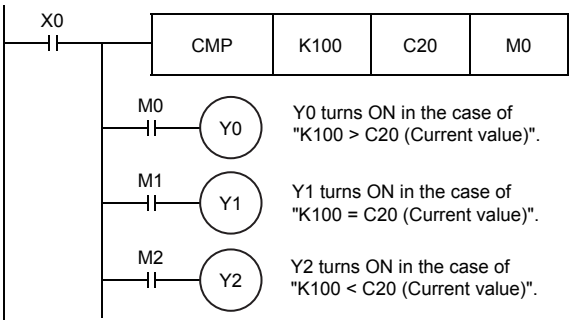
(1): Even if the command input turns OFF and the CMP instruction is not executed, (d) to (d)+2 latches the status just before the command input turns from ON to OFF.

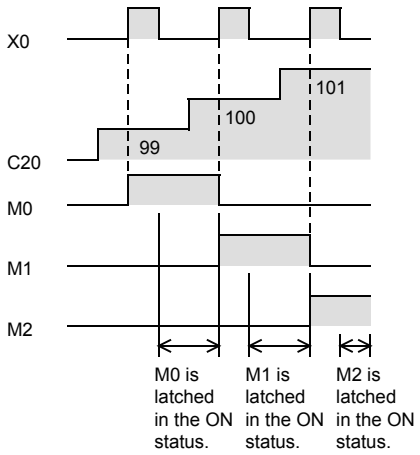
Precautions

Three devices are occupied from the device specified in (d). Make sure that these devices are not used in other controls.

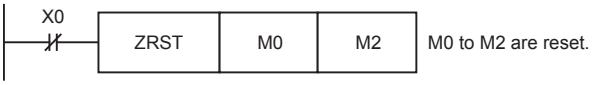
Program example

- When comparing the current value of a counter





If it is necessary to clear the comparison result when the instruction is not executed, add the following contents under the above program.



Operation error

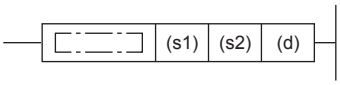
Error code (SD0/SD8067)	Description
2820H	The range of 3 points of data starting from the device specified by (d) exceeds said device.

Comparison output 32-bit binary data

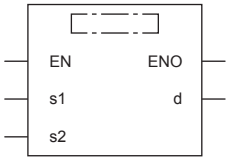
DCMP(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation between the 32-bit binary data in the devices specified by (s1) and (s2).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DCMP(P)	Comparison value data or the head device where the comparison value data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DCMP(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DCMP(P)	Comparison source data or the head device where the comparison source data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DCMP(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)		The starting bit device to which the comparison result is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

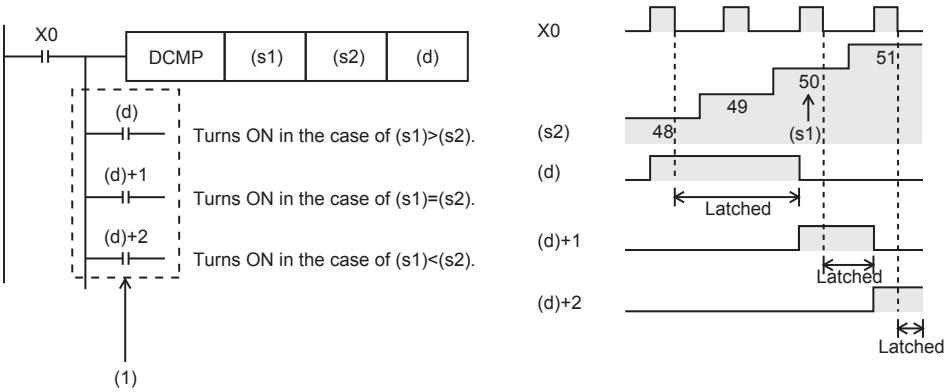
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions perform a comparison operation between the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2) and according to the result (small, equal, large), (d), (d) + 1, or (d) + 2 is turned ON.
- (s1) and (s2) are handled as binary values within the range of above data setting.
- Large and small comparison is executed algebraically.
  - With sign…-125400 (FFFE1628H) < 224566 (00036D36H)
  - Without sign…16776690 (00FFFDF2H) < 4294967176 (FFFFFF88H)



(1): Even if the command input turns OFF and the DCMP instruction is not executed, (d) to (d)+2 latches the status just before the command input turns from ON to OFF.

Precautions

Three devices are occupied from the device specified in (d). Make sure that these devices are not used in other controls.

Operation error

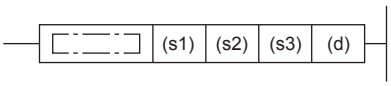
Error code (SD0/SD8067)	Description
2820H	The range of 3 points of data starting from the device specified by (d) exceeds said device.

Comparing 16-bit binary data band

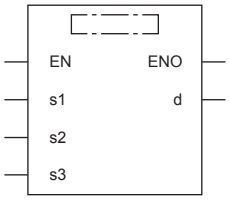
ZCP(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation on the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2) with the 16-bit binary data in the device specified by comparison source (s3), and output the comparison result (below, within zone, above) to the device specified by (d) onwards.

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	ZCP(P)	Lower limit comparison data or the device where the comparison data is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	ZCP(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	ZCP(P)	Upper limit comparison data or the device where the comparison data is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	ZCP(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s3)	ZCP(P)	Comparison source data or the device where the comparison source data is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	ZCP(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)		The starting bit device to which the comparison result is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

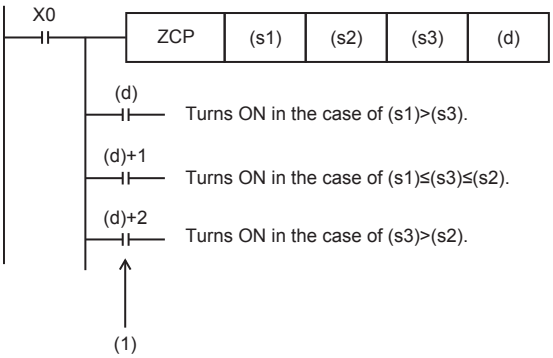
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions perform a comparison operation on the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2) with the 16-bit binary data in the device specified by comparison source (s3), and according to the comparison result (below, within zone, above), (d), (d) + 1, or (d) + 2 is turned ON. (s1), (s2), and (s3) are handled as binary values within the range of above data setting. Large and small comparison is executed algebraically.

- Large and small comparison is executed algebraically.
  - With sign…-10 (FFF6H) < 2 (0002H) < 10 (000AH)
  - Without sign…0 (0000H) < 32767 (7FFFH) < 40000 (9C40H)

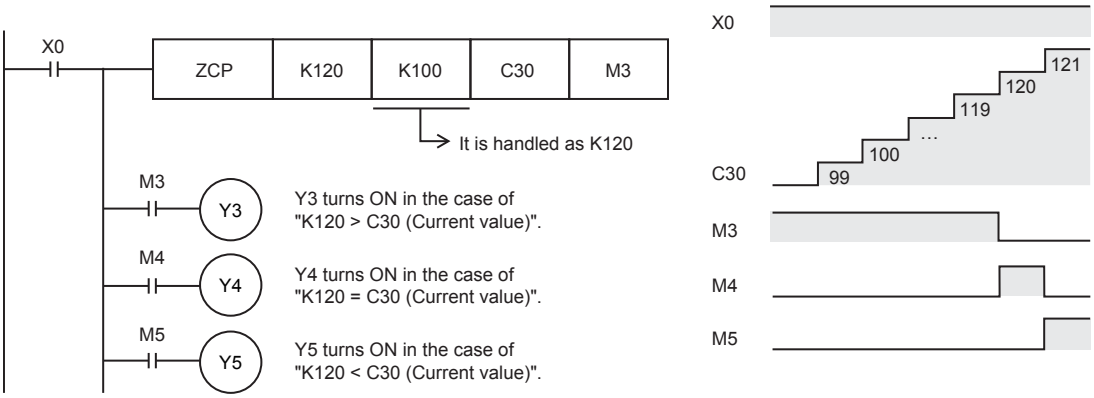


(1): Even if the command input turns OFF and the ZCP instruction is not executed, (d) to (d)+2 latches the status just before the command input turns from ON to OFF.

Precautions

- Set (s1) to a value less than (s2).

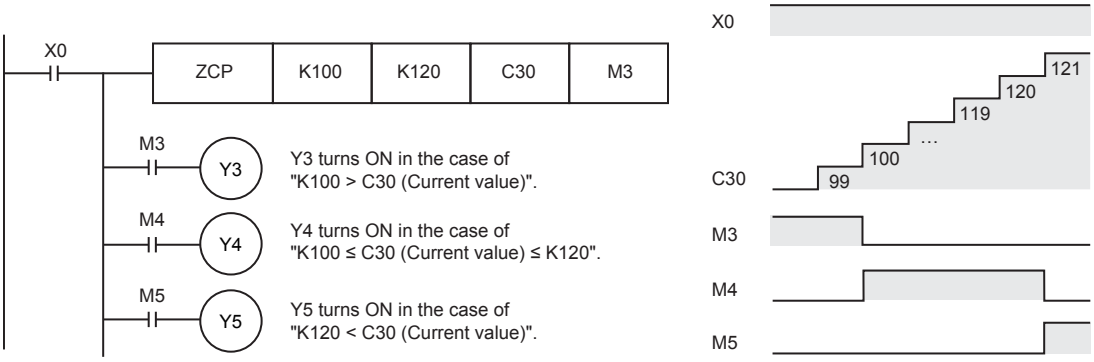
When the lower comparison value (s1) is larger than the upper comparison value (s2)



- Three devices are occupied from the device specified in (d). Make sure that these devices are not used in other controls.

Program example

When the lower comparison value (s1) is smaller than the upper comparison value (s2)



Operation error

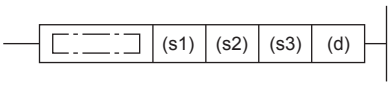
Error code (SD0/SD8067)	Description
2820H	The range of 3 points of data starting from the device specified by (d) exceeds said device.

Comparing 32-bit binary data band

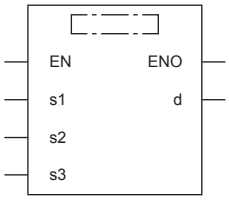
DZCP(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation on the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2) with the 32-bit binary data in the device specified by comparison source (s3), and output the comparison result (below, within zone, above) to the device specified by (d) onwards.

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DZCP(P)	Lower limit comparison data or the head device where the comparison data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DZCP(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DZCP(P)	Upper limit comparison data or the head device where the comparison data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DZCP(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s3)	DZCP(P)	Comparison source data or the head device where the comparison source data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DZCP(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)		The starting bit device to which the comparison result is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

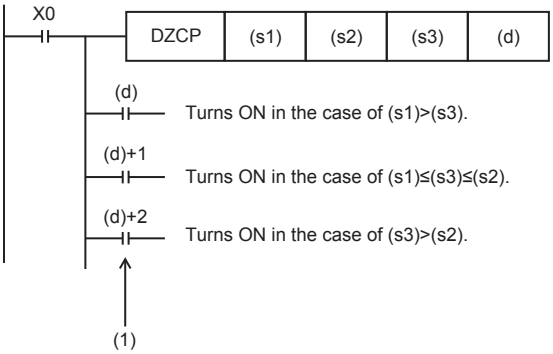
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions perform a comparison operation on the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2) with the 32-bit binary data in the device specified by comparison source (s3), and according to the comparison result (below, within zone, above), (d), (d) + 1, or (d) + 2 is turned ON. (s1), (s2), and (s3) are handled as binary values within the range of above data setting.
- Large and small comparison is executed algebraically.
  - With sign…-125400 (FFFE1628H) < 22466 (000057C2H) < 1015444 (000F7E94H)
  - Without sign…0 (00000000H) < 2147483647 (7FFFFFFFH) < 4026531840 (F0000000H)



(1): Even if the command input turns OFF and the DZCP instruction is not executed, (d) to (d)+2 latches the status just before the command input turns from ON to OFF.

Precautions

- Set (s1) to a value less than (s2).
- Three devices are occupied from the device specified in (d). Make sure that these devices are not used in other controls.

Operation error

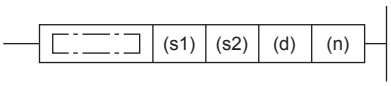
Error code (SD0/SD8067)	Description
2820H	The range of 3 points of data starting from the device specified by (d) exceeds said device.

Comparing 16-bit binary block data

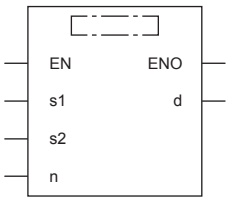
BKCMP□(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation between (n) point(s) of 16-bit binary data in the device starting from the one specified by (s1) and (n) point(s) of 16-bit binary data in the device starting from the one specified by (s2), and store the operation result in the device specified by (d).

Ladder diagram	Structured text[*1]	
	ENO:=BKCMP_□(EN,s1,s2,n,d); ENO:=BKCMP_□P(EN,s1,s2,n,d);	ENO:=BKCMP_□_U(EN,s1,s2,n,d); ENO:=BKCMP_□P_U(EN,s1,s2,n,d);
("BKCMP=(P)(_U)", "BKCMP<>(P)(_U)", "BKCMP>(P)(_U)", "BKCMP<=(P)(_U)", "BKCMP<(P)(_U)", "BKCMP>=(P)(_U)" enters □.)	("EQ", "NE", "GT", "LE", "LT", "GE" enters □.)[*2]	

FBD/LD



("BKCMP_EQ(P)(_U)", "BKCMP_NE(P)(_U)", "BKCMP_GT(P)(_U)", "BKCMP_LE(P)(_U)", "BKCMP_LT(P)(_U)", "BKCMP_GE(P)(_U)" enters □.)[*2]

*1 Supported by engineering tool version "1.035M" and later.
*2 EQ is =, NE is <>, GT is >, LE is <=, LT is <, and GE is >=.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	BKCMP□(P)	Comparison data or the device where the comparison data is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	BKCMP□(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	BKCMP□(P)	Device where the comparison source data is stored	—	16-bit signed binary	ANY16_S
	BKCMP□(P)_U		—	16-bit unsigned binary	ANY16_U
(d)		Head device storing comparison result	—	Bit	ANY_BOOL
(n)		Number of data to be compared	0 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

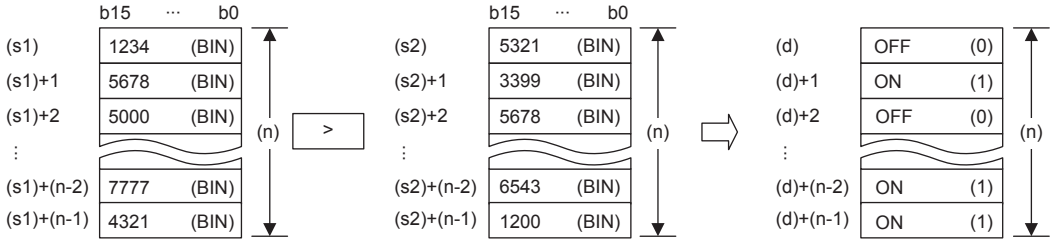
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○[*1]	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

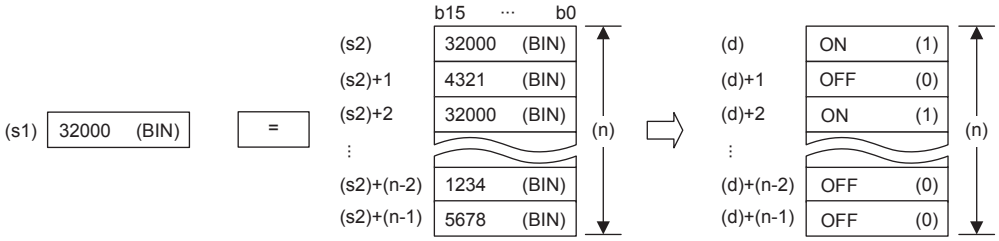
*1 T, ST, and C cannot be used.

Processing details

- These instructions perform a comparison operation between (n) point(s) of 16-bit binary data in the device starting from the one specified by (s1) and (n) point(s) of 16-bit binary data in the device starting from the one specified by (s2), and store the comparison result in (n) point(s) of data starting from the device specified by (d).
- The relevant devices of (n) point(s) of data starting from the device specified by (d) are turned ON when the comparison conditions are met and turned OFF when the comparison conditions are not met.



- Comparison operation is performed in units of 16 bits.
- A constant can be directly specified in (s1).



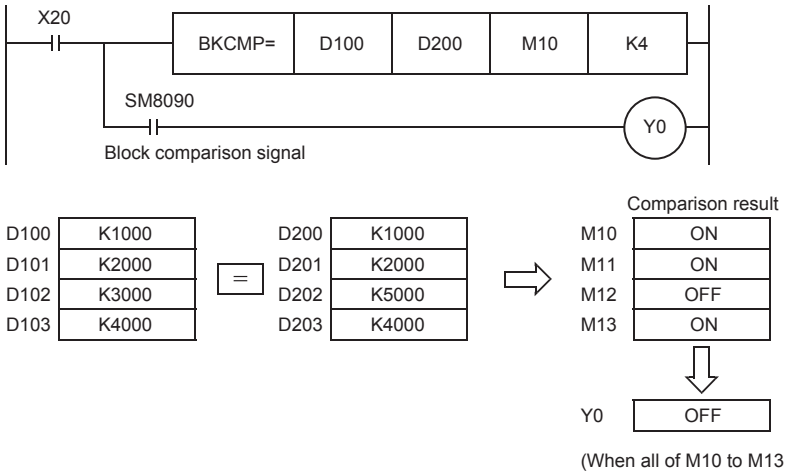
- The following table lists the comparison operation results of each instruction.

Instruction symbol	Condition	Result
BKCMP=(P)(_U)	(s1)=(s2)	On(1)
BKCMP<>(P)(_U)	(s1)≠(s2)	
BKCMP>(P)(_U)	(s1)>(s2)	
BKCMP<=(P)(_U)	(s1)≤(s2)	
BKCMP<(P)(_U)	(s1)<(s2)	
BKCMP>=(P)(_U)	(s1)≥(s2)	
BKCMP=(P)(_U)	(s1)≠(s2)	Off(0)
BKCMP<>(P)(_U)	(s1)=(s2)	
BKCMP>(P)(_U)	(s1)≤(s2)	
BKCMP<=(P)(_U)	(s1)>(s2)	
BKCMP<(P)(_U)	(s1)≥(s2)	
BKCMP>=(P)(_U)	(s1)<(s2)	

- When the comparison operation result is all ON (1) in all (n) point(s) starting from (d), SM704 and SM8090 (block comparison signal) turns ON.

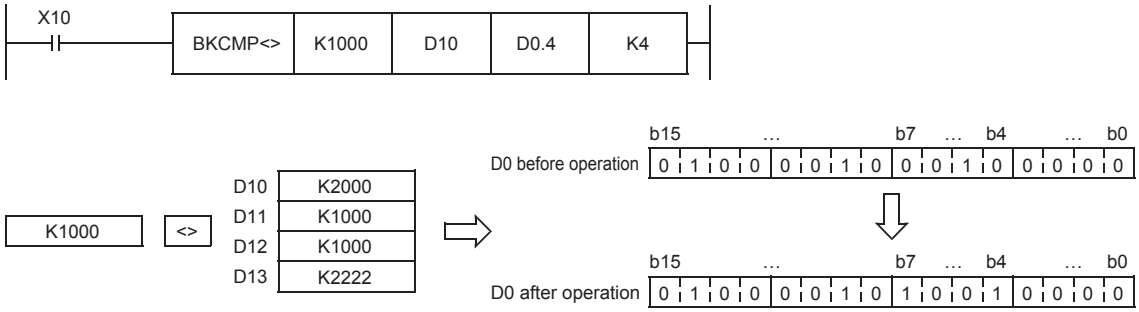
Program example

- In the program shown below, four 16-bit binary data starting from D100 are compared with four 16-bit binary data starting from D200 by BKCMP= instruction when X20 is set to ON, and the comparison result is stored in four points starting from M10. When the comparison result is "ON (1)" in all of the four points starting from M10, SM8090 turns ON and Y0 is set to ON.



(When all of M10 to M13 are ON, Y0 is set to ON.)

- In the program shown below, the constant K1000 is compared with four data starting from D10 when X10 is set to ON, and the comparison result is stored in b4 to b7 of D0.



Operation error

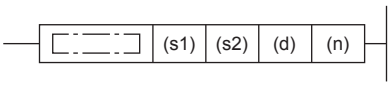
Error code (SD0/SD8067)	Description
2820H	The (n) point(s) starting from the device specified by (s1), (s2), and (d) exceeds said device.
2821H	When (d) specifies "D□.b", the data register of (d) and the (n) point(s) of data starting from the device specified by (s1) overlap.
	When (d) specifies "D□.b", the data register of (d) and the (n) point(s) of data starting from the device specified by (s2) overlap.

Comparing 32-bit binary block data

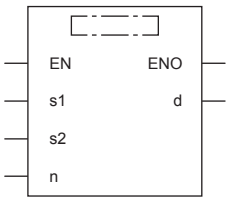
DBKCMP□(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation between the (n) point(s) of 32-bit binary data starting from the device specified by (s1) and the (n) point(s) of 32-bit binary data starting from the device specified by (s2), and store the operation result in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=DBKCMP_□(EN,s1,s2,n,d); ENO:=DBKCMP_□P(EN,s1,s2,n,d);	ENO:=DBKCMP_□_U(EN,s1,s2,n,d); ENO:=DBKCMP_□P_U(EN,s1,s2,n,d);
("DBKCMP=(P)(_U)", "DBKCMP<>(P)(_U)", "DBKCMP>(P)(_U)", "DBKCMP<=(P)(_U)", "DBKCMP<(P)(_U)", "DBKCMP>=(P)(_U)" enters □.)	("EQ", "NE", "GT", "LE", "LT", "GE" enters □.)*2	

FBD/LD



("DBKCMP_EQ(P)(_U)", "DBKCMP_NE(P)(_U)", "DBKCMP_GT(P)(_U)", "DBKCMP_LE(P)(_U)", "DBKCMP_LT(P)(_U)", "DBKCMP_GE(P)(_U)" enters □.)*2

*1 Supported by engineering tool version "1.035M" and later.
*2 EQ is =, NE is <>, GT is >, LE is <=, LT is <, and GE is >=.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DBKCMP□(P)	Comparison data or the head device where the comparison data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DBKCMP□(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DBKCMP□(P)	Head device where the comparison source data is stored	—	32-bit signed binary	ANY32_S
	DBKCMP□(P)_U		—	32-bit unsigned binary	ANY32_U
(d)		Head device storing comparison result	—	Bit	ANY_BOOL
(n)		Number of data to be compared	0 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

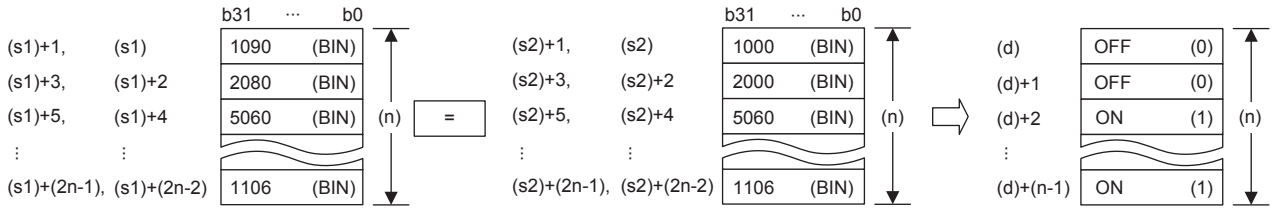
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	—	—	○	—	○	○	—	—	—
(s2)	—	○	—	—	○	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

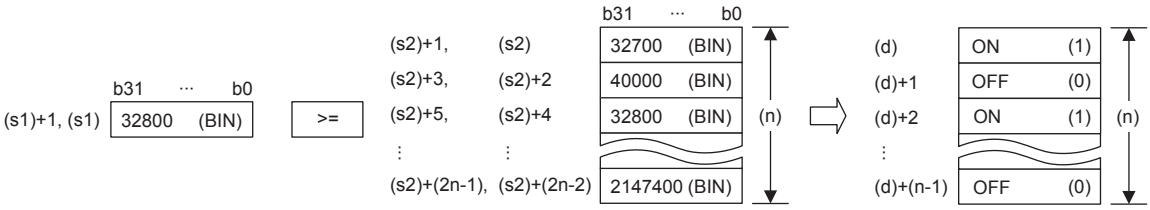
*1 T, ST, and C cannot be used.

Processing details

- These instructions perform a comparison operation between (n) point(s) of 32-bit binary data starting from the device specified by (s1) and (n) point(s) of 32-bit binary data starting from the device specified by (s2), and store the comparison result in (n) point(s) of data starting from the device specified by (d).
- The relevant devices of (n) point(s) of data starting from the device specified by (d) are turned ON when the comparison conditions are met and turned OFF when the comparison conditions are not met.



- Comparison operation is performed in units of 32 bits.
- A constant can be directly specified in (s1).



- (d) is specified outside the device range of (n) point(s) of data starting from the one specified by (s1) and outside the device range of (n) point(s) of data starting from the one specified by (s2).
- The following table lists the comparison operation results of each instruction.

Instruction symbol	Condition	Result
DBKCMP=(P)(_U)	(s1)=(s2)	On(1)
DBKCMP<>(P)(_U)	(s1)≠(s2)	
DBKCMP>(P)(_U)	(s1)>(s2)	
DBKCMP<=(P)(_U)	(s1)≤(s2)	
DBKCMP<(P)(_U)	(s1)<(s2)	
DBKCMP>=(P)(_U)	(s1)≥(s2)	
DBKCMP=(P)(_U)	(s1)≠(s2)	Off(0)
DBKCMP<>(P)(_U)	(s1)=(s2)	
DBKCMP>(P)(_U)	(s1)≤(s2)	
DBKCMP<=(P)(_U)	(s1)>(s2)	
DBKCMP<(P)(_U)	(s1)≥(s2)	
DBKCMP>=(P)(_U)	(s1)<(s2)	

- When the comparison operation result is all ON (1) in all (n) point(s) starting from (d), SM704 and SM8090 (block comparison signal) turns ON.

Precautions

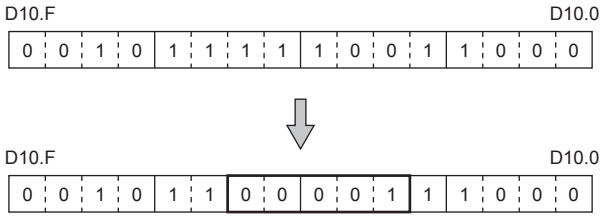
If a 32-bit counter (high-speed counter included) is used, make sure to compare using the 32-bit operation (DBKCMP=, DBKCMP>, DBKCMP<, etc.).

Operation error

Error code (SD0/SD8067)	Description
2820H	The (n) × 2 points of data starting from the device specified by (s1) and (s2) or the (n) point(s) of data starting from the device specified by (d) exceeds said device.
2821H	When (d) specifies "D□.b", the (n) point(s) of data starting from the device specified by (d) and the device range of the (n) × 2 points of data starting from the device specified by (s1) overlap.
	When (d) specifies "D□.b", the (n) point(s) of data starting from the device specified by (d) and the device range of the (n) × 2 points of data starting from the device specified by (s2) overlap.

Point

When bit is specified for word device, devices other than the bit-specified word devices where operation result is stored will not change.



7.2 Arithmetic Operation Instructions

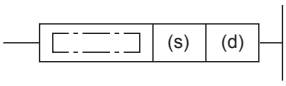
Adding 16-bit binary data

+(P)(_U) instruction and ADD(P)(_U) instruction can be used for addition of 16-bit binary data.

+(P)(_U) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions add the 16-bit binary data in the device specified by (d) and the 16-bit binary data in the device specified by (s), and store the result in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 210 +(P)(_U) [using three operands]

FBD/LD
Not supported. ☞ Page 210 +(P)(_U) [using three operands]

Setting data

■Descriptions, ranges, and data types

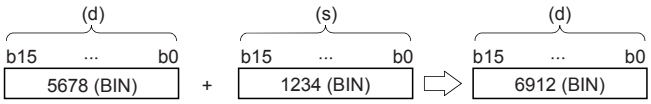
Operand		Description	Range	Data type	Data type (label)
(s)	+(P)	Addend data or the device where the data that is added to another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	+(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	+(P)	Device where the data to which another is added is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	+(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

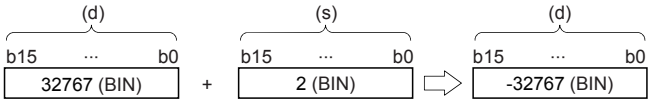
Processing details

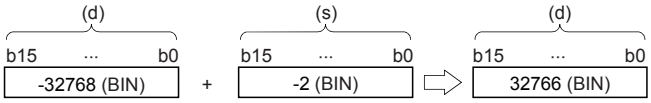
- These instructions add the 16-bit binary data in the device specified by (s) to the 16-bit binary data in the device specified by (d), and store the addition result in the device specified by (d).



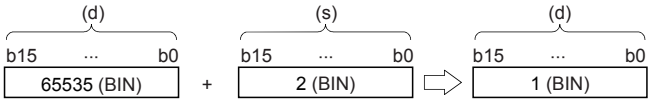
- When underflow or overflow occurs in the operation result, the following processing is executed. In this case, the carry flag (SM700, SM8022) does not turn ON.

In case of +(P)





In case of +(P)(_U)



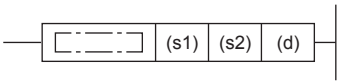
Operation error

There is no operation error.

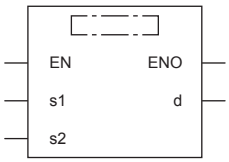
+(P)(_U) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions add the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text	
	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



("PLUS", "PLUSP", "PLUS_U", "PLUSP_U" enters □.)

Setting data

■Descriptions, ranges, and data types

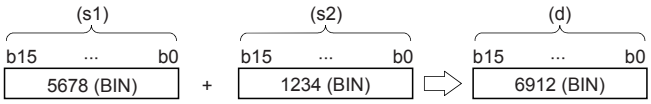
Operand		Description	Range	Data type	Data type (label)
(s1)	+(P)	Augend data or the device where the data to which another is added is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	+(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	+(P)	Addend data or the device where the data that is added to another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	+(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	+(P)	Device for storing the operation result	—	16-bit signed binary	ANY16_S
	+(P)_U		—	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

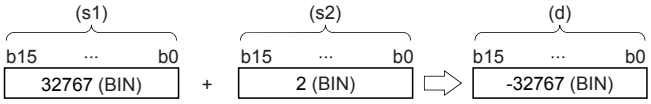
Processing details

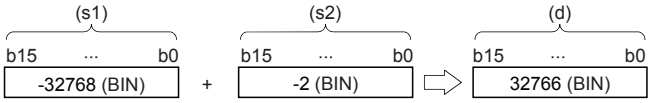
- These instructions add the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2), and store the addition result in the device specified by (d).



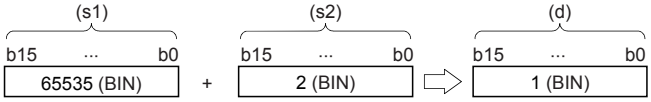
- When underflow or overflow occurs in the operation result, the following processing is executed. In this case, the carry flag (SM700, SM8022) does not turn ON.

In case of +(P)





In case of +(P)(_U)



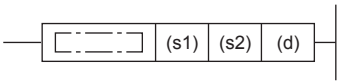
Operation error

There is no operation error.

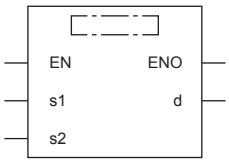
ADD(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=ADDP(EN,s1,s2,d);	ENO:=ADD_U(EN,s1,s2,d); ENO:=ADDP_U(EN,s1,s2,d);

FBD/LD*1



("ADDP", "ADD_U", "ADDP_U" enters □.)

*1 The ADD instruction is not supported by the ST language and the FBD/LD language. Use ADD of the standard function.
☞ Page 1300 ADD(_E)

Setting data

■Descriptions, ranges, and data types

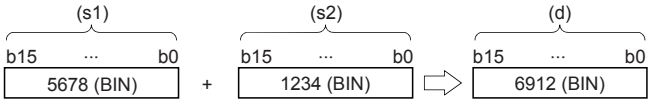
Operand		Description	Range	Data type	Data type (label)
(s1)	ADD(P)	Addend data or the device where the data that is added to another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	ADD(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	ADD(P)	Addend data or the device where the data that is added to another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	ADD(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	ADD(P)	Device for storing the operation result	—	16-bit signed binary	ANY16_S
	ADD(P)_U		—	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

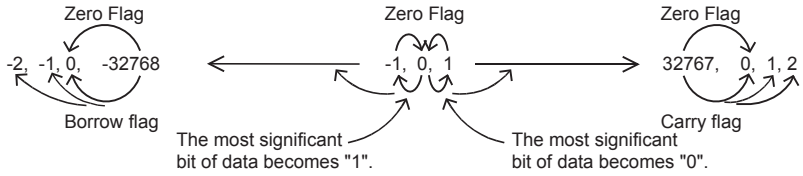
Processing details

- These instructions add the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2), and store the addition result in the device specified by (d).



■Relationship between the flag operation and the sign (positive or negative) of a numeric value

Device	Name	Description
SM700, SM8022	Carry	When the operation result exceeds the upper limit of the data setting range, the carry flag is turned ON.
SM8020	Zero	When the operation result is 0, the zero flag is turned ON.
SM8021	Borrow	When the operation result is less than the lower limit of the data setting range, the borrow flag is turned ON.



Precautions

■When specifying the same device in the source and destination

The same device number can be specified for both the source and the destination. In this case, note that the addition result changes in every operation cycle if a continuous operation type ADD instruction is used.

■Difference between ADD(P) instruction, +(P) instruction, and INC(P) instruction in a program for adding "+1"

When ADD(P) instruction is used to add 1 to the contents of D0 every time X1 turns from OFF to ON, ADD(P) instruction is similar to +(P) instruction and INC(P) instruction described later except for the contents shown in the table below

		ADD(P) instruction	+(P) instruction, INC(P) instruction
Flag (zero, borrow or carry)		Operates	Does not operate
Operation result	(s)+1=(d)	+32767 → 0 → +1 → +2 →…	+32767 → -32768 → -32767 →…

Program example

Operation error

There is no operation error.

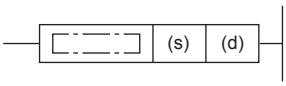
Subtracting 16-bit binary data

-(P)(_U) instruction and SUB(P)(_U) instruction can be used for subtraction of 16-bit binary data.

-(P)(_U) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions subtract the 16-bit binary data in the device specified by (d) and the 16-bit binary data in the device specified by (s), and store the result in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 216 -(P)(_U) [using three operands]

FBD/LD
Not supported. ☞ Page 216 -(P)(_U) [using three operands]

Setting data

■Descriptions, ranges, and data types

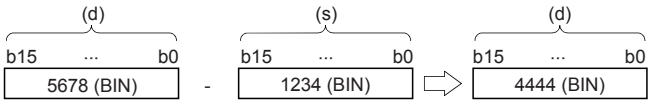
Operand		Description	Range	Data type	Data type (label)
(s)	-(P)	Subtrahend data or the device where the data to be subtracted from another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	-(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	-(P)	Device where the data from which another is to be subtracted is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	-(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

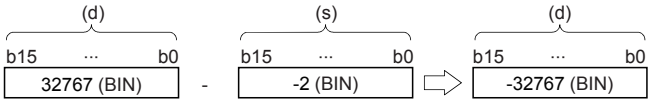
Processing details

- These instructions subtract the 16-bit binary data in the device specified by (d) and the 16-bit binary data in the device specified by (s), and store the subtraction result in the device specified by (d).

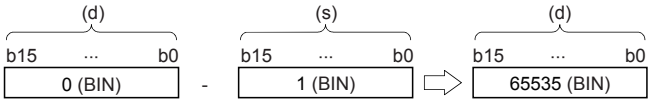


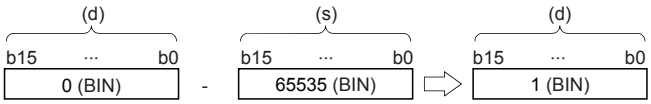
- When underflow or overflow occurs in the operation result, the following processing is executed. In this case, the carry flag (SM700, SM8022) does not turn ON.

In case of -(P)



In case of -(P)(_U)





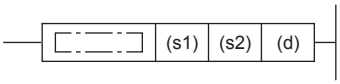
Operation error

There is no operation error.

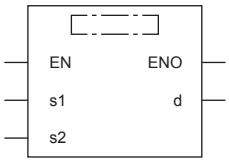
-(P)(_U) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions subtract the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text	
	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



("MINUS", "MINUSP", "MINUS_U", "MINUSP_U" enters □.)

Setting data

■Descriptions, ranges, and data types

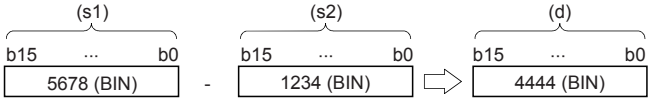
Operand		Description	Range	Data type	Data type (label)
(s1)	-(P)	Minuend data or the device where the data from which another is to be subtracted is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	-(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	-(P)	Subtrahend data or the device where the data to be subtracted from another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	-(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	-(P)	Device for storing the operation result	—	16-bit signed binary	ANY16_S
	-(P)_U		—	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

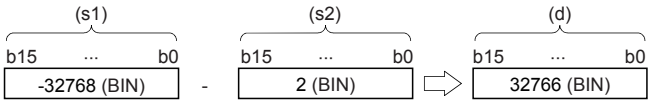
Processing details

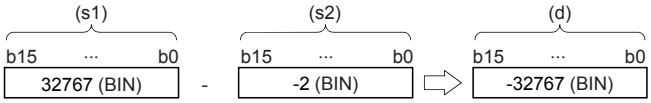
- These instructions subtract the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2), and store the subtraction result in the device specified by (d).



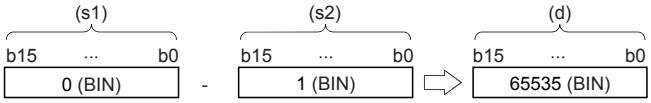
- When underflow or overflow occurs in the operation result, the following processing is executed. In this case, the carry flag (SM700, SM8022) does not turn ON.

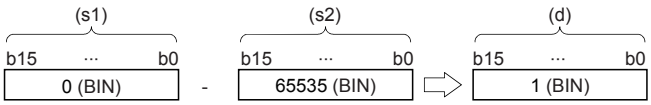
In case of -(P)





In case of -(P)(_U)





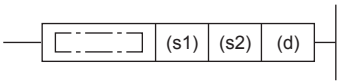
Operation error

There is no operation error.

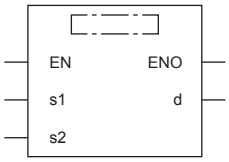
SUB(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions subtract the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=SUBP(EN,s1,s2,d);	ENO:=SUB_U(EN,s1,s2,d); ENO:=SUBP_U(EN,s1,s2,d);

FBD/LD*1



("SUBP", "SUB_U", "SUBP_U" enters □.)

*1 The SUB instruction is not supported by the ST language and the FBD/LD language. Use SUB of the standard function.
☞ Page 1304 SUB(_E)

Setting data

■Descriptions, ranges, and data types

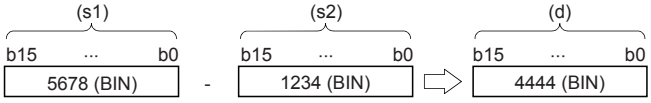
Operand		Description	Range	Data type	Data type (label)
(s1)	SUB(P)	Minuend data or the device where the data from which another is to be subtracted is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	SUB(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	SUB(P)	Subtrahend data or the device where the data to be subtracted from another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	SUB(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	SUB(P)	Device for storing the operation result	—	16-bit signed binary	ANY16_S
	SUB(P)_U		—	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

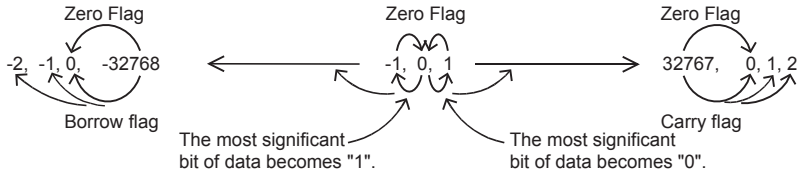
Processing details

- These instructions subtract the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2), and store the subtraction result in the device specified by (d).



■Relationship between the flag operation and the sign (positive or negative) of a numeric value

Device	Name	Description
SM700, SM8022	Carry	When the operation result exceeds the upper limit of the data setting range, the carry flag is turned ON.
SM8020	Zero	When the operation result is 0, the zero flag is turned ON.
SM8021	Borrow	When the operation result is less than the lower limit of the data setting range, the borrow flag is turned ON.



Precautions

■When specifying the same device in the source and destination

The same device number can be specified for both the source and the destination. In this case, note that the subtraction result changes in every operation cycle if a continuous operation type SUB instruction is used.

■Difference between SUB(P) instruction, -(P) instruction, and DEC(P) instruction in a program for subtracting "-1"

When SUB(P) instruction is used to subtract 1 from the contents of D0 every time X1 turns from OFF to ON, SUB(P) instruction is similar to -(P) instruction and DEC(P) instruction described later except for the contents shown in the table below

		SUB(P) instruction	-(P) instruction, DEC(P) instruction
Flag (zero, borrow or carry)		Operates	Does not operate
Operation result	(s)-1=(d)	-32768 → 0 → -1 → -2 →…	-32768 → +32767 → +32766 →…

Program example

Operation error

There is no operation error.

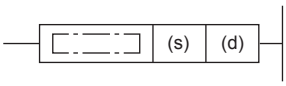
Adding 32-bit binary data

D+(P)(_U) instruction and DADD(P)(_U) instruction can be used for addition of 32-bit binary data.

D+(P)(_U) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions add the 32-bit binary data in the device specified by (d) and the 32-bit binary data in the device specified by (s), and store the result in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 222 D+(P)(_U) [using three operands]

FBD/LD
Not supported. ☞ Page 222 D+(P)(_U) [using three operands]

Setting data

■Descriptions, ranges, and data types

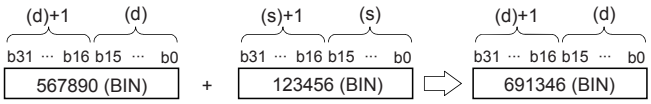
Operand		Description	Range	Data type	Data type (label)
(s)	D+(P)	Addend data or the head device where the data that is added to another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D+(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	D+(P)	Head device where the data to which another is added is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D+(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

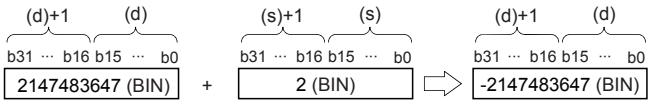
Processing details

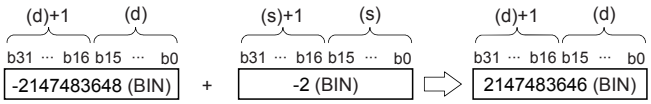
- These instructions add the 32-bit binary data in the device specified by (d) and the 32-bit binary data in the device specified by (s), and store the addition result in the device specified by (d).



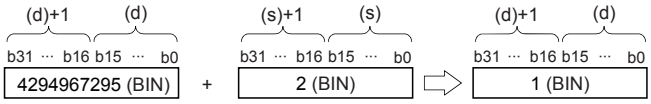
- When underflow or overflow occurs in the operation result, the following processing is executed. In this case, the carry flag (SM700, SM8022) does not turn ON.

In case of D+(P)





In case of D+(P)(_U)



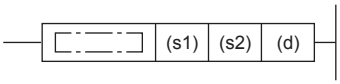
Operation error

There is no operation error.

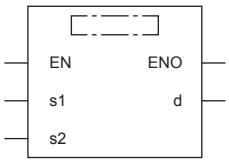
D+(P)(_U) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions add the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=DPLUS(EN,s1,s2,d); ENO:=DPLUSP(EN,s1,s2,d);	ENO:=DPLUS_U(EN,s1,s2,d); ENO:=DPLUSP_U(EN,s1,s2,d);

FBD/LD



("DPLUS", "DPLUSP", "DPLUS_U", "DPLUSP_U" enters □.)

Setting data

■Descriptions, ranges, and data types

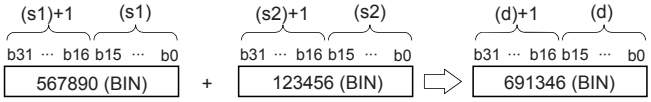
Operand		Description	Range	Data type	Data type (label)
(s1)	D+(P)	Augend data or the head device where the data to which another is added is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D+(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	D+(P)	Addend data or the head device where the data that is added to another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D+(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	D+(P)	Head device for storing the operation result	—	32-bit signed binary	ANY32_S
	D+(P)_U		—	32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

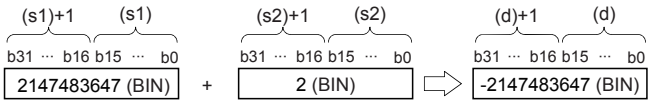
Processing details

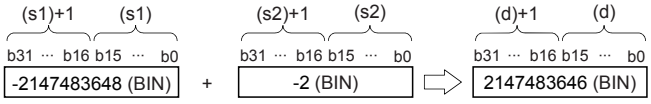
- These instructions add the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the addition result in the device specified by (d).



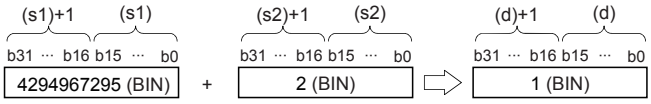
- When underflow or overflow occurs in the operation result, the following processing is executed. In this case, the carry flag (SM700, SM8022) does not turn ON.

In case of D+(P)





In case of D+(P)(_U)



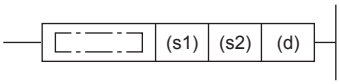
Operation error

There is no operation error.

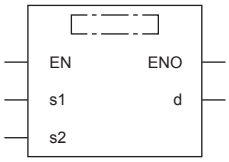
DADD(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=DADD(EN,s1,s2,d); ENO:=DADDP(EN,s1,s2,d);	ENO:=DADD_U(EN,s1,s2,d); ENO:=DADDP_U(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

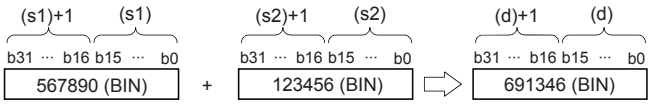
Operand		Description	Range	Data type	Data type (label)
(s1)	DADD(P)	Addend data or the head device where the data that is added to another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DADD(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DADD(P)	Addend data or the head device where the data that is added to another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DADD(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DADD(P)	Head device for storing the operation result	—	32-bit signed binary	ANY32_S
	DADD(P)_U		—	32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

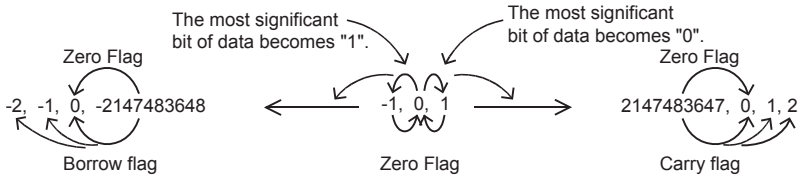
Processing details

- These instructions add the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the addition result in the device specified by (d).



■Relationship between the flag operation and the sign (positive or negative) of a numeric value

Device	Name	Description
SM700, SM8022	Carry	When the operation result exceeds the upper limit of the data setting range, the carry flag is turned ON.
SM8020	Zero	When the operation result is 0, the zero flag is turned ON.
SM8021	Borrow	When the operation result is less than the lower limit of the data setting range, the borrow flag is turned ON.



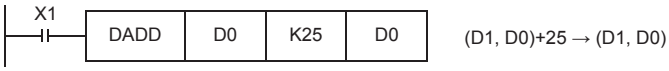
Precautions

■When DADD instruction is used

When specifying word devices, a device for the lower-order 16-bits is specified first, and then a word device with the next device number is set for the higher-order 16 bits. To prevent number overlap, it is recommended to always specify an even number.

■When specifying the same device in the source and destination

The same device number can be specified for both the source and the destination. In this case, note that the addition result changes in every operation cycle if a continuous operation type ADD instruction is used.



■Difference between DADD(P) instruction, D+(P) instruction, and DINC(P) instruction in a program for adding "+1"

When DADD(P) instruction is used to add 1 to the contents of D0 every time X1 turns from OFF to ON, DADD(P) instruction is similar to D+(P) instruction and DINC(P) instruction described later except for the contents shown in the table below.

		DADD(P) instruction	D+(P) instruction, DINC(P) instruction
Flag (zero, borrow or carry)		Operates	Does not operate
Operation result	(s)+1=(d)	+2147483647 → 0 → +1 → +2 →…	+2147483647 → -2147483648 → -2147483647 →…

Operation error

There is no operation error.

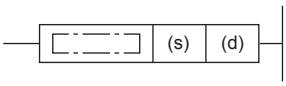
Subtracting 32-bit binary data

D-(P)(_U) instruction and DSUB(P)(_U) instruction can be used for subtraction of 32-bit binary data.

D-(P)(_U) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions subtract the 16-bit binary data in the device specified by (d) and the 16-bit binary data in the device specified by (s), and store the result in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 228 D-(P)(_U) [using three operands]

FBD/LD

Not supported.
☞ Page 228 D-(P)(_U) [using three operands]

Setting data

■Descriptions, ranges, and data types

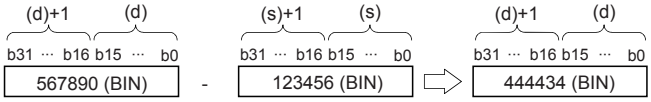
Operand		Description	Range	Data type	Data type (label)
(s)	D-(P)	Subtrahend data or the head device where the data to be subtracted from another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D-(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	D-(P)	Head device where the data from which another is to be subtracted is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D-(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

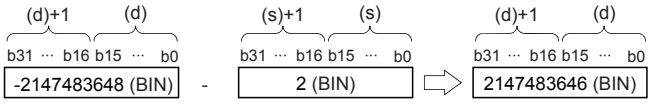
Processing details

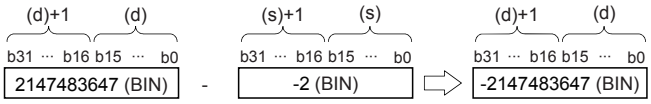
- These instructions subtract the 32-bit binary data in the device specified by (d) and the 32-bit binary data in the device specified by (s), and store the subtraction result in the device specified by (d).



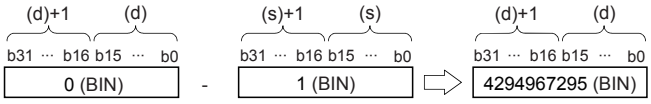
- When underflow or overflow occurs in the operation result, the following processing is executed. In this case, the carry flag (SM700, SM8022) does not turn ON.

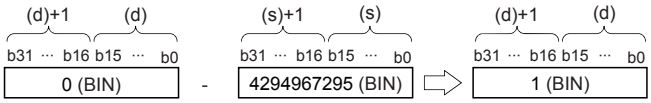
In case of D-(P)





In case of D-(P)(_U)





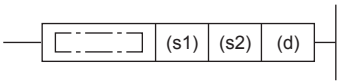
Operation error

There is no operation error.

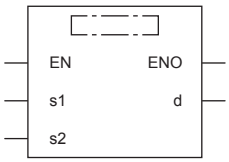
D-(P)(_U) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions subtract the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text	
	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



("DMINUS", "DMINUSP", "DMINUS_U", "DMINUSP_U" enters □.)

Setting data

■Descriptions, ranges, and data types

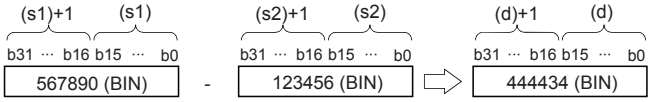
Operand		Description	Range	Data type	Data type (label)
(s1)	D-(P)	Minuend data or the head device where the data from which another is to be subtracted is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D-(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	D-(P)	Subtrahend data or the head device where the data to be subtracted from another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D-(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	D-(P)	Head device for storing the operation result	—	32-bit signed binary	ANY32_S
	D-(P)_U		—	32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

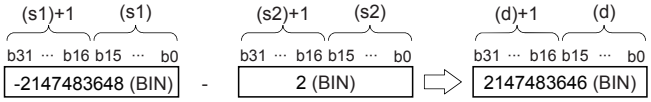
Processing details

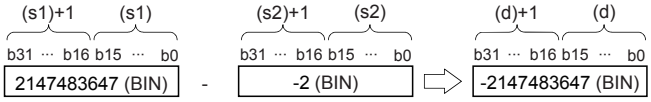
- These instructions subtract the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the subtraction result in the device specified by (d).



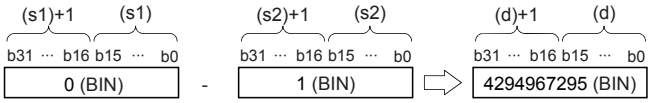
- When underflow or overflow occurs in the operation result, the following processing is executed. In this case, the carry flag (SM700, SM8022) does not turn ON.

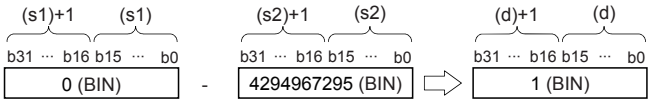
In case of D-(P)





In case of D-(P)(_U)





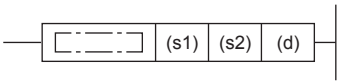
Operation error

There is no operation error.

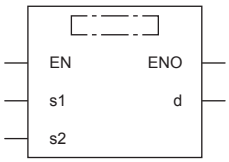
DSUB(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions subtract the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=DSUB(EN,s1,s2,d); ENO:=DSUBP(EN,s1,s2,d);	ENO:=DSUB_U(EN,s1,s2,d); ENO:=DSUBP_U(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

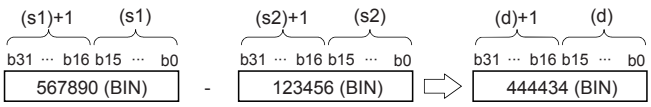
Operand		Description	Range	Data type	Data type (label)
(s1)	DSUB(P)	Minuend data or the head device where the data from which another is to be subtracted is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DSUB(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DSUB(P)	Subtrahend data or the head device where the data to be subtracted from another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DSUB(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DSUB(P)	Head device for storing the operation result	—	32-bit signed binary	ANY32_S
	DSUB(P)_U		—	32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

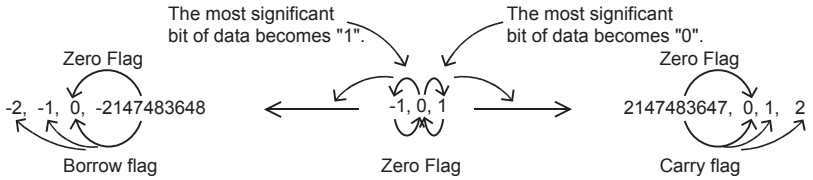
Processing details

- These instructions subtract the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the subtraction result in the device specified by (d).



■Relationship between the flag operation and the sign (positive or negative) of a numeric value

Device	Name	Description
SM700, SM8022	Carry	When the operation result exceeds the upper limit of the data setting range, the carry flag is turned ON.
SM8020	Zero	When the operation result is 0, the zero flag is turned ON.
SM8021	Borrow	When the operation result is less than the lower limit of the data setting range, the borrow flag is turned ON.



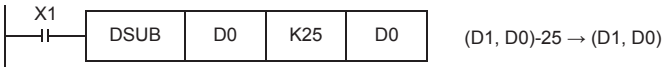
Precautions

■When the DSUB instruction is used

When specifying word devices, a device is specified for the lower-order 16-bits first, and then a word device with the next device number is set for the higher-order 16 bits. To prevent number overlap, it is recommended to always specify an even number.

■When specifying the same device in the source and destination

The same device number can be specified for both the source and the destination. In this case, note that the subtraction result changes in every operation cycle if a continuous operation type SUB instruction is used.



■Difference between DSUB(P) instruction, D-(P) instruction, and DDEC(P) instruction in a program for subtracting "-1"

When DSUB(P) instruction is used to subtract 1 from the contents of D0 every time X1 turns from OFF to ON, SUB(P) instruction is similar to D-(P) instruction and DDEC(P) instruction described later except for the contents shown in the table below:

		DSUB(P) instruction	D-(P) instruction, DDEC(P) instruction
Flag (zero, borrow or carry)		Operates	Does not operate
Operation result	(s)-1=(d)	-2147483648 → 0 → -1 → -2 →…	-2147483648 → +2147483647 → +2147483646 →…

Operation error

There is no operation error.

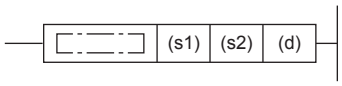
Multiplying 16-bit binary data

*(P)(_U) instruction and MUL(P)(_U) instruction can be used for multiplication of 16-bit binary data.

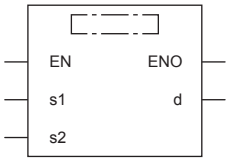
*(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions multiply the 16-bit binary data in the device specified by (s1) by the 16-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text[*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



("MULTI", "MULTIP", "MULTI_U", "MULTIP_U" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

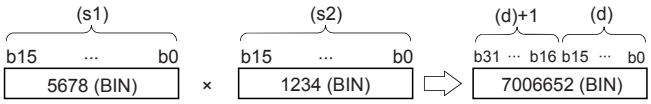
Operand		Description	Range	Data type	Data type (label)
(s1)	*(P)	Multiplicand data or the device where the data to be multiplied by another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	*(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	*(P)	Multiplier data or the device where the data by which another is to be multiplied is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	*(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	*(P)	Head device for storing the operation result	—	32-bit signed binary	ANY32_S
	*(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions multiply the 16-bit binary data in the device specified by (s1) by the 16-bit binary data in the device specified by (s2), and store the multiplication result in the device specified by (d).



- When (d) is a bit device, lower-order bit is specified first.

Ex.

Multiplication result when (d) is a bit device

- K1 ... Lower 4 bits (b0 to b3)
- K4 ... Lower 16 bits (b0 to b15)
- K8 ... Lower 32 bits (b0 to b31)

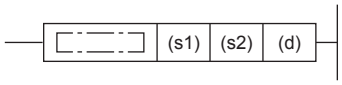
Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.

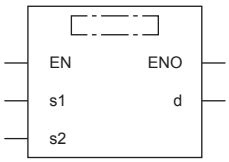
MUL(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions multiply the 16-bit binary data in the device specified by (s1) by the 16-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=MULP(EN,s1,s2,d);	ENO:=MUL_U(EN,s1,s2,d); ENO:=MULP_U(EN,s1,s2,d);

FBD/LD*1



("MULP", "MUL_U", "MULP_U" enters □.)

*1 The MUL instruction is not supported by the ST language and the FBD/LD language. Use MUL of the standard function.
☞ Page 1302 MUL(_E)

Setting data

■Descriptions, ranges, and data types

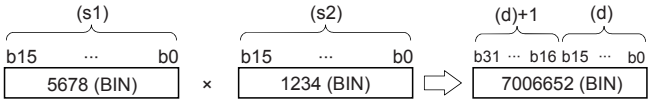
Operand		Description	Range	Data type	Data type (label)
(s1)	MUL(P)	Multiplicand data or the device where the data to be multiplied by another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	MUL(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	MUL(P)	Multiplier data or the device where the data by which another is to be multiplied is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	MUL(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	MUL(P)	Head device for storing the operation result	—	32-bit signed binary	ANY32_S
	MUL(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

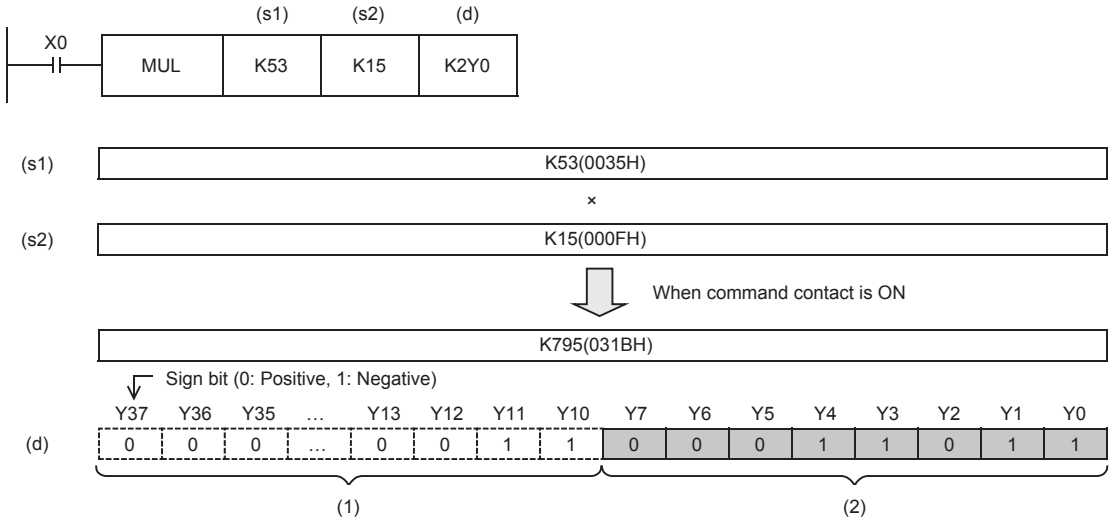
- These instructions multiply the 16-bit binary data in the device specified by (s1) by the 16-bit binary data in the device specified by (s2), and store the multiplication result in the device specified by (d).



• Nibble can be specified ranging from K1 to K8 for (d).

Ex.

For example, when K2 is specified, only the lower-order 8 bits can be obtained out of the product (32 bits).



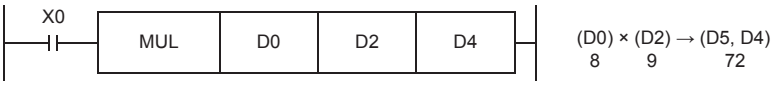
(1): Not output

(2): K2Y0 operation result is output.

■Related flag

Device	Name	Description
SM8304	Zero	When the operation result is 0, the zero flag is turned ON.

Program example



Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.

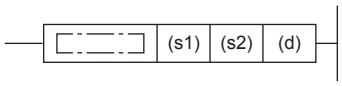
Dividing 16-bit binary data

/(P)(_U) instruction and DIV(P)(_U) instruction can be used for division of 16-bit binary data.

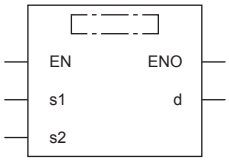
/(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions divide the 16-bit binary data in the device specified by (s1) by the 16-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text*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



("DIVISON", "DIVISIONP", "DIVISION_U", "DIVISIONP_U" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

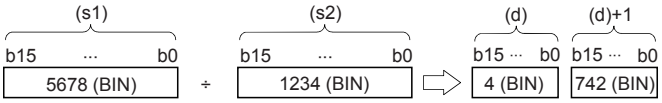
Operand		Description	Range	Data type	Data type (label)
(s1)	/(P)	Dividend data or the device where the data to be divided by another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	/(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	/(P)	Divisor data or the device where the data by which another is to be divided is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	/(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	/(P)	Head device for storing the operation result	—	32-bit signed binary	ANY16_S_ARRAY (Number of elements: 2)
	/(P)_U			32-bit unsigned binary	ANY16_U_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions divide the 16-bit binary data in the device specified by (s1) by the 16-bit binary data in the device specified by (s2), and store the division result in the device specified by (d).



(d): Quotient
(d)+1: Remainder

- For the division result, 32-bit is used for word device to store the quotient and remainder and 16-bit is used for bit device to store quotient only.
  - Quotient…… Stored in the lower 16 bits.
  - Remainder…… Stored in the upper 16 bits. (This data can be stored for word device only.)

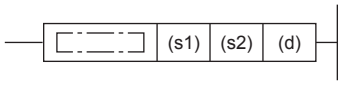
Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
3400H	0 is specified for (s2) value.

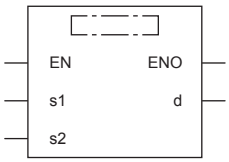
DIV(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions divide the 16-bit binary data in the device specified by (s1) by the 16-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=DIVP(EN,s1,s2,d);	ENO:=DIV_U(EN,s1,s2,d); ENO:=DIVP_U(EN,s1,s2,d);

FBD/LD*1



("DIVP", "DIV_U", "DIVP_U" enters □.)

*1 The DIV instruction is not supported by the ST language and the FBD/LD language. Use DIV of the standard function.
☞ Page 1306 DIV(_E)

Setting data

■Descriptions, ranges, and data types

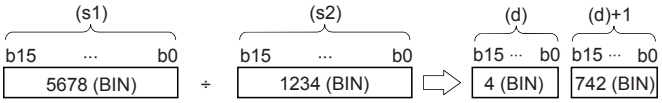
Operand		Description	Range	Data type	Data type (label)
(s1)	DIV(P)	Dividend data or the device where the data to be divided by another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	DIV(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	DIV(P)	Divisor data or the device where the data by which another is to be divided is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	DIV(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	DIV(P)	Head device for storing the operation result (quotient, remainder)	—	32-bit signed binary	ANY16_S_ARRAY (Number of elements: 2)
	DIV(P)_U			32-bit unsigned binary	ANY16_U_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions divide the 16-bit binary data in the device specified by (s1) by the 16-bit binary data in the device specified by (s2), and store the division result in the device specified by (d).



(d): Quotient
(d)+1: Remainder

- Two devices in total starting from the one specified by (d) are used to store the division result. Make sure that these two devices are not used for another control.
  - Quotient…… Stored in the lower 16 bits.
  - Remainder…… Stored in the upper 16 bits.

■Related flag

Device	Name	Description
SM700	Carry	When the operation result of the signed operation exceeds 32767, the carry flag is turned ON.
SM8304	Zero	When the operation result is 0, the zero flag is turned ON.
SM8306	Carry	When the operation result of the signed operation exceeds 32767, the carry flag is turned ON.

Precautions

■Operation result

- The most significant bit of the quotient and remainder indicates the sign (positive: 0, negative: 1), respectively.
- The quotient is negative when either (s1) or (s2) is negative. The remainder is negative when the (s1) is negative.

■Device specified by (d)

- The remainder is not obtained when a bit device is specified with digit specification.

Program example

Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
3400H	0 is specified for (s2) value.
3403H	The data type of the data setting is signed data and the operation result exceeds 32767.

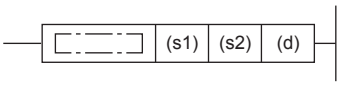
Multiplying 32-bit binary data

D*(P)(_U) instruction and DMUL(P)(_U) instruction can be used for multiplication of 32-bit binary data.

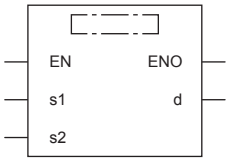
D*(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions multiply the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=DMULTI(EN,s1,s2,n,d); ENO:=DMULTIP(EN,s1,s2,n,d);	ENO:=DMULTI_U(EN,s1,s2,n,d); ENO:=DMULTIP_U(EN,s1,s2,n,d);

FBD/LD



("DMULTI", "DMULTIP", "DMULTI_U", "DMULTIP_U" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

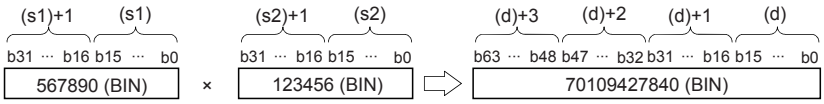
Operand		Description	Range	Data type	Data type (label)
(s1)	D*(P)	Multiplicand data or the head device where the data to be multiplied by another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D*(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	D*(P)	Multiplier data or the head device where the data by which another is to be multiplied is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D*(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	D*(P)	Head device for storing the operation result	—	64-bit signed binary	ANY32_S_ARRAY (Number of elements: 2)
	D*(P)_U			64-bit unsigned binary	ANY32_U_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	—	—	○	—	○	—	—	—	—

Processing details

- These instructions multiply the 32-bit binary data in the device specified by (s1) by the 32-bit binary data in the device specified by (s2), and store the multiplication result in the device specified by (d).



- When (d) is a bit device, only the lower 32 bits of the multiplication result are stored and the upper 32 bits cannot be specified. If the upper 32 bits data of the multiplication operation result are required, temporarily store the result in a word device, and transfer the data stored in word device ((d)+2) and ((d)+3) to the specified bit devices.

Ex.

Multiplication result when (d) is a bit device

- K1 ... Lower 4 bits (b0 to b3)
- K4 ... Lower 16 bits (b0 to b15)
- K8 ... Lower 32 bits (b0 to b31)

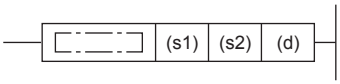
Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.

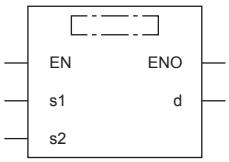
DMUL(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions multiply the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=DMUL(EN,s1,s2,d); ENO:=DMULP(EN,s1,s2,d);	ENO:=DMUL_U(EN,s1,s2,d); ENO:=DMULP_U(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

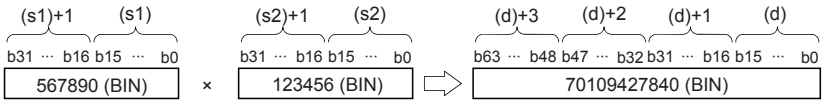
Operand		Description	Range	Data type	Data type (label)
(s1)	DMUL(P)	Multiplicand data or the head device where the data to be multiplied by another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DMUL(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DMUL(P)	Multiplier data or the head device where the data by which another is to be multiplied is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DMUL(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DMUL(P)	Head device for storing the operation result	—	64-bit signed binary	ANY32_S_ARRAY (Number of elements: 2)
	DMUL(P)_U			64-bit unsigned binary	ANY32_U_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

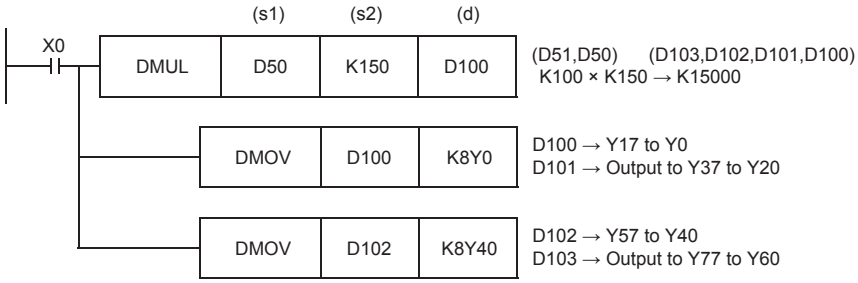
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions multiply the 32-bit binary data in the device specified by (s1) by the 32-bit binary data in the device specified by (s2), and store the multiplication result in the device specified by (d).



- When nibble is specified ranging from K1 to K8 for (d), the result is obtained only for the lower-order 32 bits, and is not obtained for the higher-order 32 bits. Transfer the data to word devices once, then execute the operation.



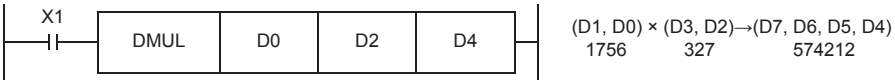
■Related flag

Device	Name	Description
SM8304	Zero	When the operation result is 0, the zero flag is turned ON.

Precautions

- Even if word devices are used, the operation result (64 bits binary data) cannot be monitored at one time. In such a case, a floating point operation is recommended.

Program example



Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.

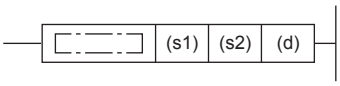
Dividing 32-bit binary data

D/(P)(_U) instruction and DDIV(P)(_U) instruction can be used for division of 32-bit binary data.

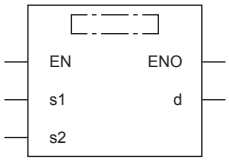
D/(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions divide the 32-bit binary data in the device specified by (s1) by the 32-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text*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



("DDIVISION", "DDIVISIONP", "DDIVISION_U", "DDIVISIONP_U" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

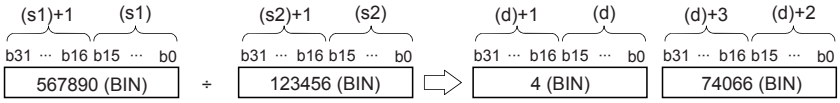
Operand		Description	Range	Data type	Data type (label)
(s1)	D/(P)	Dividend data or the head device where the data to be divided by another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D/(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	D/(P)	Divisor data or the head device where the data by which another is to be divided is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D/(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	D/(P)	Head device for storing the operation result	—	64-bit signed binary	ANY32_ARRAY (Number of elements: 2)
	D/(P)_U			64-bit unsigned binary	
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	—	—	○	—	○	—	—	—	—

Processing details

- These instructions divide the 32-bit binary data in the device specified by (s1) by the 32-bit binary data in the device specified by (s2), and store the division result in the device specified by (d).



- For the division result of word device, 64-bit binary is used to store the quotient and remainder. For bit device, 32-bit binary is used to store quotient only.

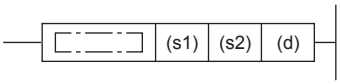
Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
3400H	0 is specified for (s2) value.

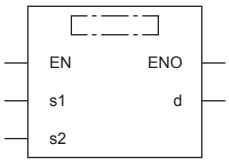
DDIV(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions divide the 32-bit binary data in the device specified by (s1) by the 32-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=DDIV(EN,s1,s2,d); ENO:=DDIVP(EN,s1,s2,d);	ENO:=DDIV_U(EN,s1,s2,d); ENO:=DDIVP_U(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

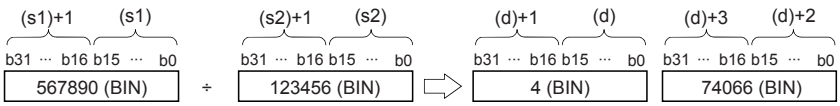
Operand		Description	Range	Data type	Data type (label)
(s1)	DDIV(P)	Dividend data or the head device where the data to be divided by another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DDIV(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DDIV(P)	Divisor data or the head device where the data by which another is to be divided is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DDIV(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DDIV(P)	Head device for storing the operation result	—	64-bit signed binary	ANY32_S_ARRAY (Number of elements: 2)
	DDIV(P)_U			64-bit unsigned binary	ANY32_U_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions divide the 32-bit binary data in the device specified by (s1) by the 32-bit binary data in the device specified by (s2), and store the division result in the device specified by (d).



■Related flag

Device	Name	Description
SM700	Carry	When the operation result of the signed operation exceeds 32767, the carry flag is turned ON.
SM8304	Zero	When the operation result is 0, the zero flag is turned ON.
SM8306	Carry	When the operation result of the signed operation exceeds 32767, the carry flag is turned ON.

Precautions

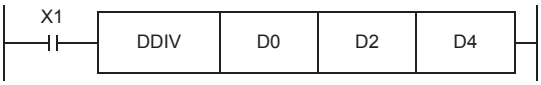
■Operation result

- The most significant bit of the quotient and remainder indicates the sign (positive: 0, negative: 1), respectively.
- The quotient is negative when either (s1) or (s2) is negative. The remainder is negative when the (s1) is negative.

■Device specified by (d)

- The remainder is not obtained when a bit device is specified with digit specification.

Program example



Dividend	Divisor	Quotient	Remainder
(D1, D0)	÷ (D3, D2)	→(D5, D4)	⋯ (D7, D6)
100000	3333	30	10

Operation error

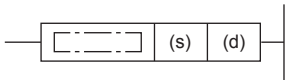
Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
3400H	0 is specified for (s2) value.
3403H	Signed operation is performed and the operation result exceeds 2147483647.

Adding BCD 4-digit data

B+(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions add the BCD 4-digit data in the device specified by (d) and the BCD 4-digit data in the device specified by (s), and store the result in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 249 B+(P) [using three operands]

FBD/LD
Not supported. ☞ Page 249 B+(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

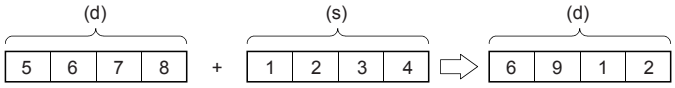
Operand	Description	Range	Data type	Data type (label)
(s)	Addend data or the device where the data that is added to another is stored	0 to 9999	BCD 4-digit	ANY16
(d)	Device where the data to which another is added is stored	0 to 9999	BCD 4-digit	ANY16

■Applicable devices

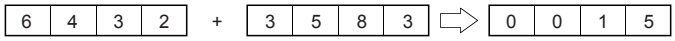
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions add the BCD 4-digit data in the device specified by (d) and the BCD 4-digit data in the device specified by (s), and store the addition result in the device specified by (d).



- If the addition result exceeds 9999, carry is ignored. In this case, the carry flag (SM700) does not turn ON.



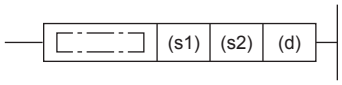
Operation error

Error code (SD/SD8067)	Description
3405H	BCD data in the device specified by (s) is outside of the valid range (0 to 9999).
	BCD data in the device specified by (d) is outside of the valid range (0 to 9999).

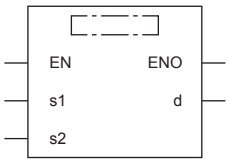
B+(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions add the BCD 4-digit data in the device specified by (s1) and the BCD 4-digit data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text
	ENO:=BPLUS(EN,s1,s2,d); ENO:=BPLUSP(EN,s1,s2,d);

FBD/LD



("BPLUS", "BPLUSP" enters ☐.)

Setting data

■Descriptions, ranges, and data types

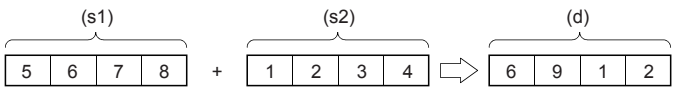
Operand	Description	Range	Data type	Data type (label)
(s1)	Augend data or the device where the data to which another is added is stored	0 to 9999	BCD 4-digit	ANY16
(s2)	Addend data or the device where the data that is added to another is stored	0 to 9999	BCD 4-digit	ANY16
(d)	Device for storing the operation result	0 to 9999	BCD 4-digit	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

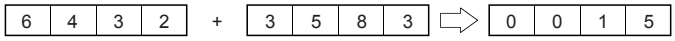
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U☐\G☐	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions add the BCD 4-digit data in the device specified by (s1) and the BCD 4-digit data in the device specified by (s2), and store the addition result in the device specified by (d).



- If the addition result exceeds 9999, carry is ignored. In this case, the carry flag (SM700) does not turn ON.



Operation error

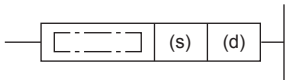
Error code (SD0/SD8067)	Description
3405H	BCD data in the device specified by (s1) is outside of the valid range (0 to 9999).
	BCD data in the device specified by (s2) is outside of the valid range (0 to 9999).

Subtracting BCD 4-digit data

B-(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions subtract the BCD 4-digit data in the device specified by (d) and the BCD 4-digit data in the device specified by (s), and store the result in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 251 B-(P) [using three operands]

FBD/LD
Not supported. ☞ Page 251 B-(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

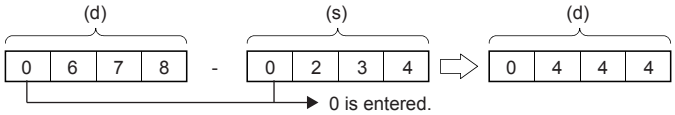
Operand	Description	Range	Data type	Data type (label)
(s)	Subtrahend data or the device where the data to be subtracted from another is stored	0 to 9999	BCD 4-digit	ANY16
(d)	Device where the data from which another is to be subtracted is stored	0 to 9999	BCD 4-digit	ANY16

■Applicable devices

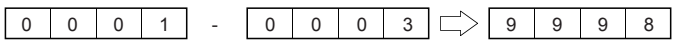
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions subtract the BCD 4-digit data in the device specified by (s) and the BCD 4-digit data in the device specified by (d), and store the subtraction result in the device specified by (d).



- If an underflow occurs, the result will be as follows. In this case, the carry flag (SM700) does not turn ON.



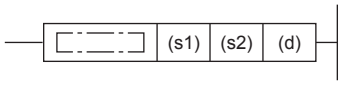
Operation error

Error code (SD0/SD8067)	Description
3405H	BCD data in the device specified by (s) is outside of the valid range (0 to 9999).
	BCD data in the device specified by (d) is outside of the valid range (0 to 9999).

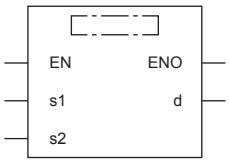
B-(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions subtract the BCD 4-digit data in the device specified by (s1) and the BCD 4-digit data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text
	ENO:=BMINUS(EN,s1,s2,d); ENO:=BMINUSP(EN,s1,s2,d);

FBD/LD



("BMINUS", "BMINUSP" enters □.)

Setting data

■Descriptions, ranges, and data types

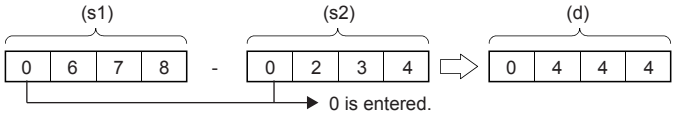
Operand	Description	Range	Data type	Data type (label)
(s1)	Minuend data or the device where the data from which another is to be subtracted is stored	0 to 9999	BCD 4-digit	ANY16
(s2)	Subtrahend data or the device where the data to be subtracted from another is stored	0 to 9999	BCD 4-digit	ANY16
(d)	Device for storing the operation result	0 to 9999	BCD 4-digit	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

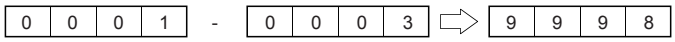
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions subtract the BCD 4-digit data in the device specified by (s1) and the BCD 4-digit data in the device specified by (s2), and store the subtraction result in the device specified by (d).



- If an underflow occurs, the result will be as follows. In this case, the carry flag (SM700) does not turn ON.



Operation error

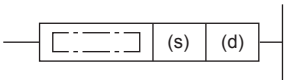
Error code (SD0/SD8067)	Description
3405H	BCD data in the device specified by (s1) is outside of the valid range (0 to 9999).
	BCD data in the device specified by (s2) is outside of the valid range (0 to 9999).

Adding BCD 8-digit data

DB+(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions add the BCD 8-digit data in the device specified by (d) and the BCD 8-digit data in the device specified by (s), and store the result in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 254 DB+(P) [using three operands]

FBD/LD
Not supported. ☞ Page 254 DB+(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

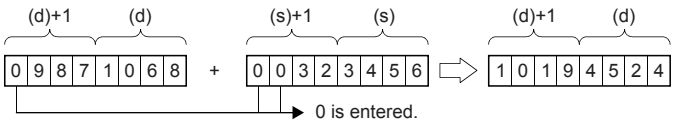
Operand	Description	Range	Data type	Data type (label)
(s)	Addend data or the head device where the data that is added to another is stored	0 to 99999999	BCD 8-digit	ANY32
(d)	Head device where the data to which another data is added is stored	0 to 99999999	BCD 8-digit	ANY32

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

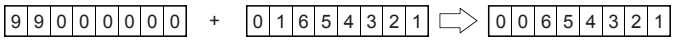
Processing details

- These instructions add the BCD 8-digit data in the device specified by (d) and the BCD 8-digit data in the device specified by (s), and store the addition result in the device specified by (d).



(d)+1, (s)+1	(d), (s)
Upper 4 digits	Lower 4 digits

- If the addition result exceeds 99999999, carry is ignored. In this case, the carry flag (SM700) does not turn ON.



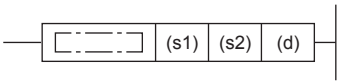
Operation error

Error code (SD0/SD8067)	Description
3405H	BCD data in the device specified by (s) is outside of the valid range (0 to 99999999).
	BCD data in the device specified by (d) is outside of the valid range (0 to 99999999).

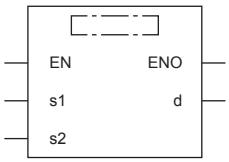
DB+(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions add the BCD 8-digit data in the device specified by (s1) and the BCD 8-digit data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DBPLUS(EN,s1,s2,d); ENO:=DBPLUSP(EN,s1,s2,d);

FBD/LD



("DBPLUS", "DBPLUSP" enters □.)

Setting data

■Descriptions, ranges, and data types

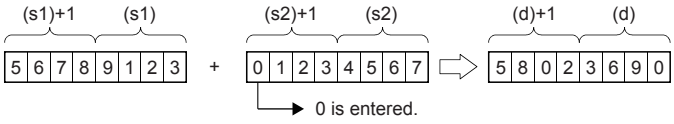
Operand	Description	Range	Data type	Data type (label)
(s1)	Augend data or the head device where the data to which another is added is stored	0 to 99999999	BCD 8-digit	ANY32
(s2)	Addend data or the head device where the data that is added to another is stored	0 to 99999999	BCD 8-digit	ANY32
(d)	Head device for storing the operation result	0 to 99999999	BCD 8-digit	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

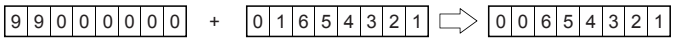
Processing details

- These instructions add the BCD 8-digit data in the device specified by (s1) and the BCD 8-digit data in the device specified by (s2), and store the addition result in the device specified by (d).



(d)+1, (s1)+1, (s2)+1	(d), (s1), (s2)
Upper 4 digits	Lower 4 digits

- If the addition result exceeds 99999999, carry is ignored. In this case, the carry flag (SM700) does not turn ON.



Operation error

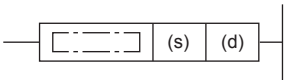
Error code (SD0/SD8067)	Description
3405H	BCD data in the device specified by (s1) is outside of the valid range (0 to 99999999).
	BCD data in the device specified by (s2) is outside of the valid range (0 to 99999999).

Subtracting BCD 8-digit data

DB-(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions subtract the BCD 8-digit data in the device specified by (d) and the BCD 8-digit data in the device specified by (s), and store the result in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 257 DB-(P) [using three operands]

FBD/LD
Not supported. ☞ Page 257 DB-(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

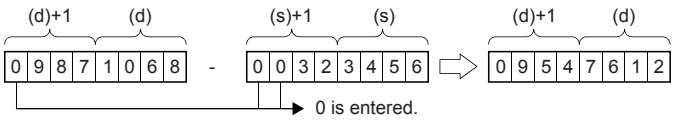
Operand	Description	Range	Data type	Data type (label)
(s)	Subtrahend data or the head device where the data to be subtracted from another is stored	0 to 99999999	BCD 8-digit	ANY32
(d)	Minuend data or the head device where the data from which another is to be subtracted is stored	0 to 99999999	BCD 8-digit	ANY32

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

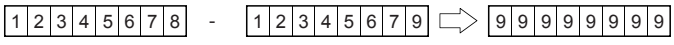
Processing details

- These instructions subtract the BCD 8-digit data specified by (d) and the BCD 8-digit data specified by (s), and store the results in the device specified by (d).



(d)+1, (s)+1	(d), (s)
Upper 4 digits	Lower 4 digits

- If an underflow occurs, the result will be as follows. In this case, the carry flag (SM700) does not turn ON.



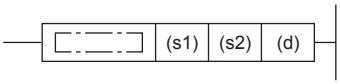
Operation error

Error code (SD0/SD8067)	Description
3405H	BCD data in the device specified by (s) is outside of the valid range (0 to 99999999).
	BCD data in the device specified by (d) is outside of the valid range (0 to 99999999).

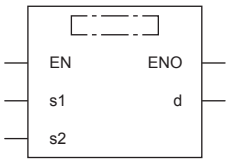
DB-(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions subtract the BCD 8-digit data specified by (s1) and the BCD 8-digit data specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DBMINUS(EN,s1,s2,d); ENO:=DBMINUSP(EN,s1,s2,d);

FBD/LD



("DBMINUS", "DBMINUSP" enters □.)

Setting data

■Descriptions, ranges, and data types

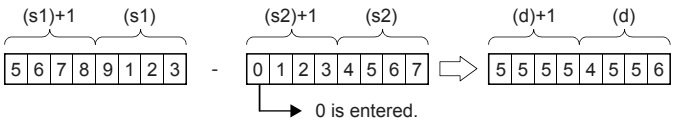
Operand	Description	Range	Data type	Data type (label)
(s1)	Minuend data or the head device where the data from which another is to be subtracted is stored	0 to 99999999	BCD 8-digit	ANY32
(s2)	Subtrahend data or the head device where the data to be subtracted from another is stored	0 to 99999999	BCD 8-digit	ANY32
(d)	Head device for storing the operation result	0 to 99999999	BCD 8-digit	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

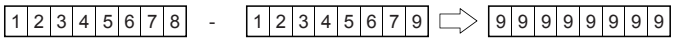
Processing details

- These instructions subtract the BCD 8-digit data specified by (s1) and the BCD 8-digit data specified by (s2), and store the results in the device specified by (d).



(d)+1, (s1)+1, (s2)+1	(d), (s1), (s2)
Upper 4 digits	Lower 4 digits

- If an underflow occurs, the result will be as follows. In this case, the carry flag (SM700) does not turn ON.



Operation error

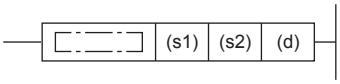
Error code (SD0/SD8067)	Description
3405H	BCD data in the device specified by (s1) is outside of the valid range (0 to 99999999).
	BCD data in the device specified by (s2) is outside of the valid range (0 to 99999999).

Multiplying BCD 4-digit data

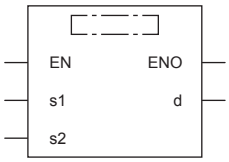
B*(P)

FX5S FX5UJ FX5U FX5UC

These instructions multiply the BCD 4-digit data specified by (s1) and the BCD 4-digit data specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported

FBD/LD



("BMULTI", "BMULTIP" enters □.)

Setting data

■Descriptions, ranges, and data types

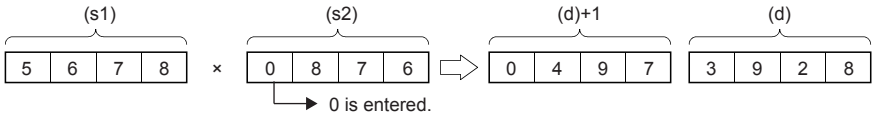
Operand	Description	Range	Data type	Data type (label)
(s1)	Multiplicand data or the device where the data to be multiplied by another is stored	0 to 9999	BCD 4-digit	ANY16
(s2)	Multiplier data or the device where the data by which another is to be multiplied is stored	0 to 9999	BCD 4-digit	ANY16
(d)	Head device for storing the operation result	—	BCD 8-digit	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions multiply the BCD 4-digit data specified by (s1) and the BCD 4-digit data specified by (s2), and store the multiplication results in the device specified by (d). ((d)+1: Upper 4 digits, (d): Lower 4 digits)



Operation error

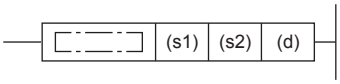
Error code (SD0/SD8067)	Description
2820H	The device specified by (d) exceeds the corresponding device range.
3405H	BCD data in the device specified by (s1) is outside of the valid range (0 to 9999).
	BCD data in the device specified by (s2) is outside of the valid range (0 to 9999).

Dividing BCD 4-digit data

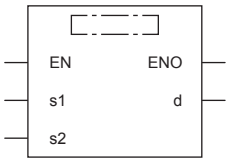
B/(P)

FX5S FX5UJ FX5U FX5UC

These instructions divide the BCD 4-digit data specified by (s1) by the BCD 4-digit data specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported

FBD/LD



("BDIVISION", "BDIVISIONP" enters □.)

Setting data

■Descriptions, ranges, and data types

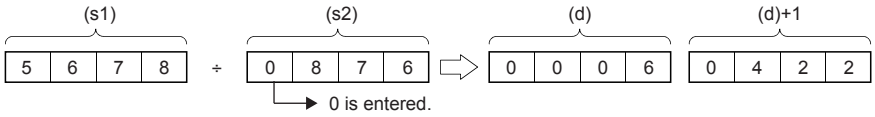
Operand	Description	Range	Data type	Data type (label)
(s1)	Dividend data or the device where the data to be divided by another is stored	0 to 9999	BCD 4-digit	ANY16
(s2)	Divisor data or the device where the data by which another is to be divided is stored	0 to 9999	BCD 4-digit	ANY16
(d)	Head device for storing the operation result	—	BCD 8-digit	ANY16_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions divide the BCD 4-digit data specified by (s1) by the BCD 4-digit data specified by (s2), and store the results of division in the device specified by (d).



(d): Quotient
(d)+1: Remainder

- The results of division are stored as quotient and remainder using 32 bit(s).
  - Quotient (BCD 4-digit): Stored in lower 16 bit(s).
  - Remainder (BCD 4-digit): Stored in upper 16 bit(s).
- If (d) is specified by bit device, remainder of division results is not stored.

Operation error

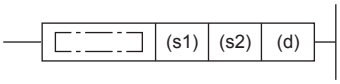
Error code (SD0/SD8067)	Description
2820H	The device specified by (d) exceeds the corresponding device range.
3400H	0 is specified for (s2) value.
3405H	BCD data in the device specified by (s1) is outside of the valid range (0 to 9999).
	BCD data in the device specified by (s2) is outside of the valid range (0 to 9999).

Multiplying BCD 8-digit data

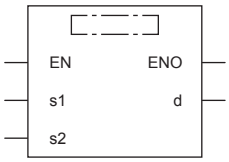
DB*(P)

FX5S FX5UJ FX5U FX5UC

These instructions multiply the BCD 8-digit data specified by (s1) and the BCD 8-digit data specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported

FBD/LD



("DBMULTI", "DBMULTIP" enters □.)

7

Setting data

■Descriptions, ranges, and data types

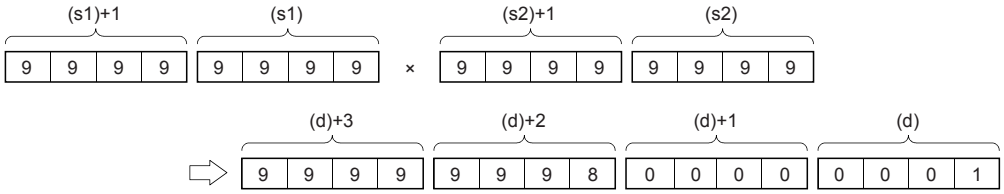
Operand	Description	Range	Data type	Data type (label)
(s1)	Multiplicand data or the head device where the data to be multiplied by another is stored	0 to 99999999	BCD 8-digit	ANY32
(s2)	Multiplier data or the head device where the data by which another is to be multiplied is stored	0 to 99999999	BCD 8-digit	ANY32
(d)	Head device for storing the operation result	—	BCD 16-digit	ANY32_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	—	—	○	—	○	—	—	—	—

Processing details

- These instructions multiply the BCD 8-digit data specified by (s1) and the BCD 8-digit data specified by (s2), and store the multiplication results in the device specified by (d).



- When (d) is a bit device, only the lower 8 nibbles (32 bits) of the multiplication result are stored, and the higher 8 nibbles (32 bits) cannot be specified.

Ex.

Multiplication result when (d) is a bit device

- K1 ... Lower 1 nibble (b0 to b3)
- K4 ... Lower 4 nibble (b0 to b15)
- K8 ... Lower 8 nibble (b0 to b31)

Operation error

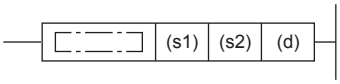
Error code (SD0/SD8067)	Description
2820H	The device specified by (d) exceeds the corresponding device range.
3405H	BCD data in the device specified by (s1) is outside of the valid range (0 to 99999999).
	BCD data in the device specified by (s2) is outside of the valid range (0 to 99999999).

Dividing BCD 8-digit data

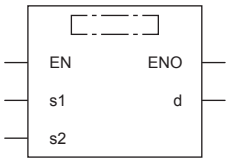
DB/(P)

FX5S FX5UJ FX5U FX5UC

These instructions divide the BCD 8-digit data specified by (s1) by the BCD 8-digit data specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported

FBD/LD



("DBDIVISION", "DBDIVISIONP" enters □.)

Setting data

■Descriptions, ranges, and data types

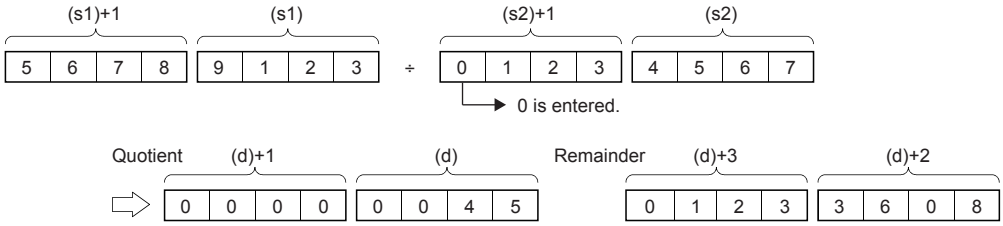
Operand	Description	Range	Data type	Data type (label)
(s1)	Dividend data or the head device where the data to be divided by another is stored	0 to 99999999	BCD 8-digit	ANY32
(s2)	Divisor data or the head device where the data by which another is to be divided is stored	0 to 99999999	BCD 8-digit	ANY32
(d)	Head device for storing the operation result	—	BCD 16-digit	ANY32_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	—	—	○	—	○	—	—	—	—

Processing details

- These instructions divide the BCD 8-digit data specified by (s1) by the BCD 8-digit data specified by (s2), and store the results of division in the device specified by (d).



(d)+1, (d)+3	(d), (d)+2
Upper 4 digits	Lower 4 digits

- The results of division are stored as quotient and remainder using 64 bit(s) binary.
  - Quotient (BCD 8-digit): Stored in lower 32 bit(s).
  - Remainder (BCD 8-digit): Stored in upper 32 bit(s).
- If (d) is specified by bit device, remainder of division results is not stored.

Operation error

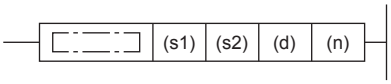
Error code (SD0/SD8067)	Description
2820H	The device specified by (d) exceeds the corresponding device range.
3400H	0 is specified for (s2) value.
3405H	BCD data in the device specified by (s1) is outside of the valid range (0 to 99999999).
	BCD data in the device specified by (s2) is outside of the valid range (0 to 99999999).

Adding 16-bit binary block data

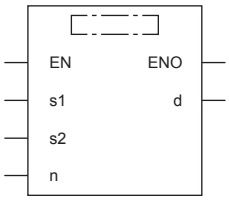
BK+(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add (n) point(s) of 16-bit binary data from the device specified by (s1) and the (n) point(s) of 16-bit binary data from the device specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=BKPLUS(EN,s1,s2,n,d); ENO:=BKPLUSP(EN,s1,s2,n,d);	ENO:=BKPLUS_U(EN,s1,s2,n,d); ENO:=BKPLUSP_U(EN,s1,s2,n,d);

FBD/LD



("BKPLUS", "BKPLUSP", "BKPLUS_U", "BKPLUSP_U" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	BK+(P)	Head device where the data to which another data is added is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	BK+(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	BK+(P)	Addend data or the head device where the data that is added to another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	BK+(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	BK+(P)	Head device for storing the operation result	—	16-bit signed binary	ANY16_S
	BK+(P)_U		—	16-bit unsigned binary	ANY16_U
(n)		Number of addition data	0 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

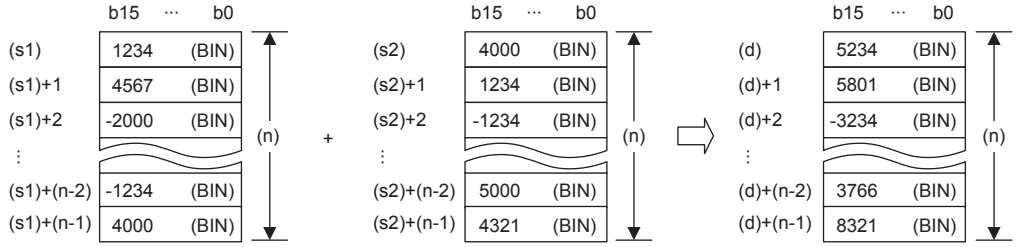
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	○	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

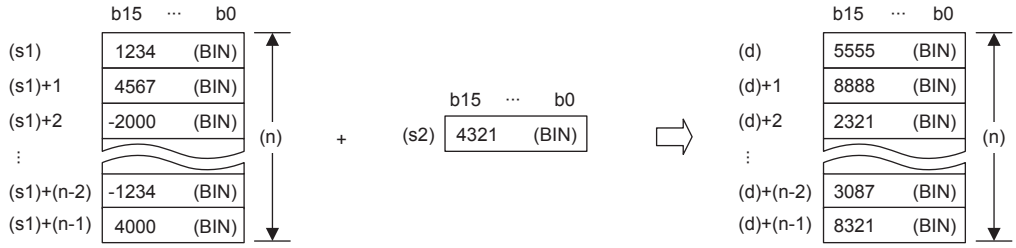
- These instructions add (n) point(s) of 16-bit binary data from the device specified by (s1) and the (n) point(s) of 16-bit binary data from the device specified by (s2), and store the results of addition in the device specified by (d).
- Block addition is performed in units of 16-bits.

Ex.

If device is specified for (s2) (signed)



If constant is specified for (s2) (signed)

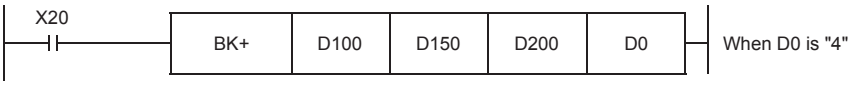


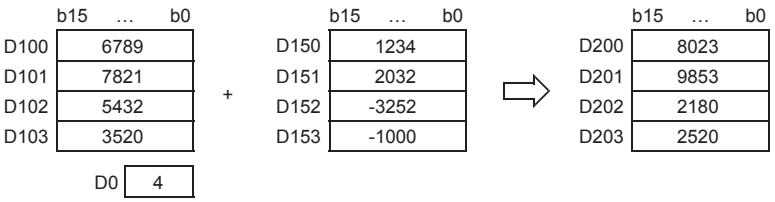
- If an underflow or overflow occurs for operation result, the result will be as follows. In this case, the carry flag (SM700) does not turn ON.

If signed is specified	If unsigned is specified
K32767 (7FFFH) + K2 (0002H) → K-32767 (8001H) K-32767 (8001H) + K-2 (FFFEH) → K32767 (7FFFH)	K65535 (FFFFH) + K1 (0001H) → K0 (0000H)

Program example

In the program shown below, the specified number of data stored in D150 to D0 are added to the specified number of data stored in D100 to D0 when X20 is set to ON, and the operation result is stored in D200 and later.





Operation error

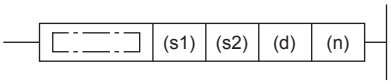
Error code (SD0/SD8067)	Description
2820H	The range of (n) point(s) of data starting from the device specified by (s1), (s2), or (d) exceed the corresponding device range.
2821H	The device range for (n) point(s) beginning from (s1) overlaps with that of (n) point(s) starting from (d). (Does not apply when same device has been specified for (s1) and (d).)
	The device range for (n) point(s) beginning from (s2) overlaps with that of (n) point(s) starting from (d). (Does not apply when same device has been specified for (s2) and (d).)

Subtracting 16-bit binary block data

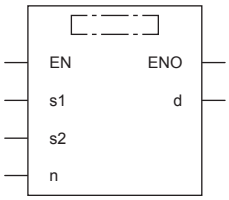
BK-(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions subtract (n) point(s) of 16-bit binary data from the device specified by (s1) and the (n) point(s) of 16-bit binary data from the device specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text*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



("BKMINUS", "BKMINUSP", "BKMINUS_U", "BKMINUSP_U" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	BK-(P)	Head device where the data from which another is to be subtracted is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	BK-(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	BK-(P)	Subtrahend data or the head device where the data to be subtracted from another is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	BK-(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	BK-(P)	Head device for storing the operation result	—	16-bit signed binary	ANY16_S
	BK-(P)_U		—	16-bit unsigned binary	ANY16_U
(n)		Number of subtraction data	0 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

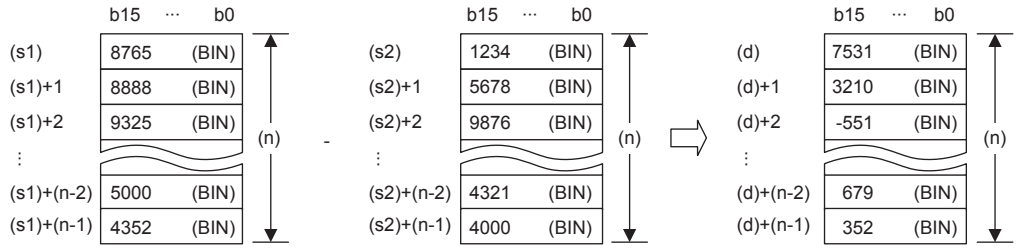
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	○	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

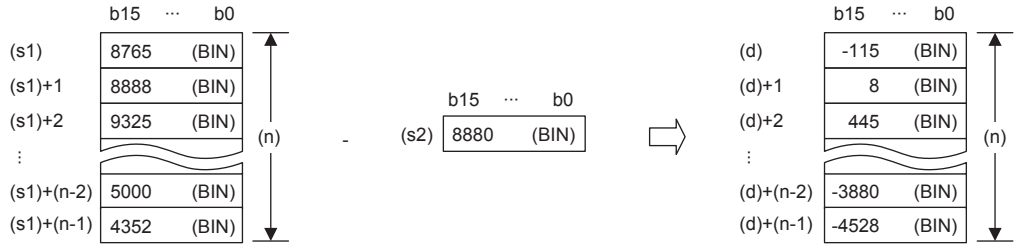
- These instructions subtract (n) point(s) of 16-bit binary data from the device specified by (s1) and the (n) point(s) of 16-bit binary data from the device specified by (s2), and store the subtraction results in the device specified by (d).
- Block subtraction is performed in 16-bit units.

Ex.

If device has been specified for (s2)



If constant is specified for (s2)

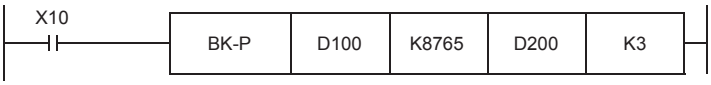


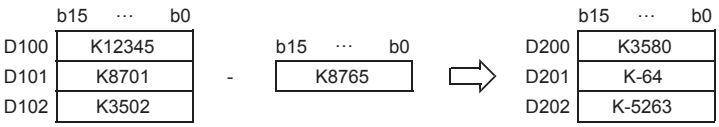
- If an underflow or overflow occurs for operation result, the result will be as follows. In this case, the carry flag (SM700) does not turn ON.

If signed is specified	If unsigned is specified
K-32767 (8001H) - K2 (0002H) ⇨ K32766 (7FFEH) K32767 (7FFFH) - K-2 (FFFEH) ⇨ K-32767 (8001H)	K0 (0000H) - K1 (0001H) ⇨ K65535 (FFFFH)

Program example

In the program shown below, the constant "K8765" is subtracted from the data stored in D100 to D102 when X10 is set to ON, and the operation result is stored in D200 to D202.





Operation error

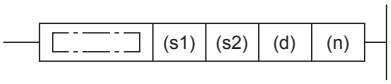
Error code (SD0/SD8067)	Description
2820H	The range of (n) point(s) of data starting from the device specified by (s1), (s2), or (d) exceed the corresponding device range.
2821H	The device range for (n) point(s) beginning from (s1) overlaps with that of (n) point(s) starting from (d). (Does not apply when same device has been specified for (s1) and (d).)
	The device range for (n) point(s) beginning from (s2) overlaps with that of (n) point(s) starting from (d). (Does not apply when same device has been specified for (s2) and (d).)

Adding 32-bit binary block data

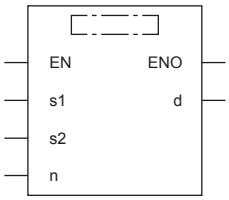
DBK+(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add (n) point(s) of 32-bit binary data from the device specified by (s1) and the (n) point(s) of 32-bit binary data from the device specified by (s2), and store the results of addition in the device specified by (d).

Ladder diagram	Structured text*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



("DBKPLUS", "DBKPLUSP", "DBKPLUS_U", "DBKPLUSP_U" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DBK+(P)	Head device where the data to which another is added is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DBK+(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DBK+(P)	Addend data or the head device where the data that is added to another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DBK+(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DBK+(P)	Head device for storing the operation result	—	32-bit signed binary	ANY32_S
	DBK+(P)_U		—	32-bit unsigned binary	ANY32_U
(n)		Number of addition data	0 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

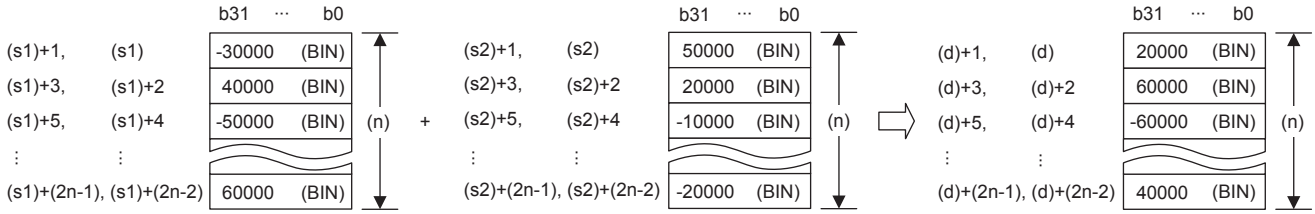
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	—	—	○	—	○	—	—	—	—
(s2)	—	○	—	—	○	—	○	○	—	—	—
(d)	—	○	—	—	○	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

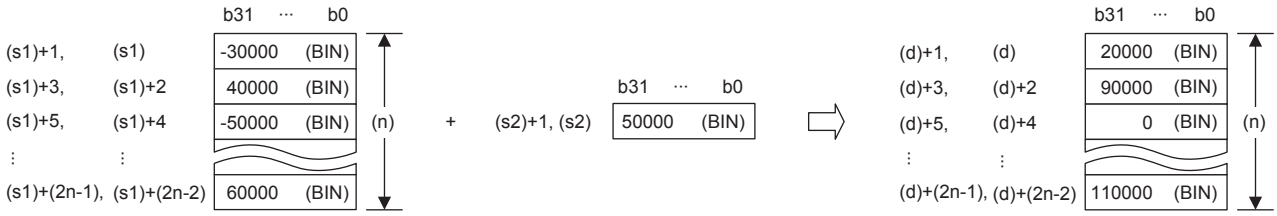
- These instructions add (n) point(s) of 32-bit binary data from the device specified by (s1) and the (n) point(s) of 32-bit binary data from the device specified by (s2), and store the results of addition in the device specified by (d).
- Block addition is performed in 32-bit units.

Ex.

If device is specified for (s2) (signed)



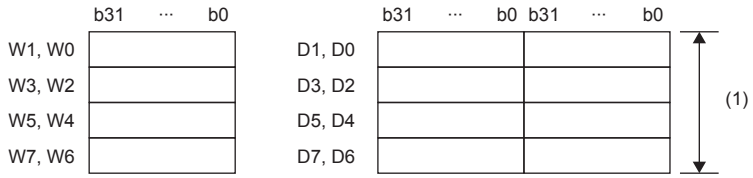
If constant is specified for (s2) (signed)



- Operation is enabled when (s1) or (s2) have been specified by same device as (d) (perfect match). An error occurs if the device range of (n) point(s) from (s1) or (s2) partially matches (overlaps) the device range of (n) point(s) from (d).

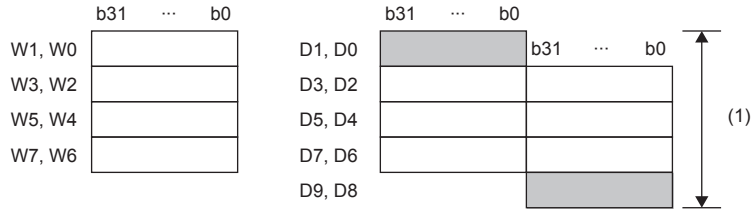
Ex.

If 4 points of the device from (s2) and (d) match



(1)Because it is a perfect match, operation is possible.

If 4 points of the device from (s2), (d) match partially



(1)An operation error occurs if they partially match.

- If the value specified for (n) is 0, processing is not performed.
- If an underflow or overflow occurs for operation result, the result will be as follows. In this case, the carry flag (SM700) does not turn ON.

If signed is specified	If unsigned is specified
K2147483647 (7FFFFFFFH) + K2 (00000002H) → K-2147483647 (80000001H) K-2147483647 (80000001H) + K-2 (FFFFFFFEH) → K2147483647 (7FFFFFFFH)	K4294967295 (FFFFFFFFH) + K1 (00000001H) → K0 (00000000H)

Operation error

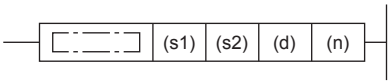
Error code (SD0/SD8067)	Description
2820H	The range of (n) point(s) of data starting from the device specified by (s1), (s2), or (d) exceed the corresponding device range.
2821H	The device range for (n) point(s) beginning from (s1) overlaps with that of (n) point(s) starting from (d). (Does not apply when same device has been specified for (s1) and (d).)
	The device range for (n) point(s) beginning from (s2) overlaps with that of (n) point(s) starting from (d). (Does not apply when same device has been specified for (s2) and (d).)

Subtracting 32-bit binary block data

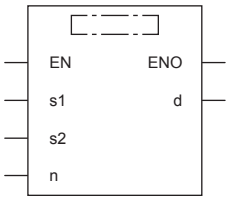
DBK-(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions subtract (n) point(s) of 32-bit binary data from the device specified by (s1) and the (n) point(s) of 32-bit binary data from the device specified by (s2), and store the results of subtraction in the device specified by (d).

Ladder diagram	Structured text*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



("DBKMINUS", "DBKMINUSP", "DBKMINUS_U", "DBKMINUSP_U" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DBK-(P)	Head device where the data from which another is to be subtracted is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DBK-(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DBK-(P)	Subtrahend data or the head device where the data to be subtracted from another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DBK-(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DBK-(P)	Head device for storing the operation result	—	32-bit signed binary	ANY32_S
	DBK-(P)_U		—	32-bit unsigned binary	ANY32_U
(n)		Number of subtraction data	0 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

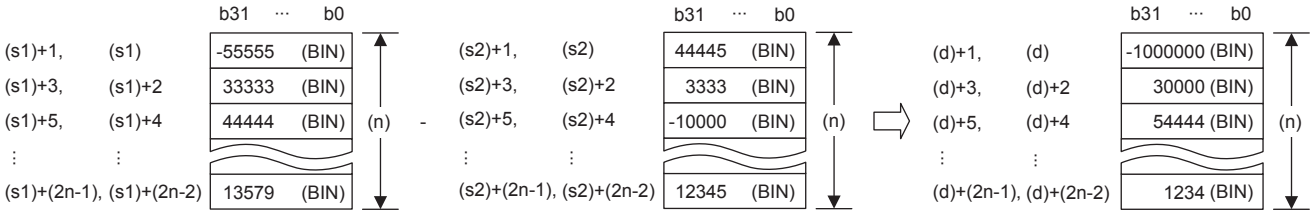
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	—	—	○	—	○	—	—	—	—
(s2)	—	○	—	—	○	—	○	○	—	—	—
(d)	—	○	—	—	○	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

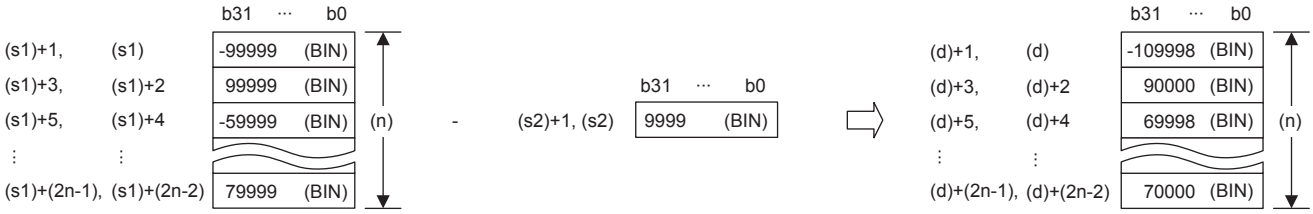
- These instructions subtract (n) point(s) of 32-bit binary data from the device specified by (s1) and the (n) point(s) of 32-bit binary data from the device specified by (s2), and store the results of subtraction in the device specified by (d).
- Block subtraction is performed in 32-bit units.

Ex.

If device is specified for (s2) (signed)



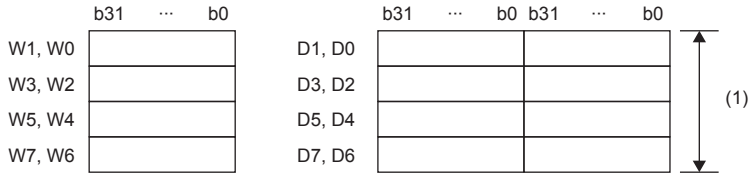
If constant is specified for (s2) (signed)



- Operation is enabled when (s1) or (s2) have been specified by same device as (d) (perfect match). An error occurs if the device range of (n) point(s) from (s1) or (s2) partially matches (overlaps) the device range of (n) point(s) from (d).

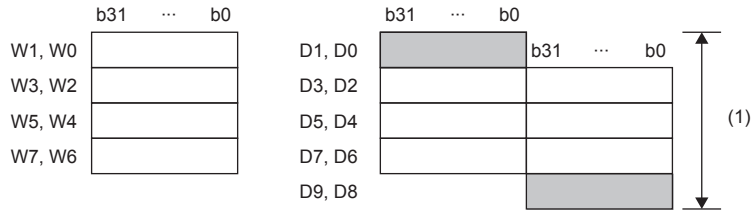
Ex.

If 4 points of the device from (s2) and (d) match



(1)Because it is a perfect match, operation is possible.

If 4 points of the device from (s2), (d) match partially



(1)An operation error occurs if they partially match.

- If the value specified for (n) is 0, processing is not performed.
- If an underflow or overflow occurs for operation result, the result will be as follows. In this case, the carry flag (SM700) does not turn ON.

If signed is specified	If unsigned is specified
K2147483647 (7FFFFFFFH) - K-2 (FFFFFFFEH) → K-2147483647 (80000001H) K-2147483647 (80000001H) - K2 (00000002H) → K2147483647 (7FFFFFFFH)	K0 (00000000H) - K1 (00000001H) → K4294967295 (FFFFFFFFH)

Operation error

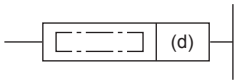
Error code (SD0/SD8067)	Description
2820H	The range of (n) point(s) of data starting from the device specified by (s1), (s2), or (d) exceed the corresponding device range.
2821H	The device range for (n) point(s) beginning from (s1) overlaps with that of (n) point(s) starting from (d). (Does not apply when same device has been specified for (s1) and (d).)
	The device range for (n) point(s) beginning from (s2) overlaps with that of (n) point(s) starting from (d). (Does not apply when same device has been specified for (s2) and (d).)

Incrementing 16-bit binary data

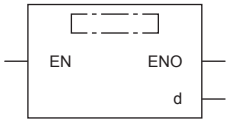
INC(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add 1 to the device (16-bit binary data) specified by (d).

Ladder diagram	Structured text	
	ENO:=INC(EN,d); ENO:=INCP(EN,d);	ENO:=INC_U(EN,d); ENO:=INCP_U(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

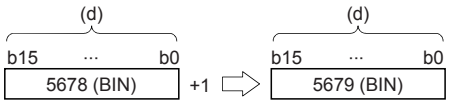
Operand		Description	Range	Data type	Data type (label)
(d)	INC(P)	Device to be incremented by +1	-32768 to +32767	16-bit signed binary	ANY16_S
	INC(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions add 1 to the device (16-bit binary data) specified by (d).

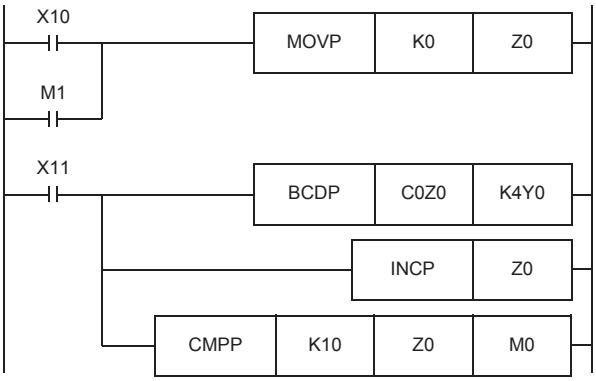


- If INC(P) instruction is executed when contents of device specified by (d) is 32767, -32768 is stored in the device specified by (d). (If signed is specified)
- If INC(P)_U instruction is executed when contents of device specified by (d) is 65535, 0 is stored in the device specified by (d). (If unsigned is specified)
- Flags (zero, carry and borrow) are not activated at this time.

Precautions

Note that data is incremented in every operation cycle in a continuous operation type (INC) instruction.

Program example



Z0 is cleared by the reset input X10.

The current values of counters C0 to C9 are converted into BCD format, and output to K4Y0.

Every time X11 is set to ON, the current values of C0, C1 ... C9 are output one at a time.

Operation error

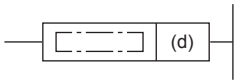
There is no operation error.

Decrementing 16-bit binary data

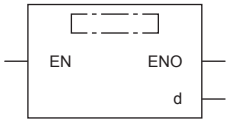
DEC(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions subtract 1 from the device (16-bit binary data) specified by (d).

Ladder diagram	Structured text	
	ENO:=DEC(EN,d); ENO:=DECP(EN,d);	ENO:=DEC_U(EN,d); ENO:=DECP_U(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

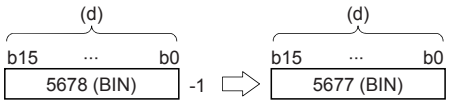
Operand		Description	Range	Data type	Data type (label)
(d)	DEC(P)	Device to be decremented by -1	-32768 to +32767	16-bit signed binary	ANY16_S
	DEC(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions subtract 1 from the device (16-bit binary data) specified by (d).



- If DEC(P) instruction is executed when contents of device specified by (d) is -32768, 32767 is stored in the device specified by (d). (If signed is specified)
- If DEC(P)_U instruction is executed when contents of device specified by (d) is 0, 65535 is stored in the device specified by (d). (If unsigned is specified)
- Flags (zero, carry and borrow) are not activated at this time.

Precautions

Note that data is decremented in every operation cycle in a continuous operation type (DEC) instruction.

Operation error

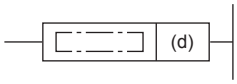
There is no operation error.

Incrementing 32-bit binary data

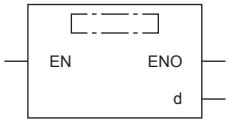
DINC(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add 1 to the device (32-bit binary data) specified by (d).

Ladder diagram	Structured text	
	ENO:=DINC(EN,d); ENO:=DINCP(EN,d);	ENO:=DINC_U(EN,d); ENO:=DINCP_U(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

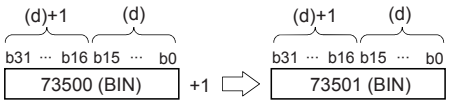
Operand		Description	Range	Data type	Data type (label)
(d)	DINC(P)	Head device to be incremented by +1	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DINC(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions add 1 to the contents of device (32-bit binary data) specified by (d).



- If DINC(P) instruction is executed when contents of device specified by (d) is 2147483647, -2147483648 is stored in the device specified by (d). (If signed is specified)
- If DINC(P)_U instruction is executed when contents of device specified by (d) is 4294967295, 0 is stored in the device specified by (d). (If unsigned is specified)
- Flags (zero, carry and borrow) are not activated at this time.

Precautions

Note that data is incremented in every operation cycle in a continuous operation type instruction.

Operation error

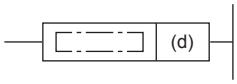
There is no operation error.

Decrementing 32-bit binary data

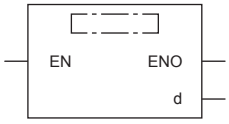
DDEC(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions subtract 1 from the device (32-bit binary data) specified by (d).

Ladder diagram	Structured text	
	ENO:=DDEC(EN,d); ENO:=DDECP(EN,d);	ENO:=DDEC_U(EN,d); ENO:=DDECP_U(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

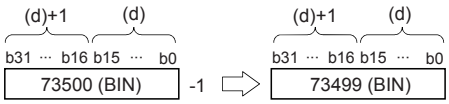
Operand		Description	Range	Data type	Data type (label)
(d)	DDEC(P)	Head device to be decremented by 1	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DDEC(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions subtract 1 from the contents of device (32-bit binary data) specified by (d).



- If DDEC(P) instruction is executed when contents of device specified by (d) is 0, -1 is stored in the device specified by (d). (If signed is specified)
- If DDEC(P)_U instruction is executed when contents of device specified by (d) is 0, 4294967295 is stored in the device specified by (d). (If unsigned is specified)
- Flags (zero, carry and borrow) are not activated at this time.

Precautions

Note that data is decremented in every operation cycle in a continuous operation type (DDEC) instruction.

Operation error

There is no operation error.

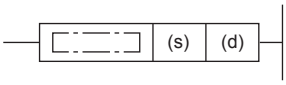
7.3 Logical Operation Instructions

Performing an AND operation on 16-bit data

WAND(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions AND each bit of 16-bit binary data from the device specified by (d) and each bit of 16-bit binary data from device specified by (s), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 281 WAND(P) [using three operands]

FBD/LD
Not supported. ☞ Page 281 WAND(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

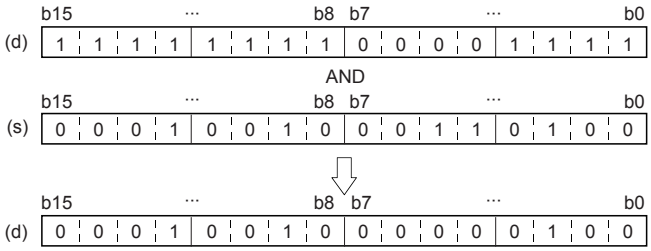
Operand	Description	Range	Data type	Data type (label)
(s)	Data for AND or device where the data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Device for storing AND results	-32768 to +32767	16-bit signed binary	ANY16

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions AND each bit of 16-bit binary data from the device specified by (d) and each bit of 16-bit binary data from device specified by (s), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

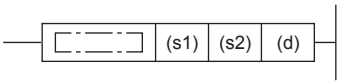
Operation error

There is no operation error.

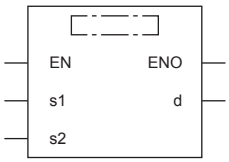
WAND(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions AND each bit of 16-bit binary data from the device specified by (s1) and each bit of 16-bit binary data from device specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	ENO:=WAND(EN,s1,s2,d); ENO:=WANDP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

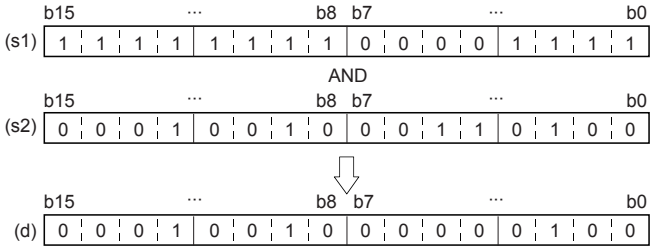
Operand	Description	Range	Data type	Data type (label)
(s1)	Data for AND or device where the data is stored	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Data for AND or device where the data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Device for storing AND results	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions AND each bit of 16-bit binary data from the device specified by (s1) and each bit of 16-bit binary data from device specified by (s2), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

Operation error

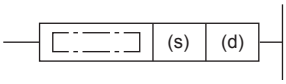
There is no operation error.

Performing an AND operation on 32-bit data

DAND(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions AND each bit of 32-bit binary data from the device specified by (d) and each bit of 32-bit binary data from device specified by (s), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 283 DAND(P) [using three operands]

FBD/LD
Not supported. ☞ Page 283 DAND(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

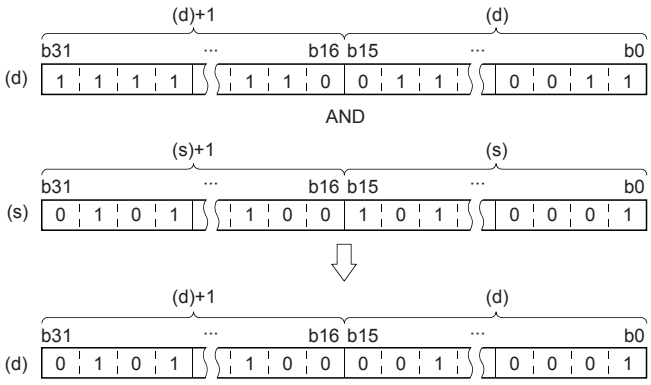
Operand	Description	Range	Data type	Data type (label)
(s)	Data for AND or head device where the data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Head device for storing AND results	-2147483648 to +2147483647	32-bit signed binary	ANY32

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions AND each bit of 32-bit binary data from the device specified by (d) and each bit of 32-bit binary data from device specified by (s), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

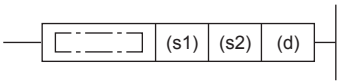
Operation error

There is no operation error.

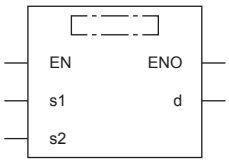
DAND(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions AND each bit of 32-bit binary data from the device specified by (s1) and each bit of 32-bit binary data from device specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DAND(EN,s1,s2,d); ENO:=DANDP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

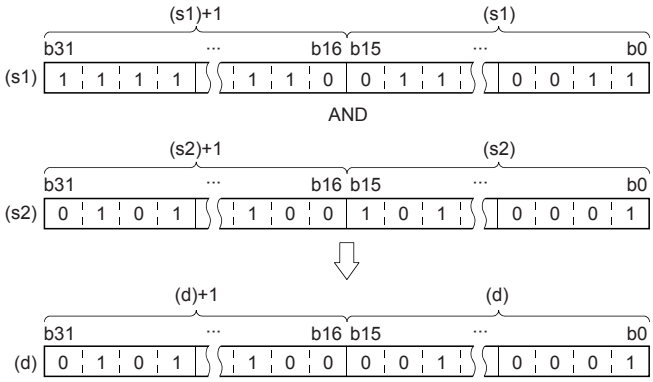
Operand	Description	Range	Data type	Data type (label)
(s1)	Data for AND or head device where the data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Data for AND or head device where the data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Head device for storing AND results	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions AND each bit of 32-bit binary data from the device specified by (s1) and each bit of 32-bit binary data from device specified by (s2), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

Operation error

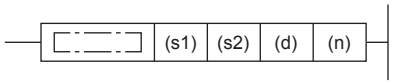
There is no operation error.

Performing an AND operation on 16-bit block data

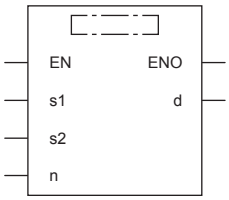
BKAND(P)

FX5S FX5UJ FX5U FX5UC

These instructions AND contents of (n) point(s) from the device specified by (s1) and (n) point(s) from the device specified by (s2), and store the results in the devices specified by (d) onwards.

Ladder diagram	Structured text
	ENO:=BKAND(EN,s1,s2,n,d); ENO:=BKANDP(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device that stores data for AND	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Data for AND or head device where the data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device for storing AND results	—	16-bit signed binary	ANY16
(n)	Number of data	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

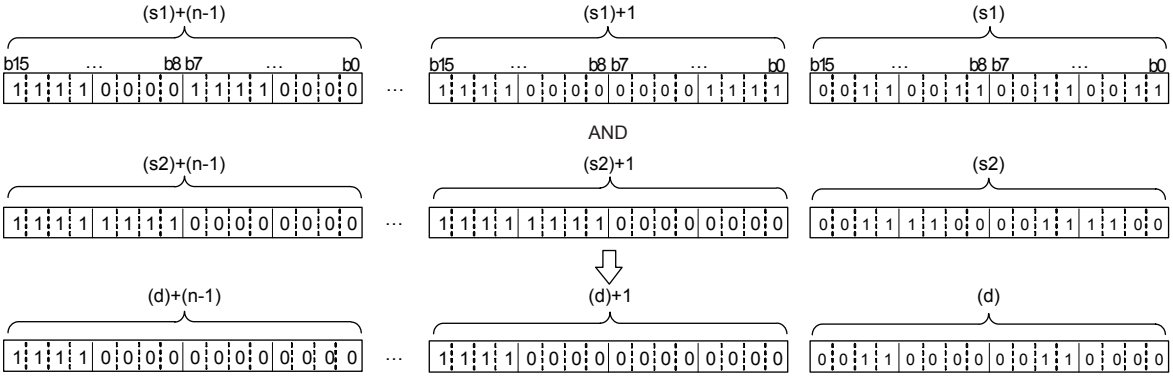
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)*1	—	○	—	—	—	—	○	—	—	—	—
(s2)*1	—	○	—	—	—	—	○	○	—	—	—
(d)*1	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 The same device number can be specified for (s1) and (d) or (s2) and (d).

Processing details

- These instructions AND contents of (n) point(s) from the device specified by (s1) and (n) point(s) from the device specified by (s2), and store the results in the devices specified by (d) onwards.



Operation error

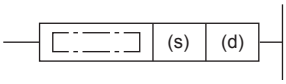
Error code (SD0/SD8067)	Description
2820H	The range of (n) point(s) of data starting from the device specified by (s1), (s2), or (d) exceed the corresponding device range.
2821H	Device range of (n) point(s) from (s1) partially overlaps with device range of (n) point(s) from (d). (Does not apply when same device has been specified for (s1) and (d).)
	Device range of (n) point(s) from (s2) partially overlaps with device range of (n) point(s) from (d). (Does not apply when same device has been specified for (s2) and (d).)

Performing an OR operation on 16-bit data

WOR(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions OR each bit of 16-bit binary data from the device specified by (d) and each bit of 16-bit binary data from device specified by (s), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 288 WOR(P) [using three operands]

FBD/LD
Not supported. ☞ Page 288 WOR(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

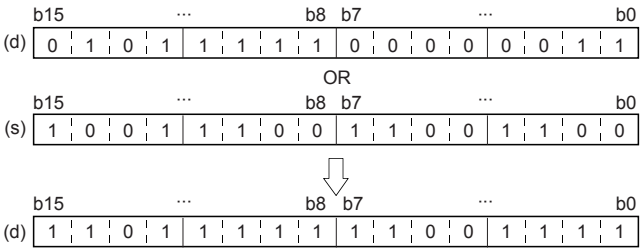
Operand	Description	Range	Data type	Data type (label)
(s)	Data for OR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device for storing the OR results	-32768 to +32767	16-bit signed binary	ANY16

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions OR each bit of 16-bit binary data from the device specified by (d) and each bit of 16-bit binary data from device specified by (s), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

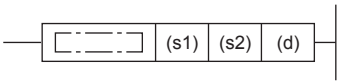
Operation error

There is no operation error.

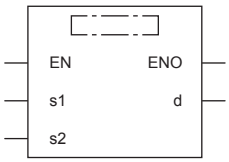
WOR(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions OR each bit of 16-bit binary data from the device specified by (s1) and each bit of 16-bit binary data from device specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	ENO:=WOR(EN,s1,s2,d); ENO:=WORP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

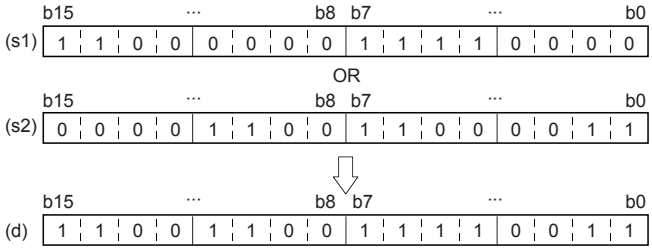
Operand	Description	Range	Data type	Data type (label)
(s1)	Data for OR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Data for OR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device for storing the OR results	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions OR each bit of 16-bit binary data from the device specified by (s1) and each bit of 16-bit binary data from device specified by (s2), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

Operation error

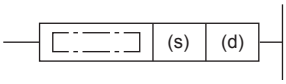
There is no operation error.

Performing an OR operation on 32-bit data

DOR(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions OR each bit of 32-bit binary data from the device specified by (d) and each bit of 32-bit binary data from device specified by (s), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 290 DOR(P) [using three operands]

FBD/LD
Not supported. ☞ Page 290 DOR(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

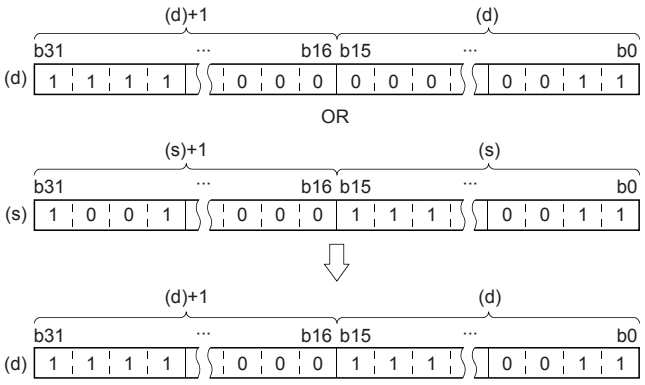
Operand	Description	Range	Data type	Data type (label)
(s)	Data for OR or head device where data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Head device for storing the OR results	-2147483648 to +2147483647	32-bit signed binary	ANY32

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions OR each bit of 32-bit binary data from the device specified by (d) and each bit of 32-bit binary data from device specified by (s), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

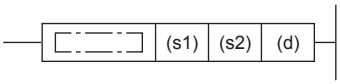
Operation error

There is no operation error.

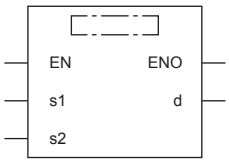
DOR(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions OR each bit of 32-bit binary data from the device specified by (s1) and each bit of 32-bit binary data from device specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DOR(EN,s1,s2,d); ENO:=DORP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

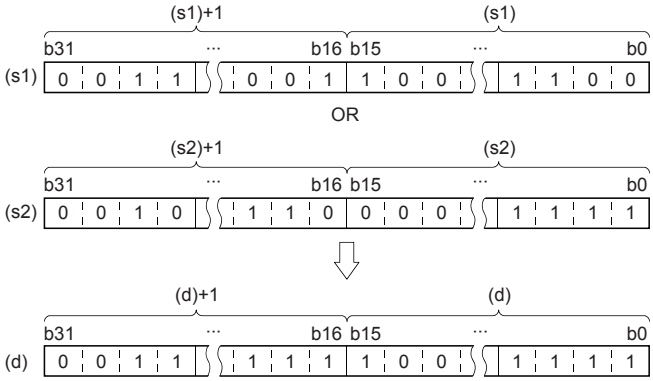
Operand	Description	Range	Data type	Data type (label)
(s1)	Data for OR or head device where data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Data for OR or head device where data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Head device for storing the OR results	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions OR each bit of 32-bit binary data from the device specified by (s1) and each bit of 32-bit binary data from device specified by (s2), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

Operation error

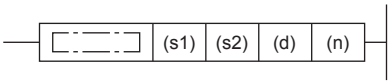
There is no operation error.

Performing an OR operation on 16-bit block data

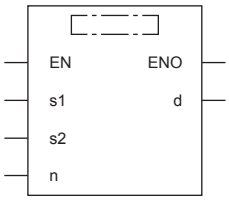
BKOR(P)

FX5S FX5UJ FX5U FX5UC

These instructions OR contents of (n) point(s) from the device specified by (s1) and (n) point(s) from the device specified by (s2), and store the results in the devices specified by (d) onwards.

Ladder diagram	Structured text
	ENO:=BKOR(EN,s1,s2,n,d); ENO:=BKORP(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device where the logical operation data is stored	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Logical operation data or the head device where the logical operation data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device for storing the operation result	—	16-bit signed binary	ANY16
(n)	Number of data	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

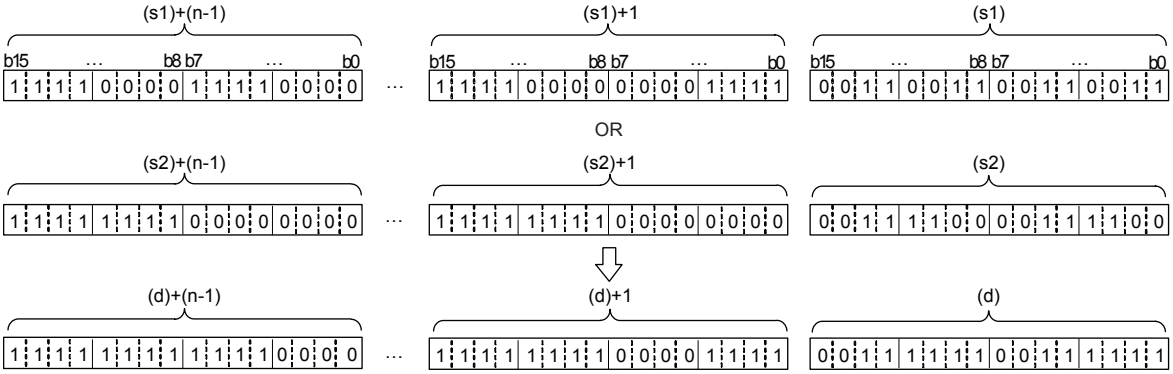
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)*1	—	○	—	—	—	—	○	—	—	—	—
(s2)*1	—	○	—	—	—	—	○	○	—	—	—
(d)*1	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 The same device number can be specified for (s1) and (d) or (s2) and (d).

Processing details

- These instructions OR contents of (n) point(s) from the device specified by (s1) and (n) point(s) from the device specified by (s2), and store the results in the devices specified by (d) onwards.



Operation error

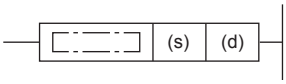
Error code (SD0/SD8067)	Description
2820H	The range of (n) point(s) of data starting from the device specified by (s1), (s2), or (d) exceed the corresponding device range.
2821H	Device range of (n) point(s) from (s1) partially overlaps with device range of (n) point(s) from (d). (Does not apply when same device has been specified for (s1) and (d).)
	Device range of (n) point(s) from (s2) partially overlaps with device range of (n) point(s) from (d). (Does not apply when same device has been specified for (s2) and (d).)

Performing an XOR operation on 16-bit data

WXOR(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions exclusive OR each bit of 16-bit binary data from the device specified by (d) and each bit of 16-bit binary data from device specified by (s), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 295 WXOR(P) [using three operands]

FBD/LD
Not supported. ☞ Page 295 WXOR(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Data for exclusive OR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device for storing exclusive OR results	-32768 to +32767	16-bit signed binary	ANY16

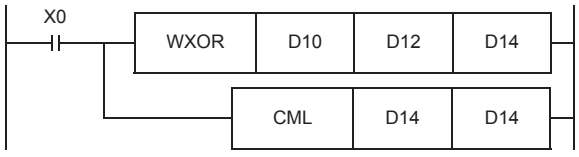
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions exclusive OR each bit of 16-bit binary data from the device specified by (d) and each bit of 16-bit binary data from device specified by (s), and store the results in the device specified by (d).

- Bit devices subsequent to number of points by digit specification are calculated as 0.
- By combining WXOR and CML instructions, the exclusive logical sum not (XORNOT) operation can be executed as well as the WXNR instruction.



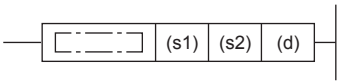
Operation error

There is no operation error.

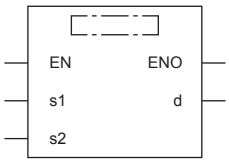
WXOR(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions exclusive OR each bit of 16-bit binary data from the device specified by (s1) and each bit of 16-bit binary data from device specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	ENO:=WXOR(EN,s1,s2,d); ENO:=WXORP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

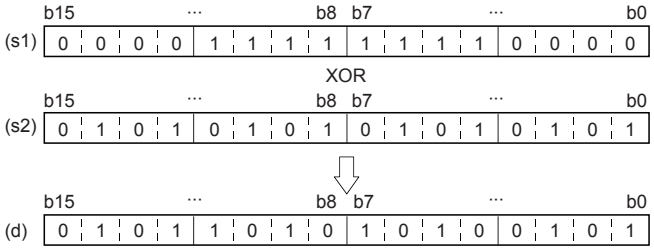
Operand	Description	Range	Data type	Data type (label)
(s1)	Data for exclusive OR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Data for exclusive OR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device for storing exclusive OR results	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions exclusive OR each bit of 16-bit binary data from the device specified by (s1) and each bit of 16-bit binary data from device specified by (s2), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

Operation error

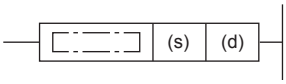
There is no operation error.

Performing an XOR operation on 32-bit data

DXOR(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions exclusive OR each bit of 32-bit binary data from the device specified by (d) and each bit of 32-bit binary data from device specified by (s), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 297 DXOR(P) [using three operands]

FBD/LD
Not supported. ☞ Page 297 DXOR(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

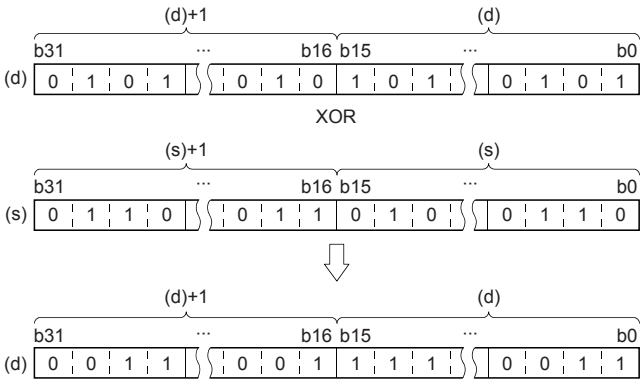
Operand	Description	Range	Data type	Data type (label)
(s)	Data for exclusive OR or head device where data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Head device for storing exclusive OR results	-2147483648 to +2147483647	32-bit signed binary	ANY32

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions exclusive OR each bit of 32-bit binary data from the device specified by (d) and each bit of 32-bit binary data from device specified by (s), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

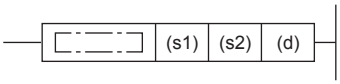
Operation error

There is no operation error.

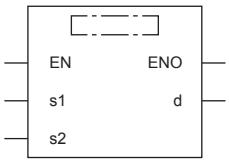
DXOR(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions exclusive OR each bit of 32-bit binary data from the device specified by (s1) and each bit of 32-bit binary data from device specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DXOR(EN,s1,s2,d); ENO:=DXORP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

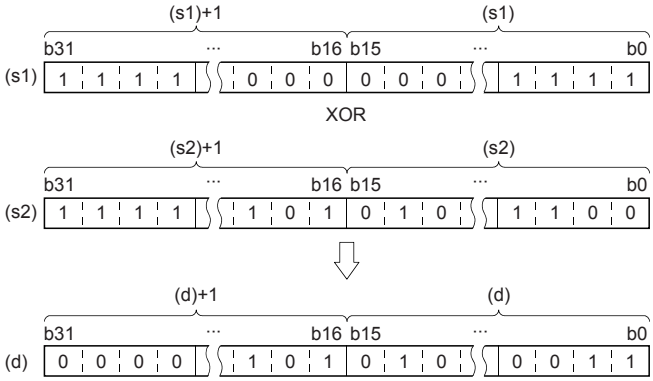
Operand	Description	Range	Data type	Data type (label)
(s1)	Data for exclusive OR or head device where data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Data for exclusive OR or head device where data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Head device for storing exclusive OR results	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions exclusive OR each bit of 32-bit binary data from the device specified by (s1) and each bit of 32-bit binary data from device specified by (s2), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

Operation error

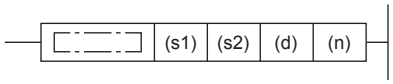
There is no operation error.

Performing an XOR operation on 16-bit block data

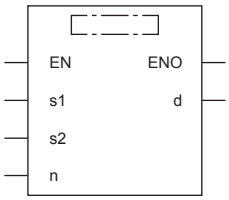
BKXOR(P)

FX5S FX5UJ FX5U FX5UC

These instructions exclusive OR contents of (n) point(s) from the device specified by (s1) and (n) point(s) from the device specified by (s2), and store the results in the devices specified by (d) onwards.

Ladder diagram	Structured text
	ENO:=BKXOR(EN,s1,s2,n,d); ENO:=BKXORP(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Data for exclusive OR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Data for exclusive OR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device for storing the operation result	—	16-bit signed binary	ANY16
(n)	Number of data	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

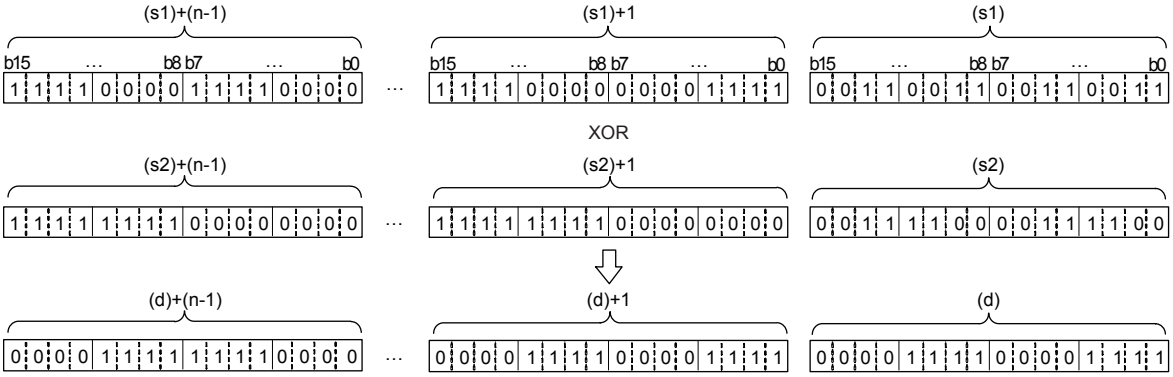
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)*1	—	○	—	—	—	—	○	—	—	—	—
(s2)*1	—	○	—	—	—	—	○	○	—	—	—
(d)*1	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 The same device number can be specified for (s1) and (d) or (s2) and (d).

Processing details

- These instructions exclusive OR contents of (n) point(s) from the device specified by (s1) and (n) point(s) from the device specified by (s2), and store the results in the devices specified by (d) onwards.



Operation error

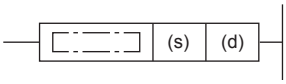
Error code (SD0/SD8067)	Description
2820H	The range of (n) point(s) of data starting from the device specified by (s1), (s2), or (d) exceed the corresponding device range.
2821H	Device range of (n) point(s) from (s1) partially overlaps with device range of (n) point(s) from (d). (Does not apply when same device has been specified for (s1) and (d).)
	Device range of (n) point(s) from (s2) partially overlaps with device range of (n) point(s) from (d). (Does not apply when same device has been specified for (s2) and (d).)

Performing an XNOR operation on 16-bit data

WXNR(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions exclusive NOR each bit of 16-bit binary data from the device specified by (d) and each bit of 16-bit binary data from device specified by (s), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 302 WXNR(P) [using three operands]

FBD/LD
Not supported. ☞ Page 302 WXNR(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Data for exclusive NOR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device for storing exclusive NOR results	-32768 to +32767	16-bit signed binary	ANY16

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions exclusive NOR each bit of 16-bit binary data from the device specified by (d) and each bit of 16-bit binary data from device specified by (s), and store the results in the device specified by (d).

- Bit devices subsequent to number of points by digit specification are calculated as 0.

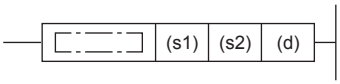
Operation error

There is no operation error.

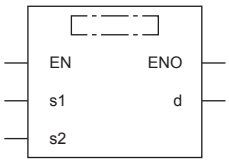
WXNR(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions exclusive NOR each bit of 16-bit binary data from the device specified by (s1) and each bit of 16-bit binary data from device specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	ENO:=WXNR(EN,s1,s2,d); ENO:=WXNRP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

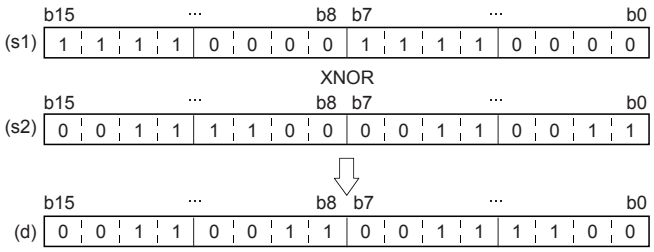
Operand	Description	Range	Data type	Data type (label)
(s1)	Data for exclusive NOR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Data for exclusive NOR or head device where data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device for storing exclusive NOR results	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions exclusive NOR each bit of 16-bit binary data from the device specified by (s1) and each bit of 16-bit binary data from device specified by (s2), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

Operation error

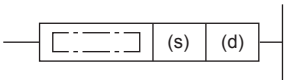
There is no operation error.

Performing an XNOR operation on 32-bit data

DXNR(P) [using two operands]

FX5S FX5UJ FX5U FX5UC

These instructions exclusive NOR each bit of 32-bit binary data from the device specified by (d) and each bit of 32-bit binary data from device specified by (s), and store the results in the device specified by (d).

Ladder diagram	Structured text
	Not supported ☞ Page 304 DXNR(P) [using three operands]

FBD/LD
Not supported. ☞ Page 304 DXNR(P) [using three operands]

Setting data

■Descriptions, ranges, and data types

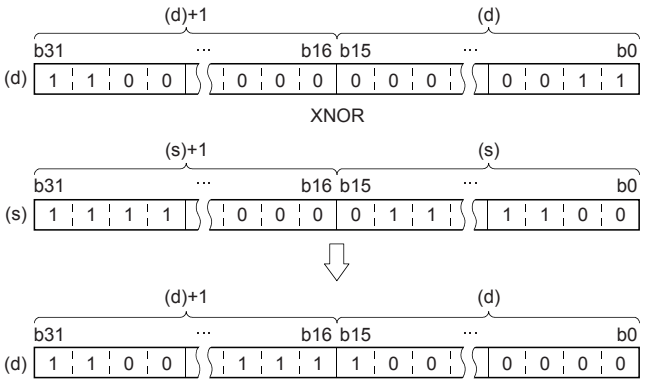
Operand	Description	Range	Data type	Data type (label)
(s)	Data for exclusive NOR or head device where data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Head device for storing exclusive NOR results	-2147483648 to +2147483647	32-bit signed binary	ANY32

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions exclusive NOR each bit of 32-bit binary data from the device specified by (d) and each bit of 32-bit binary data from device specified by (s), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

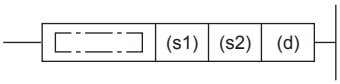
Operation error

There is no operation error.

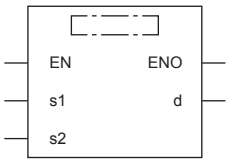
DXNR(P) [using three operands]

FX5S FX5UJ FX5U FX5UC

These instructions exclusive NOR each bit of 32-bit binary data from the device specified by (s1) and each bit of 32-bit binary data from device specified by (s2), and store the results in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DXNR(EN,s1,s2,d); ENO:=DXNRP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

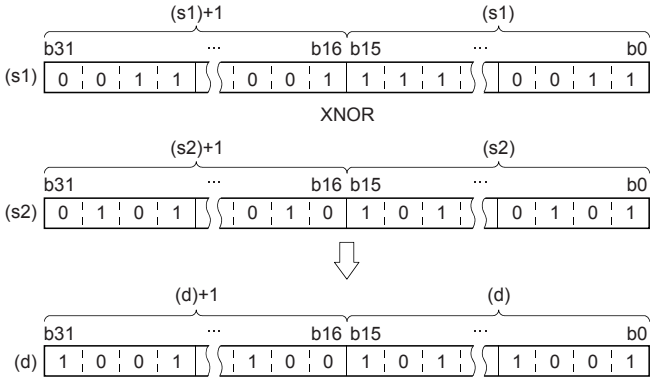
Operand	Description	Range	Data type	Data type (label)
(s1)	Data for exclusive NOR or head device where data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Data for exclusive NOR or head device where data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Head device for storing exclusive NOR results	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions exclusive NOR each bit of 32-bit binary data from the device specified by (s1) and each bit of 32-bit binary data from device specified by (s2), and store the results in the device specified by (d).



- Bit devices subsequent to number of points by digit specification are calculated as 0.

Operation error

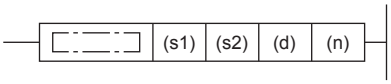
There is no operation error.

Performing an XNOR operation on 16-bit block data

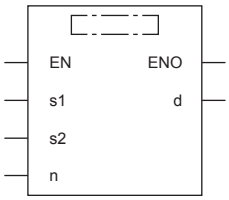
BKXNR(P)

FX5S FX5UJ FX5U FX5UC

These instructions exclusive NOR contents of (n) point(s) from the device specified by (s1) and (n) point(s) from the device specified by (s2), and store the results in the devices specified by (d) onward.

Ladder diagram	Structured text
	ENO:=BKXNR(EN,s1,s2,n,d); ENO:=BKXNRP(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device where the logical operation data is stored	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Logical operation data or the head device where the logical operation data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device for storing the operation result	—	16-bit signed binary	ANY16
(n)	Number of data	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

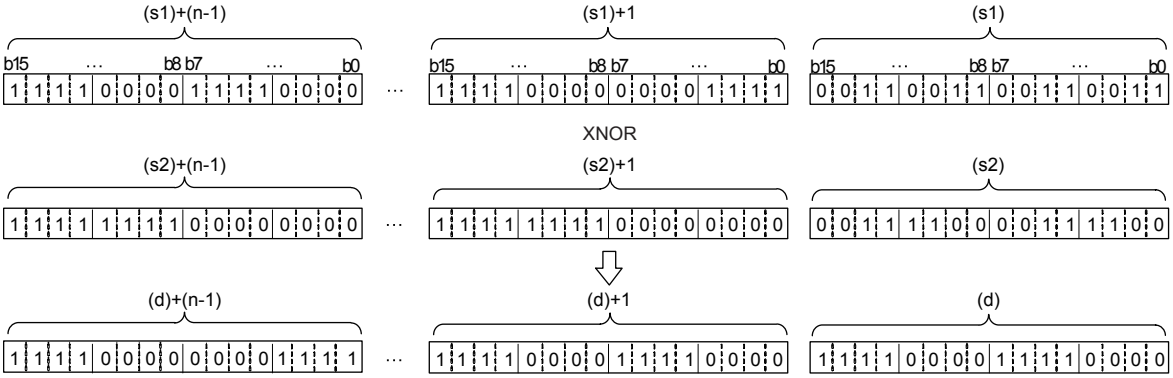
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)*1	—	○	—	—	—	—	○	—	—	—	—
(s2)*1	—	○	—	—	—	—	○	○	—	—	—
(d)*1	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 The same device number can be specified for (s1) and (d) or (s2) and (d).

Processing details

- These instructions exclusive NOR contents of (n) point(s) from the device specified by (s1) and (n) point(s) from the device specified by (s2), and store the results in the devices specified by (d) onward.



Operation error

Error code (SD0/SD8067)	Description
2820H	The range of (n) point(s) of data starting from the device specified by (s1), (s2), or (d) exceed the corresponding device range.
2821H	Device range of (n) point(s) from (s1) partially overlaps with device range of (n) point(s) from (d). (Does not apply when same device has been specified for (s1) and (d).)
	Device range of (n) point(s) from (s2) partially overlaps with device range of (n) point(s) from (d). (Does not apply when same device has been specified for (s2) and (d).)

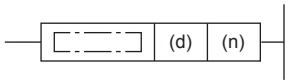
7.4 Bit Processing Instructions

Setting a bit in the word device

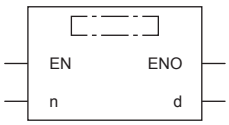
BSET(P)

FX5S FX5UJ FX5U FX5UC

These instructions set (to 1) (n)th bit of word device specified by (d).

Ladder diagram	Structured text
	ENO:=BSET(EN,n,d); ENO:=BSETP(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

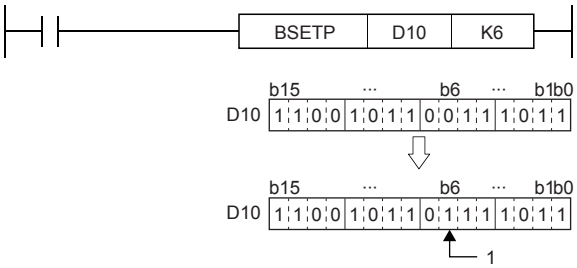
Operand	Description	Range	Data type	Data type (label)
(d)	Head device for which bit is to be set	—	16-bit signed binary	ANY16
(n)	Number of bit(s) to be set	0 to 15	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions set (to 1) (n)th bit of word device specified by (d).
- If (n) exceeds 15, the processing will be done based on the lower 4 bits of (n).



Operation error

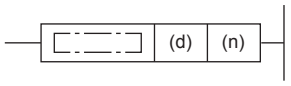
There is no operation error.

Resetting a bit in the word device

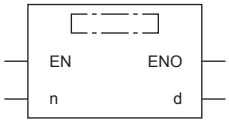
BRST(P)

FX5S FX5UJ FX5U FX5UC

These instructions reset (to 0) (n)th bit of word device specified by (d).

Ladder diagram	Structured text
	ENO:=BRST(EN,n,d); ENO:=BRSTP(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

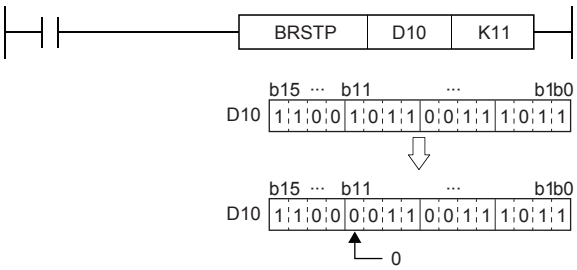
Operand	Description	Range	Data type	Data type (label)
(d)	Head device for which bit is to be reset	—	16-bit signed binary	ANY16
(n)	Number of bit(s) to be reset	0 to 15	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions reset (to 0) (n)th bit of word device specified by (d).
- If (n) exceeds 15, the processing will be done based on the lower 4 bits of (n).



Operation error

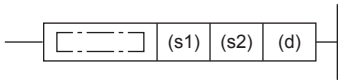
There is no operation error.

Performing a 16-bit test

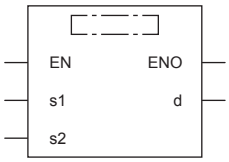
TEST(P)

FX5S FX5UJ FX5U FX5UC

These instructions take bit data at position specified by (s2) from device specified by (s1) and write to bit device specified by (d).

Ladder diagram	Structured text
	ENO:=TEST(EN,s1,s2,d); ENO:=TESTP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Device number where bit data to be extracted is stored	—	16-bit signed binary	ANY16
(s2)	Position of bit data to be extracted	0 to 15	16-bit unsigned binary	ANY16
(d)	Bit device number where extracted bit data is to be stored	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

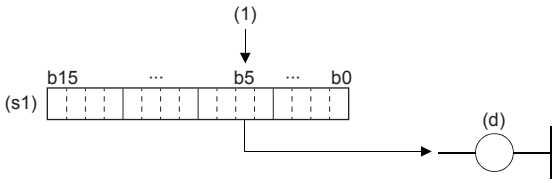
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	—	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○*1	○	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions take bit data at position specified by (s2) from device specified by (s1) and write to bit device specified by (d).



(1): (s2) bit (When (s2)=5)

- If relevant bit is "0", device specified by (d) is turned OFF, and if it is "1", device is turned ON.
- For (s2) specify the bit position (0 to 15) of word data. If 16 or more is specified for (s2), the value of the remainder of (s2)÷16 is the bit position.

Ex.

For (s2) = 18, the remainder for 18÷16 is "2", so it becomes data of b2.

Operation error

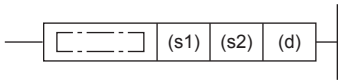
There is no operation error.

Performing a 32-bit test

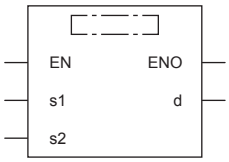
DTEST(P)

FX5S FX5UJ FX5U FX5UC

These instructions take bit data at position specified by (s2) from device specified by (s1) and write to bit device specified by (d).

Ladder diagram	Structured text
	ENO:=DTEST(EN,s1,s2,d); ENO:=DTESTP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Device number where bit data to be extracted is stored	—	32-bit signed binary	ANY32
(s2)	Position of bit data to be extracted	0 to 31	16-bit unsigned binary	ANY16
(d)	Bit device number where extracted bit data is to be stored	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

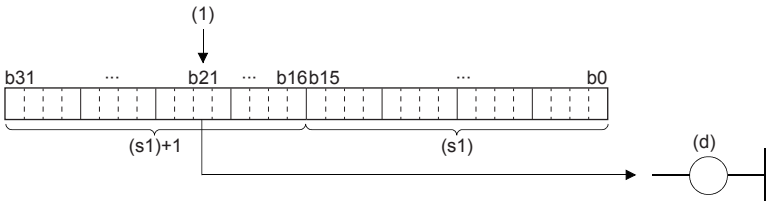
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	—	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○*1	○	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions take bit data at position specified by (s2) from device specified by (s1), (s1) +1 and write to bit device specified by (d).



(1): (s2) bit (When (s2)=21)

- If relevant bit is "0", device specified by (d) is turned OFF, and if it is "1", device is turned ON.
- For (s2) specify the bit position (0 to 31) of double word data. If 32 or more is specified for (s2), the value of the remainder of (s2)÷32 is the bit position.

Ex.

For (s2) = 34, the remainder for 34÷32 is "2", so it becomes data of b2.

Operation error

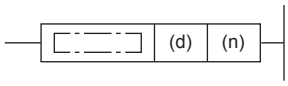
There is no operation error.

Batch-resetting bit devices

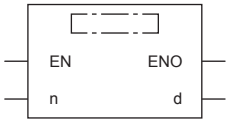
BKRST(P)

FX5S FX5UJ FX5U FX5UC

These instructions reset (n) point(s) bit devices from the bit device specified by (d).

Ladder diagram	Structured text
	ENO:=BKRST(EN,n,d); ENO:=BKRSTP(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device to be reset	—	Bit	ANY_BOOL
(n)	Number of devices to be reset	—	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	—	—	○	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions reset (n) point(s) bit devices from the bit device specified by (d).
- Reset status of bit device is as follows.

Device	Status
Annunciator (F)	• (n) point(s) from annunciator (F) number specified by (d) are turned OFF. • Annunciator numbers from SD64 to SD79 that were turned OFF are deleted and the subsequent numbers are shifted forward. • The number of annunciators stored in SD64 to SD79 is stored in SD63.
Timer (T), Counter (C)	• Current value of (n) point(s) from timer (T) or counter (C) number specified by (d) is set to 0, and coil contact is turned OFF.
Bit devices other than given above	• Coils and contacts of (n) point(s) from the device specified by (d) are turned OFF.

- If specified devices are OFF, device status remains unchanged.

Operation error

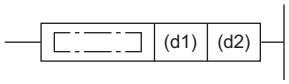
Error code (SD0/SD8067)	Description
2820H	(n) point(s) of data starting from the device specified by (d) exceed the corresponding device range.

Batch-resetting devices

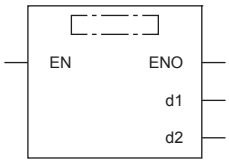
ZRST(P)

FX5S FX5UJ FX5U FX5UC

These instructions reset all data among devices of same type specified by (d1) and (d2). Use these instructions for restarting operation from the beginning after pause or after resetting control data.

Ladder diagram	Structured text
	ENO:=ZRST(EN, d1, d2); ENO:=ZRSTP(EN, d1, d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d1)	Head bit or word device number to be reset	—	Bit/16-bit signed binary	ANY_ELEMENTARY*1
(d2)	Last bit or word device number to be reset	—	Bit/16-bit signed binary	ANY_ELEMENTARY*1
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

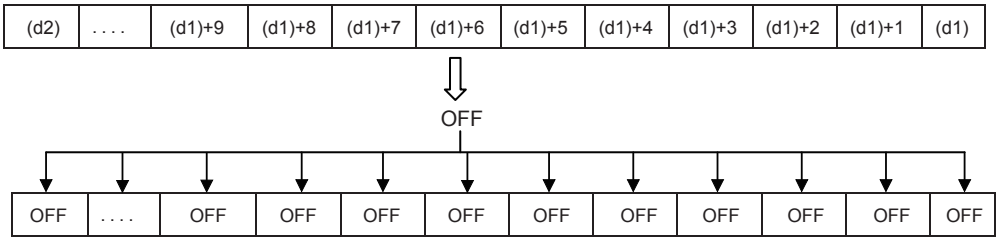
*1 Digit specified bit type label cannot be used.

■Applicable devices

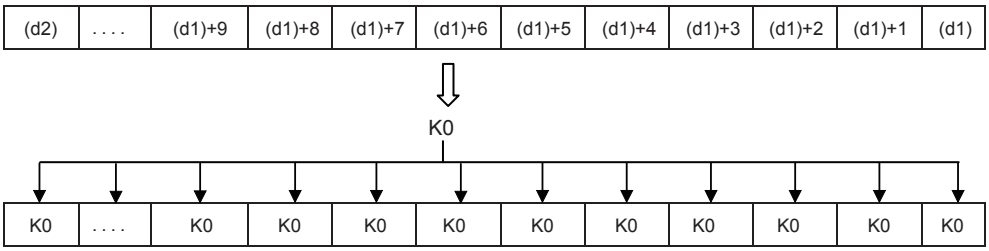
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d1)	○	○	○	○	○	○	○	—	—	—	—
(d2)	○	○	○	○	○	○	○	—	—	—	—

Processing details

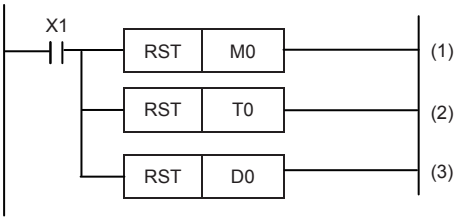
- These instructions reset all data among devices of same type specified by (d1) and (d2).
- OFF (reset) is written to the entire range of devices from (d1) to (d2) all at once if (d1) and/or (d2) are bit devices.



- K0 is written to the entire range of devices from (d1) to (d2) all at once if (d1) and/or (d2) are word devices.

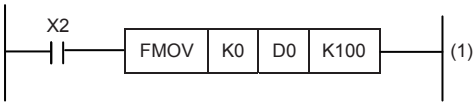


- As a reset instruction for individual devices, the RST instruction can be used for bit devices and word devices.



(1): M0 is reset.
(2): The current value of T0 is reset.
(3): D0 is reset.

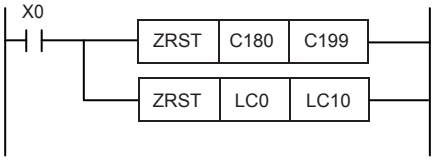
- The FMOV(P) instruction is a batch write instruction for a constant (K0 for example) that can write "0" for word devices (including digit specification of bit devices).



(1): K0 is written to D0 to D99.

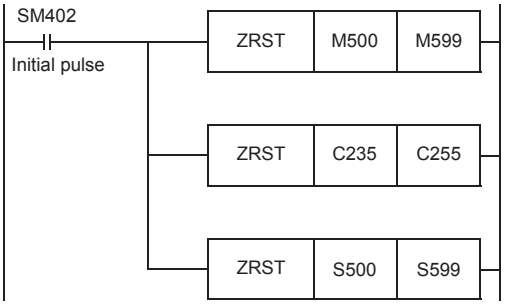
Precautions

- Specify the same type of device for (d1) and (d2) so that (d1) number is less than (d2) number. If the (d1) number≥(d2) number, only the device specified by (d1) is reset.
- The ZRST(P) instruction is a 16-bit instruction, but long counter (LC) and long index register (LZ) can be specified for (d1) and (d2).



Program example

In the program shown below, when the power to the CPU module is turned ON or when the PLC mode is changed to RUN, the specified ranges of bit devices and word devices are reset at one time.



M500 to M599 are reset at one time.

C235 to C255 are reset at one time.
("0" is written to them, and their contacts are reset.)

S500 to S599 are reset at one time.

Operation error

Error code (SD0/SD8067)	Description
2820H	The number of devices to be reset is 32768 or more when module access device has been specified for (d1) and/or (d2).
3405H	Device type specified by (d1) differs from type specified by (d2).
	Module number for (d1) and (d2) differ when module access device is specified.

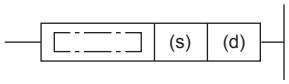
7.5 Data Conversion Instructions

Converting binary data to BCD 4-digit data

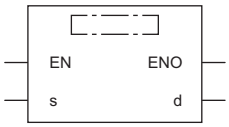
BCD(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the binary data in the device specified by (s) to BCD data, and store the converted data in the device specified by (d).
Binary data is used in operations in CPU module. Use this instruction to display numeric values on seven-segment display unit equipped with BCD decoder.

Ladder diagram	Structured text
	ENO:=BCD(EN,s,d); ENO:=BCDP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

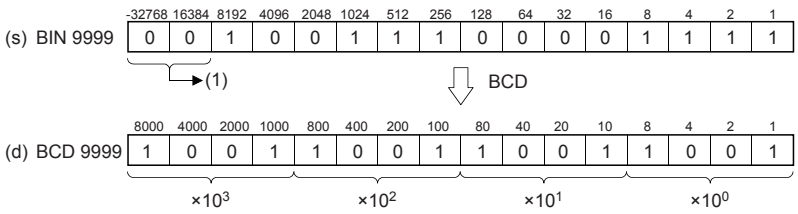
Operand	Description	Range	Data type	Data type (label)
(s)	Binary data or the head device where the binary data is stored	0 to 9999	16-bit signed binary	ANY16
(d)	Head device for storing the BCD data	—	BCD 4-digit	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

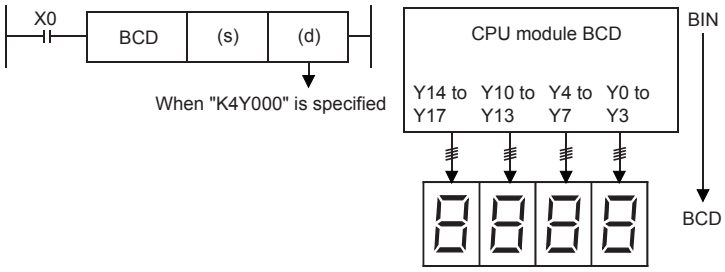
- These instructions convert the 16-bit binary data (0 to 9999) in the device specified by (s) to BCD 4-digit data, and store the converted data in the device specified by (d).



(1): Set 0s.

- Data specified by (s) can be converted if it is within the range from K0 to K9999 BCD (decimal).

• The table below shows digit specification for the data in the device specified by (s) and (d).



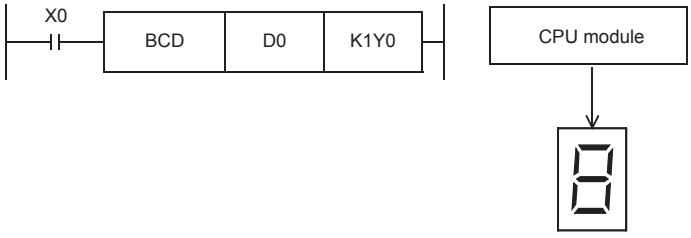
(d)	Number of digits	Data range
K1Y0	1-digit	0 to 9
K2Y0	2-digit	00 to 99
K3Y0	3-digit	000 to 999
K4Y0	4-digit	0000 to 9999

Precautions

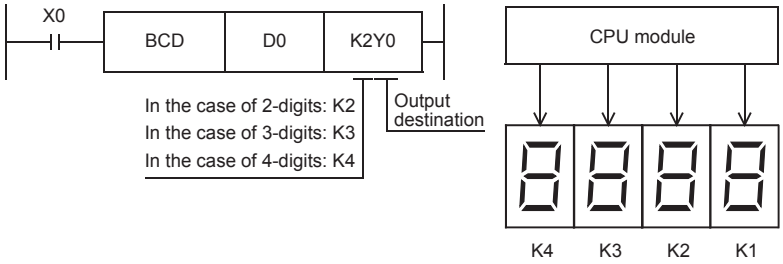
• Binary data is used in all operations in CPU module including arithmetic operations (+-×÷), increment and decrement instructions. When receiving digital switch information in binary-coded decimal (BCD) format into a CPU module, use the BIN(P) instructions (for converting BCD data into binary data). Furthermore, to output data to seven-segment display unit handling binary-coded decimal (BCD) data, use the BCD(P) instructions (for converting binary data into BCD data).

Program example

• When the seven-segment display unit has 1-digit



• When the seven-segment display unit has 2 to 4-digits



Operation error

Error code (SD0/SD8067)	Description
3401H	Data in the device specified by (s) is out of the valid range (0 to 9999).

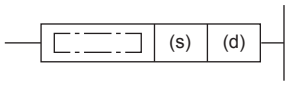
Converting binary data to BCD 8-digit data

DBCD(P)

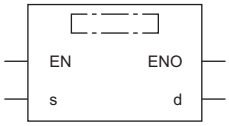
FX5S FX5UJ FX5U FX5UC

These instructions convert the binary data in the device specified by (s) to BCD data, and store the converted data in the device specified by (d).

Binary data is used in operations in CPU module. Use this instruction to display numeric values on seven-segment display unit equipped with BCD decoder.

Ladder diagram	Structured text
	ENO:=DBCD(EN,s,d); ENO:=DBCDP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

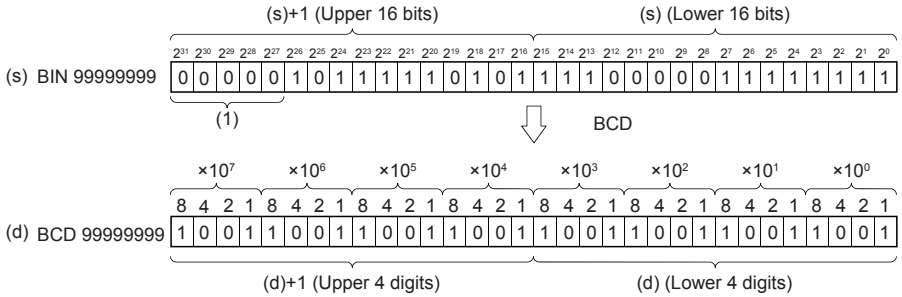
Operand	Description	Range	Data type	Data type (label)
(s)	Binary data or the head device where the binary data is stored	0 to 99999999	32-bit signed binary	ANY32
(d)	Head device for storing the BCD data	—	BCD 8-digit	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

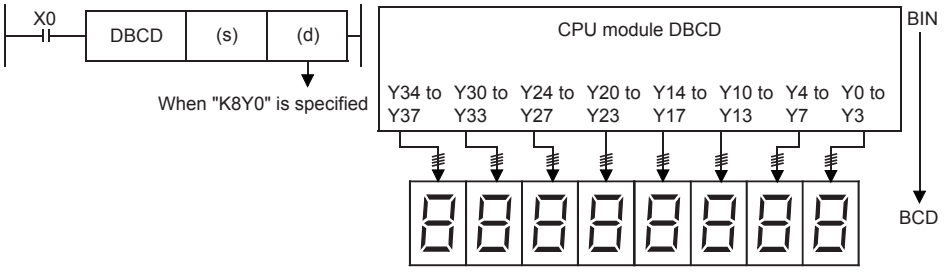
- These instructions convert 32-bit binary data (0 to 99999999) in device specified by (s) to BCD 8-digit data, and store the converted data in the device specified by (d).



(1): Set 0s to the upper 5 bits.

- Data specified by (s) can be converted if it is within the range from K0 to K99999999 BCD (decimal).

• The table below shows digit specification for the data in the device specified by (s) and (d).



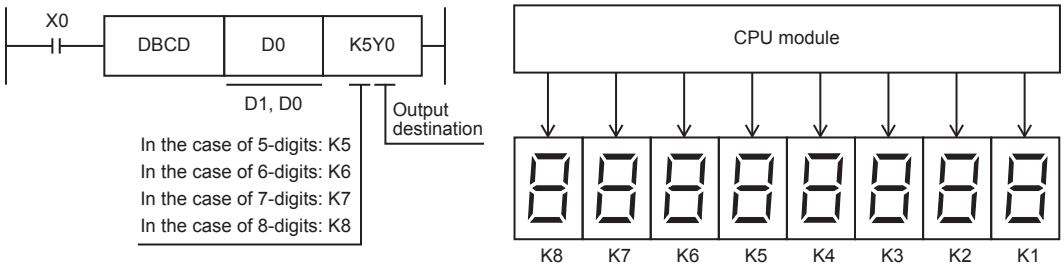
(d)+1, (d)	Number of digits	Data range
K1Y0	1-digit	0 to 9
K2Y0	2-digit	00 to 99
K3Y0	3-digit	000 to 999
K4Y0	4-digit	0000 to 9999
K5Y0	5-digit	00000 to 99999
K6Y0	6-digit	000000 to 999999
K7Y0	7-digit	0000000 to 9999999
K8Y0	8-digit	00000000 to 99999999

Precautions

• Binary data is used in all operations in CPU module including arithmetic operations (+-×÷), increment and decrement instructions. When receiving digital switch information in binary-coded decimal (BCD) format into a CPU module, use the BIN(P) instructions (for converting BCD data into binary data). Furthermore, to output data to seven-segment display unit handling binary-coded decimal (BCD) data, use the BCD(P) instructions (for converting binary data into BCD data).

Program example

• When the seven-segment display unit has 5 to 8-digits



Operation error

Error code (SD0/SD8067)	Description
3401H	Data in the device specified by (s) is out of the valid range (0 to 99999999).

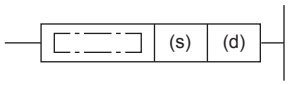
Converting BCD 4-digit data to binary data

BIN(P)

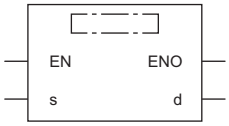
FX5S FX5UJ FX5U FX5UC

These instructions convert the binary-coded decimal data in the device specified by (s) to binary data, and store the converted data in the device specified by (d).

Use this instruction to convert a binary-coded decimal (BCD) value such as a value set by a digital switch into binary (BIN) data and to receive the converted binary data so that the data can be handled in operations in CPU module.

Ladder diagram	Structured text
	ENO:=BIN(EN,s,d); ENO:=BINP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

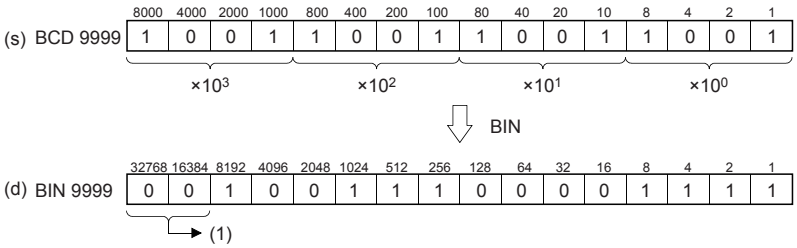
Operand	Description	Range	Data type	Data type (label)
(s)	Binary-coded decimal data or the head device where the binary-coded decimal data is stored	0 to 9999	BCD 4-digit	ANY16
(d)	Head device for storing the binary data	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

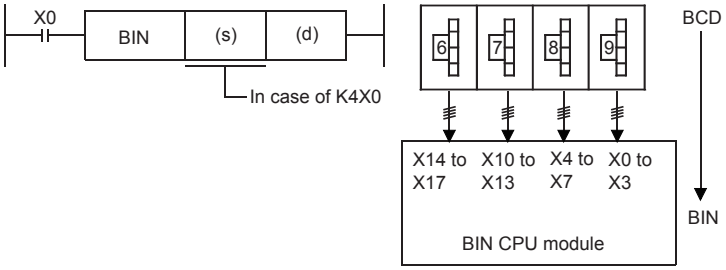
- These instructions convert the BCD 4-digit data (0 to 9999) in the device specified by (s) to 16-bit binary data, and store the converted data in the device specified by (d).



(1): Filled with 0s.

- The data in the device specified by (s) can be converted if it is in the range from 0 to 9999 (BCD).

• The table below shows digit specification for the data in the device specified by (s) and (d).



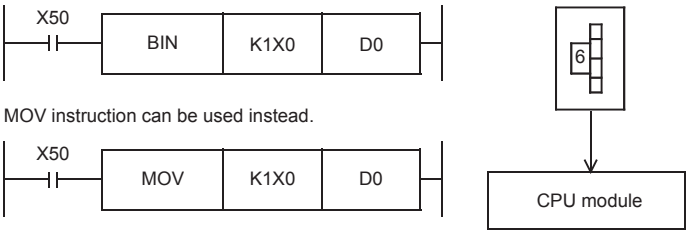
(d)	Number of digits	Data range
K1X0	1-digit	0 to 9
K2X0	2-digit	00 to 99
K3X0	3-digit	000 to 999
K4X0	4-digit	0000 to 9999

Precautions

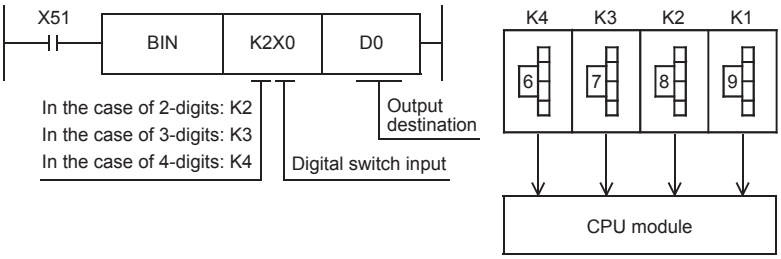
• Binary data is used in all operations in CPU module including arithmetic operations (+-×÷), increment and decrement instructions. When receiving digital switch information in binary-coded decimal (BCD) format into a CPU module, use the BIN(P) instructions (for converting BCD data into binary data). Furthermore, to output data to seven-segment display unit handling binary-coded decimal (BCD) data, use the BCD(P) instructions (for converting binary data into BCD data).

Program example

• When the digital switch has 1-digit



• When the digital switch has 2 to 4-digits



Operation error

Error code (SD0/SD8067)	Description
3401H	The value of each digit of the device specified by (s) is other than 0 to 9. (The data is not binary-coded decimal data.)

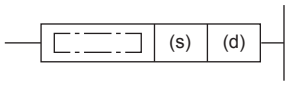
Converting BCD 8-digit data to binary data

DBIN(P)

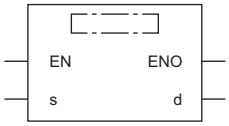
FX5S FX5UJ FX5U FX5UC

These instructions convert the binary-coded decimal data in the device specified by (s) to binary data, and store the converted data in the device specified by (d).

Use this instruction to convert a binary-coded decimal (BCD) value such as a value set by a digital switch into binary (BIN) data and to receive the converted binary data so that the data can be handled in operations in CPU module.

Ladder diagram	Structured text
	ENO:=DBIN(EN,s,d); ENO:=DBINP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

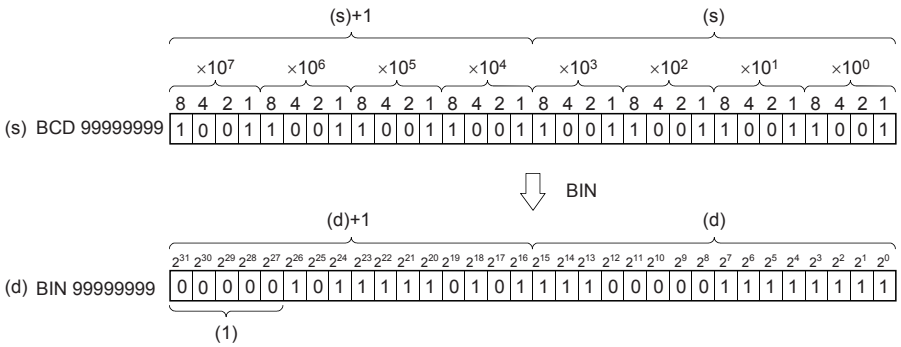
Operand	Description	Range	Data type	Data type (label)
(s)	Binary-coded decimal data or the head device where the binary-coded decimal data is stored	0 to 99999999	BCD 8-digit	ANY32
(d)	Head device for storing the binary data	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

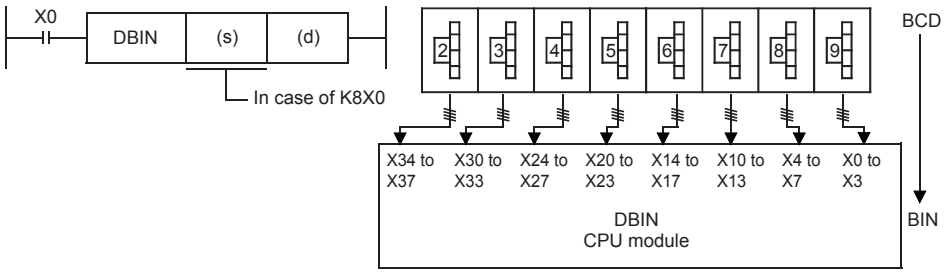
- These instructions convert the BCD 8-digit data (0 to 99999999) in the device specified by (s) to 32-bit binary data, and store the converted data in the device specified by (d).



(1): Filled with 0s.

- The data in the device specified by (s) can be converted if it is in the range from 0 to 99999999 (BCD).

• The table below shows digit specification for the data in the device specified by (s) and (d).



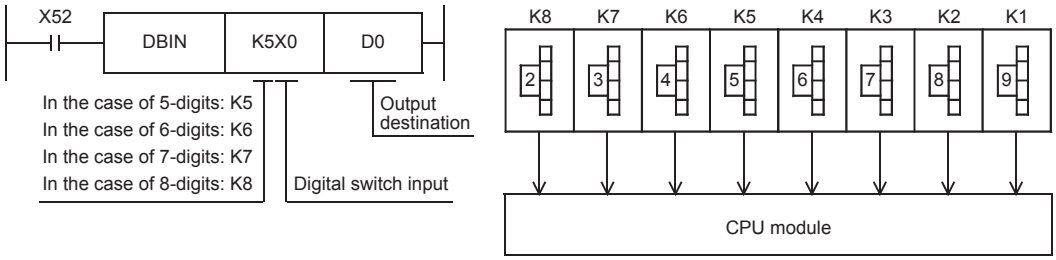
(s)+1, (s)	Number of digits	Data range
K1X0	1-digit	0 to 9
K2X0	2-digit	00 to 99
K3X0	3-digit	000 to 999
K4X0	4-digit	0000 to 9999
K5X0	5-digit	00000 to 99999
K6X0	6-digit	000000 to 999999
K7X0	7-digit	0000000 to 9999999
K8X0	8-digit	00000000 to 99999999

Precautions

• Binary data is used in all operations in CPU module including arithmetic operations (+-×÷), increment and decrement instructions. When receiving digital switch information in binary-coded decimal (BCD) format into a CPU module, use the BIN instruction (for converting BCD data into binary data). Furthermore, to output data to seven-segment display unit handling binary-coded decimal (BCD) data, use the BCD instruction (for converting binary data into BCD data).

Program example

• When the digital switch has 5 to 8-digits



Operation error

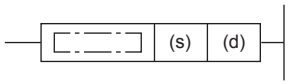
Error code (SD0/SD8067)	Description
3401H	The value of each digit of the device specified by (s) is other than 0 to 9. (The data is not binary-coded decimal data.)

Converting single-precision real number to 16-bit signed binary data

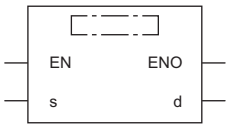
FLT2INT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the single-precision real number in the device specified by (s) to 16-bit signed binary data, and store the converted data in the device specified by (d). After conversion, the digits after the decimal point of the single-precision real number, specified with (s), are truncated.

Ladder diagram	Structured text*1
	ENO:=FLT2INT(EN,s,d); ENO:=FLT2INTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

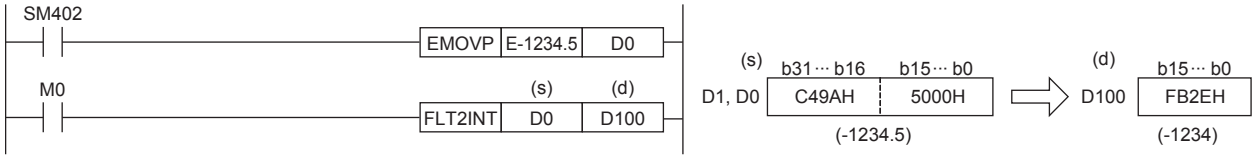
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	-32768 to +32767	Single-precision real number	ANYREAL_32
(d)	Data after conversion	—	16-bit signed binary	ANY16_S
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions convert the single-precision real number in the device specified by (s) to 16-bit signed binary data, and store the converted data in the device specified by (d). After conversion, the digits after the decimal point of the single-precision real number, specified with (s), are truncated.



- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The input value is out of range.	The operation result is not reflected to (d) and the carry flag (SM700) turns on.
SM8020	Zero	The input value is 0.	The value of (d) becomes 0 and the zero flag (SM8020) turns on.
SM8021	Borrow	The input value is ignored when it is less than 1.	The value of (d) becomes 0 and the borrow flag (SM8021) turns on.
SM8022	Carry	The input value is out of range.	The operation result is not reflected to (d) and the carry flag (SM8022) turns on.

Operation error

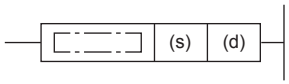
Error code (SD0/SD8067)	Description
3402H	When the contents of the specified device are outside the following range: 0, $2^{-126} \leq$ |specified value (stored value)| $< 2^{128}$
	The specified device value is -0, denormalized number, NaN (not a number), or $\pm\infty$.

Converting single-precision real number to 16-bit unsigned binary data

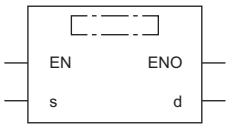
FLT2UINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the single-precision real number in the device specified by (s) to 16-bit unsigned binary data, and store the converted data in the device specified by (d). After conversion, the digits after the decimal point of the single-precision real number, specified with (s), are truncated.

Ladder diagram	Structured text*1
	ENO:=FLT2UINT(EN,s,d); ENO:=FLT2UINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

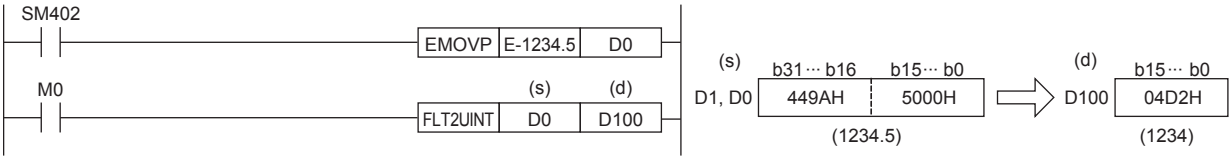
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	0 to 65535	Single-precision real number	ANYREAL_32
(d)	Data after conversion	—	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions convert the single-precision real number in the device specified by (s) to 16-bit unsigned binary data, and store the converted data in the device specified by (d). After conversion, the digits after the decimal point of the single-precision real number, specified with (s), are truncated.



- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The input value is out of range.	The operation result is not reflected to (d) and the carry flag (SM700) turns on.
SM8020	Zero	The input value is 0.	The value of (d) becomes 0 and the zero flag (SM8020) turns on.
SM8021	Borrow	The input value is ignored when it is less than 1.	The value of (d) becomes 0 and the borrow flag (SM8021) turns on.
SM8022	Carry	The input value is out of range.	The operation result is not reflected to (d) and the carry flag (SM8022) turns on.

Operation error

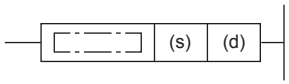
Error code (SD0/SD8067)	Description
3402H	When the contents of the specified device are outside the following range: 0, $2^{-126} \leq$ |specified value (stored value)| $< 2^{128}$
	The specified device value is -0, denormalized number, NaN (not a number), or $\pm\infty$.

Converting single-precision real number to 32-bit signed binary data

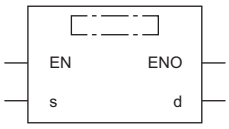
FLT2DINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the single-precision real number in the device specified by (s) to 32-bit signed binary data, and store the converted data in the device specified by (d). After conversion, the digits after the decimal point of the single-precision real number, specified with (s), are truncated.

Ladder diagram	Structured text*1
	ENO:=FLT2DINT(EN,s,d); ENO:=FLT2DINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

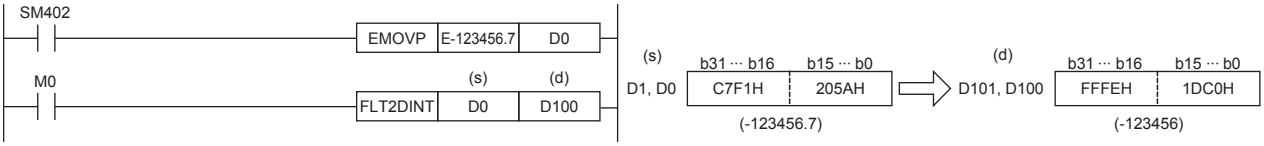
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	-2147483648 to +2147483647	Single-precision real number	ANYREAL_32
(d)	Data after conversion	—	32-bit signed binary	ANY32_S
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions convert the single-precision real number in the device specified by (s) to 32-bit signed binary data, and store the converted data in the device specified by (d). After conversion, the digits after the decimal point of the single-precision real number, specified with (s), are truncated.



- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The input value is out of range.	The operation result is not reflected to (d) and the carry flag (SM700) turns on.
SM8020	Zero	The input value is 0.	The value of (d) becomes 0 and the zero flag (SM8020) turns on.
SM8021	Borrow	The input value is ignored when it is less than 1.	The value of (d) becomes 0 and the borrow flag (SM8021) turns on.
SM8022	Carry	The input value is out of range.	The operation result is not reflected to (d) and the carry flag (SM8022) turns on.

Operation error

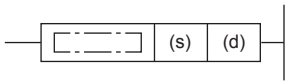
Error code (SD0/SD8067)	Description
3402H	When the contents of the specified device are outside the following range: 0, $2^{-126} \leq$ |specified value (stored value)| $< 2^{128}$
	The specified device value is -0, denormalized number, NaN (not a number), or $\pm\infty$.

Converting single-precision real number to 32-bit unsigned binary data

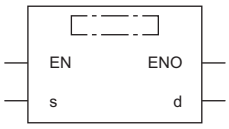
FLT2UDINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the single-precision real number in the device specified by (s) to 32-bit unsigned binary data, and store the converted data in the device specified by (d). After conversion, the digits after the decimal point of the single-precision real number, specified with (s), are truncated.

Ladder diagram	Structured text*1
	ENO:=FLT2UDINT(EN,s,d); ENO:=FLT2UDINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

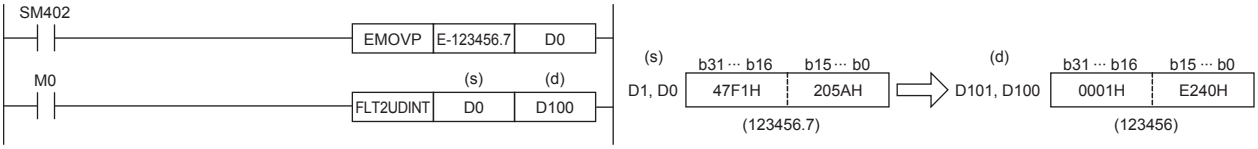
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	0 to 4294967295	Single-precision real number	ANYREAL_32
(d)	Data after conversion	—	32-bit unsigned binary	ANY32_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions convert the single-precision real number in the device specified by (s) to 32-bit unsigned binary data, and store the converted data in the device specified by (d). After conversion, the digits after the decimal point of the single-precision real number, specified with (s), are truncated.



- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The input value is out of range.	The operation result is not reflected to (d) and the carry flag (SM700) turns on.
SM8020	Zero	The input value is 0.	The value of (d) becomes 0 and the zero flag (SM8020) turns on.
SM8021	Borrow	The input value is ignored when it is less than 1.	The value of (d) becomes 0 and the borrow flag (SM8021) turns on.
SM8022	Carry	The input value is out of range.	The operation result is not reflected to (d) and the carry flag (SM8022) turns on.

Operation error

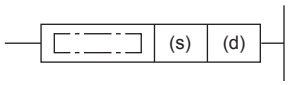
Error code (SD0/SD8067)	Description
3402H	When the contents of the specified device are outside the following range: 0, $2^{-126} \leq$ |specified value (stored value)| $< 2^{128}$
	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.

Converting 16-bit signed binary data to 16-bit unsigned binary data

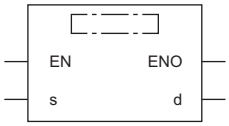
INT2UINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit signed binary data in the device specified by (s) to 16-bit unsigned binary data, and store the result in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=INT2UINT(EN,s,d); ENO:=INT2UINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

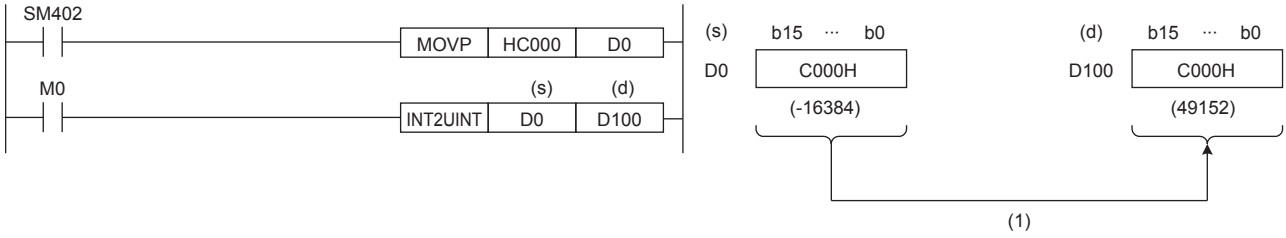
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	-32768 to +32767	16-bit signed binary	ANY16_S
(d)	Data after conversion	—	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions convert the 16-bit signed binary data in the device specified by (s) to 16-bit unsigned binary data, and store the result in the device specified by (d).



(1): The value is stored as is.

Operation error

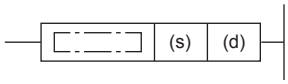
There is no operation error.

Converting 16-bit signed binary data to 32-bit signed binary data

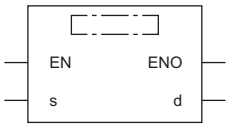
INT2DINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit signed binary data in the device specified by (s) to 32-bit signed binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=INT2DINT(EN,s,d); ENO:=INT2DINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

7

Setting data

■Descriptions, ranges, and data types

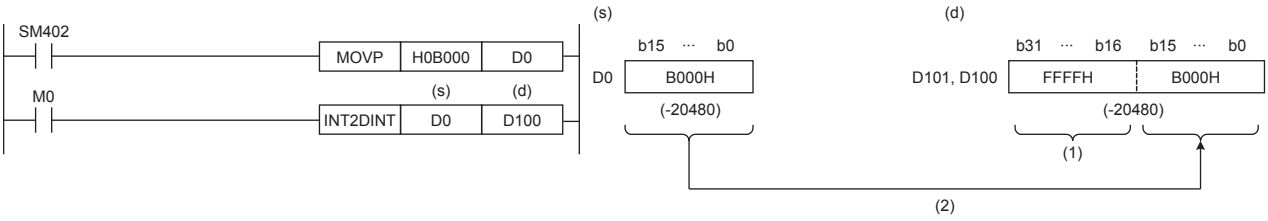
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	-32768 to +32767	16-bit signed binary	ANY16_S
(d)	Data after conversion	—	32-bit signed binary	ANY32_S
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions convert the 16-bit signed binary data in the device specified by (s) to 32-bit signed binary data, and store the converted data in the device specified by (d).



(1): The most significant bit of data before conversion is stored.
(2): Data before conversion is stored in the lower 16 bits.

Operation error

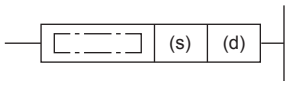
There is no operation error.

Converting 16-bit signed binary data to 32-bit unsigned binary data

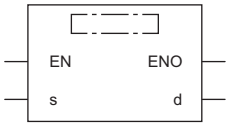
INT2UDINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit signed binary data in the device specified by (s) to 32-bit unsigned binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=INT2UDINT(EN,s,d); ENO:=INT2UDINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

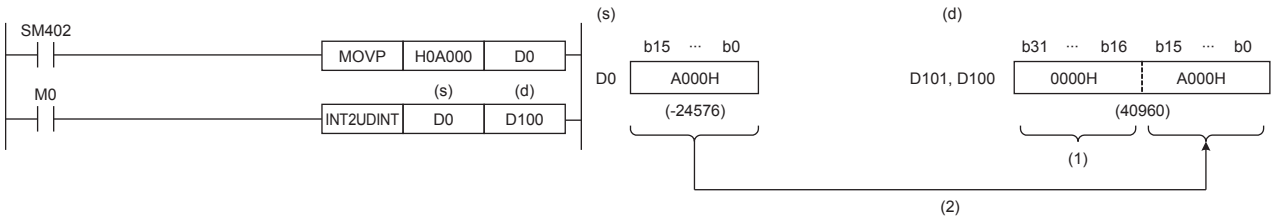
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	-32768 to +32767	16-bit signed binary	ANY16_S
(d)	Data after conversion	—	32-bit unsigned binary	ANY32_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions convert the 16-bit signed binary data in the device specified by (s) to 32-bit unsigned binary data, and store the converted data in the device specified by (d).



(1): The value 0 is stored.
(2): Data before conversion is stored in the lower 16 bits.

Operation error

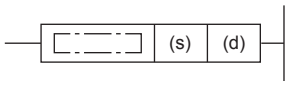
There is no operation error.

Converting 16-bit unsigned binary data to 16-bit signed binary data

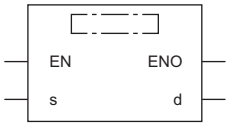
UINT2INT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit unsigned binary data in the device specified by (s) to 16-bit signed binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=UINT2INT(EN,s,d); ENO:=UINT2INTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

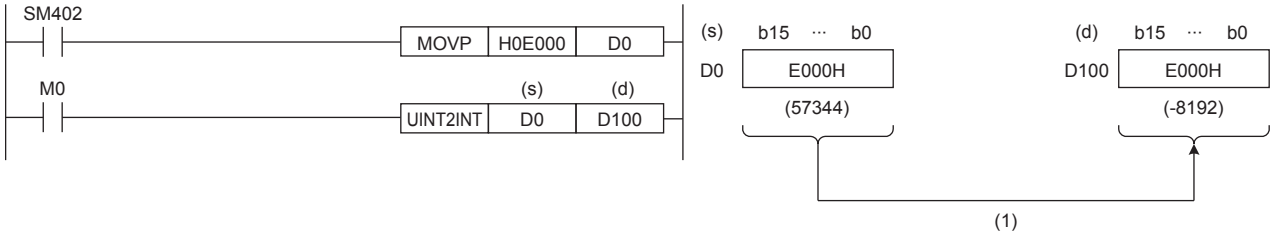
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	0 to 65535	16-bit unsigned binary	ANY16_U
(d)	Data after conversion	—	16-bit signed binary	ANY16_S
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions convert the 16-bit unsigned binary data in the device specified by (s) to 16-bit signed binary data, and store the converted data in the device specified by (d).



(1): The value is stored as is.

Operation error

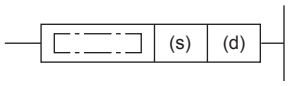
There is no operation error.

Converting 16-bit unsigned binary data to 32-bit signed binary data

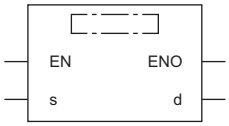
UINT2DINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit unsigned binary data in the device specified by (s) to 32-bit signed binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=UINT2DINT(EN,s,d); ENO:=UINT2DINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

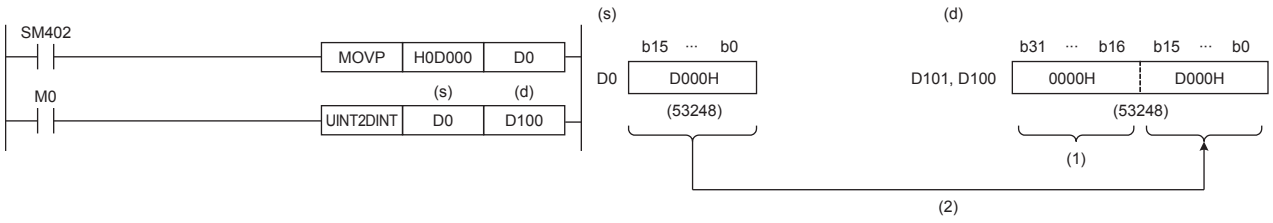
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	0 to 65535	16-bit unsigned binary	ANY16_U
(d)	Data after conversion	—	32-bit signed binary	ANY32_S
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions convert the 16-bit unsigned binary data in the device specified by (s) to 32-bit signed binary data, and store the converted data in the device specified by (d).



(1): The value 0 is stored.
(2): Data before conversion is stored in the lower 16 bits.

Operation error

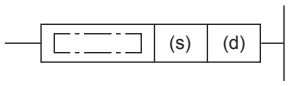
There is no operation error.

Converting 16-bit unsigned binary data to 32-bit unsigned binary data

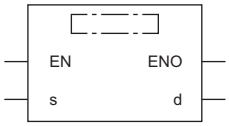
UINT2UDINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit unsigned binary data in the device specified by (s) to 32-bit unsigned binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=UINT2UDINT(EN,s,d); ENO:=UINT2UDINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

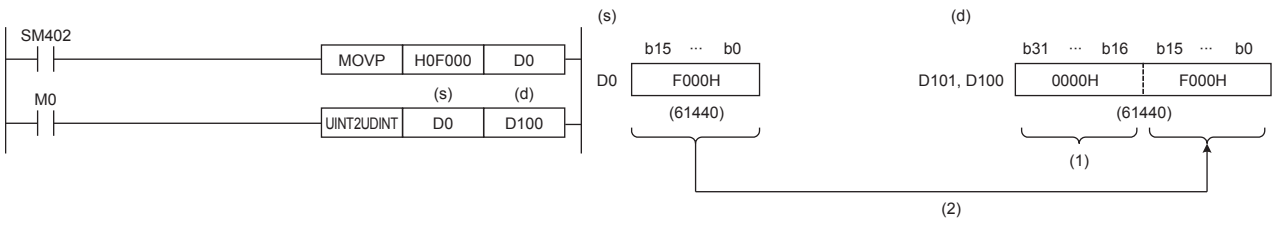
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	0 to 65535	16-bit unsigned binary	ANY16_U
(d)	Data after conversion	—	32-bit unsigned binary	ANY32_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions convert the 16-bit unsigned binary data in the device specified by (s) to 32-bit unsigned binary data, and store the converted data in the device specified by (d).



(1): The value 0 is stored.
(2): Data before conversion is stored in the lower 16 bits.

Operation error

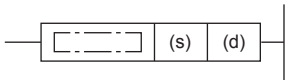
There is no operation error.

Converting 32-bit signed binary data to 16-bit signed binary data

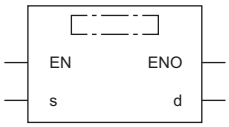
DINT2INT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit signed binary data in the device specified by (s) to 16-bit signed binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=DINT2INT(EN,s,d); ENO:=DINT2INTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

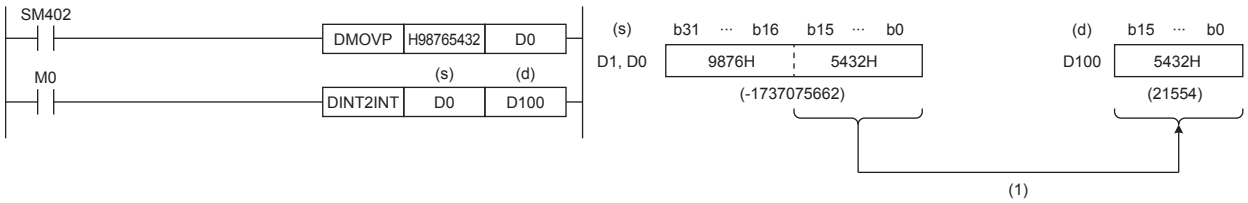
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
(d)	Data after conversion	—	16-bit signed binary	ANY16_S
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions convert the 32-bit signed binary data in the device specified by (s) to 16-bit signed binary data, and store the converted data in the device specified by (d).



(1): Data before conversion is stored in the lower 16 bits.

Operation error

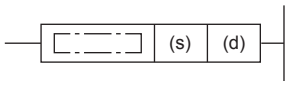
There is no operation error.

Converting 32-bit signed binary data to 16-bit unsigned binary data

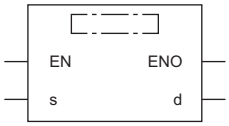
DINT2UINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit signed binary data in the device specified by (s) to 16-bit unsigned binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=DINT2UINT(EN,s,d); ENO:=DINT2UINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

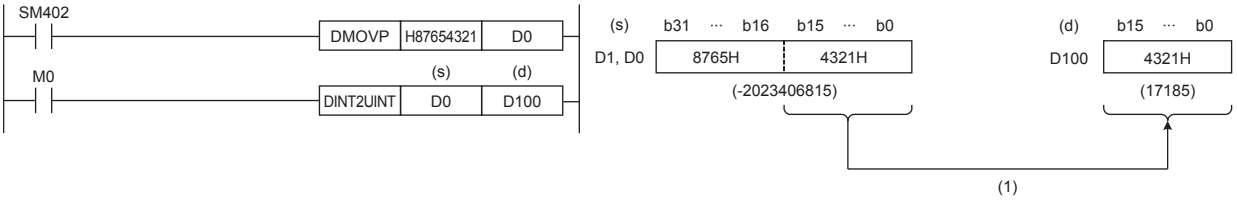
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
(d)	Data after conversion	—	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions convert the 32-bit signed binary data in the device specified by (s) to 16-bit unsigned binary data, and store the converted data in the device specified by (d).



(1): Data before conversion is stored in the lower 16 bits.

Operation error

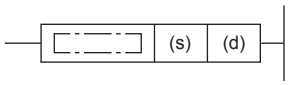
There is no operation error.

Converting 32-bit signed binary data to 32-bit unsigned binary data

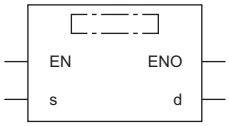
DINT2UDINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit signed binary data in the device specified by (s) to 32-bit unsigned binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=DINT2UDINT(EN,s,d); ENO:=DINT2UDINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

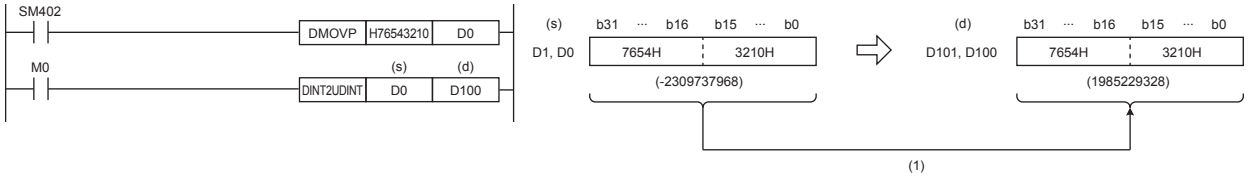
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
(d)	Data after conversion	—	32-bit unsigned binary	ANY32_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions convert the 32-bit signed binary data in the device specified by (s) to 32-bit unsigned binary data, and store the converted data in the device specified by (d).



(1): The value is stored as is.

Operation error

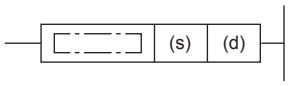
There is no operation error.

Converting 32-bit unsigned binary data to 16-bit signed binary data

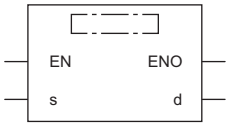
UDINT2INT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit unsigned binary data in the device specified by (s) to 16-bit signed binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=UDINT2INT(EN,s,d); ENO:=UDINT2INTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

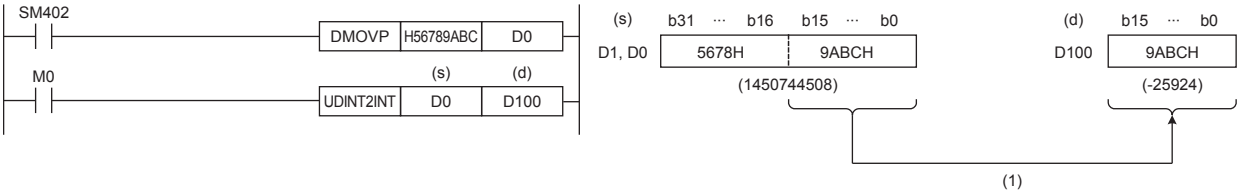
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	Data after conversion	—	16-bit signed binary	ANY16_S
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions convert the 32-bit unsigned binary data in the device specified by (s) to 16-bit signed binary data, and store the converted data in the device specified by (d).



(1): Data before conversion is stored in the lower 16 bits.

Operation error

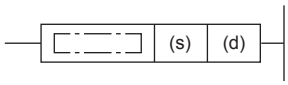
There is no operation error.

Converting 32-bit unsigned binary data to 16-bit unsigned binary data

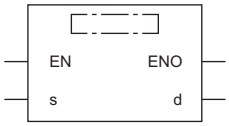
UDINT2UINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit unsigned binary data in the device specified by (s) to 16-bit unsigned binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=UDINT2UINT(EN,s,d); ENO:=UDINT2UINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

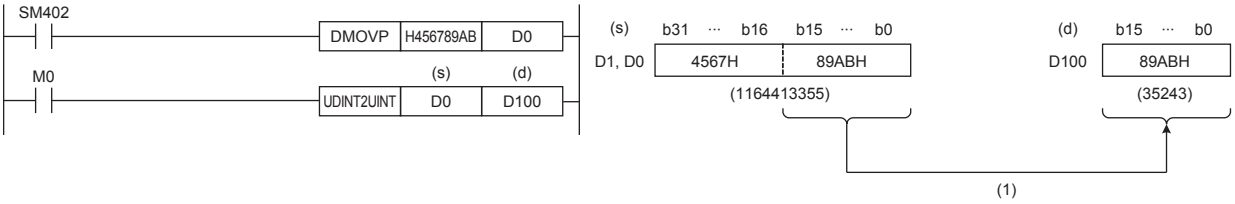
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	Data after conversion	—	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions convert the 32-bit unsigned binary data in the device specified by (s) to 16-bit unsigned binary data, and store the converted data in the device specified by (d).



(1): Data before conversion is stored in the lower 16 bits.

Operation error

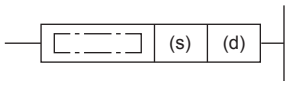
There is no operation error.

Converting 32-bit unsigned binary data to 32-bit signed binary data

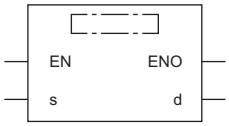
UDINT2DINT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit unsigned binary data in the device specified by (s) to 32-bit signed binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=UDINT2DINT(EN,s,d); ENO:=UDINT2DINTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

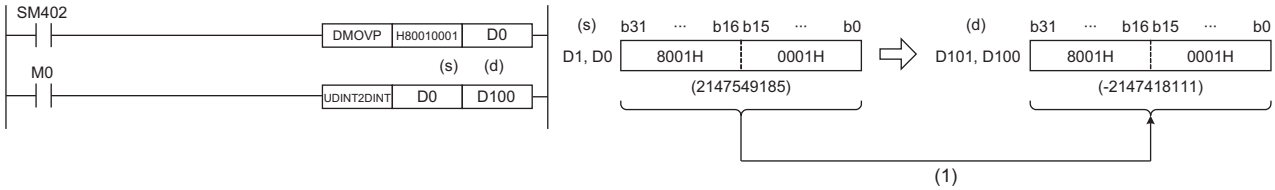
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	Data after conversion	—	32-bit signed binary	ANY32_S
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions convert the 32-bit unsigned binary data in the device specified by (s) to 32-bit signed binary data, and store the converted data in the device specified by (d).



(1): The value is stored as is.

Operation error

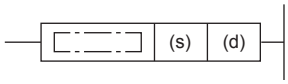
There is no operation error.

Converting 16-bit binary data to Gray code

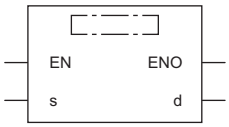
GRY(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit binary data in the device specified by (s) to 16-bit binary gray code data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=GRY(EN,s,d); ENO:=GRYP(EN,s,d);	ENO:=GRY_U(EN,s,d); ENO:=GRYP_U(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

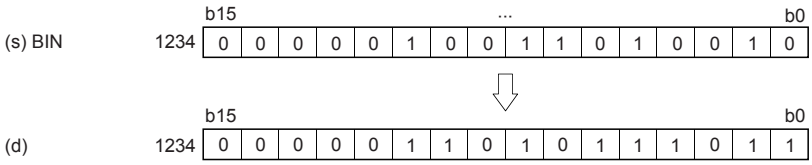
Operand		Description	Range	Data type	Data type (label)
(s)	GRY(P)	Binary data or the head device where the binary data is stored	0 to 32767	16-bit signed binary	ANY16_S
	GRY(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	GRY(P)	Head device for storing the gray code data after conversion	—	16-bit signed binary	ANY16_S
	GRY(P)_U			16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions convert the 16-bit binary data in the device specified by (s) to 16-bit binary gray code data, and store the converted data in the device specified by (d).



Precautions

The data conversion speed depends on the scan time of the CPU module.

Operation error

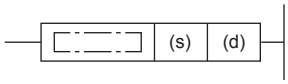
There is no operation error.

Converting 32-bit binary data to Gray code

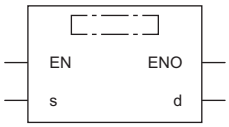
DGRY(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit binary data in the device specified by (s) to 32-bit binary gray code data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=DGRY(EN,s,d); ENO:=DGRYP(EN,s,d);	ENO:=DGRY_U(EN,s,d); ENO:=DGRYP_U(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

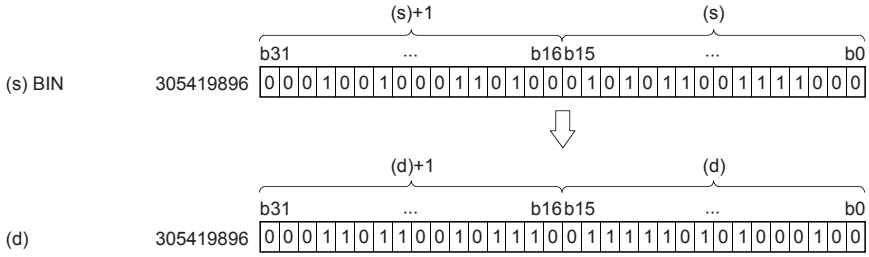
Operand		Description	Range	Data type	Data type (label)
(s)	DGRY(P)	Binary data or the head device where the binary data is stored	0 to 2147483647	32-bit signed binary	ANY32_S
	DGRY(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DGRY(P)	Head device for storing the gray code data after conversion	—	32-bit signed binary	ANY32_S
	DGRY(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions convert the 32-bit binary data in the device specified by (s) to 32-bit binary gray code data, and store the converted data in the device specified by (d).



(s)+1: Upper 16 bits
(s): Lower 16 bits

Precautions

The data conversion speed depends on the scan time of the CPU module.

Operation error

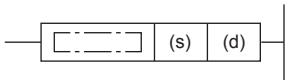
There is no operation error.

Converting Gray code to 16-bit binary data

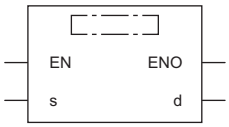
GBIN(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit binary gray code data in the device specified by (s) to 16-bit binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=GBIN(EN,s,d); ENO:=GBINP(EN,s,d);	ENO:=GBIN_U(EN,s,d); ENO:=GBINP_U(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

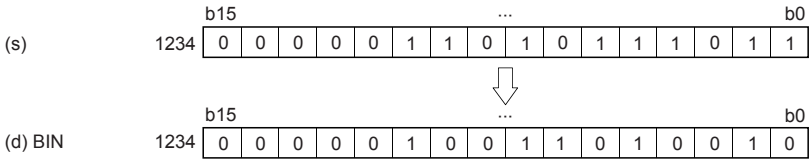
Operand		Description	Range	Data type	Data type (label)
(s)	GBIN(P)	Gray code data or head device storing the gray code data	0 to 32767	16-bit signed binary	ANY16_S
	GBIN(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	GBIN(P)	Head device for storing the binary data after conversion	—	16-bit signed binary	ANY16_S
	GBIN(P)_U			16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions convert the 16-bit binary gray code data in the device specified by (s) to 16-bit binary data, and store the converted data in the device specified by (d).



Precautions

When an input relay (X) is specified as (s), the response delay will be "Scan time of CPU module + Input filter constant".

Operation error

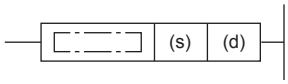
There is no operation error.

Converting Gray code to 32-bit binary data

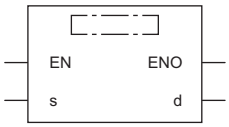
DGBIN(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit binary gray code data in the device specified by (s) to 32-bit binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=DGBIN(EN,s,d); ENO:=DGBINP(EN,s,d);	ENO:=DGBIN_U(EN,s,d); ENO:=DGBINP_U(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

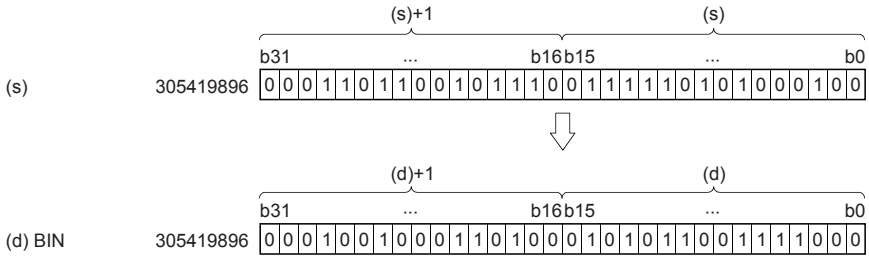
Operand		Description	Range	Data type	Data type (label)
(s)	DGBIN(P)	Gray code data or head device storing the gray code data	0 to 2147483647	32-bit signed binary	ANY32_S
	DGBIN(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DGBIN(P)	Head device for storing the binary data after conversion	—	32-bit signed binary	ANY32_S
	DGBIN(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions convert the 32-bit binary gray code data in the device specified by (s) to 32-bit binary data, and store the converted data in the device specified by (d).



(s)+1: Upper 16 bits
(s): Lower 16 bits

Precautions

When an input relay (X) is specified as (s), the response delay will be "Scan time of CPU module + Input filter constant".

Operation error

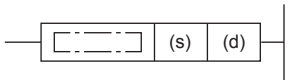
There is no operation error.

Converting decimal ASCII to 16-bit binary data

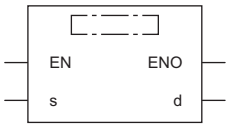
DABIN(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert the decimal ASCII data in the device areas specified by (s) and later to 16-bit binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=DABIN(EN,s,d); ENO:=DABINP(EN,s,d);	ENO:=DABIN_U(EN,s,d); ENO:=DABINP_U(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)		ASCII data or the head device where the ASCII data is stored	—	Character string	ANYSTRING_SINGLE
(d)	DABIN(P)	Head device for storing the converted data	—	16-bit signed binary	ANY16_S
	DABIN(P)_U			16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

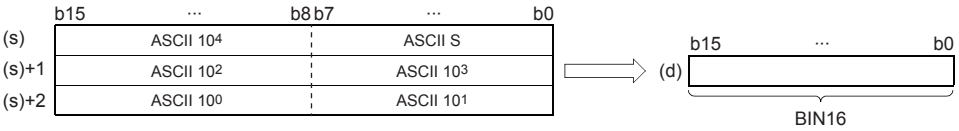
- These instructions convert the decimal ASCII data in the device areas specified by (s) and later to 16-bit binary data, and store the converted data in the device specified by (d).
- The setting method of the decimal ASCII data to be set in (s) depends on the status of SM705 (Number of conversion digits selection).

Status of SM705*1	Setting method of (s)	Reference
OFF	Set (s) with a fixed number of digits (a sign + 5 digits in the numeric part).	Page 353 Setting method of (s) for when SM705 (Number of conversion digits selection) is off
ON	Set (s) with a desired number of digits (maximum: a sign + 5 digits in the numeric part).	Page 354 Setting method of (s) for when SM705 (Number of conversion digits selection) is on

*1 For the firmware version of the CPU module supporting SM705, refer to the following.
MELSEC iQ-F FX5 User's Manual (Application)
A CPU module which does not support SM705 operates in the same way as SM705 is off even if it is turned on.

■Setting method of (s) for when SM705 (Number of conversion digits selection) is off

Set decimal ASCII data with the fixed number of digits in (s) to (s)+2.

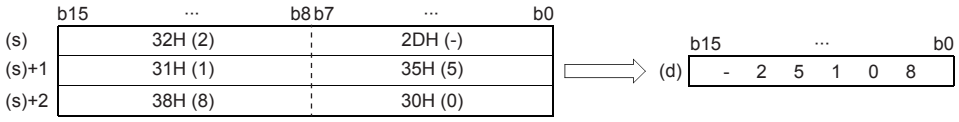


ASCII S: ASCII code for sign
ASCII 10^0: ASCII code for ones place
ASCII 10^1: ASCII code for the tens place
ASCII 10^2: ASCII code for the hundreds place
ASCII 10^3: ASCII code for the thousands place
ASCII 10^4: ASCII code for the ten-thousands place

- The ASCII data in the device specified by (s) to (s)+2 is within the range from -32768 to 32767 when signed data is specified, and it is within the range from 0 to 65535 when unsigned data is specified.
- The data of (s)+3 or later is ignored.
- As sign data, set 20H (space) when the ASCII data is positive, and set 2DH (-) when the data is negative. (If a value other than 20H and 2DH is set, the data will be processed as positive data.)
- A value from 30H to 39H can be set in each place of ASCII code.
- If a value 20H or 00H is set in each place of ASCII code, the value will be processed as 30H.

Ex.

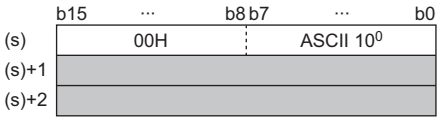
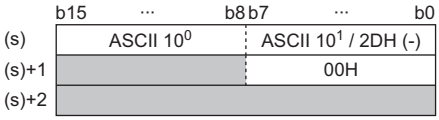
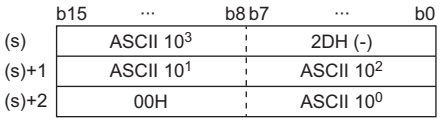
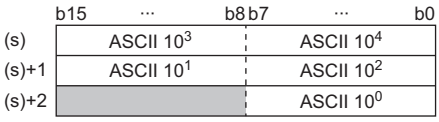
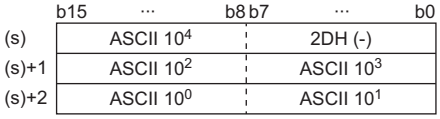
When the ASCII data, -25108 (signed), is specified by (s)



■Setting method of (s) for when SM705 (Number of conversion digits selection) is on

Set decimal ASCII data with a desired number of digits (including 00H (NULL code)) in (s). Note that 00H (NULL code) is not required to be set if the integral part has the maximum number of digits (5 digits).

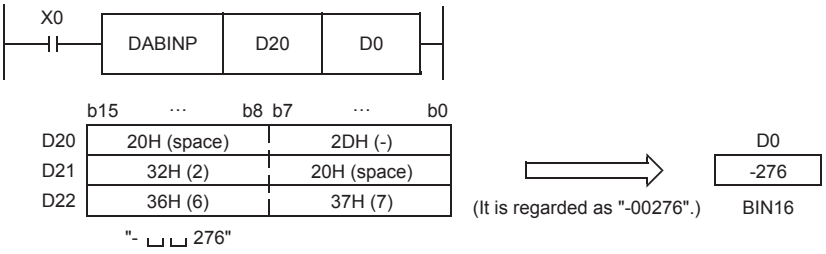
The following table lists the setting method of (s).

<table>
<tr><th>Value to be set in (s)</th><th>Data of (s) to (s)+2</th><th>Value to be set in (s)</th><th>Data of (s) to (s)+2</th></tr>
<tr><td>• 0
• Positive value (1 digit in numeric part)</td><td>• Set 00H in the upper byte of (s).
• The data of (s)+1 or later is ignored.
  </td><td>• Positive value (2 digits in numeric part)
• Negative value (1 digit in numeric part)</td><td>• Set 00H in the lower byte of (s)+1.
• The data of the upper byte of (s)+1 or later is ignored.
  </td></tr>
<tr><td colspan="4">⋮</td></tr>
<tr><td>• Negative value (4 digits in numeric part)</td><td>• Set 00H in the upper byte of (s)+2.
  </td><td>• Positive value (5 digits in numeric part)</td><td>• The data of the upper byte of (s)+2 or later is ignored. Since the number of digits is the maximum, 00H is not required to be set.
  </td></tr>
<tr><td>• Negative value (5 digits in numeric part)</td><td>• The data of (s)+3 or later is ignored. Since the number of digits is the maximum, 00H is not required to be set.
  </td><td colspan="2">ASCII 10^0: ASCII code for ones place
ASCII 10^1: ASCII code for tens place
⋮
ASCII 10^4: ASCII code for the ten-thousands place</td></tr>
</table>

- The ASCII data in the device specified by (s) to (s)+2 is within the range from -32768 to 32767 when signed data is specified, and it is within the range from 0 to 65535 when unsigned data is specified.
- Set 2DH (-) to lower byte of (s)+0 as sign data when the ASCII data is negative. Set an ASCII code of the uppermost digit instead of setting sign data when the ASCII data is 0 or positive.
- A value from 30H to 39H can be set in each place of ASCII code.
- If the value is positive and the numeric part has 5 digits, the data of the upper byte of (s)+2 or later is ignored. If the value is negative and the numeric part has 5 digits, the data of (s)+3 or later is ignored.
- If a value 20H is set in each place of ASCII code, the value is processed as 30H. If a value 00H is set, the value is processed as the end of the decimal ASCII data.
- In the following cases, 0 is stored.
  - The first character is 00H (NULL).
  - The first character is 2DH (-) and the second character is 00H (NULL).

Program example

In the program below, the signed decimal ASCII codes in five digits stored in D20 to D22 are converted into 16-bit binary data and stored in D0 when X0 is set to ON.



Operation error

Error code (SD0/SD8067)	Description
2820H	The device specified by (s) exceeds the corresponding device range.
3401H	The sign data is other than 20H, 2DH.*1
	A value specified by (s) to (s)+2 for each place of the ASCII code is other than "30H" to "39H", "20H", and "00H".
	The ASCII data in the device specified by (s) to (s)+2 is out of the valid range (-32768 to +32767) (when a signed data is specified).
	The ASCII data in the device specified by (s) to (s)+2 is out of the valid range (0 to 65535) (when unsigned data is specified).

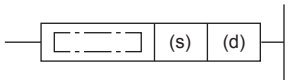
*1 This error occurs when SM705 is on. When SM705 is off, sign data other than "2DH (-)" is ignored.

Converting decimal ASCII to 32-bit binary data

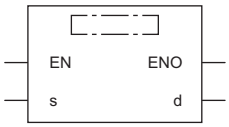
DDABIN(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert the decimal ASCII data in the device numbers specified by (s) and later to 32-bit binary data, and store the converted data in the device specified by (d).

Ladder diagram	Structured text	
	ENO:=DDABIN(EN,s,d); ENO:=DDABINP(EN,s,d);	ENO:=DDABIN_U(EN,s,d); ENO:=DDABINP_U(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)		ASCII data or the head device where the ASCII data is stored	—	Character string	ANYSTRING_SINGLE
(d)	DDABIN(P)	Head device for storing the converted data	—	32-bit signed binary	ANY32_S
	DDABIN(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	○	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

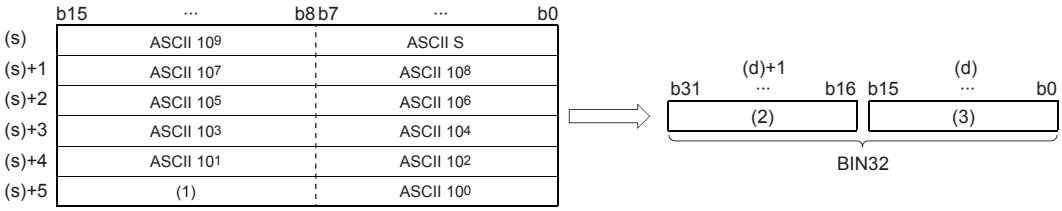
- These instructions convert the decimal ASCII data in the device numbers specified by (s) and later to 32-bit binary data, and store the converted data in the device specified by (d).
- The setting method of the decimal ASCII data to be set in (s) depends on the status of SM705 (Number of conversion digits selection).

Status of SM705*1	Setting method of (s)	Reference
OFF	Set (s) with a fixed number of digits (a sign + 10 digits in the numeric part).	Page 357 Setting method of (s) for when SM705 (Number of conversion digits selection) is off
ON	Set (s) with a desired number of digits (maximum: a sign + 10 digits in the numeric part).	Page 358 Setting method of (s) for when SM705 (Number of conversion digits selection) is on

*1 For the firmware version of the CPU module supporting SM705, refer to the following.
MELSEC iQ-F FX5 User's Manual (Application)
A CPU module which does not support SM705 operates in the same way as SM705 is off even if it is turned on.

■Setting method of (s) for when SM705 (Number of conversion digits selection) is off

Set decimal ASCII data with the fixed number of digits in (s) to (s)+5.

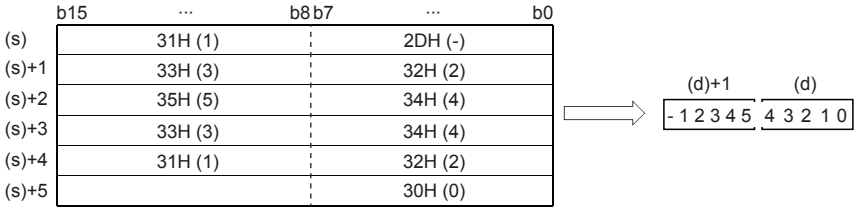


ASCII S: ASCII code for sign
ASCII 10^0: ASCII code for ones place
ASCII 10^1: ASCII code for the tens place
ASCII 10^2: ASCII code for the hundreds place
ASCII 10^3: ASCII code for the thousands place
ASCII 10^4: ASCII code for the ten-thousands place
ASCII 10^5: ASCII code for the hundred-thousands place
ASCII 10^6: ASCII code for the millions place
ASCII 10^7: ASCII code for the ten-millions place
ASCII 10^8: ASCII code for the hundred-millions place
ASCII 10^9: ASCII code for the billions place
(1): Ignore
(2): Upper 16 bits
(3): Lower 16 bits

- The ASCII data in the device specified by (s) to (s)+5 is within the range from -2147483648 to +2147483647 when signed data is specified, and it is within the range from 0 to 4294967295 when unsigned data is specified. Any data stored in the upper byte of the device specified by (s)+5 is ignored.
- As signed data, set 20H if the ASCII data is positive, and set 2DH if the data is negative. (If a value other than 20H and 2DH is set, the data will be processed as positive data.)
- A value from 30H to 39H can be set in each place of ASCII code.
- If a value 20H or 00H is set in each place of ASCII code, the value will be processed as 30H.

Ex.

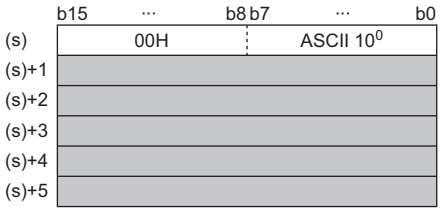
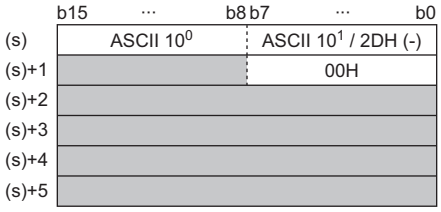
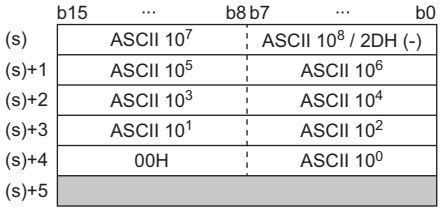
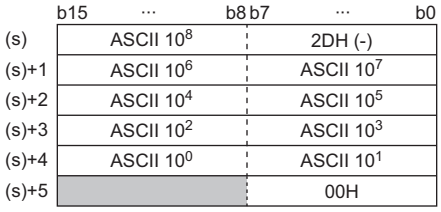
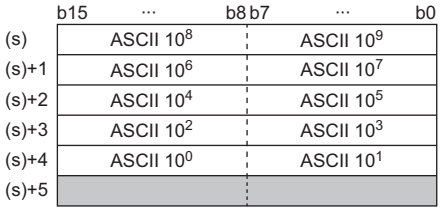
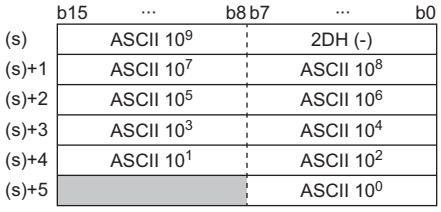
When the ASCII data, -1234543210 (signed), is specified by (s)



■Setting method of (s) for when SM705 (Number of conversion digits selection) is on

Set decimal ASCII data with a desired number of digits (including 00H (NULL code)) in (s). Note that 00H (NULL code) is not required to be set if the integral part has the maximum number of digits (10 digits).

The following table lists the setting method of (s).

<table>
<tr><th>Value to be set in (s)</th><th>Data of (s) to (s)+5</th><th>Value to be set in (s)</th><th>Data of (s) to (s)+5</th></tr>
<tr><td>• 0
• Positive value (1 digit in numeric part)</td><td>• Set 00H in the upper byte of (s).
• The data of (s)+1 or later is ignored.
  </td><td>• Positive value (2 digits in numeric part)
• Negative value (1 digit in numeric part)</td><td>• Set 00H in the lower byte of (s)+1.
• The data of the upper byte of (s)+1 or later is ignored.
  </td></tr>
<tr><td colspan="4">⋮</td></tr>
<tr><td>• Positive value (9 digits in numeric part)
• Negative value (8 digits in numeric part)</td><td>• Set 00H in the upper byte of (s)+4.
• (s)+5 is ignored.
  </td><td>• Negative value (9 digits in numeric part)</td><td>• Set 00H in the lower byte of (s)+5.
• The data of the upper byte of (s)+5 is ignored.
  </td></tr>
<tr><td>• Positive value (10 digits in numeric part)</td><td>• (s)+5 is ignored. Since the number of digits is the maximum, 00H is not required to be set.
  </td><td>• Negative value (10 digits in numeric part)</td><td>• The data of the upper byte of (s)+5 is ignored. Since the number of digits is the maximum, 00H is not required to be set.
  </td></tr>
</table>

ASCII 10^0: ASCII code for the ones place
ASCII 10^1: ASCII code for the tens place
⋮
ASCII 10^9: ASCII code for the billions place

- The ASCII data in the device specified by (s) to (s)+5 is within the range from -2147483648 to 2147483647 when signed data is specified, and it is within the range from 0 to 4294967295 when unsigned data is specified.
- Set 2DH (-) to lower byte of (s)+0 as sign data when the ASCII data is negative. Set an ASCII code of the uppermost digit instead of setting sign data when the ASCII data is 0 or positive.
- A value from 30H to 39H can be set in each place of ASCII code.
- If the value is positive and the numeric part has 10 digits, the data stored in (s)+5 or later is ignored. If the value is negative and the numeric part has 10 digits, the data stored in the upper byte of (s)+5 or later is ignored.
- If a value 20H is set in each place of ASCII code, the value is processed as 30H. If a value 00H is set, the value is processed as the end of the decimal ASCII data.
- In the following cases, "0" is stored in (d).
  - The first character is 00H (NULL).
  - The first character is 2DH (-) and the second character is 00H (NULL).

Operation error

Error code (SD0/SD8067)	Description
2820H	The device specified by (s) exceeds the corresponding device range.
3401H	The sign data is other than 20H, 2DH.*1
	A value specified by (s) to (s)+2 for each place of the ASCII code is other than "30H" to "39H", "20H", and "00H".
	The ASCII data in the device specified by (s) to (s)+5 is out of the valid range (-2147483648 to +2147483647) (when a signed data is specified).
	The ASCII data in the device specified by (s) to (s)+5 is out of the valid range (0 to 4294967295) (when unsigned data is specified).

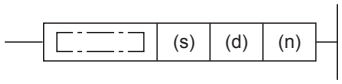
*1 This error occurs when SM705 is on. When SM705 is off, sign data other than "2DH (-)" is ignored.

Converting ASCII to HEX

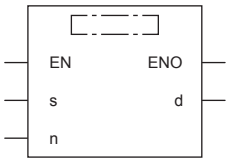
HEXA(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the ASCII data stored in the number of characters specified by (n) in the device numbers specified by (s) and later to HEX code data, and store the converted data in the device numbers specified by (d) and later.

Ladder diagram	Structured text
	ENO:=HEXA(EN,s,n,d); ENO:=HEXAP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device for storing the ASCII data to be converted to hexadecimal code	—	Character string	ANYSTRING_SINGLE
(d)	Head device for storing the hexadecimal code after conversion	—	16-bit signed binary	ANY16
(n)	Number of characters (number of bytes) of ASCII data to be converted	1 to 16383	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○*1	○	—	—	—	○	—	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

These instructions convert the ASCII data stored in the number of characters specified by (n) in the device numbers specified by (s) and later to HEX code data, and store the converted data in the device numbers specified by (d) and later. 16-bit conversion mode and 8-bit conversion mode options are available for these instructions. For operation in each mode, refer to the succeeding pages.

• 16-bit conversion mode (while SM8161 is OFF)

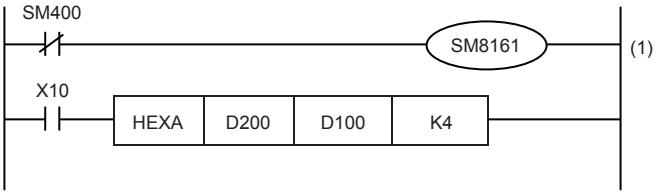
The ASCII data stored in high-order 8 bits and low-order 8 bits (byte) of the device specified by (s) is converted to hexadecimal code, and transferred to the device specified by (d) in units of 4 digits. The number of characters to be converted is specified by (n).

SM8161 is also used for the ASCI(P), CRC(P), and CCD(P) instructions. When using the 16-bit conversion mode, set SM8161 to normally OFF.

SM8161 is cleared when the CPU module mode is changed from RUN to STOP.

Moreover, when using the 16-bit conversion mode, the ASCII data must also be stored in high-order 8 bits of the device specified by (s).

In the following program, conversion is executed as follows:



(1): 16-bit conversion mode

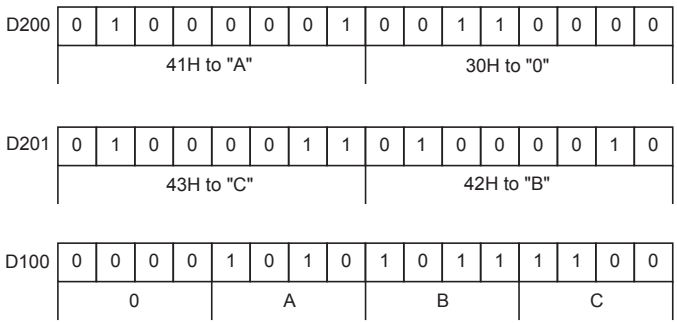
Conversion source data

(s)	ASCII data	Hexadecimal code
D200 low-order	30H	0
D200 high-order	41H	A
D201 low-order	42H	B
D201 high-order	43H	C
D202 low-order	31H	1
D202 high-order	32H	2
D203 low-order	33H	3
D203 high-order	34H	4
D204 low-order	35H	5

Number of specified characters and conversion result "·" indicates "0".

When (n)=K4

(n)	(d)		
	D102	D101	D100
1	(1)		··· 0H
2			·· 0AH
3			· 0ABH
4			0ABCH
5		··· 0H	ABC1H
6		·· 0AH	BC12H
7		· 0ABH	C123H
8		0ABCH	1234H
9	··· 0H	ABC1H	2345H



(1): The data remain the same.

• 8-bit conversion mode (while SM8161 is on)

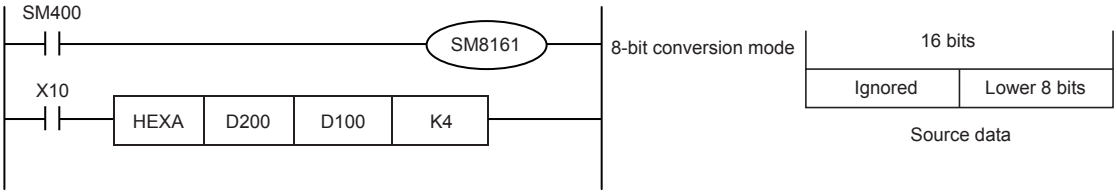
The ASCII data stored in low-order 8 bits of the device specified by (s) is converted to hexadecimal code, and transferred to the device specified by (d) in units of 4 digits.

The number of characters to be converted is specified by (n).

SM8161 is also used for the ASCI(P), CRC(P), and CCD(P) instructions. When using the 8-bit conversion mode, set SM8161 to normally on.

SM8161 is cleared when the CPU module mode is changed from RUN to STOP.

In the following program, conversion is executed as follows:



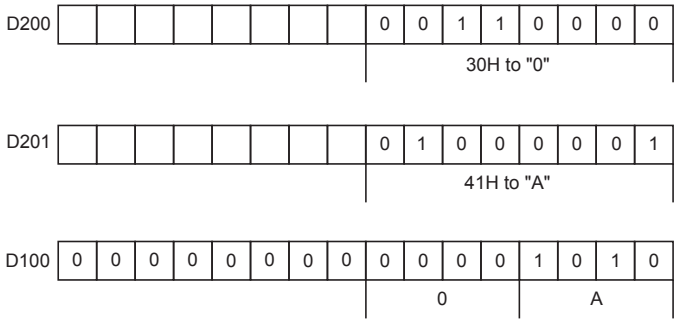
Conversion source data

(s)	ASCII data	Hexadecimal code
D200	30H	0
D201	41H	A
D202	42H	B
D203	43H	C
D204	31H	1
D205	32H	2
D206	33H	3
D207	34H	4
D208	35H	5

Number of specified characters and conversion result "." indicates "0".

When (n)=K2

(n)	(d)		
	D102	D101	D100
1	(1)		··· 0H
2			·· 0AH
3			· 0ABH
4			0ABCH
5		··· 0H	ABC1H
6		·· 0AH	BC12H
7		· 0ABH	C123H
8		0ABCH	1234H
9	···0H	ABC1H	2345H



(1): The data remain the same.

Precautions

- Make sure that only ASCII codes "0" to "9" and "A" to "F" are stored in the device specified by (s).
- If ASCII data is not stored in the device specified for (s) by the HEXA(P) instructions, an operation error occurs and conversion into hexadecimal code is disabled. Especially, note that when SM8161 is OFF (16-bit conversion mode), ASCII code should be stored in high-order 8 bits of the device specified by (s).
- The number of points occupied by the device specified by (d) varies depending on the ON/OFF status of SM8161. When SM8161 is on (8-bit conversion mode), as many points as the number of characters are occupied, and when SM8161 is OFF (16-bit conversion mode) as many points as the (number of characters ÷2) are occupied.
- The SM8161 flag is also used for the ASCI(P), CRC(P), and CCD(P) instructions. When using these instructions and the HEXA(P) instructions in the same program, make sure to set SM8161 to ON or OFF just before each instruction so that SM8161 does not affect another instruction.

Operation error

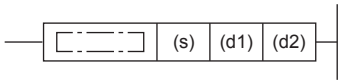
Error code (SD0/SD8067)	Description
2820H	The (n) number of devices specified by (s) and (d) exceeds the corresponding device range.
2821H	The range specified by (s) and (d) overlaps.
3401H	An ASCII code other than 30H to 39H, and 41H to 46H is set in the device specified by (s).
3405H	The value specified in (n) is outside the range specified below. 1 to 16383

Converting character string to 16-bit binary data

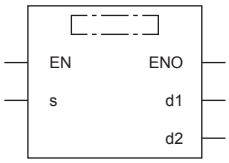
VAL(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert the character string in the device numbers specified by (s) and later to 16-bit binary data, and store the number of digits in the device specified by (d1) and the binary data in the device specified by (d2).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)		Character string to be converted to binary data, or head device for storing the character string.	—	Character string	ANYSTRING_SINGLE
(d1)	VAL(P)	Head device for storing the number of digits of the binary data after conversion	—	16-bit signed binary	ANY16_S_ARRAY (Number of elements: 2)
	VAL(P)_U			16-bit unsigned binary	ANY16_U_ARRAY (Number of elements: 2)
(d2)	VAL(P)	Head device for storing the binary data after conversion	—	16-bit signed binary	ANY16_S
	VAL(P)_U			16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

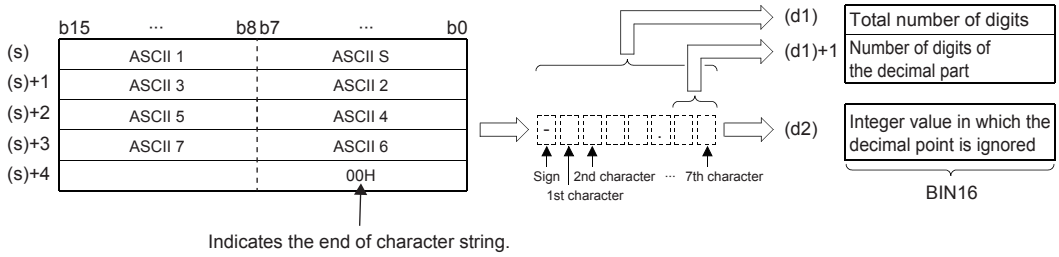
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	○	—
(d1)	○	○	—	—	—	—	○	—	—	—	—
(d2)	○	○	○	○	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

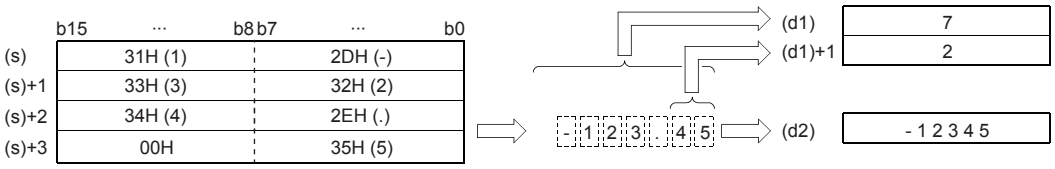
Processing details

- These instructions convert the character string in the device numbers specified by (s) and later to 16-bit binary data, and store the number of digits in the device specified by (d1) and the binary data in the device specified by (d2). When converting a character string into binary data, the data from the device number specified by (s) to a device number storing "00H" is handled as a character string.
- The total number of digits stored in (d1) is the total number of characters (including the sign and decimal point) representing the numeric value. The number of digits in the decimal part stored in (d1)+1 is the number of characters representing the decimal part after 2EH (.). The 16-bit binary data stored in (d2) is binary value converted from a character string with the decimal point ignored.



Ex.

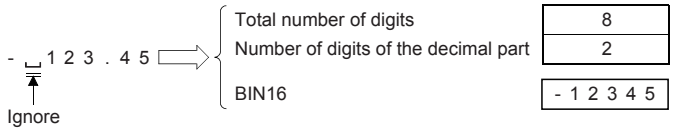
When the character string "-123.45" (signed) is specified by (s) and later



- The total number of characters of the character string specified by (s) is 2 to 8 characters.
- In the character string specified by (s), the number of characters that form the decimal part is 0 to 5 characters. However, be sure to specify "Total number of digits - 3" or below.
- The range of the character string of the numeric value that can be converted to a binary value is -32768 to +32767 for a signed value with the decimal point ignored, and 0 to 65535 for an unsigned value. A character string of a numeric value excluding the sign and decimal point can be specified only within the range of 30H to 39H. (Value with the decimal point ignored ... "-12345.6" becomes "-123456".)
- When representing a positive numeric value, 20H is set in the sign, and when representing a negative numeric value, 2DH is set.
- 2EH is set in the decimal point.
- When "20H (space)" or "30H (0)" exists between the sign and the first non-zero number in a character string specified by (s), "20H" or "30H" is ignored during conversion to a binary value.

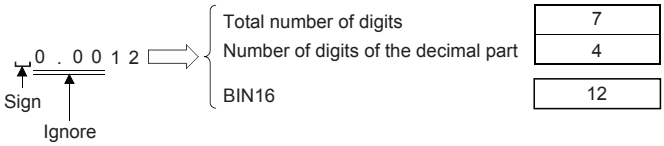
Ex.

When "20H" exists between the sign and the first non-zero number (a signed value is specified)



Ex.

When "30H" exists between the sign and the first non-zero number



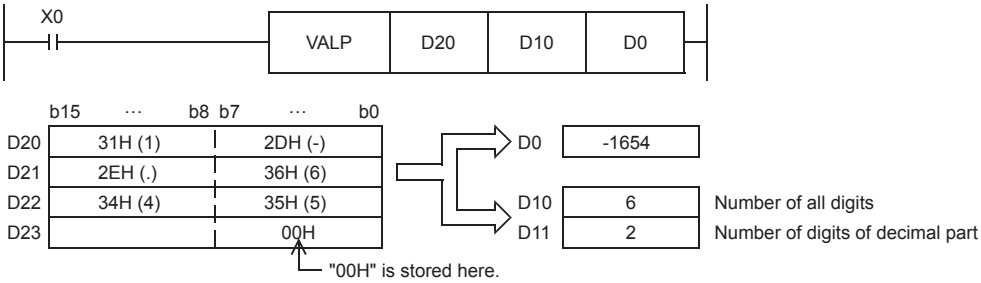
- Do not set 2EH(.) to the last character of (s). (Example: 1234.)

Precautions

- Store signed data, "space (20H)" or "- (2DH)" only in the 1st byte (low-order 8 bits of the head device set in (s)). Only the ASCII data "0 (30H)" to "9 (39H)", "space (20H)" and "decimal point (2EH)" can be stored from the 2nd byte to the "00H" at the end of the character string in (s). If "- (2DH)" is stored in the 2nd byte or later, an operation error occurs.

Program example

In the program below, the character string data stored in D20 to D22 is regarded as an integer value. This value is converted into a binary value, and stored in D0 when X0 is set to ON.



Operation error

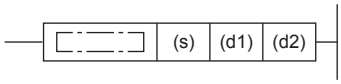
Error code (SD0/SD8067)	Description
2820H	The device specified by (d1) exceeds the corresponding device range.
	When "00H" is not set in the corresponding device range after the device specified in (s).
3401H	The number of characters of the character string specified by (s) is other than 2 to 8 characters.
	The number of characters of the decimal part of the character string specified by (s) is other than 0 to 5 characters.
	The relationship between the total number of characters specified by (s) and the number of characters of the decimal part is other than that described below. Total number of characters - 3≥Number of characters in the decimal part
	When the VAL(P) instruction is used, an ASCII code other than "20H" and "2DH" is set as a sign.
	When the VAL(P)_U instruction is used, an ASCII code other than "20H" is set as a sign.
	An ASCII code other than 30H to 39H, and 2EH (decimal point) is set in the digits of each number
	Two or more decimal points are set.
	The converted binary value exceeds the range that can be converted by each instruction. Signed operation: -32768 to +32767, unsigned operation: 0 to 65535

Converting character string to 32-bit binary data

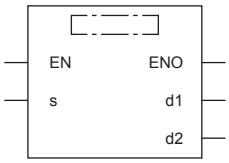
DVAL(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert the character string in the device numbers specified by (s) and later to 32-bit binary data, and store the number of digits in the device specified by (d1) and the binary data in the device specified by (d2).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)		Character string to be converted to binary data, or head device for storing the character string.	—	Character string	ANYSTRING_SINGLE
(d1)	DVAL(P)	Head device for storing the number of digits of the binary data after conversion	—	16-bit signed binary	ANY16_S_ARRAY (Number of elements: 2)
	DVAL(P)_U			16-bit unsigned binary	ANY16_U_ARRAY (Number of elements: 2)
(d2)	DVAL(P)	Head device for storing the binary data after conversion	—	32-bit signed binary	ANY32_S
	DVAL(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

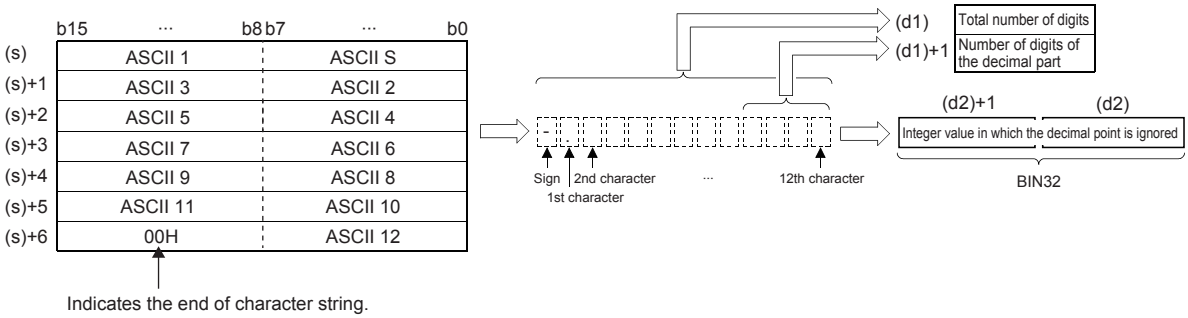
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	○	—
(d1)	○	○	—	—	—	—	○	—	—	—	—
(d2)	○	○	○	○	○	○	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

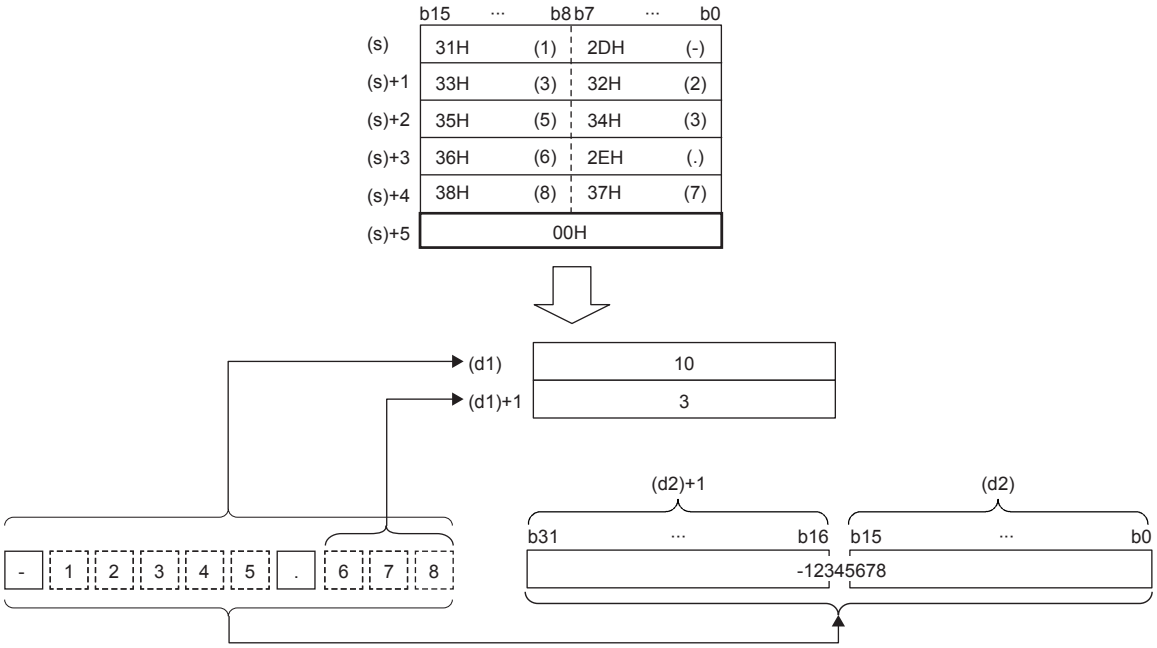
- These instructions convert the character string in the device numbers specified by (s) and later to 32-bit binary data, and store the number of digits in the device specified by (d1) and the binary data in the device specified by (d2). When converting a character string into binary data, the data from the device number specified by (s) to a device number storing "00H" is handled as a character string.

- The total number of digits stored in (d1) is the total number of characters (including the sign and decimal point) representing the numeric value. The number of digits in the decimal part stored in (d1)+1 is the number of characters representing the decimal part after 2EH (.). The 32-bit binary data stored in (d2) is binary value converted from a character string with the decimal point ignored.



Ex.

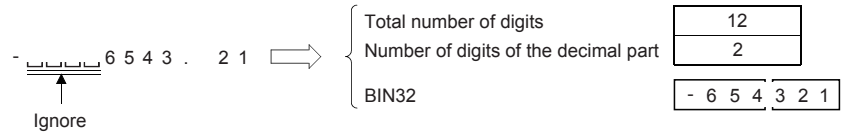
When the character string "-12345.678" (signed) is specified by (s) and later



- The total number of characters of the character string specified by (s) is 2 to 13 characters.
- In the character string specified by (s), the number of characters that form the decimal part is 0 to 10 characters. However, be sure to specify "Total number of digits - 3" or below.
- The range of the character string of the numeric value that can be converted to a binary value is -2147483648 to 2147483647 for a signed value with the decimal point ignored, and 0 to 4294967295 for an unsigned value. A character string of a numeric value excluding the sign and decimal point can be specified only within the range of 30H to 39H. (Value with the decimal point ignored ... "-12345.6" becomes "-123456".)
- When representing a positive numeric value, 20H is set in the sign, and when representing a negative numeric value, 2DH is set.
- Set 2EH in the decimal point.
- The total number of digits stored in (d1) includes all characters (including signs and decimal points) that represent a numerical value. The number of digits in the decimal part stored in (d1)+1 is the number of characters representing the decimal part after 2EH (.). The binary data to be stored in (d2) is the binary value converted from the character string with the decimal point ignored.
- When "20H (space)" or "30H (0)" exists between the sign and the first non-zero number in a character string specified by (s), "20H" or "30H" is ignored during conversion to a binary value.

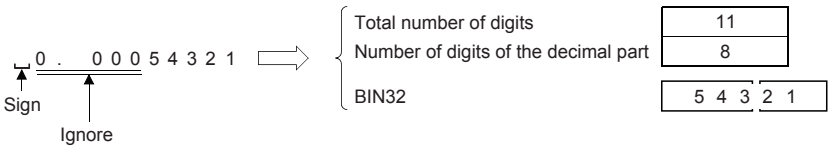
Ex.

When "20H" exists between the sign and the first non-zero number (a signed value is specified)



Ex.

When "30H" exists between the sign and the first non-zero number



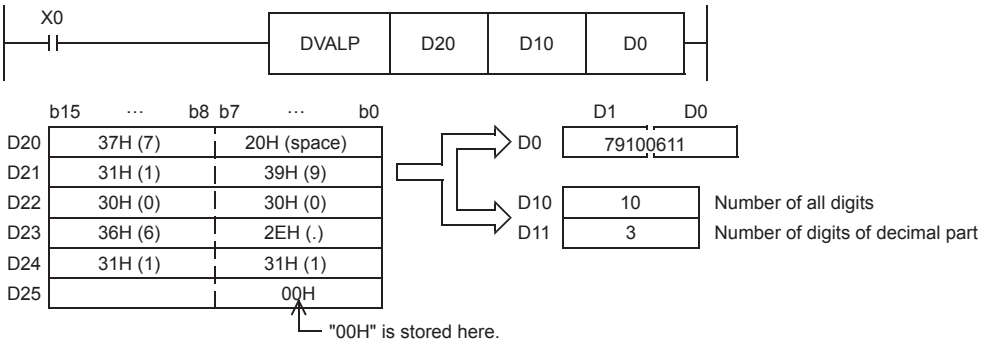
- Do not set 2EH(.) to the last character of (s). (Example: 1234.)

Precautions

- Store sign data, "space (20H)" or "- (2DH)" in the 1st byte (low-order 8 bits of the head device set in (s)). Only the ASCII data "0 (30H)" to "9 (39H)", "space (20H)" and "decimal point (2EH)" can be stored from the 2nd byte to the "00H" at the end of the character string in (s). If "- (2DH)" is stored in the 2nd byte or later, an operation error occurs.

Program example

In the program below, the character string data stored in D20 to D24 is regarded as an integer value. This value is converted into 32-bit binary data, and stored in D0 and D1 when X0 is set to ON.



Operation error

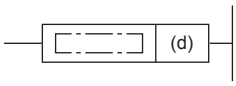
Error code (SD0/SD8067)	Description
2820H	The device specified by (d1) exceeds the corresponding device range.
	When "00H" is not set in the corresponding device range after the device specified in (s).
3401H	The number of characters of the character string specified by (s) is other than 2 to 13 characters.
	The number of characters of the decimal part of the character string specified by (s) is other than 0 to 10 characters.
	The relationship between the total number of characters specified by (s) and the number of characters of the decimal part is other than that described below. Total number of characters - 3≥Number of characters in the decimal part
	When the DVAL(P) instruction is used, an ASCII code other than "20H" and "2DH" is set as a sign.
	When the DVAL(P)_U instruction is used, an ASCII code other than "20H" is set as a sign.
	An ASCII code other than 30H to 39H, and 2EH (decimal point) is set in the digits of each number.
	Two or more decimal points are set.
	The converted binary value exceeds the range that can be converted by each instruction. Signed operation: -2147483648 to +2147483647, unsigned operation: 0 to 4294967295
	2EH(.) is set to the last character of (s).

Two's complement of 16-bit binary data (sign inversion)

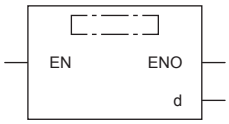
NEG(P)

FX5S FX5UJ FX5U FX5UC

These instructions invert the sign of the 16-bit binary data in the device specified by (d), and store the resultant data in the device specified by (d).

Ladder diagram	Structured text
	ENO:=NEG(EN,d); ENO:=NEGP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

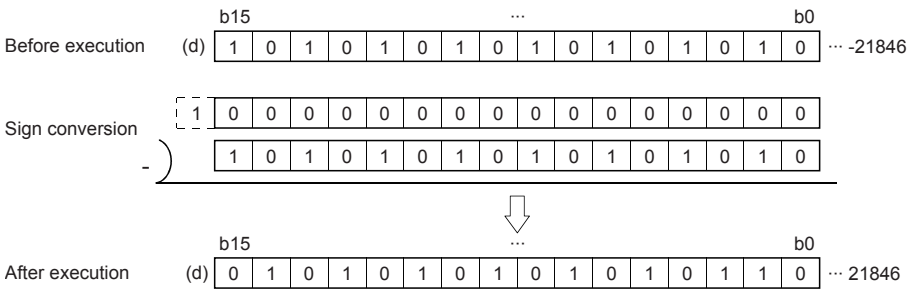
Operand	Description	Range	Data type	Data type (label)
(d)	Head device for storing the data that performs two's complement	-32768 to +32767	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions invert the sign of the 16-bit binary data in the device specified by (d), and store the resultant data in the device specified by (d).
- They are used when a positive or negative sign is to be inverted.



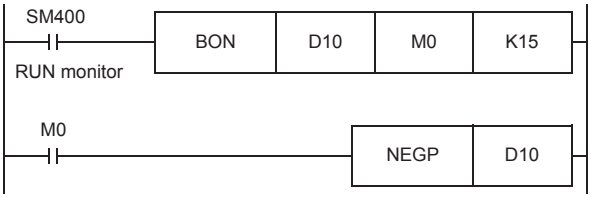
Precautions

Note that data is inverted in every operation cycle in a continuous operation type (NEG) instruction.

Program example

The program examples below are provided to obtain the absolute value of a negative binary 16-bit data.

- Obtaining the absolute value of a negative value using NEG instruction



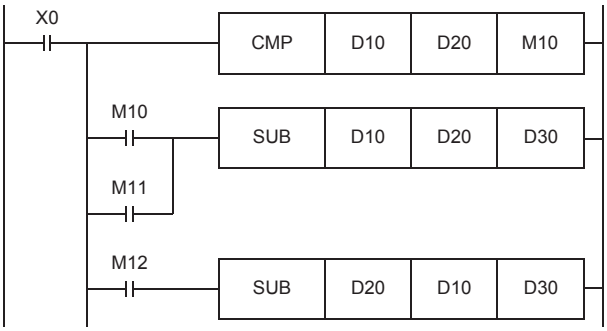
In BON (ON bit check) instruction, M0 turns ON when the bit 15 (b15 among b0 to b15) of D10 is "1".

NEGP instruction is executed for D10 only when M0 turns ON.

$\overline{D10} \rightarrow D10$

- Obtaining the absolute value by SUB (subtraction) instruction

Even if NEG instruction is not used, D30 always stores the absolute value of the difference.



(D10)>(D20) (D10)=(D20) (D10)<(D20)
M10=ON M11=ON M12=ON

In the case of "(D10) ≥ (D20)", (D10) - (D20) → (D30).

In the case of "(D10) < (D20)", (D20) - (D10) → (D30).

Operation error

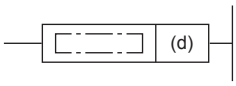
There is no operation error.

Two's complement of 32-bit binary data (sign inversion)

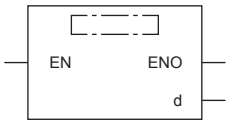
DNEG(P)

FX5S FX5UJ FX5U FX5UC

These instructions invert the sign of the 32-bit binary data in the device specified by (d), and store the resultant data in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DNEG(EN,d); ENO:=DNEGP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

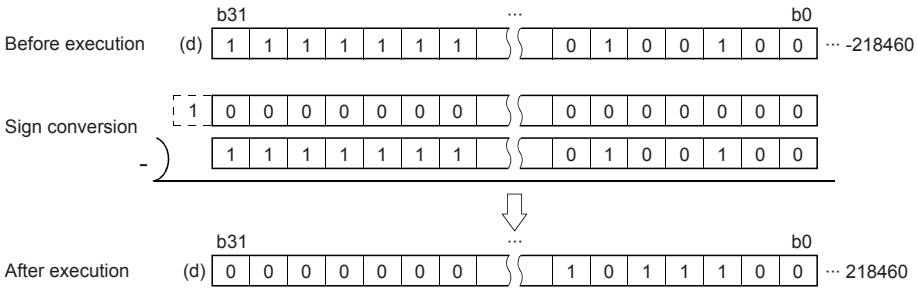
Operand	Description	Range	Data type	Data type (label)
(d)	Head device for storing the data that performs two's complement	-2147483648 to +2147483647	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions invert the sign of the 32-bit binary data in the device specified by (d), and store the resultant data in the device specified by (d).
- They are used when a positive or negative sign is to be inverted.



Precautions

Note that data is inverted in every operation cycle in a continuous operation type (DNEG) instruction.

Operation error

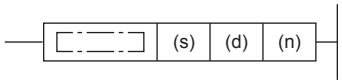
There is no operation error.

Decoding from 8 to 256 bits

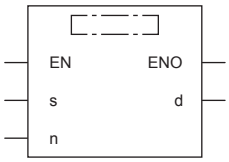
DECO(P)

FX5S FX5UJ FX5U FX5UC

These instructions decode the lower-order (n) bits of the device specified by (s), and store the result in the 2 (to the power (n)) bit from the device specified by (d).

Ladder diagram	Structured text
	ENO:=DECO(EN,s,n,d); ENO:=DECOP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Decode data or device number for storing the decode data	—	Bit/16-bit signed binary	ANY_ELEMENTARY
(d)	Head device for storing the decode result	—	Bit/16-bit unsigned binary	ANY_ELEMENTARY*1
(n)	Valid bit length	1 to 8	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

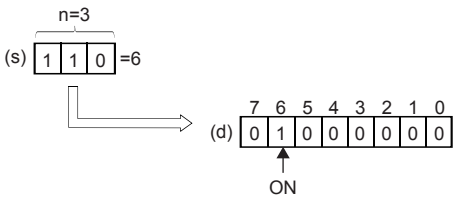
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○*1	○*2	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 Digit specified bit device cannot be used.
*2 Bit-specified word device cannot be used.

Processing details

- These instructions turn ON the bit position of the device specified by (d) in correspondence to the BIN value specified by the lower-order (n) bits of (s).

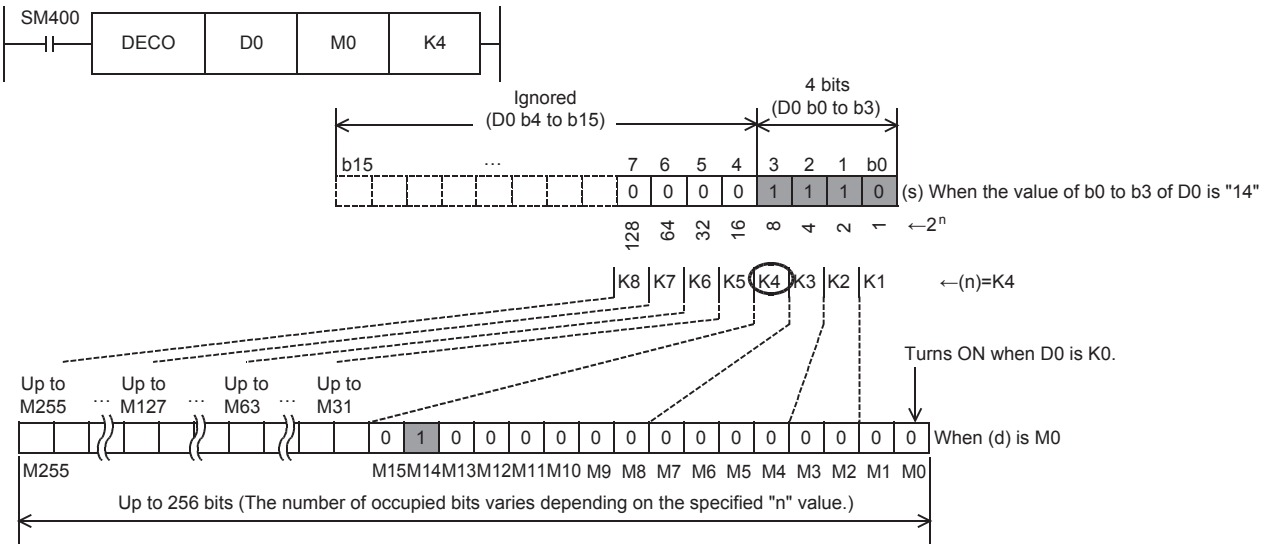


- When (n) is 0, no processing is performed, and the contents of the device specified by (d) do not change.
- The bit device is handled as a device storing one-bit data and the word device is handled as a device storing 16-bit data.

Program example

• When setting bit devices to ON according to the value of a data register

The value of D0 (whose current value is "14" in this example) is decoded to M0 to M15.



• When the value of b0 to b3 of D0 is "14 (= 0 + 2 + 4 + 8)", M14 (which is the 15th from M0) becomes "1" (turn ON).

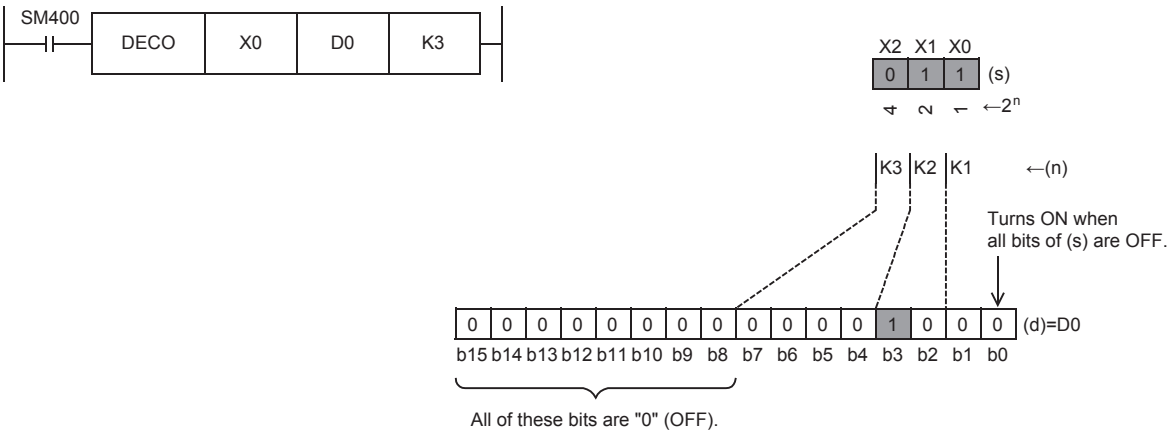
• When the value of D0 is "0", M0 becomes "1" (turns ON).

• When "n" is set to "K4", either one point among M0 to M15 turns ON according to the value of D0 (0 to 15).

• By changing "n" from K1 to K8, D0 can correspond to numeric values from 0 to 255. However, because the device range of (d) is occupied for decoding accordingly, such device range should not be used for another control.

• Turning ON the bit out of word devices according to the contents of bit devices

The value expressed by X0 to X2 is decoded to D0 (X0 and X1 are ON, and X2 is OFF in this example.).



• When the values expressed by X0 to X2 are "3 (= 1 + 2 + 0)", b3 (which is the 4th from b0) becomes 1 (turns ON).

• When all of X0 to X2 are "0" (OFF), b0 becomes "1" (turns ON).

Operation error

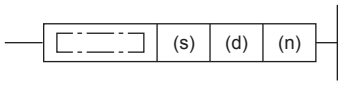
Error code (SD0/SD8067)	Description
2820H	The device specified by (s) exceeds the corresponding device range.
	The device specified by (d) exceeds the corresponding device range.
3401H	(d) is specified as a bit device and (n) is other than 0 to 8.
	(d) is specified as a word device and (n) is other than 0 to 4.

Encoding from 256 to 8 bits

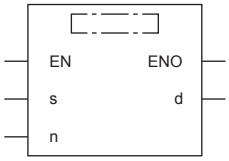
ENCO(P)

FX5S FX5UJ FX5U FX5UC

These instructions encode the 2(n) bits of data from the device specified by (s), and store it in (d).

Ladder diagram	Structured text
	ENO:=ENCO(EN,s,n,d); ENO:=ENCOP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device for storing the encode data	—	Bit/16-bit unsigned binary	ANY_ELEMENTARY*1
(d)	Device number for storing the encoding result	—	16-bit signed binary	ANY_ELEMENTARY
(n)	Valid bit length	1 to 8	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

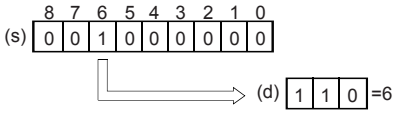
*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	—	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions store into (d) the binary value corresponding to the bit whose value is 1 in the data with $2^{(n)}$ bits.



- When (n) is 0, no processing is performed, and the contents of the device specified by (d) do not change.
- The bit device is handled as a device storing one-bit data and the word device is handled as a device storing 16-bit data.
- If two or more bits are 1, the higher bit position is processed.

Operation error

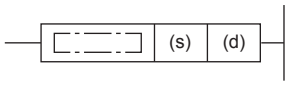
Error code (SD0/SD8067)	Description
2820H	The device specified by (s) exceeds the corresponding device range.
3401H	The entire data from (s) to $2^{(n)}$ number of bits is 0.
	(s) is specified as a bit device and (n) is other than 0 to 8.
	(s) is specified as a word device and (n) is other than 0 to 4.

Seven-segment decoding

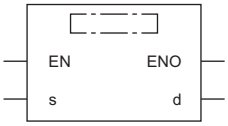
SEGD(P)

FX5S FX5UJ FX5U FX5UC

This instruction decodes data, and turns the seven-segment display unit (1 digit) ON.

Ladder diagram	Structured text
	ENO:=SEGD(EN,s,d); ENO:=SEGDP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device to be decoded	-32768 to +32767	16-bit signed binary	ANY16
(d)	Device number storing the data to be displayed in the seven-segment display unit	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

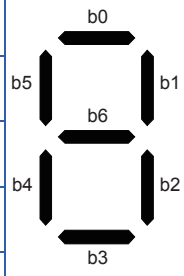
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○*1	○	○	○	—	—	○	—	—	—	—

*1 X cannot be used.

Processing details

- "0" to "F" (hexadecimal numbers) in low-order 4 bits (1 digit) of (s) are decoded to data for the seven-segment display unit, and stored in the low-order 8 bits of (d). Low-order 8 bits of (d) are occupied, and high-order 8 bits do not change.
- Seven-segment decode table is as follows.

(s)					Seven-segment data components	(d)											Display data
Hexadecimal	b3	b2	b1	b0		b15	to	b8	b7	b6	b5	b4	b3	b2	b1	b0	
0	0	0	0	0		0	0	0	0	0	1	1	1	1	1	1	
1	0	0	0	1		0	0	0	0	0	0	0	0	1	1	0	
2	0	0	1	0		0	0	0	0	1	0	1	1	0	1	1	
3	0	0	1	1		0	0	0	0	1	0	0	1	1	1	1	
4	0	1	0	0		0	0	0	0	1	1	0	0	1	1	0	
5	0	1	0	1		0	0	0	0	1	1	0	1	1	0	1	
6	0	1	1	0		0	0	0	0	1	1	1	1	1	0	1	
7	0	1	1	1		0	0	0	0	0	1	0	0	1	1	1	
8	1	0	0	0		0	0	0	0	1	1	1	1	1	1	1	
9	1	0	0	1		0	0	0	0	1	1	0	1	1	1	1	
A	1	0	1	0		0	0	0	0	1	1	1	0	1	1	1	
B	1	0	1	1		0	0	0	0	1	1	1	1	1	0	0	
C	1	1	0	0		0	0	0	0	0	1	1	1	0	0	1	
D	1	1	0	1		0	0	0	0	1	0	1	1	1	1	0	
E	1	1	1	0		0	0	0	0	1	1	1	1	0	0	1	
F	1	1	1	1		0	0	0	0	1	1	1	0	0	0	1	

Operation error

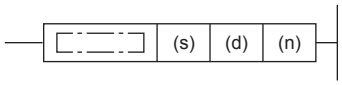
There is no operation error.

Seven Segment With Latch

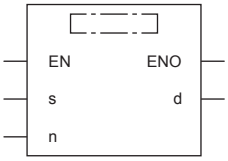
SEGL

FX5S FX5UJ FX5U FX5UC

This instruction controls one or two sets of 4-digit seven-segment display units having the latch function.

Ladder diagram	Structured text
	ENO:=SEGL(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device converted into the BCD format	0 to 9999	16-bit signed binary	ANY16
(d)	Head Y number to be output	—	bit	ANY_BOOL
(n)	Parameter number	0 to 7	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○*1	—	—	—	—	—	—	—	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—

*1 Only Y can be used.

Processing details

The 4-digit numeric value stored in (s) is converted into BCD data, and each digit is output to the seven-segment display unit with the BCD decoder by the time division method. For (s), binary data ranging from 0 to 9999 is valid.
Set the parameters (n) as follows according to the positive/negative logic of the PLC and the positive/negative logic of the seven-segment display module.

PLC output logic	Data input	Strobe signal	Parameter "n"	
			4 digits × 1 set	4 digits × 2 set
Negative logic	Negative logic (match)	Negative logic (match)	0	4
		Positive logic (mismatch)	1	5
	Positive logic (mismatch)	Negative logic (match)	2	6
		Positive logic (mismatch)	3	7
Positive logic	Positive logic (match)	Negative logic (mismatch)	0	4
		Positive logic (match)	1	5
	Negative logic (mismatch)	Negative logic (mismatch)	2	6
		Positive logic (match)	3	7

■When using one set of 4 digits (n = K0 to K3)

A 4-digit numeric value stored in (s) is converted from binary into BCD, and each digit is output in turn from (d) to (d)+3 by the time division method. The strobe signal is output in turn from (d)+4 to (d)+7 by the time division method also to latch one set of 4-digit seven-segment display unit.

■When using two sets of 4 digits (n = K4 to K7)

1st set of 4 digits

A 4-digit numeric value stored in (s) is converted from binary into BCD, and its each digit is output in turn from (d) to (d)+3 by the time division method. The strobe signal is output in turn from (d)+4 to (d)+7 by the time division method also to latch the first set of 4-digit seven-segment display unit.

2nd set of 4 digits

A 4-digit numeric value stored in (s)+1 is converted from binary into BCD, and its each digit is output in turn from (d)+10 to (d)+13 by the time division method. The strobe signal is output in turn from (d)+4 to (d)+7 by the time division method also to latch the second set of 4-digit seven-segment display unit.

For the connection example of two seven-segment display units, refer to the following manual.

MELSEC iQ-F FX5S/FX5UJ/FX5U/FX5UC User's Manual (Hardware)

Precautions

- The scan time (operation cycle) multiplied by 12 is required to update (one or two sets of) the 4-digit display.
- While the command input is ON, the operation is repeated. When the command contact is set to OFF in the middle of an operation, the operation is paused. When the command contact is set to ON again, the operation is started from the beginning.
- When one set of 4 digits is used, one device is occupied from the device specified in (s) and eight devices are occupied from the device specified in (d).
- When two sets of 4 digits are used, two devices are occupied from the device specified in (s) and twelve devices are occupied from the device specified in (d).
- SEGL instruction is executed in synchronization with the scan time (operation cycle) of the CPU module. For achieving a series of display, the scan time of the CPU module should be 10 ms or more. If the scan time is less than 10 ms, use the constant scan mode so that the scan time exceeds 10 ms.
- Use a transistor output type CPU module.
- The SEGL instruction can only be executed four times in a program.
- When a constant (K or H) is specified as (s), operate as shown below.
  When one set of 4 digits is used: It operates considering the constant specified by (s) as the 1st set.
  When two sets of 4 digits are used: It operates considering the constant specified by (s) as the 1st set, and the 2nd set fixed to 0.

Operation error

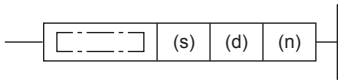
Error code (SD0/SD8067)	Description
2820H	The device specified by (s) or (d) exceeds the corresponding device range.
3405H	(n) is other than 0 to 7.
	The value specified by (s), (s)+1 is other than 0 to 9999.
1811H	The number of the SEGL instructions which are used simultaneously exceeds four.

Separating 4 bits from 16-bit data

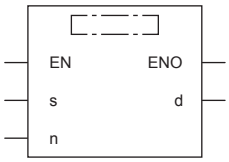
DIS(P)

FX5S FX5UJ FX5U FX5UC

These instructions store the data equivalent of the (n) nibbles (1-nibble/ 4-bits) of the 16-bit binary data specified by (s) in to the lower-order 4 bits of (n) number of devices starting from the one specified by (d).

Ladder diagram	Structured text
	ENO:=DIS(EN,s,n,d); ENO:=DISP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

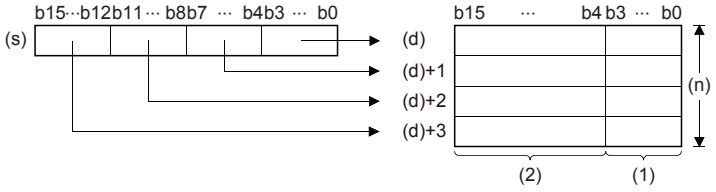
Operand	Description	Range	Data type	Data type (label)
(s)	Head device for storing the data to be separated	—	16-bit signed binary	ANY16
(d)	Head device storing separated data	—	16-bit signed binary	ANY16
(n)	Number of separations (0 indicates no processing is performed)	1 to 4	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions store the data equivalent of the (n) nibbles (1-nibble/ 4-bits) of the 16-bit binary data specified by (s) in to the lower-order 4 bits of (n) number of devices starting from the one specified by (d).

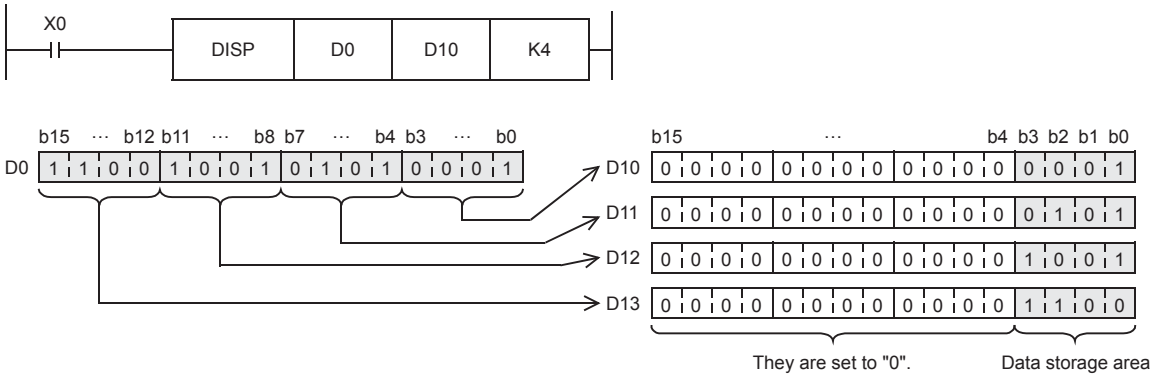


(1): Data storage area

- The higher-order 12 bits of (n) number of devices starting from the one specified by (s) becomes 0.
- When (n) is 0, no processing is performed, and the contents of the (n) number of devices starting from the one specified by (d) do not change.

Program example

In the program below, D0 is separated into 4 bit units and stored in D10 to D13 when X0 turns ON.



Operation error

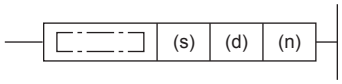
Error code (SD0/SD8067)	Description
2820H	The range of (n) number of points from (d) exceed the corresponding device range.
3401H	(n) is other than 0 to 4.

Connecting 4 bits to 16-bit data

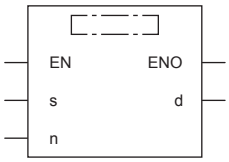
UNI(P)

FX5S FX5UJ FX5U FX5UC

These instructions link the lower-order 4 bits of the 16-bit binary data of the (n) number of devices starting from the one specified by (s) to the device storing 16-bit binary data specified by (d).

Ladder diagram	Structured text
	ENO:=UNI(EN,s,n,d); ENO:=UNIP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

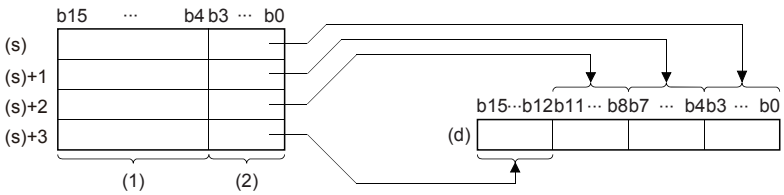
Operand	Description	Range	Data type	Data type (label)
(s)	Head device for storing the data to be linked	—	16-bit signed binary	ANY16
(d)	Head device for storing the linked data	—	16-bit signed binary	ANY16
(n)	Number of links	1 to 4	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions link the lower-order 4 bits of the 16-bit binary data of the (n) number of devices starting from the one specified by (s) to the device storing 16-bit binary data specified by (d).

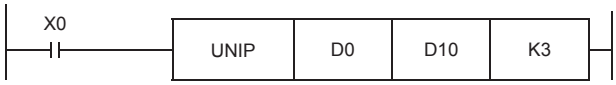


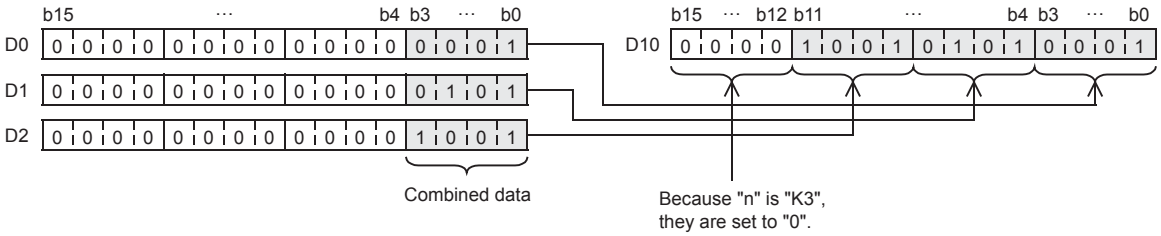
(1): Ignored
(2): Data to be connected

- The higher-order (4-n) nibble bits of the device specified by (d) becomes 0.
- When (n) is 0, no processing is performed, and the contents of the device specified by (d) do not change.

Program example

In the program below, the low-order 4 bits of D0 to D2 are combined and stored in D10 when X0 turns ON.





Operation error

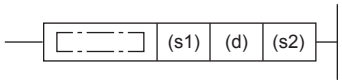
Error code (SD0/SD8067)	Description
2820H	The range of (n) number of points from (d) exceed the corresponding device range.
3401H	(n) is other than 0 to 4.

Separating the specified number of bits

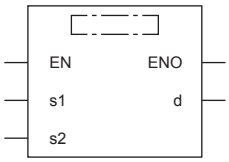
NDIS(P)

FX5S FX5UJ FX5U FX5UC

These instructions separate each bit of the data in the device numbers specified by (s1) onwards into bit units specified by (s2), and store the separated data in the device number specified by (d) onwards.

Ladder diagram	Structured text
	ENO:=NDIS(EN,s1,s2,d); ENO:=NDISP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

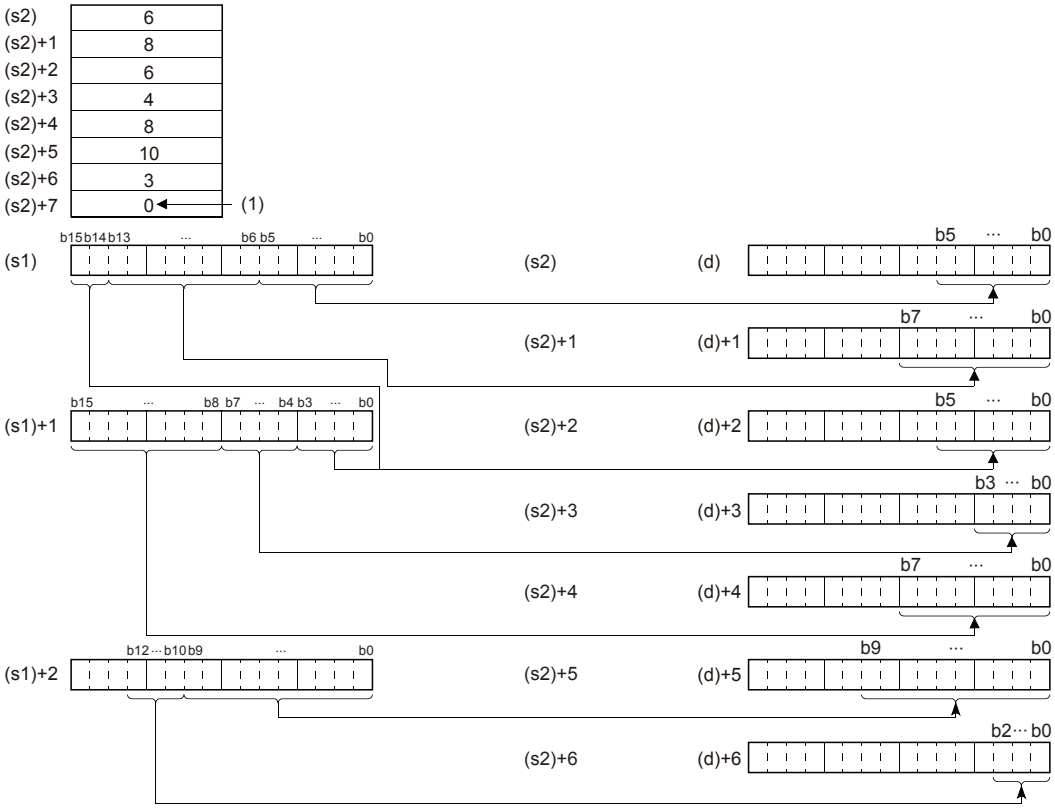
Operand	Description	Range	Data type	Data type (label)
(s1)	Head device for storing the data to be separated	—	16-bit signed binary	ANY16
(d)	Head device for storing the separated data	—	16-bit signed binary	ANY16
(s2)	Head device for storing the separation unit	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—

Processing details

- These instructions separate each bit of the data in the device numbers specified by (s1) and later into bit units specified by (s2), and store the separated data in the device numbers starting from the one specified by (d).



(s2) to (s2)+6: Number of bits specified by (s2) to (s2)+6
(1): The value 0 indicates the end of setting.

- The number of separation bits specified by (s2) can be specified within the range of 1 to 16 bits.
- The number of bits specified in devices from the device number specified by (s2) up to the device number in which "0" is stored are processed as the number of separation bits.
- If the device numbers specified by (s1), (s2), (d) are partially overlapping, an operation error occurs.

Operation error

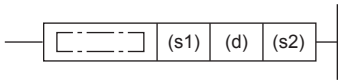
Error code (SD0/SD8067)	Description
2820H	The usage range of the device specified by (s1) or (d) exceeds the corresponding device range due to the specification of the number of separation bits specified by (s2).
2821H	The (s1), (s2) devices are overlapping.
	The (s1), (d) devices are overlapping.
	The (s2), (d) devices are overlapping.
3401H	The specification of the number of separation bits specified by (s2) is not set within the range of 1 to 16 bits.
	0 is not set in the range between the device specified by (s2) up to the corresponding device range.

Connecting the specified number of bits

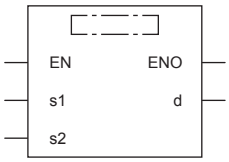
NUNI(P)

FX5S FX5UJ FX5U FX5UC

These instructions link each bit of the data in the device numbers specified by (s1) onwards into bit units specified by (s2), and store the connected data in the device number specified by (d) onwards.

Ladder diagram	Structured text
	ENO:=NUNI(EN,s1,s2,d); ENO:=NUNIP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

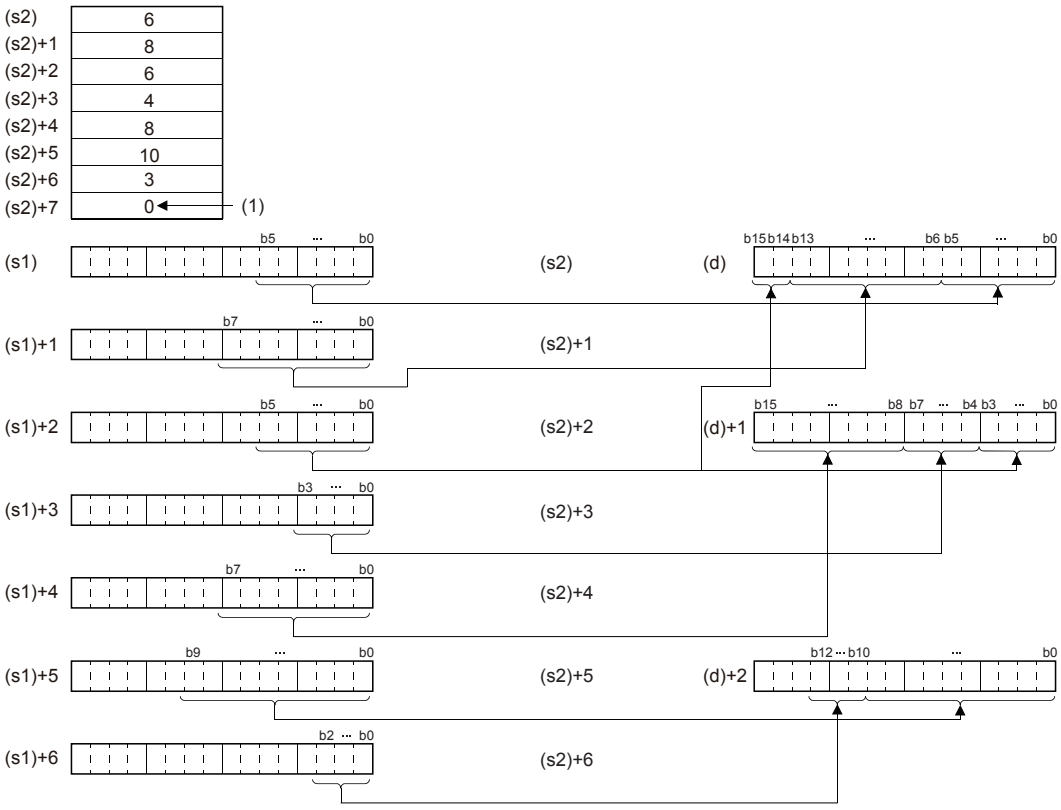
Operand	Description	Range	Data type	Data type (label)
(s1)	Head device for storing the data to be linked	—	16-bit signed binary	ANY16
(d)	Head device for storing the linked data	—	16-bit signed binary	ANY16
(s2)	Head device for storing the link unit size	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—

Processing details

- These instructions link each bit of the data in the device numbers specified by (s1) onwards into bit units specified by (s2), and store the linked data in the device number specified by (d).



(s2) to (s2)+6: Number of bits specified by (s2) to (s2)+6
(1): The value 0 indicates the end of setting.

- The number of link bits specified by (s2) can be specified within the range of 1 to 16 bits.
- The number of bits specified in devices from the device number specified by (s2) up to the device number in which "0" is stored are processed as the number of connection bits.
- If the device numbers specified by (s1), (s2), (d) are partially overlapping, an operation error occurs.

Operation error

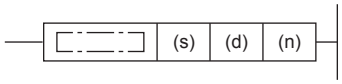
Error code (SD0/SD8067)	Description
2820H	The usage range of the device specified by (s1) or (d) exceeds the corresponding device range due to the specification of the number of link bits specified by (s2).
2821H	The (s1), (s2) devices are overlapping.
	The (s1), (d) devices are overlapping.
	The (s2), (d) devices are overlapping.
3401H	The specification of the number of link bits specified by (s2) is not set within the range of 1 to 16 bits.
	0 is not set in the range between the device specified by (s2) up to the corresponding device range.

Separating data in byte units

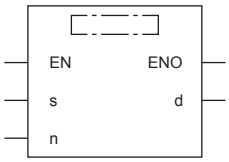
WTOB(P)

FX5S FX5UJ FX5U FX5UC

These instructions separate the 16-bit binary data in the device numbers starting from the one specified by (s) onwards into (n) byte units, and store the separated data in the device number specified by (d) onwards.

Ladder diagram	Structured text
	ENO:=WTOB(EN,s,n,d); ENO:=WTOBP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

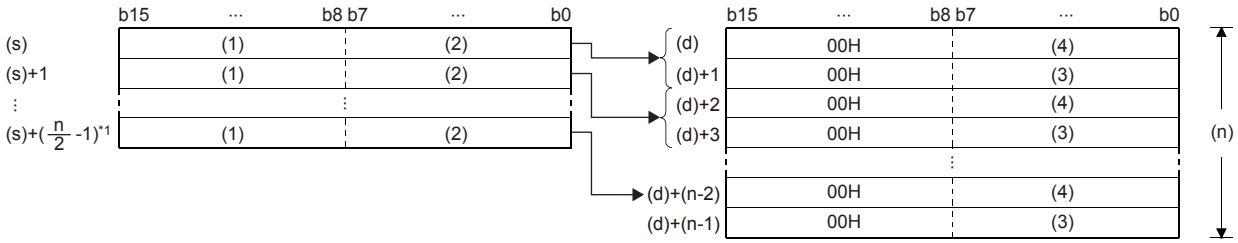
Operand	Description	Range	Data type	Data type (label)
(s)	Head device where the separation target data is stored	—	16-bit signed binary	ANY16
(d)	Head device for storing the result of separation in byte unit	—	16-bit signed binary	ANY16
(n)	Number of byte units	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions separate the 16-bit binary data in the device numbers starting from the one specified by (s) onwards into (n) byte units, and store the separated data in the device number specified by (d) onwards.

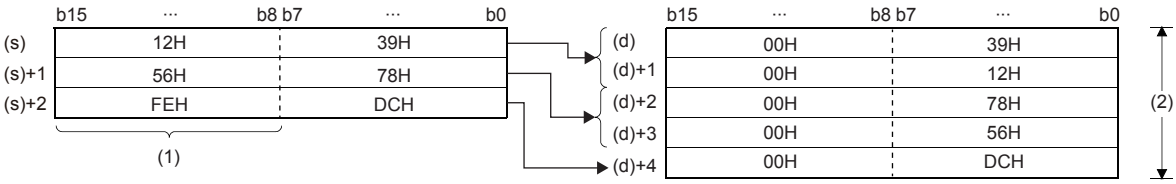


(1): Upper byte
(2): Lower byte
(3): Upper byte data
(4): Lower byte data

*1 Values after the decimal point are rounded up.

Ex.

For example, when (n) is 5, data starting from (s) to the lower 8 bits of (s)+2 is stored into (d) through (d)+4.

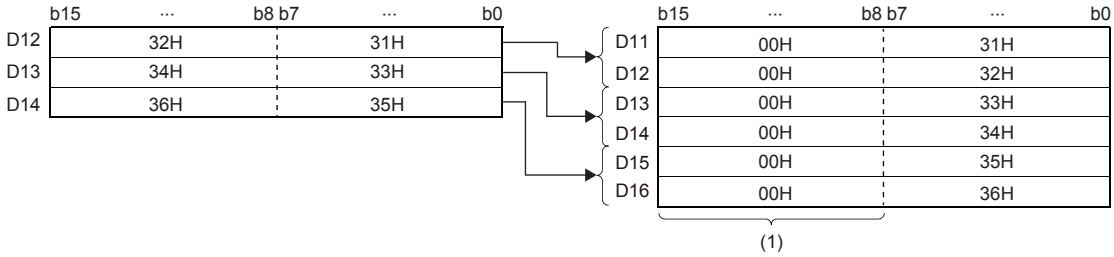


(1): The data FEH is ignored when (n) is 5.
(2): When (n) is 5

- Setting the number of bytes by (n) automatically determines the 16-bit binary data range specified by (s) and the device range specified by (d) for storing the separated byte data.
- If (n) is 0, no processing is performed.
- In the upper 8 bits of the devices specified by (d) to hold byte data, 00H is automatically stored.

Ex.

To store data in D12 to D14 into the lower 8 bits of D11 to D16



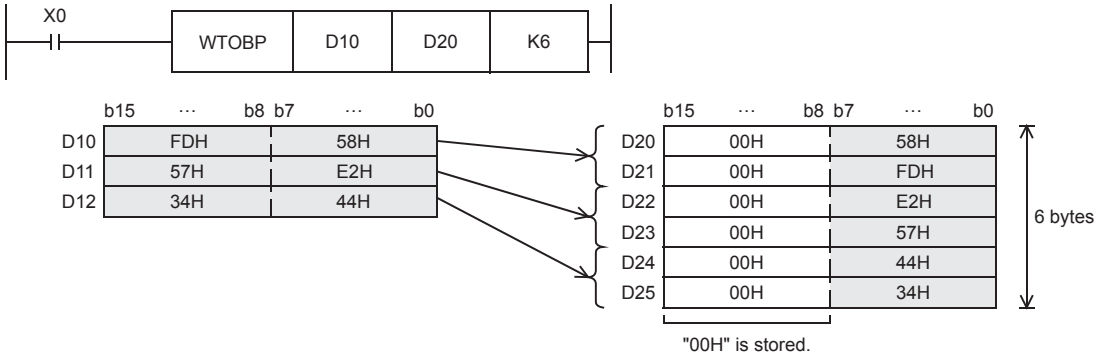
(1): 00H is automatically stored.

- Even if the device range of the data to be separated and the device range for storing the separated data overlap, the processing is performed normally.

Device range where the data to be separated is stored	Device range for storing the separated data
(s) to (s)+($\frac{n}{2}$ -1)	(d)+0 to (d)+(n)-1

Program example

In the program shown below, the data stored in D10 to D12 is separated in byte units, and stored in D20 to D25 when X0 is set to ON.



Operation error

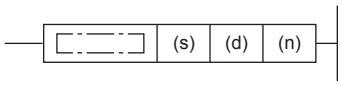
Error code (SD0/SD8067)	Description
2820H	The range of no. of bytes specified in (n) from the device number specified in (s) onwards exceed the corresponding device range.
	The range of (n) points of devices from the device number specified in (d) onwards exceed the corresponding device range.

Connecting data in byte units

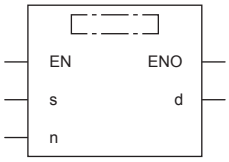
BTOW(P)

FX5S FX5UJ FX5U FX5UC

These instructions link the lower-order 8 bits of the 16-bit binary data of (n) number of bytes stored in the device numbers starting from the one specified by (s) onwards into word units, and store the linked data in the device numbers starting from the one specified by (d) onwards.

Ladder diagram	Structured text
	ENO:=BTOW(EN,s,n,d); ENO:=BTOWP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

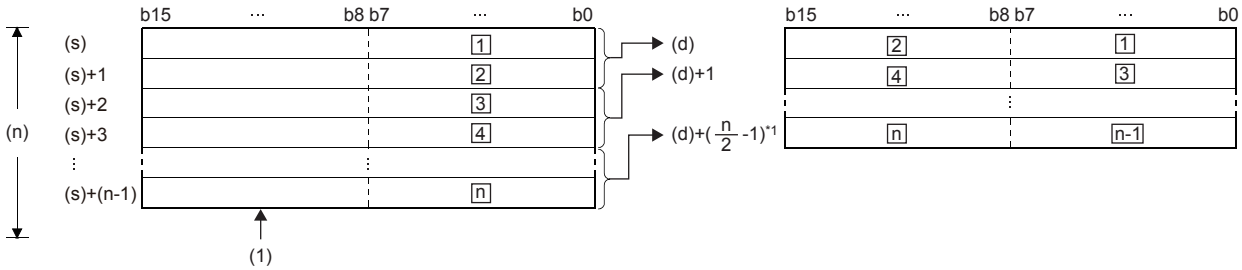
Operand	Description	Range	Data type	Data type (label)
(s)	Head device for storing the data to be linked in byte units	—	16-bit signed binary	ANY16
(d)	Head device storing data acquired by combination in byte units	—	16-bit signed binary	ANY16
(n)	Number of byte data to be linked	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions link the lower-order 8 bits of the 16-bit binary data of (n) number of bytes stored in the device numbers starting from the one specified by (s) onwards into word units, and store the linked data in the device numbers starting from the one specified by (d) onwards.
- The higher-order 8 bits of the data of (n) words stored in device numbers starting from the one specified by (s) are ignored. If (n) is an odd number, 0 is stored in the higher-order 8 bits of the device for storing the data of the (n)th byte.

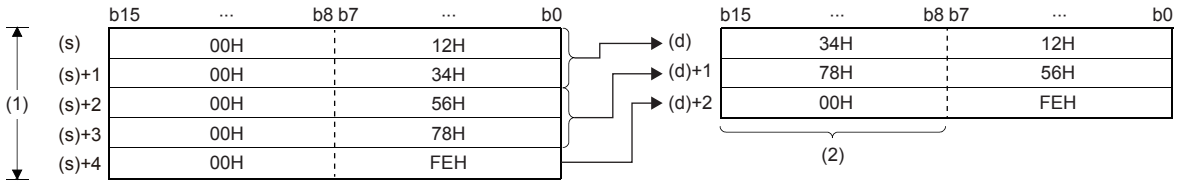


□: □th byte data
(1): The upper byte data are ignored.

*1 Values after the decimal point are rounded up.

Ex.

For example, when (n) is 5, lower 8 bits of data from (s) through (s+4) is stored into (d) through (d)+2.

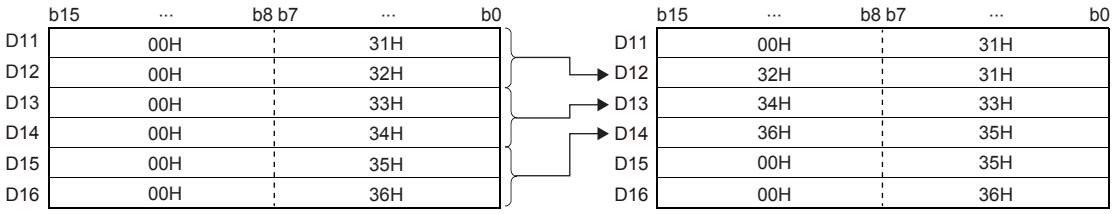


(1): When (n) is 5
(2): Filled with 00H.

- Setting the number of bytes by (n) automatically determines the byte data range specified by (s) and the device range specified by (d) for storing the linked data.
- If (n) is 0, no processing is performed.
- The higher-order 8 bits of the device specified by (s) for storing byte data are ignored, and only the lower-order 8 bits are applicable.

Ex.

To store data in lower 8 bits of D11 to D16 into D12 to D14

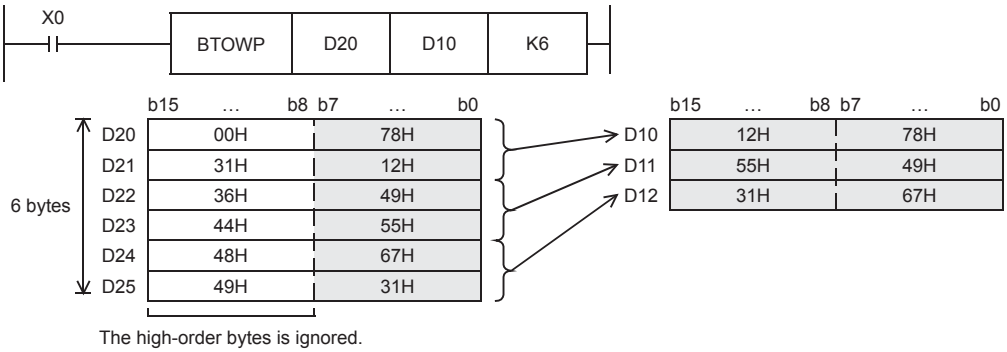


- Even if the device range of the data to be linked and the device range for storing the linked data overlap, the processing is performed normally.

Device range where the data to be linked is stored	Device range for storing the linked data
(s)+0 to (s)+(n)-1	(d) to (d)+(n/2 -1)

Program example

In the program shown below, the low-order 8-bits data stored in D20 to D25 is combined, and stored in D10 to D12 when X0 is set to ON.



Operation error

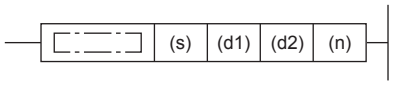
Error code (SD0/SD8067)	Description
2820H	The range of (n) points of devices from the device number specified in (s) onwards exceed the corresponding device range.
	The range of no. of bytes specified in (n) from the device number specified in (d) onwards exceed the corresponding device range.

7.6 Digital Switch

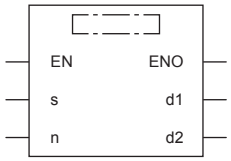
DSW

FX5S FX5UJ FX5U FX5UC

This instruction reads the set value of digital switches. This instruction can read a set of 4 digits (n = K1) or two sets of 4 digits (n = K2).

Ladder diagram	Structured text
	ENO:=DSW(EN,s,n,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Start device number to connect a digital switch	—	Bit	ANYBIT_ARRAY (Number of elements: 4)
(d1)	Start device number of strobe signal output	—	Bit	ANYBIT_ARRAY (Number of elements: 4)
(d2)	Device number storing the numeric value of a digital switch	0 to 9999	16-bit signed binary	ANY16
(n)	Total number of 4-digit switch sets	1, 2	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○*1	—	—	—	—	—	—	—	—	—	—
(d1)	○*2	—	—	—	—	—	—	—	—	—	—
(d2)	—	○	○	○	—	—	○	—	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—

*1 Only X can be used.
*2 Only Y can be used.

Processing details

The value of each digital switch connected to (s) is input by the time division method (in which the value is input in turn from the 1st digit by 100 ms interval output signal), and stored to (d2). A numeric value from 0 to 9999 (up to 4 digits) can be read, the first set is stored to (d2), and the second set is stored to (d2)+1.

When using one set of 4 digits (n = K1)

A 4-digit BCD digital switch connected to (s) to (s)+3 is read in turn by the strobe signal (d1) to (d1)+3, and stored in binary format to (d2).

When using two sets of 4 digits (n = K2)

A 4-digit BCD digital switch connected to (s) to (s)+7 is read in turn by the strobe signal (d1) to (d1)+3. (s) to (s3)+3 is stored in (d2) and (s)+4 to (s)+7 is stored in (d2)+1 as BIN format.

For the connection example of digital switch, refer to the following manual.

MELSEC iQ-F FX5S/FX5UJ/FX5U/FX5UC User's Manual (Hardware)

Precautions

- Though the contents of (d2) do not change, all of (d1) to (d1)+3 turn OFF.
- When one set of 4 digits (n = K1) are used, four devices are occupied starting from (s).
- When two sets of 4 digits (n = K2) are used, eight devices are occupied starting from (s), and two devices are occupied starting from (d2).
- When connecting a digital switch of less than 4 digits, it is not necessary to wire the strobe signal (output for digit specification) (d1) to unused digits. Because unused digits are occupied also by this instruction, however, they cannot be used for any other purpose.
- For continuously receiving digital switch values, make sure to use a transistor output type CPU module.
- Use BCD output type digital switches.
- The DSW instruction can only be executed four times in a program.

Operation error

Error code (SD0/SD8067)	Description
3405H	(n) is other than 1 or 2.
	The value specified by (s) to (s)+3 and (s)+4 to (s)+7 is other than 0 to 9.
2820H	The device range specified by (s), (d1), (d2) exceeds the corresponding device range.
1811H	The number of the DSW instructions which are used simultaneously exceeds four.

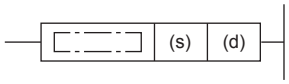
7.7 Data Transfer Instructions

Transferring 16-bit data

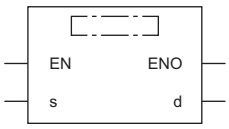
MOV(P)

FX5S FX5UJ FX5U FX5UC

These instructions transfer the 16-bit binary data in the device specified by (s) to the device specified by (d).

Ladder diagram	Structured text
	ENO:=MOV(EN,s,d); ENO:=MOVP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

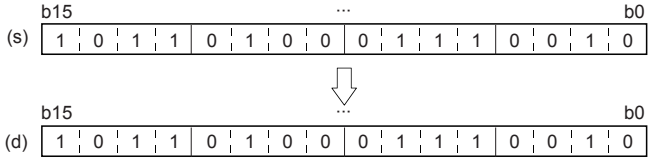
Operand	Description	Range	Data type	Data type (label)
(s)	Transfer source data or device number for storing data	-32768 to +32767	16-bit signed binary	ANY16
(d)	Transfer destination device number	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

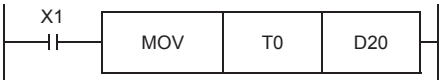
Processing details

- These instructions transfer the 16-bit binary data in the device specified by (s) to the device specified by (d).



Program example

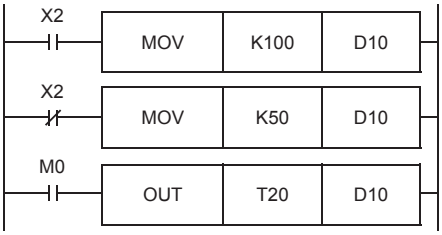
• When reading the current value of a timer and counter



(Current value of T0) → (D20)
The operation is the same as a counter.

• When indirectly specifying the set value of a timer or counter

As the set value of the timer T20, two values can be specified by turning ON or OFF the switch X2.
For specifying more than two set values, more than one switch is required.



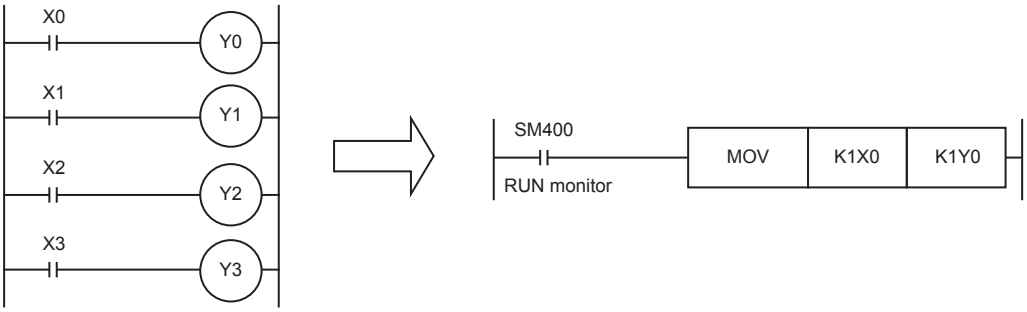
(K100)→(D10)

(K50)→(D10)

When X2 is ON, D10 = K100 (10-second timer).
When X2 is OFF, D10 = K50 (5-second timer).

• When transferring a bit device

The following sequence program can be programmed by using MOV instruction.



Operation error

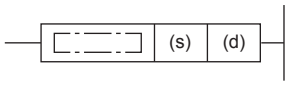
There is no operation error.

Transferring 32-bit data

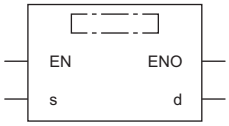
DMOV(P)

FX5S FX5UJ FX5U FX5UC

These instructions transfer the 32-bit binary data in the device specified by (s) to the device specified by (d).

Ladder diagram	Structured text
	ENO:=DMOV(EN,s,d); ENO:=DMOVP(EN,s,d)

FBD/LD



Setting data

■Descriptions, ranges, and data types

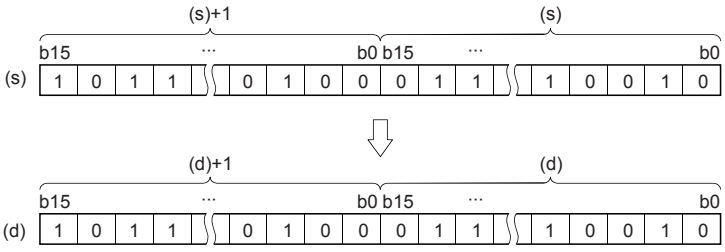
Operand	Description	Range	Data type	Data type (label)
(s)	Transfer source data or device number for storing data	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Transfer destination device number	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word		Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

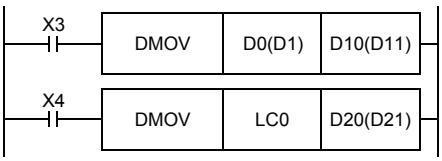
Processing details

- These instructions transfer the 32-bit binary data in the device specified by (s) to the device specified by (d).



Program example

Make sure to use DMOV instruction for transferring an instruction (such as MUL instruction) whose operation result is output in 32 bits, a 32-bit numeric value or a 32-bit device.



(D1, D0) → (D11, D10)

(Current value of LC0) → (D21, D20)

Operation error

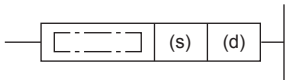
There is no operation error.

Inverting and transferring 16-bit data

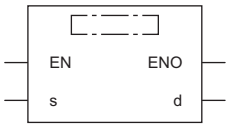
CML(P)

FX5S FX5UJ FX5U FX5UC

These instructions invert each bit of the 16-bit binary data in the device specified by (s), and transfer the result to the device specified by (d).

Ladder diagram	Structured text
	ENO:=CML(EN,s,d); ENO:=CMLP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

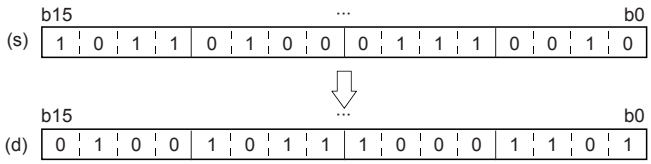
Operand	Description	Range	Data type	Data type (label)
(s)	Data to be inverted or device number in which data is stored	-32768 to +32767	16-bit signed binary	ANY16
(d)	Device number for storing the inversion result	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

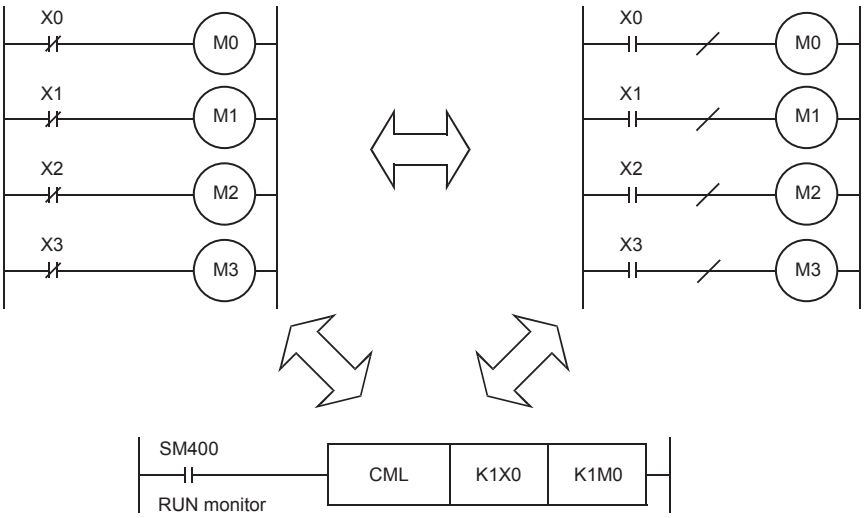
- These instructions invert each bit of the 16-bit binary data in the device specified by (s), and transfer the result to the device specified by (d).



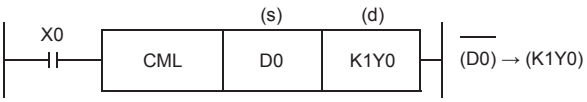
Program example

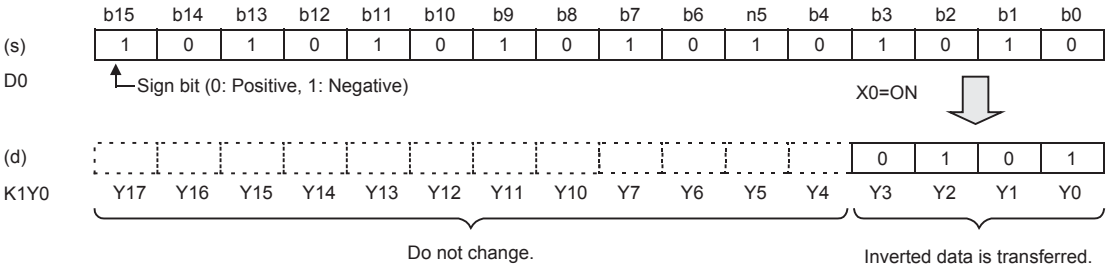
• When receiving an inverted input

The sequence program shown below can be written by CML instruction.



• When four bits are specified for a device with digit specification





Operation error

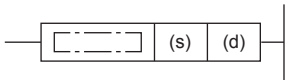
There is no operation error.

Inverting and transferring 32-bit data

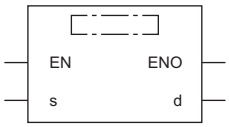
DCML(P)

FX5S FX5UJ FX5U FX5UC

These instructions invert each bit of the 32-bit binary data in the device specified by (s), and transfer the result to the device specified by (d).

Ladder diagram	Structured text
	ENO:=DCML(EN,s,d); ENO:=DCMLP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

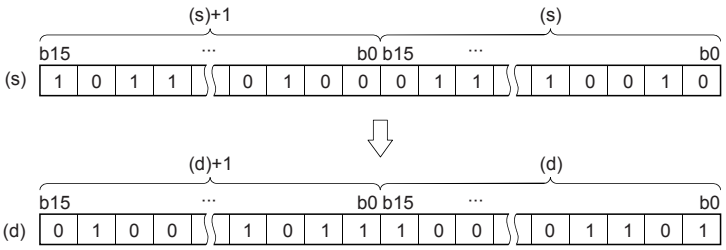
Operand	Description	Range	Data type	Data type (label)
(s)	Data to be inverted or device number in which data is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Device number for storing the inversion result	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word		Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions invert each bit of the 32-bit binary data in the device specified by (s), and store the result in the device specified by (d).



Operation error

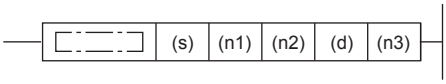
There is no operation error.

Digit move

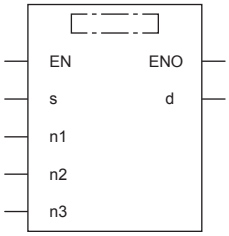
SMOV(P)

FX5S FX5UJ FX5U FX5UC

These instructions distribute and compose data in units of nibble (4 bits).

Ladder diagram	Structured text
	ENO:=SMOV(EN,s,n1,n2,n3,d); ENO:=SMOVP(EN,s,n1,n2,n3,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Word device number storing data whose nibbles will be moved	—	16-bit signed binary	ANY16
(n1)*1	Head nibble position to be moved	1 to 4	16-bit unsigned binary	ANY16_U
(n2)*1	Number of nibbles to be moved	1 to 4	16-bit unsigned binary	ANY16_U
(d)	Word device number storing data whose nibbles are moved	—	16-bit signed binary	ANY16
(n3)*1	Head digit position of movement destination	1 to 4	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Set so that n2≤n1, n2≤n3.

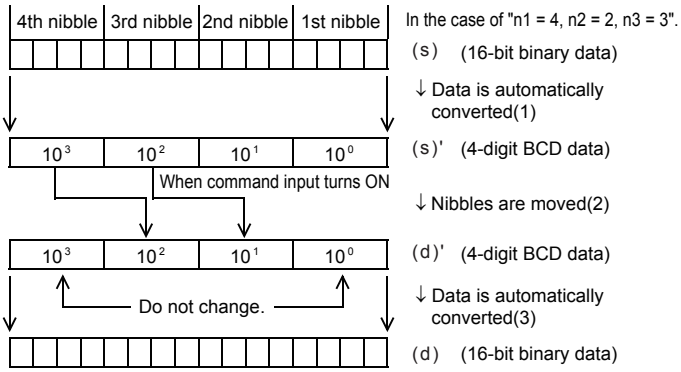
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n3)	○	○	○	○	—	—	○	○	—	—	—

Processing details

These instructions distribute and compose data in units of nibble (4 bits). The contents of the transfer source (s) and transfer destination (d) are converted into 4-digit BCD (0000 to 9999). (n2) nibbles starting from the (n1)th nibble are transferred to the transfer destination (d) starting from the (n3)th nibble, converted into binary, and then stored to the transfer destination (d).

- While the command input is OFF, the transfer destination (d) does not change.
- When the command input turns ON, only the specified digits in the transfer destination (d) are changed. The transfer source (s) and unspecified digits in the transfer destination (d) do not change.



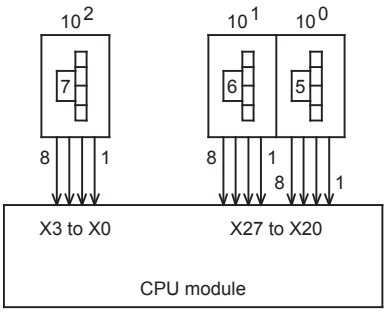
(1)(s) is converted from binary to → BCD data.
(2) (n2) digits starting from the (n1)th digit are transferred (combined) to (d)' starting from the (n3)th digit. The first and fourth digits of (d)' are not affected even if data is transferred from (s)'.
(3)The combined data (BCD) is converted into binary, and stored to (d).

■Extension function

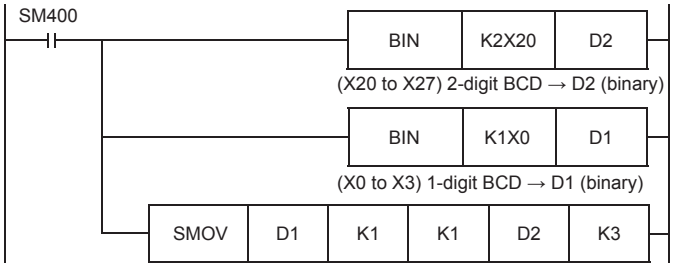
When SM8168 is set to ON first and then SMOV instruction is executed, conversion from binary to BCD is not executed. Data is moved in units of 4 bits.

Program example

The data on three-digit digital switches are composed, and stored as binary data to D2.



Data on three digital switches connected to non-consecutive input terminals are composed.



1-digit BCD data of D1 is transferred to the 3rd digit (BCD data) of D2, and automatically converted into binary data.

Operation error

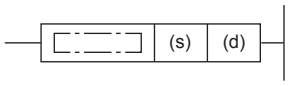
Error code (SD0/SD8067)	Description
3405H	Any one of (n1), (n2), (n3) is 0.
	Either (s) or (d) is other than 0 to 9999 when SM8168 is OFF.
	Either (n1) or (n3) is larger than 4.
	(n2) is larger than (n1) or (n3).

Inverting and transferring 1-bit data

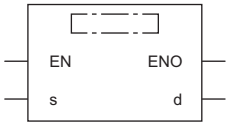
CMLB(P)

FX5S FX5UJ FX5U FX5UC

These instructions invert the bit data in the device specified by (s), and transfer the result into the device specified by (d).

Ladder diagram	Structured text
	ENO:=CMLB(EN,s,d); ENO:=CMLBP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

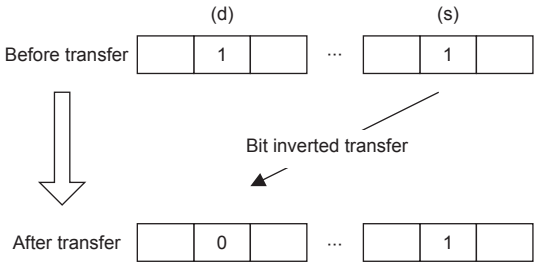
Operand	Description	Range	Data type	Data type (label)
(s)	Transfer-source data	—	Bit	ANY_BOOL
(d)	Transfer-destination data	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	○	—	—	—	—	—	—
(d)	○	○	○	—	○	—	—	—	—	—	—

Processing details

These instructions invert the bit data in the device specified by (s), and transfer the result into the device specified by (d).



Operation error

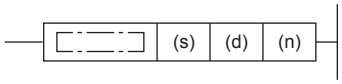
There is no operation error.

Transferring 16-bit block data (65535 points maximum)

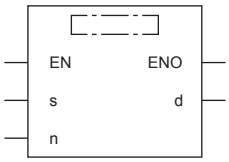
BMOV(P)

FX5S FX5UJ FX5U FX5UC

These instructions block transfer the 16-bit binary data of (n) number of devices starting from the one specified by (s) to the device specified by (d).

Ladder diagram	Structured text
	ENO:=BMOV(EN,s,n,d); ENO:=BMOVP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

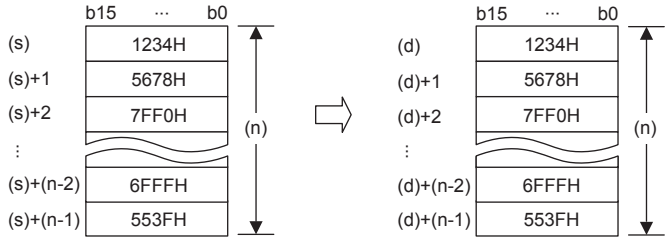
Operand	Description	Range	Data type	Data type (label)
(s)	Head device for storing the data to be transferred	—	16-bit signed binary/ 32-bit signed binary	ANY16
(d)	Head number of the transfer-destination device	—	16-bit signed binary/ 32-bit signed binary	ANY16
(n)	Number of transfers	1 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	○	—	○	—	—	—	—
(d)	○	○	○	—	○	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions block transfer the 16-bit binary data of (n) number of devices starting from the one specified by (s) to the device specified by (d).

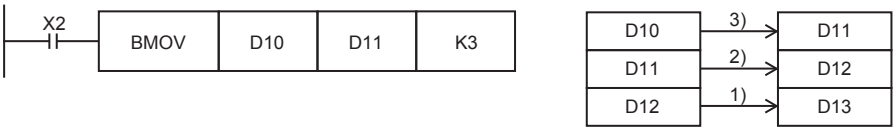


- If the device number range is exceeded, data is transferred within the possible range.
- Data can be transferred even when the device range of the transfer-source device and transfer-destination device is overlapping. To transfer data to a device having a smaller device number, transfer from (s), and to transfer data to a device having a larger device number, transfer from (s)+(n)-1.

Ex.

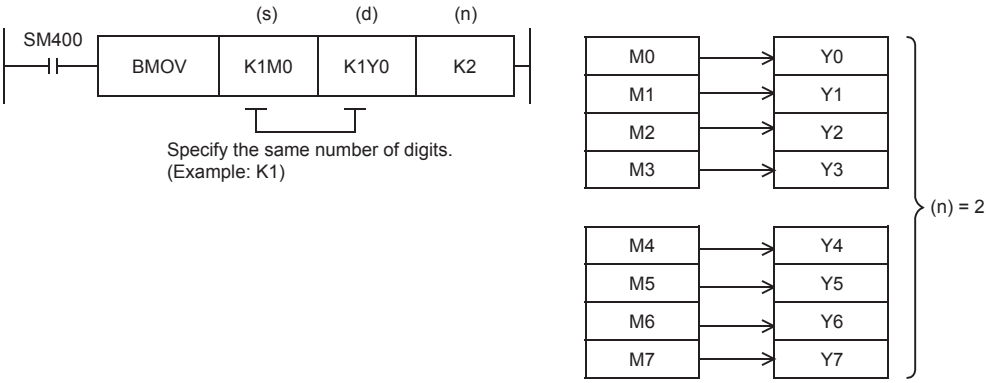
When transferring data to a device having a smaller device number

When transferring data to a device having a larger device number



Precautions

- To perform digit specification of bit device for both (s) and (d), be sure to set the same number of nibbles for (s) and (d).



- To use a module access device for (s) and (d), specify either (s) or (d).

Operation error

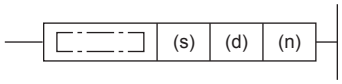
Error code (SD0/SD8067)	Description
3405H	The number of nibbles of the digit specification of bit device of (s) and (d) is different.
3420H	A module access device is specified for both (s) and (d).

Transferring identical 16-bit block data (65535 points maximum)

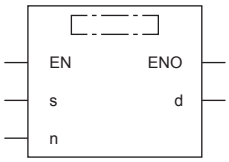
FMOV(P)

FX5S FX5UJ FX5U FX5UC

These instructions transfer (n) point(s) of data identical to the 16-bit binary data in the device specified by (s) to the device specified by (d).

Ladder diagram	Structured text
	ENO:=FMOV(EN,s,n,d); ENO:=FMOVP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

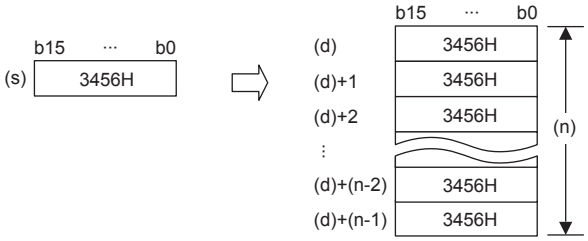
Operand	Description	Range	Data type	Data type (label)
(s)	Data to be transferred or the head device for storing the data to be transferred	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device of the transfer-destination	—	16-bit signed binary	ANY16
(n)	Number of transfers	1 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions transfer (n) point(s) of data identical to the 16-bit binary data in the device specified by (s) to the device specified by (d).



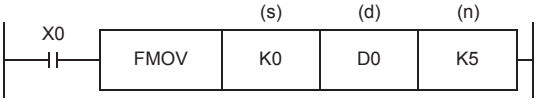
- If the number of points specified by (n) exceeds the device number range, data is transferred within the possible range.
- When a constant (K) is specified as the transfer source (s), it is automatically converted into binary.

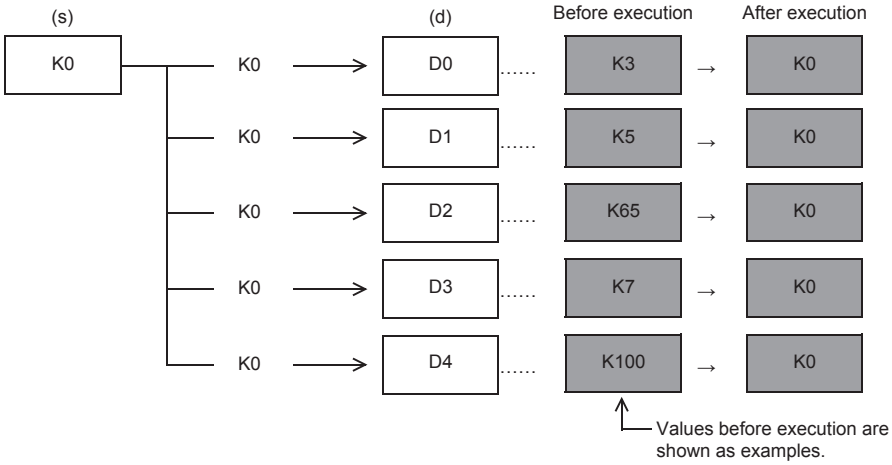
Precautions

When the value specified in (n) is 0, an operation error does not occur, but no processing is performed.

Program example

- When writing specified data to two or more devices





Operation error

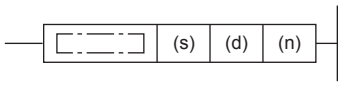
There is no operation error.

Transferring identical 32-bit block data (65535 points maximum)

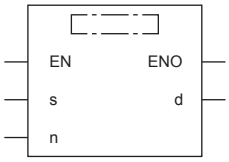
DFMOV(P)

FX5S FX5UJ FX5U FX5UC

These instructions transfer (n) point(s) of data identical to the 32-bit binary data in the device specified by (s) to the device specified by (d).
(65535 points maximum)

Ladder diagram	Structured text
	ENO:=DFMOV(EN,s,n,d); ENO:=DFMOVP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

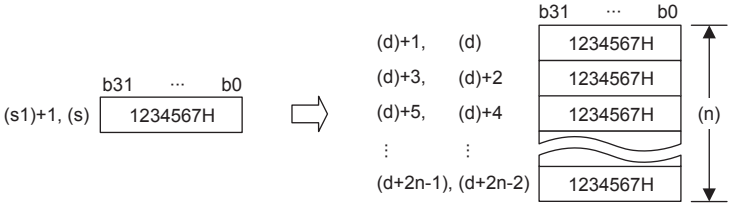
Operand	Description	Range	Data type	Data type (label)
(s)	Data to be transferred or the head device for storing the data to be transferred	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Head device of the transfer-destination	—	32-bit signed binary	ANY32
(n)	Number of transfers	1 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	—	○	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions transfer (n) point(s) of data identical to the 32-bit binary data in the device specified by (s) to the device specified by (d).



- If the number of points specified by (n) exceeds the device number range, data is transferred within the possible range.
- When a constant (K) is specified as the transfer source (s), it is automatically converted into binary.

Precautions

When the value specified in (n) is 0, an operation error does not occur, but no processing is performed.

Operation error

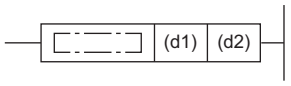
There is no operation error.

Exchanging 16-bit data

XCH(P)

FX5S FX5UJ FX5U FX5UC

These instructions exchange 16-bit binary data of (d1) and (d2).

Ladder diagram	Structured text
	ENO:=XCH(EN,d1,d2); ENO:=XCHP(EN,d1,d2);

FBD/LD

Setting data

■Descriptions, ranges, and data types

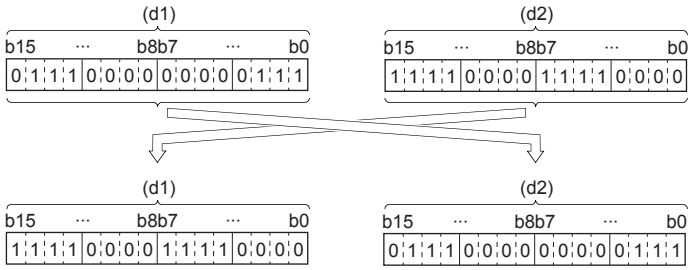
Operand	Description	Range	Data type	Data type (label)
(d1)	Head device for storing the data to be exchanged	—	16-bit signed binary	ANY16
(d2)	Head device for storing the data to be exchanged	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d1)	○	○	○	○	—	—	○	—	—	—	—
(d2)	○	○	○	○	—	—	○	—	—	—	—

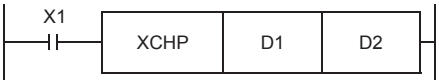
Processing details

These instructions exchange 16-bit binary data of (d1) and (d2).



Program example

This program exchanges the contents of D1 and D2 when X1 is set to ON.



Operation error

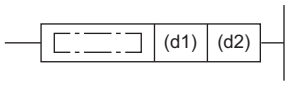
There is no operation error.

Exchanging 32-bit data

DXCH(P)

FX5S FX5UJ FX5U FX5UC

These instructions exchange 32-bit binary data of (d1) and (d2).

Ladder diagram	Structured text
	ENO:=DXCH(EN,d1,d2); ENO:=DXCHP(EN,d1,d2);

FBD/LD

Setting data

■Descriptions, ranges, and data types

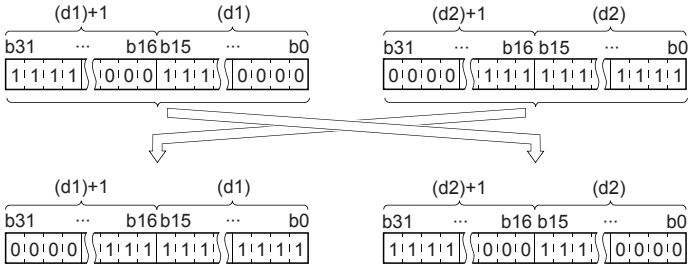
Operand	Description	Range	Data type	Data type (label)
(d1)	Head device for storing the data to be exchanged	—	32-bit signed binary	ANY32
(d2)	Head device for storing the data to be exchanged	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d1)	○	○	○	○	○	○	○	—	—	—	—
(d2)	○	○	○	○	○	○	○	—	—	—	—

Processing details

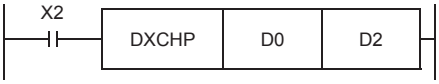
These instructions exchange 32-bit binary data of (d1), (d1)+1 and (d2), (d2)+1



7

Program example

This program exchanges the contents of D0, D1 and D2, D3 when X2 is set to ON.



Operation error

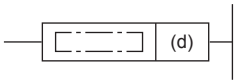
There is no operation error.

Exchanging the upper and lower bytes of 16-bit data

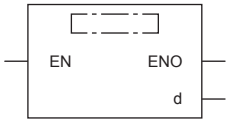
SWAP(P)

FX5S FX5UJ FX5U FX5UC

These instructions swap the value of 8 bits of the upper and lower bytes of the device specified by (d).

Ladder diagram	Structured text
	ENO:=SWAP(EN,d); ENO:=SWAPP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

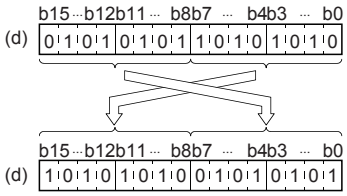
Operand	Description	Range	Data type	Data type (label)
(d)	Head device for storing the data to be exchanged	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

These instructions swap the value of 8 bits of the upper and lower bytes of the device specified by (d).



Precautions

If a continuous operation type instruction is used, swap is done in each operation cycle.

Operation error

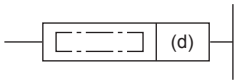
There is no operation error.

Exchanging the upper and lower bytes of 32-bit data

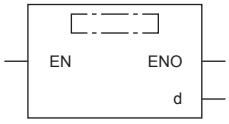
DSWAP(P)

FX5S FX5UJ FX5U FX5UC

These instructions swap the value of 8 bits of the upper and lower bytes of the devices specified by (d) and (d)+1.

Ladder diagram	Structured text
	ENO:=DSWAP(EN,d); ENO:=DSWAPP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

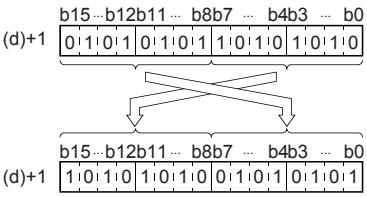
Operand	Description	Range	Data type	Data type (label)
(d)	Head device for storing the data to be exchanged	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

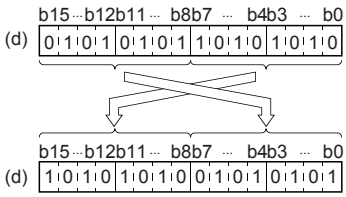
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

These instructions swap the value of each of the upper and lower 8 bits of the device specified by (d) and (d)+1.





Precautions

If a continuous operation type instruction is used, swap is done in each operation cycle.

Operation error

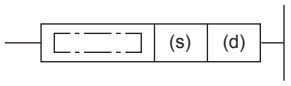
There is no operation error.

Transferring 1-bit data

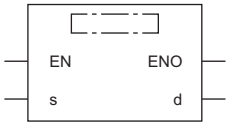
MOVB(P)

FX5S FX5UJ FX5U FX5UC

These instructions store bit data specified by (s) to (d).

Ladder diagram	Structured text
	ENO:=MOVB(EN,s,d); ENO:=MOVBP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

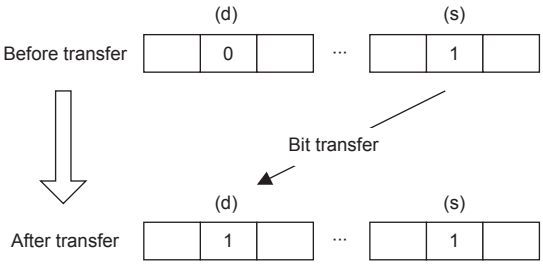
Operand	Description	Range	Data type	Data type (label)
(s)	Head device for storing the transfer-source data	—	Bit	ANY_BOOL
(d)	Head device for storing the transfer-destination data	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	○	—	—	—	—	—	—
(d)	○	○	○	—	○	—	—	—	—	—	—

Processing details

- These instructions transfer bit data specified by (s) to (d).



Operation error

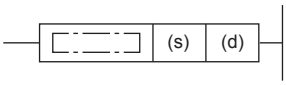
There is no operation error.

Transferring octal bits (16-bit data)

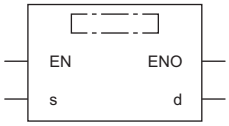
PRUN(P)

FX5S FX5UJ FX5U FX5UC

These instructions handle the device number of (s) and (d) with digit specification as octal numbers, and transfer data.

Ladder diagram	Structured text
	ENO:=PRUN(EN,s,d); ENO:=PRUNP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Digit specification*1	—	16-bit signed binary	ANY16
(d)	Device number of transfer destination*1	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Make sure that the least significant digit of a specified device number is "0".

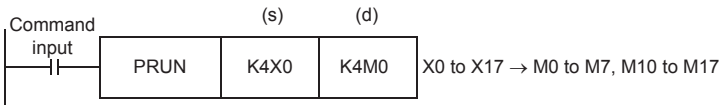
■Applicable devices

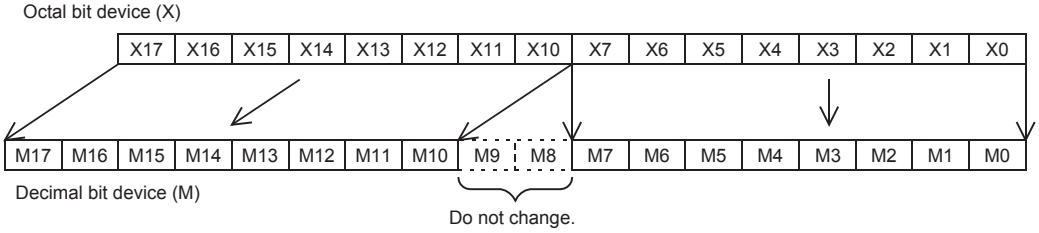
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○*1	—	—	—	—	—	○	—	—	—	—
(d)	○*1	—	—	—	—	—	○	—	—	—	—

*1 B, SB cannot be used.

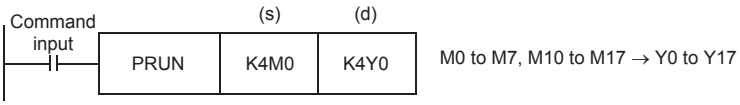
Processing details

• Octal bit device → Decimal bit device

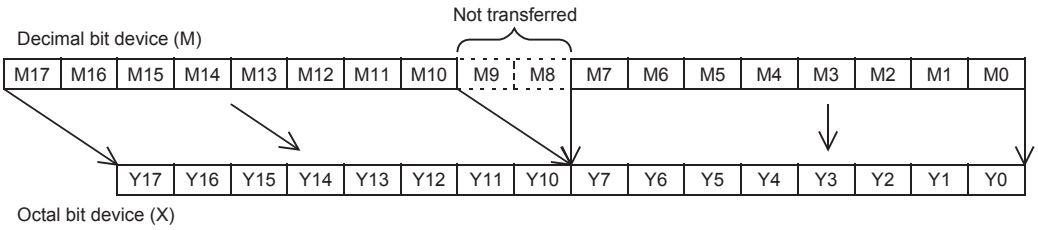




• Decimal bit device → Octal bit device



M0 to M7, M10 to M17 → Y0 to Y17



Operation error

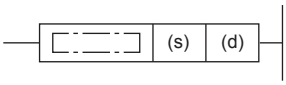
Error code (SD0/SD8067)	Description
2820H	The devices specified by (s) and (d) exceed the range of the corresponding device.

Transferring octal bits (32-bit data)

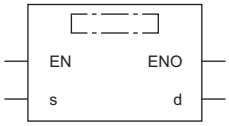
DPRUN(P)

FX5S FX5UJ FX5U FX5UC

These instructions handle the device number of (s) and (d) with digit specification as octal numbers, and transfer data.

Ladder diagram	Structured text
	ENO:=DPRUN(EN,s,d); ENO:=DPRUNP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Digit specification*1	—	32-bit signed binary	ANY32
(d)	Device number of transfer destination*1	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Make sure that the least significant digit of a specified device number is "0".

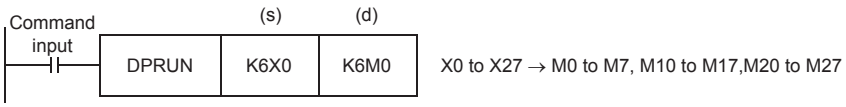
■Applicable devices

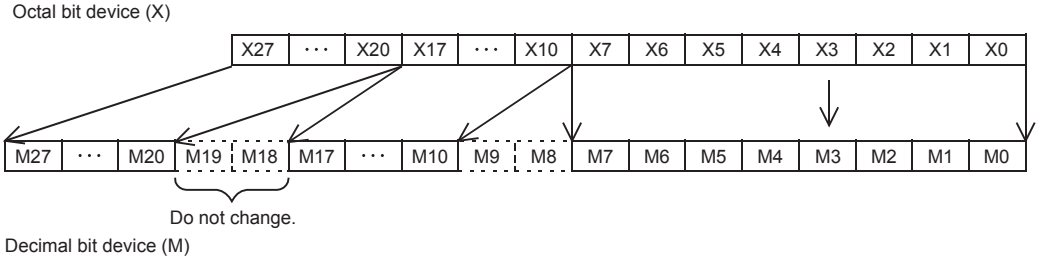
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○*1	—	—	—	—	—	○	—	—	—	—
(d)	○*1	—	—	—	—	—	○	—	—	—	—

*1 B, SB cannot be used.

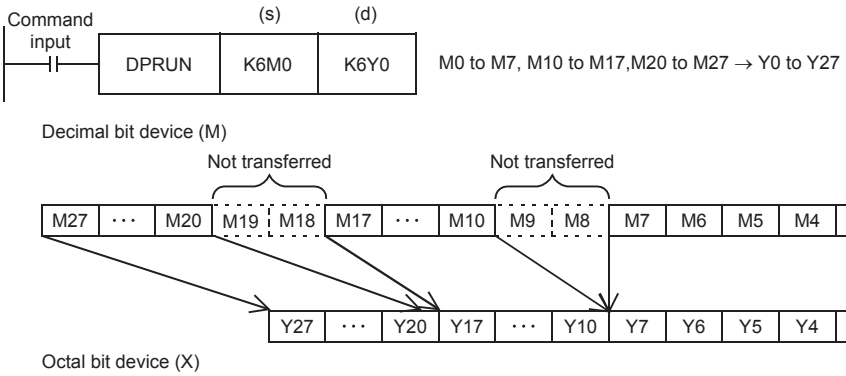
Processing details

- Octal bit device → Decimal bit device





• Decimal bit device → Octal bit device



Operation error

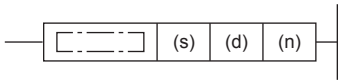
Error code (SD0/SD8067)	Description
2820H	The devices specified by (s) and (d) exceed the range of the corresponding device.

Transferring n-bit data

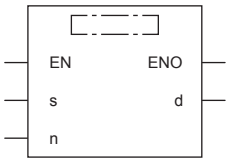
BLKMOVB(P)

FX5S FX5UJ FX5U FX5UC

These instructions block transfer the bit data of (n) point(s) from the device specified by (s) to the bit data of (n) point(s) from the device specified by (d).

Ladder diagram	Structured text
	ENO:=BLKMOVB(EN,s,n,d); ENO:=BLKMOVBP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device for storing the transfer-source bit data	—	Bit	ANY_BOOL
(d)	Head device for storing the transfer-destination bit data	—	Bit	ANY_BOOL
(n)	Number of transfers	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

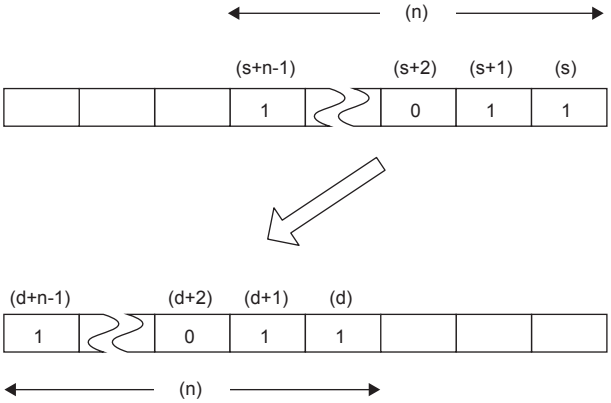
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○*1	○*2	—	—	—	—	—	—	—	—
(d)	○	○*1	○*2	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

*2 For the supported version, refer to ☞ Page 1452 Added and Changed Functions.

Processing details

- These instructions block transfer the bit data of (n) point(s) from the device specified by (s) to the bit data of (n) point(s) from the device specified by (d).
- Data can be transferred even when the device range of the transfer-source device and transfer-destination device is overlapping.



Operation error

Error code (SD0/SD8067)	Description
2820H	The range of (n) point(s) of data starting from the device specified by (s) and (d) exceed the corresponding device range.
3420H	A module access device is specified for both (s) and (d).

8 APPLICATION INSTRUCTION

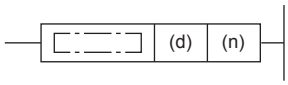
8.1 Rotation Instruction

Rotating 16-bit data to the right

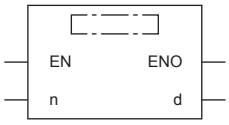
ROR(P), RCR(P)

FX5S FX5UJ FX5U FX5UC

- ROR(P): These instructions rotate the 16-bit binary data in the device specified by (d) to the right by (n) bit(s) (not including the carry flag).
- RCR(P): These instructions rotate the 16-bit binary data in the device specified by (d) to the right by (n) bit(s) (including the carry flag).

Ladder diagram	Structured text*1
	ENO:=RORP(EN,n,d); ENO:=RCR(EN,n,d); ENO:=RCRP(EN,n,d);

FBD/LD*1



*1 The ROR instruction is not supported by the ST language and the FBD/LD language. Use ROR of the standard function.
☞ Page 1320 ROR(_E)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number where the rotation target data is stored	—	16-bit signed binary	ANY16
(n)	Number of rotations	0 to 15	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

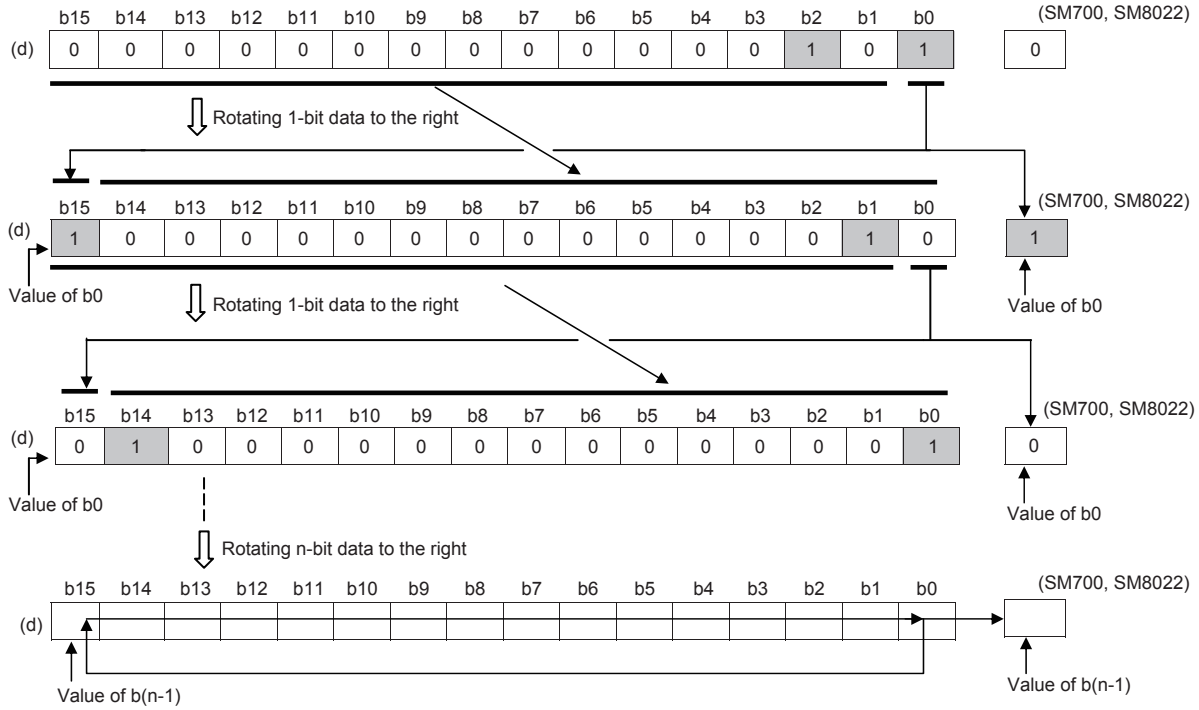
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

■ROR(P)

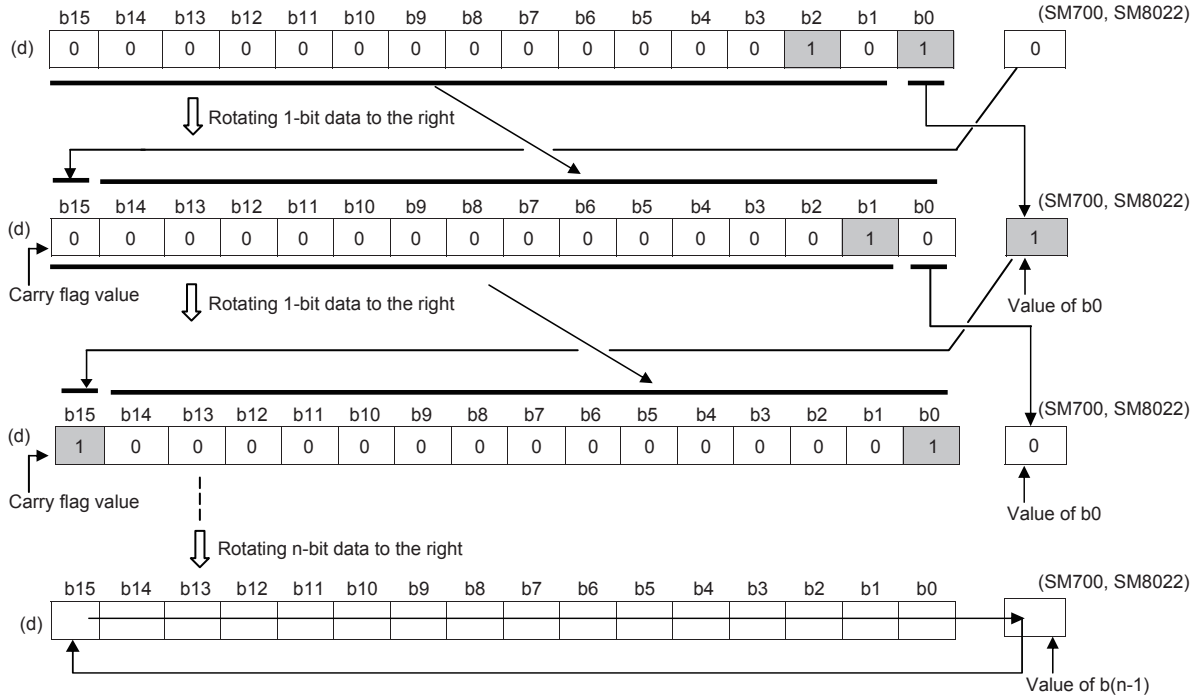
- These instructions rotate the 16-bit binary data in the device specified by (d) to the right by (n) bit(s) (not including the carry flag). The carry flag is on or off depending on the status prior to the execution of the instruction.



- When (d) is a bit device, bits are rotated to the right within the device range specified by digit specification. The number of bits actually to be rotated is the remainder of (n)÷(specified number of bits). For example, when (n) is 15 and the specified number of bits is 12, 3 bits are rotated because 15 divided by 12 equals 1 with a remainder of 3.
- Specify any value between 0 and 15 for (n). If a value 16 or bigger is specified, bits are rotated by the remainder value of n÷16. For example, when (n) is 18, 2 bits are rotated because 18 divided by 16 equals 1 with a remainder of 2.

■RCR(P)

- These instructions rotate the 16-bit binary data in the device specified by (d) to the right by (n) bit(s) (including the carry flag). The carry flag is on or off depending on the status prior to the execution of the instruction.



- When (d) is a bit device, bits are rotated to the right within the device range specified by digit specification. The number of bits actually to be rotated is the remainder of (n)÷(specified number of bits). For example, when (n) is 15 and the specified number of bits is 12, 3 bits are rotated because 15 divided by 12 equals 1 with a remainder of 3.
- Specify any value between 0 and 15 for (n). If a value 16 or bigger is specified, bits are rotated by the remainder value of n÷16. For example, when (n) is 18, 2 bits are rotated because 18 divided by 16 equals 1 with a remainder of 2.

Precautions

- Do not set a negative value to the number of bits to be rotated (n).
- In the case of continuous operation type instructions (ROR and RCR), note that shift and rotation are executed in every scan time (operation cycle).

Operation error

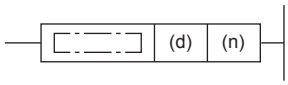
There is no operation error.

Rotating 16-bit data to the left

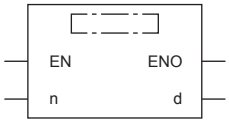
ROL(P), RCL(P)

FX5S FX5UJ FX5U FX5UC

- ROL(P): These instructions rotate the 16-bit binary data in the device specified by (d) to the left by (n) bit(s) (not including the carry flag).
- RCL(P): These instructions rotate the 16-bit binary data in the device specified by (d) to the left by (n) bit(s) (including the carry flag).

Ladder diagram	Structured text*1
	ENO:=ROLP(EN,n,d); ENO:=RCL(EN,n,d); ENO:=RCLP(EN,n,d);

FBD/LD*1



*1 The ROL instruction is not supported by the ST language and the FBD/LD language. Use ROL of the standard function.
☞ Page 1318 ROL(_E)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number where the rotation target data is stored	—	16-bit signed binary	ANY16
(n)	Number of rotations	0 to 15	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

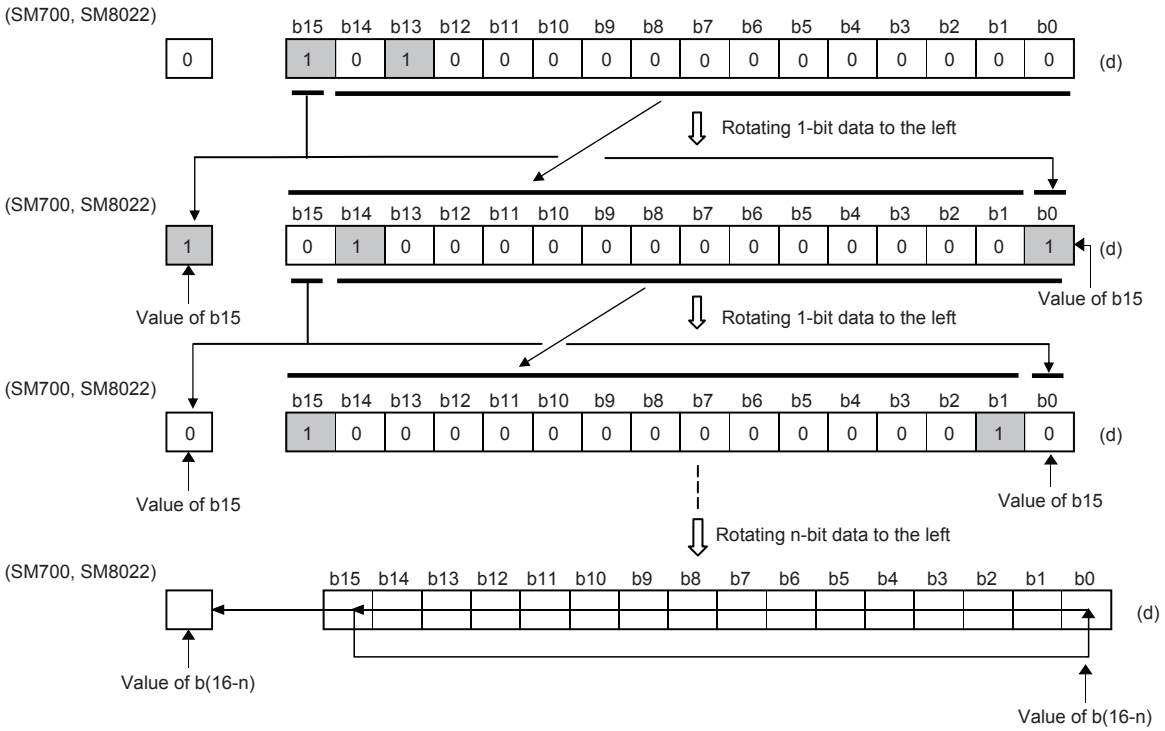
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

■ROL(P)

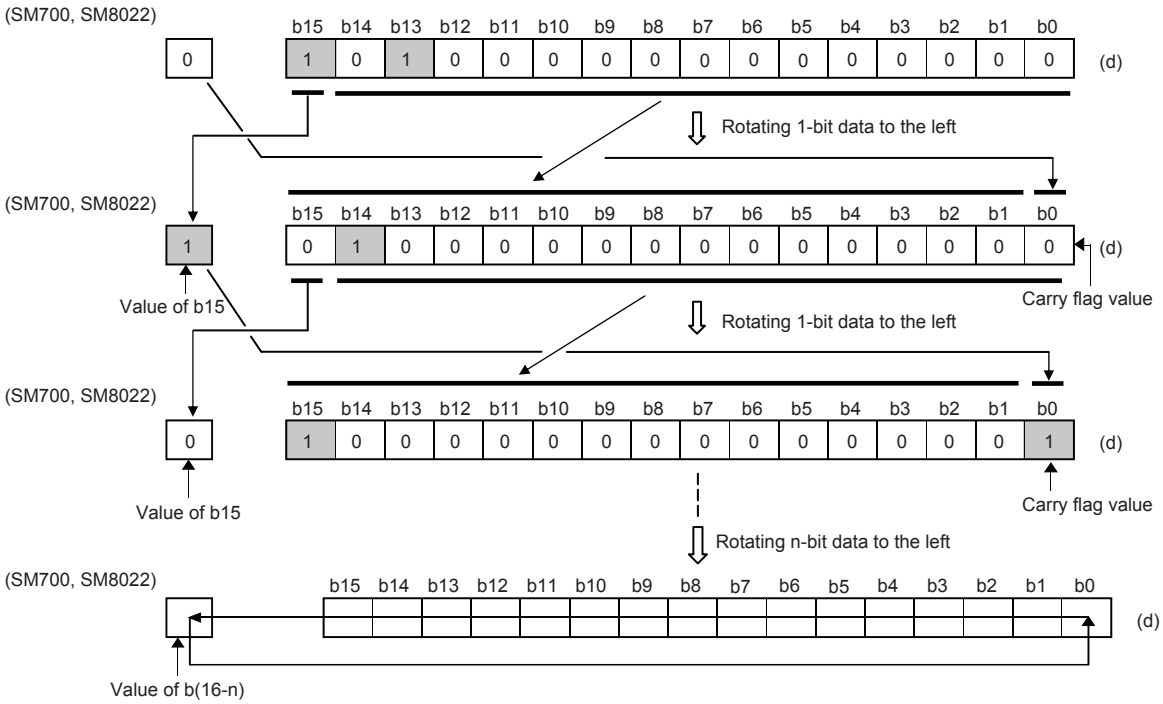
- These instructions rotate the 16-bit binary data in the device specified by (d) to the left by (n) bit(s) (not including the carry flag). The carry flag is on or off depending on the status prior to the execution of the instruction.



- When (d) is a bit device, bits are rotated to the left within the device range specified by digit specification. The number of bits actually to be rotated is the remainder of (n)÷(specified number of bits). For example, when (n) is 15 and the specified number of bits is 12, 3 bits are rotated because 15 divided by 12 equals 1 with a remainder of 3.
- Specify any value between 0 and 15 for (n). If a value 16 or bigger is specified, bits are rotated by the remainder value of n÷16. For example, when (n) is 18, 2 bits are rotated because 18 divided by 16 equals 1 with a remainder of 2.

■RCL(P)

- These instructions rotate the 16-bit binary data in the device specified by (d) to the left by (n) bit(s) (including the carry flag). The carry flag is on or off depending on the status prior to the execution of the instruction.



- When (d) is a bit device, bits are rotated to the left within the device range specified by digit specification. The number of bits actually to be rotated is the remainder of (n)÷(specified number of bits). For example, when (n) is 15 and the specified number of bits is 12, 3 bits are rotated because 15 divided by 12 equals 1 with a remainder of 3.
- Specify any value between 0 and 15 for (n). If a value 16 or bigger is specified, bits are rotated by the remainder value of n÷16. For example, when (n) is 18, 2 bits are rotated because 18 divided by 16 equals 1 with a remainder of 2.

Precautions

- Do not set a negative value to the number of bits to be rotated (n).
- In the case of continuous operation type instructions (ROL and RCL), note that shift and rotation are executed in every scan time (operation cycle).

Operation error

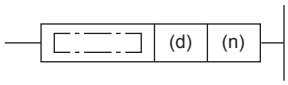
There is no operation error.

Rotating 32-bit data to the right

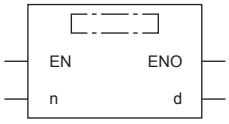
DROR(P), DRCR(P)

FX5S FX5UJ FX5U FX5UC

- DROR(P): These instructions rotate the 32-bit binary data in the device specified by (d) to the right by (n) bit(s) (not including the carry flag).
- DRCR(P): These instructions rotate the 32-bit binary data in the device specified by (d) to the right by (n) bit(s) (including the carry flag).

Ladder diagram	Structured text*1
	ENO:=DRORP(EN,n,d); ENO:=DRCR(EN,n,d); ENO:=DRCRP(EN,n,d);

FBD/LD*1



*1 The DROR instruction is not supported by the ST language and the FBD/LD language. Use ROR of the standard function.
☞ Page 1320 ROR(_E)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number where the rotation target data is stored	—	32-bit signed binary	ANY32
(n)	Number of rotations	0 to 31	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

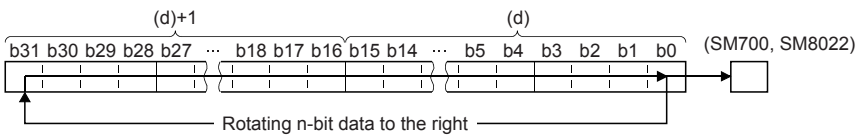
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

■DROR(P)

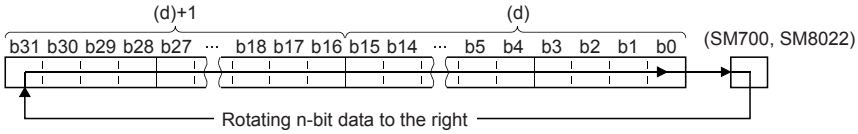
- These instructions rotate the 32-bit binary data in the device specified by (d) to the right by (n) bit(s) (not including the carry flag). The carry flag is on or off depending on the status prior to the execution of the instruction.



- When (d) is a bit device, bits are rotated to the right within the device range specified by digit specification. The number of bits actually to be rotated is the remainder of (n)÷(specified number of bits). For example, when (n) is 31 and the specified number of bits is 24, 7 bits are rotated because 31 divided by 24 equals 1 with a remainder of 7.
- Specify any value between 0 and 31 for (n). If a value 32 or bigger is specified, bits are rotated by the remainder value of n÷32. For example, when (n) is 34, 2 bits are rotated because 34 divided by 32 equals 1 with a remainder of 2.

■DRCR(P)

- These instructions rotate the 32-bit binary data in the device specified by (d) to the right by (n) bit(s) (including the carry flag). The carry flag is on or off depending on the status prior to the execution of the instruction.



- When (d) is a bit device, bits are rotated to the right within the device range specified by digit specification. The number of bits actually to be rotated is the remainder of (n)÷(specified number of bits). For example, when (n) is 31 and the specified number of bits is 24, 7 bits are rotated because 31 divided by 24 equals 1 with a remainder of 7.
- Specify any value between 0 and 31 for (n). If a value 32 or bigger is specified, bits are rotated by the remainder value of n÷32. For example, when (n) is 34, 2 bits are rotated because 34 divided by 32 equals 1 with a remainder of 2.

Precautions

- Do not set a negative value to the number of bits to be rotated (n).
- In the case of continuous operation type instructions (DROR and DRCR), note that shift and rotation are executed in every scan time (operation cycle).

Operation error

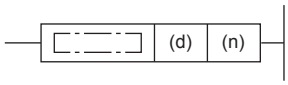
There is no operation error.

Rotating 32-bit data to the left

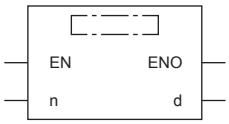
DROL(P), DRCL(P)

FX5S FX5UJ FX5U FX5UC

- DROL(P): These instructions rotate the 32-bit binary data in the device specified by (d) to the left by (n) bit(s) (not including the carry flag).
- DRCL(P): These instructions rotate the 32-bit binary data in the device specified by (d) to the left by (n) bit(s) (including the carry flag).

Ladder diagram	Structured text*1
	ENO:=DROLP(EN,n,d); ENO:=DRCL(EN,n,d); ENO:=DRCLP(EN,n,d);

FBD/LD*1



*1 The DROL instruction is not supported by the ST language and the FBD/LD language. Use ROL of the standard function.
☞ Page 1318 ROL(_E)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number where the rotation target data is stored	—	32-bit signed binary	ANY32
(n)	Number of rotations	0 to 31	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

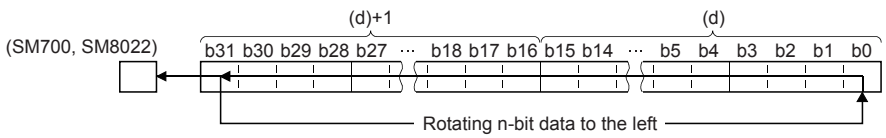
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

■DROL(P)

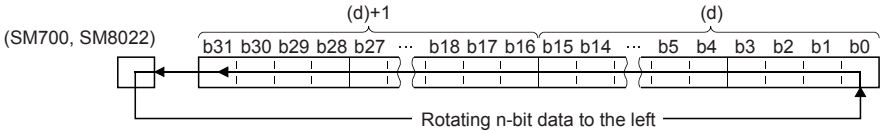
- These instructions rotate the 32-bit binary data in the device specified by (d) to the left by (n) bit(s) (not including the carry flag). The carry flag is on or off depending on the status prior to the execution of the instruction.



- When (d) is a bit device, bits are rotated to the left within the device range specified by digit specification. The number of bits actually to be rotated is the remainder of (n)÷(specified number of bits). For example, when (n) is 31 and the specified number of bits is 24, 7 bits are rotated because 31 divided by 24 equals 1 with a remainder of 7.
- Specify any value between 0 and 31 for (n). If a value 32 or bigger is specified, bits are rotated by the remainder value of n÷32. For example, when (n) is 34, 2 bits are rotated because 34 divided by 32 equals 1 with a remainder of 2.

■DRCL(P)

- These instructions rotate the 32-bit binary data in the device specified by (d) to the left by (n) bit(s) (including the carry flag). The carry flag is on or off depending on the status prior to the execution of the instruction.



- When (d) is a bit device, bits are rotated to the left within the device range specified by digit specification. The number of bits actually to be rotated is the remainder of (n)÷(specified number of bits). For example, when (n) is 31 and the specified number of bits is 24, 7 bits are rotated because 31 divided by 24 equals 1 with a remainder of 7.
- Specify any value between 0 and 31 for (n). If a value 32 or bigger is specified, bits are rotated by the remainder value of n÷32. For example, when (n) is 34, 2 bits are rotated because 34 divided by 32 equals 1 with a remainder of 2.

Precautions

- Do not set a negative value to the number of bits to be rotated (n).
- In the case of continuous operation type instructions (DROL and DRCL), note that shift and rotation are executed in every scan time (operation cycle).

Operation error

There is no operation error.

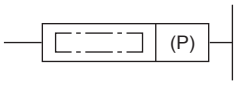
8.2 Program Branch Instruction

Pointer branch

CJ(P)

FX5S FX5UJ FX5U FX5UC

These instructions execute the program specified by the pointer number within the same program file when the jump command is on.

Ladder diagram	Structured text
	Not supported

FBD/LD
Not supported.

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(P)	Pointer number of the jump destination	—	Device name	POINTER

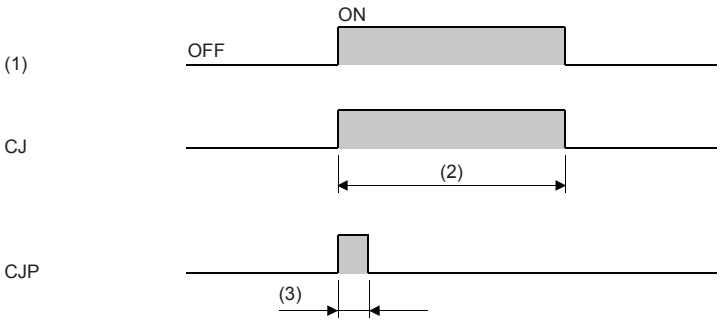
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(P)	—	—	—	—	—	—	—	—	—	—	○

Processing details

■CJ(P)

- These instructions execute the program specified by the pointer number when the execution command is on.
- When the execution command is off, the program in the next step is executed.

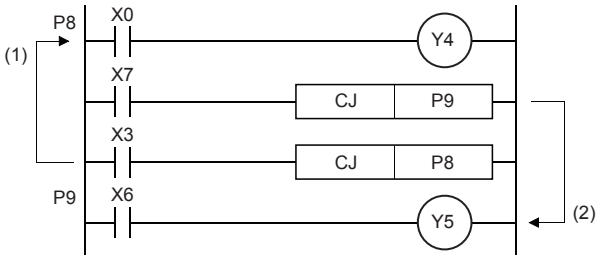


(1): Execution command
(2): Executed in each scan
(3): One Scan

Precautions

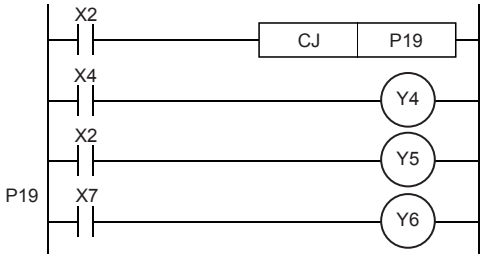
- If the timer with its coil on is skipped by these instructions, time cannot be measured correctly.
- If the OUT instruction is skipped by these instructions, the scan time will be shortened.
- If these instructions specify and jump to a later step, the scan time will be shortened.

- These instructions can specify and jump from the current step to a smaller step number. In this case, consider a method to exit a loop so that the watchdog timer does not time out.



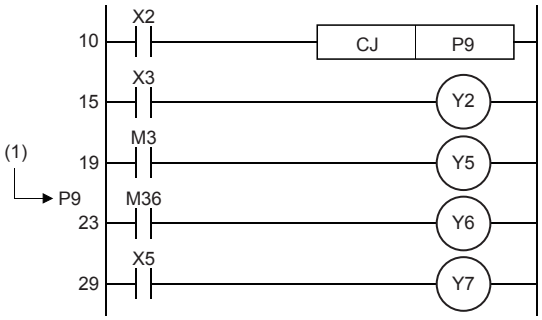
(1) While X3 is on, the loop is repeated.
(2) To exit the loop, turn on X7.

- The value in the device skipped with these instructions remains the same.



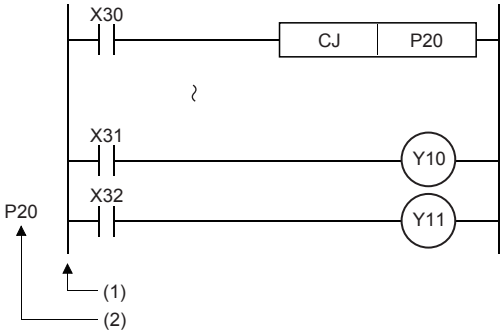
When X2 turns on, the program jumps to the label, P19.
Y4 and Y5 remain the same even if X2 and X4 turn on/off during the execution of the CJ instruction.

- A label (P□) occupies two steps.



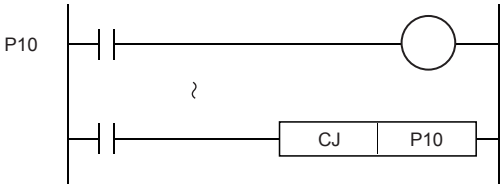
(1) A label occupies two steps.

- Only the pointer numbers within the same program file can be specified.
- During skip operation, if the program jumps to the pointer number within the skip range, the programs of the jump destination pointer number and later are executed.
- The figure below shows programming of a label. When creating a circuit program, move the cursor to the left side of the bus line in the ladder diagram, and input a label (P) at the head of the circuit block.

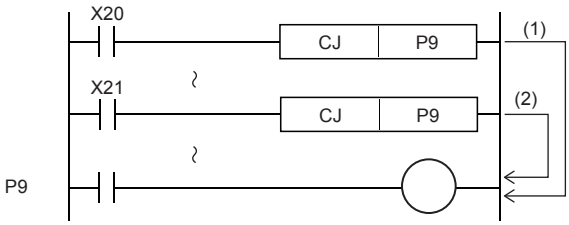


(1): Bus line
(2): Label

- A label can be programmed in a smaller number step than CJ instruction. However, note that a watchdog timer error occurs when the scan time exceeds 200 ms (default setting).

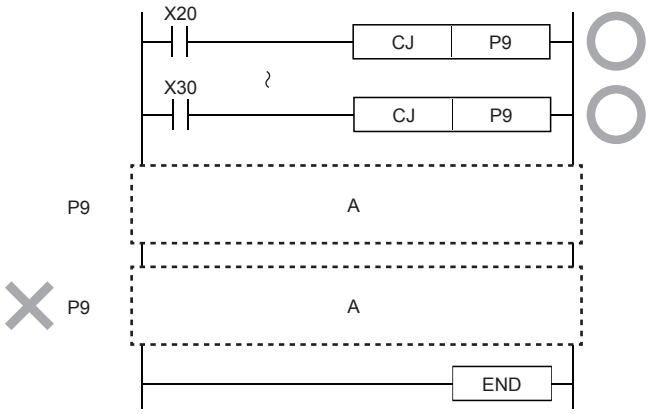


• When the pointer number in operands is same and there is one label, the following operation is caused:



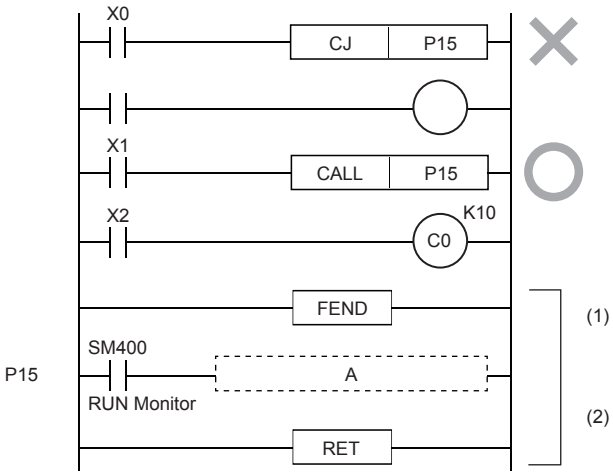
(1) When X20 turns ON, the program execution jumps from CJ instruction corresponding to X20 to the label P9.
(2) When X20 turns OFF and X21 turns ON, the program execution jumps from CJ instruction corresponding to X21 to the label P9.

• When a label number (including labels for CALL instructions described later) is used two or more times, an error is caused.



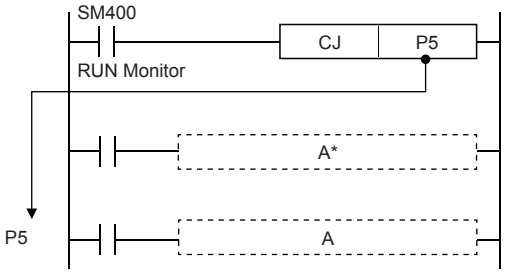
A: User program

• No label can be shared by CALL instruction and CJ instruction.



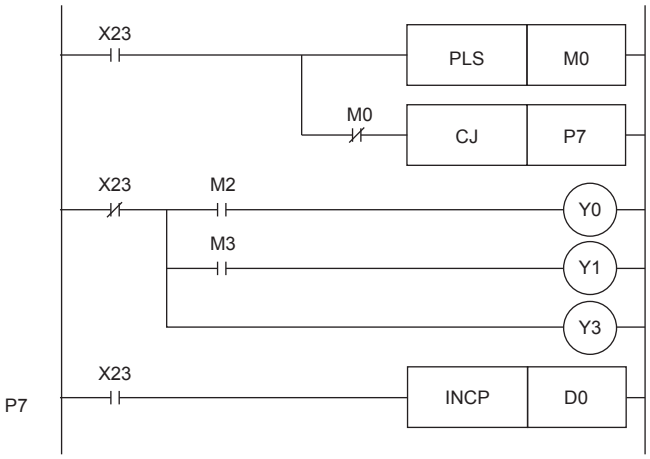
A: User program
(1): Subroutine program dedicated to CALL instruction
(2): Program a label (P) after FEND instruction.

• Because SM400/SM8000 is normally ON while the CPU module is operating, unconditional jump is applied when SM400 is used as shown in the following example:



A*: User program (It is skipped, and is not executed.)
A: User program

- In one operation cycle after X23 changes from OFF to ON, the CJ instruction becomes valid. By using this method, jump can be executed after all outputs between the CJ instruction and the label P7 turn OFF.



Operation error

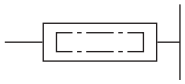
Error code (SD0/SD8067)	Description
3380H	A pointer number which is not used as a label in the same program file is specified.

Jump to END

GOEND

FX5S FX5UJ FX5U FX5UC

This instruction moves the program execution to the FEND or END instruction in the same program file.

Ladder diagram	Structured text
	Not supported

FBD/LD
Not supported.

Processing details

- This instruction moves the program execution to the FEND or END instruction in the same program file.

Precautions

- When a GOEND instruction is executed by invalid jump during interrupt program execation, it becomes the same operation as the IRET instruction.

Operation error

Error code (SD0/SD8067)	Description
3340H	After the FOR instruction is executed, the GOEND instruction is executed before the NEXT instruction is executed.
3381H	After the CALL(P) or XCALL instruction is executed, the GOEND instruction is executed before the RET instruction is executed.

8.3 Program Execution Control Instruction

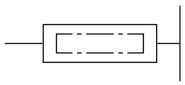
Disabling/enabling interrupt programs

DI, EI

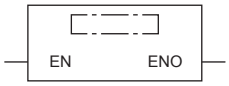
FX5S FX5UJ FX5U FX5UC

Interrupts are usually disabled in CPU module. These instructions enable interrupts in CPU module (EI instruction) or disable interrupts again (DI instruction).

- DI: Disables the execution of the interrupt program.
- EI: Releases the execution disabled state of interrupt programs.

Ladder diagram	Structured text
	ENO:=DI(EN); ENO:=EI(EN);

FBD/LD



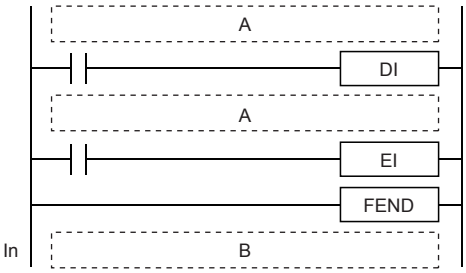
Processing details

■DI

- This instruction disables the execution of the interrupt program until the EI instruction is executed, even if the interrupt cause occurs.
- When the power is turned on or the CPU module is reset, the state in which the DI instruction is executed is applied.
- For the operation of the DI instruction (DI instruction without an argument) when using the interrupt disable instruction with a specified priority or lower (DI instruction with an argument), refer to ☞ Page 443 Disabling the interrupt program with specified priority or lower.

■EI

- This instruction releases the execution disabled state of interrupt programs when the DI instruction is executed, and enables the execution of the interrupt program with the interrupt pointer number enabled by the IMASK instruction.
- For the operation of the EI instruction when using the interrupt disable instruction with a specified priority or lower (DI instruction with an argument), refer to ☞ Page 443 Disabling the interrupt program with specified priority or lower.



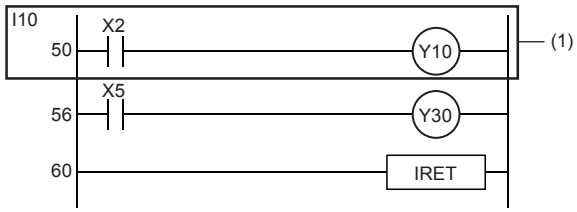
A: Sequence program

B: Interrupt Program

Even though an interrupt occurs between the DI and EI instructions, the execution of the interrupt is held until the processing between the instructions ends.

Point

- An interrupt pointer occupies two steps. (In (1) below, I10 is the step 50, X2 is the step 52, and Y10 is the step 54.)



- If the master control contains the EI or DI instruction, such an instruction is executed regardless of the execution of the MC instruction.

Precautions

Interrupts (requests) that are generated after the DI instruction execution, are processed after the EI instruction is executed.

Operation error

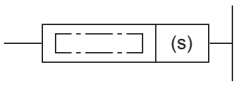
Error code (SD0/SD8067)	Description
3362H	Nesting of the DI instruction exceeds 16 levels.

Disabling the interrupt program with specified priority or lower

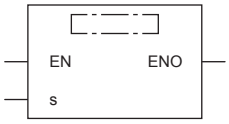
DI

FX5S FX5UJ FX5U FX5UC

This instruction disables the execution of the interrupt program with a priority specified by (s) or lower until the EI instruction is executed, even if the interrupt cause occurs.

Ladder diagram	Structured text
	ENO:=DI_1(EN,s);

FBD/LD



("DI_1" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Priority for disabling interrupts	1 to 3	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

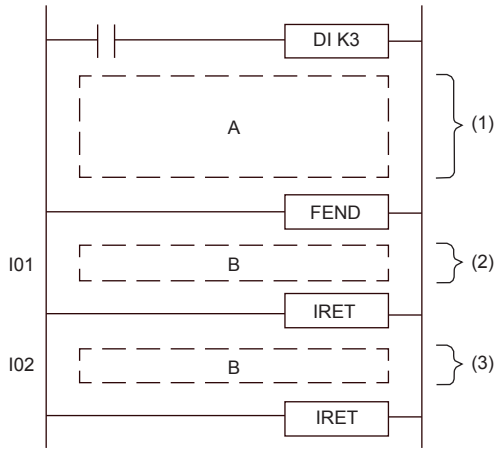
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- This instruction disables the execution of the interrupt program of the interrupt pointer number with an interrupt priority specified by (s) or lower.

Interrupt priority setting

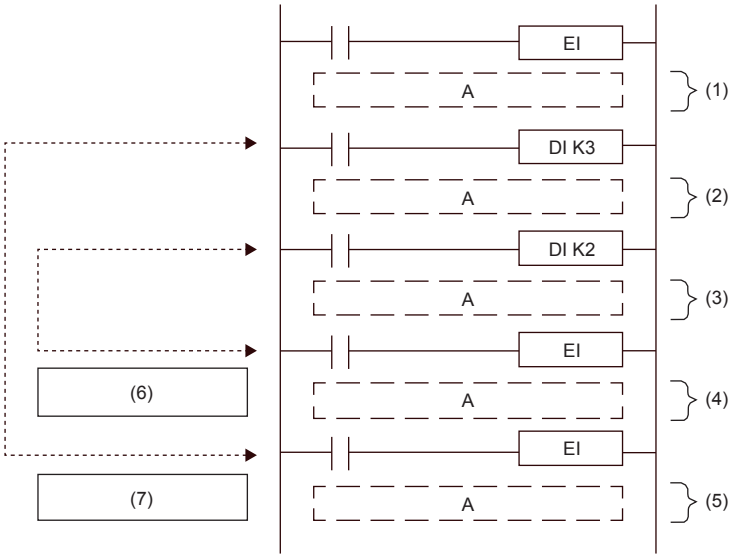
I No.	Priority
I01	2
I02	3



(1): Interrupt-disabled section for the priority 3 or lower (Interrupt-enabled section for the priority 2 or higher)
(2): Can be executed because of the priority 2.
(3): Cannot be executed because of the priority 3.

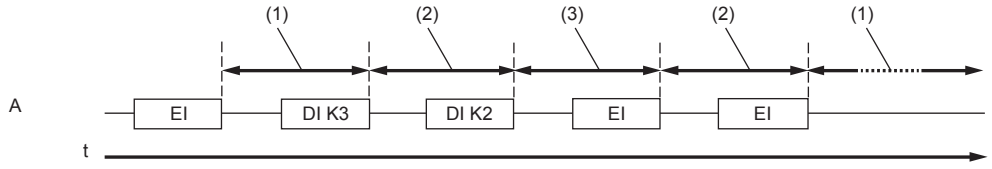
A: Sequence program
B: Interrupt Program

- By executing the EI instruction, the interrupt with the priority disabled by the counterpart DI instruction is enabled. However, when interrupts are disabled only with the DI instruction without an argument, interrupts with all the priorities are enabled by executing the EI instruction once.



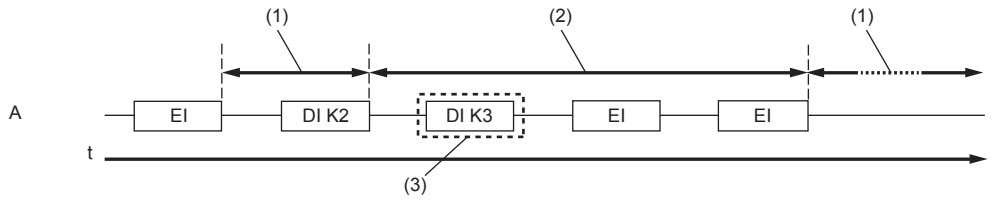
A: Sequence program
(1): Interrupt-enabled section for all priority
(2): Interrupt-disabled section for the priority 3 or lower (Interrupt-enabled section for the priority 2 or higher)
(3): Interrupt-disabled section for the priority 2 or lower (Interrupt-enabled section for the priority 1 or higher)
(4): Interrupt-disabled section for the priority 3 or lower (Interrupt-enabled section for the priority 2 or higher)
(5): Interrupt-enabled section for all priority
(6): EI, which is the counterpart of [DI K2]
(7): EI, which is the counterpart of [DI K3]

- Interrupts (requests) that are generated after the DI instruction are processed after the EI instruction is executed.
- When multiple DI instructions are executed and the argument has a priority higher than the currently disabled priority, interrupts with a priority lower than that of the argument are disabled.
- When multiple DI instructions are executed and the argument has a priority lower than the currently disabled priority, the interrupt disabled state is not changed.
- The DI instruction can be nested in up to 16 levels.
- The interrupt priority of the interrupt pointer can be set with parameters. (MELSEC iQ-F FX5 User's Manual (Application))
- The interrupt-disabled priority can be checked with SD758 (interrupt-disabling priority setting value).
- The following shows the interrupt-disabled section when the DI or EI instruction is executed.
  - When multiple DI instructions are executed (when interrupts with a priority higher than the currently disabled priority are specified and disabled)



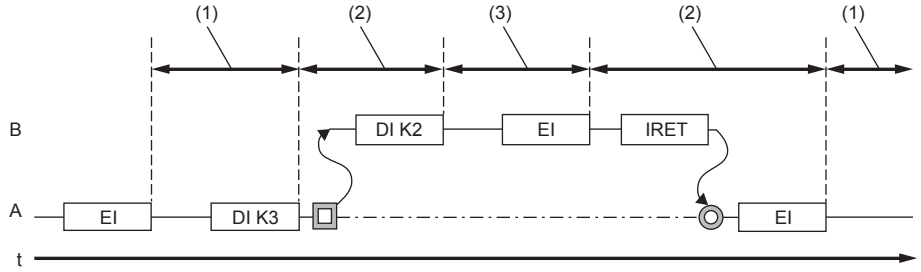
A: Scan execution type program
(1): Interrupt-enabled section for all priority
(2): Interrupt-disabled section for the priority 3 or lower (interrupt-enabled section for the priority 2 or higher)
(3): Interrupt-disabled section for the priority 2 or lower (interrupt-enabled section for the priority 1 or higher)

  - When multiple DI instructions are executed (when interrupts with a priority lower than the currently disabled priority are specified and disabled)



A: Scan execution type program
(1): Interrupt-enabled section for all priority
(2): Interrupt-disabled section for the priority 2 or lower (interrupt-enabled section for the priority 1 or higher)
(3): Because interrupts with the priority 2 or lower are already disabled, the interrupt-disabling priority is not changed.

• When the DI instruction is executed in an interrupt program



A: Scan execution type program

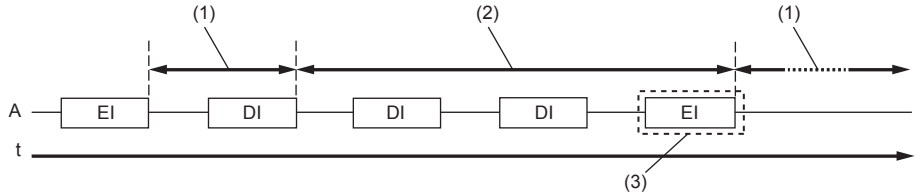
B: Interrupt Program

(1): Interrupt-enabled section for all priority

(2): Interrupt-disabled section for the priority 3 or lower (interrupt-enabled section for the priority 2 or higher)

(3): Interrupt-disabled section for the priority 2 or lower (interrupt-enabled section for the priority 1 or higher)

• When the DI instruction without an argument is executed



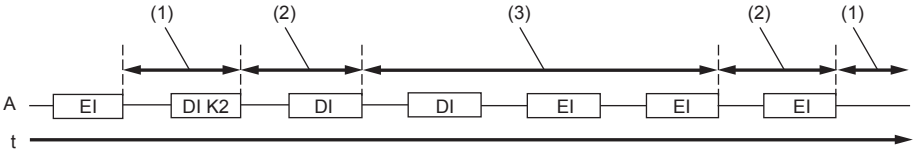
A: Scan execution type program

(1): Interrupt-enabled section for all priority

(2): Interrupt-disabled section for the priority 1 or lower (where all the interrupts are disabled)

(3): Because interrupts are disabled with the DI instruction without an argument, interrupts with all the priorities are enabled by executing the EI instruction once.

• When the DI instructions with and without an argument are executed (Execution order is DI instruction with an argument → DI instruction without an argument)



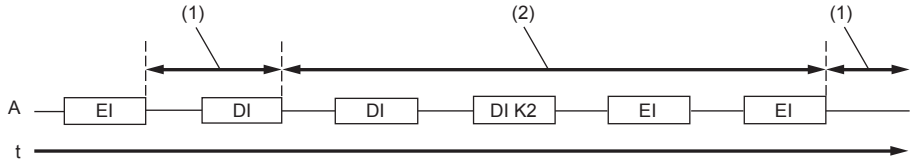
A: Scan execution type program

(1): Interrupt-enabled section for all priority

(2): Interrupt-disabled section for the priority 2 or lower (interrupt-enabled section for the priority 1 or higher)

(3): Interrupt-disabled section for the priority 1 or lower (where all the interrupts are disabled)

• When the DI instructions with and without an argument are executed (Execution order is DI instruction without an argument → DI instruction with an argument)



A: Scan execution type program

(1): Interrupt-enabled section for all priority

(2): Interrupt-disabled section for the priority 1 or lower (where all the interrupts are disabled)

Operation error

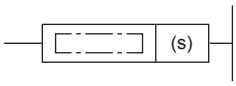
Error code (SD0/SD8067)	Description
3405H	The value specified by (s) is other than the following. 1 to 3
3362H	Nesting of the DI instruction exceeds 16 levels.

Interrupt program mask

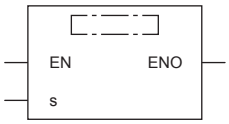
IMASK

FX5S FX5UJ FX5U FX5UC

This instruction enables or disables the execution of the interrupt program with the specified interrupt pointer number according to the 16-point bit pattern starting from the device specified in (s).

Ladder diagram	Structured text
	ENO:=IMASK(EN,s);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number where the interrupt mask data is stored The device specified in (s) and following 11 devices are used.	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 12)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—

Processing details

- This instruction enables or disables the execution of the interrupt program with the specified interrupt pointer number according to the 16-point bit pattern starting from the device specified in (s).
  - 1 (ON): The execution of interrupt programs is enabled.
  - 0 (OFF): The execution of interrupt programs is disabled.

• The following shows the assignment of the interrupt pointer numbers to each bit.

	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	-	-	-	-	I23	I22	I21	I20	I19	I18	I17	I16
(s)+2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(s)+3	I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	-	-
(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	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I177	I176

• When the power is turned on or the CPU module is reset, execution of the interrupt programs I0 to I177 is enabled.
• The states of the device (s) to (s)+11 are stored in SD1400 to SD1411 (IMASK instruction mask pattern).

Point

The IMASK instruction can enable or disable the interrupt pointers I0 to I177 in a batch.

Operation error

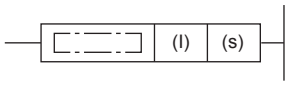
Error code (SD0/SD8067)	Description
2820H	The 16-point range starting from the device specified by (s) exceeds the corresponding device range.

Disabling/enabling the specified interrupt pointer

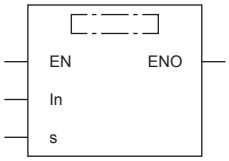
SIMASK

FX5S FX5UJ FX5U FX5UC

This instruction enables or disables the interrupt pointer number specified by (I) according to the value of (s).

Ladder diagram	Structured text
	ENO:=SIMASK(EN,In,s);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(I)*1	Interrupt pointer number for which interrupts are enabled or disabled	■FX5S CPU module I0 to I31 ■FX5UJ/FX5U/FX5UC CPU module I0 to I177	Device name	POINTER
(s)	Enabled or disabled state of the specified interrupt pointer number	0: Disabled 1: Enabled	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, I displays as In.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(I)	—	—	—	—	—	—	—	—	—	—	○
(s)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- This instruction enables or disables the execution of the interrupt program with the interrupt pointer number specified by (I) according to the data specified by (s)
- When 1 is set in (s): The execution of the interrupt program is enabled.
- When 0 is set in (s): The execution of the interrupt program is disabled.
- When the power is turned on or the CPU module is reset, execution of the interrupt programs is enabled.

■FX5S CPU module
Interrupt program: I0 to I31
■FX5UJ/FX5U/FX5UC CPU module
Interrupt program: I0 to I177

• The execution-enabled/disabled states of interrupt pointers are stored in the IMASK instruction mask pattern.

■FX5S CPU module

IMASK instruction mask pattern: SD1400 to SD1401

■FX5UJ/FX5U/FX5UC CPU module

IMASK instruction mask pattern: SD1400 to SD1411

Operation error

Error code (SD0/SD8067)	Description
3405H	The interrupt pointer number specified by (I) exceeds the range of the interrupt pointer number (I0 to I177).
	The value in (s) is other than the interrupt disabled (0) or interrupt enabled (1).

Returning from the interrupt program

IRET

FX5S FX5UJ FX5U FX5UC

This instruction indicates an end of the processing of an interrupt program.

Ladder diagram	Structured text
	Not supported

FBD/LD
Not supported.

Processing details

When an interrupt (input or timer) is generated while the main program is executing, the program execution jumps to an interrupt (I) routine. The IRET instruction returns the program execution to the main routine.
The table below shows three types of jump to an interrupt routine.

Function	Interrupt No.	Description
Interrupt from inputs (including counter)	I0 to I23	Interrupt pointer used for the CPU built-in functions (such as input interrupt, high-speed comparison match interrupt)
Interrupt setting from internal timer	I28 to I31	Interrupt pointer used for fixed-cycle interrupts of the internal timer
Interrupt from module*1	I50 to I177	Interrupt pointer used with module equipped with interruption function

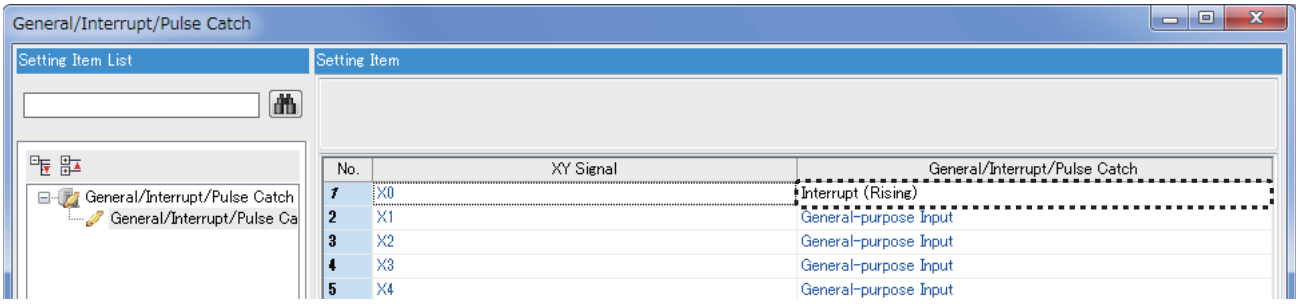
*1 FX5S does not support this function.

Program example

• Parameter setting

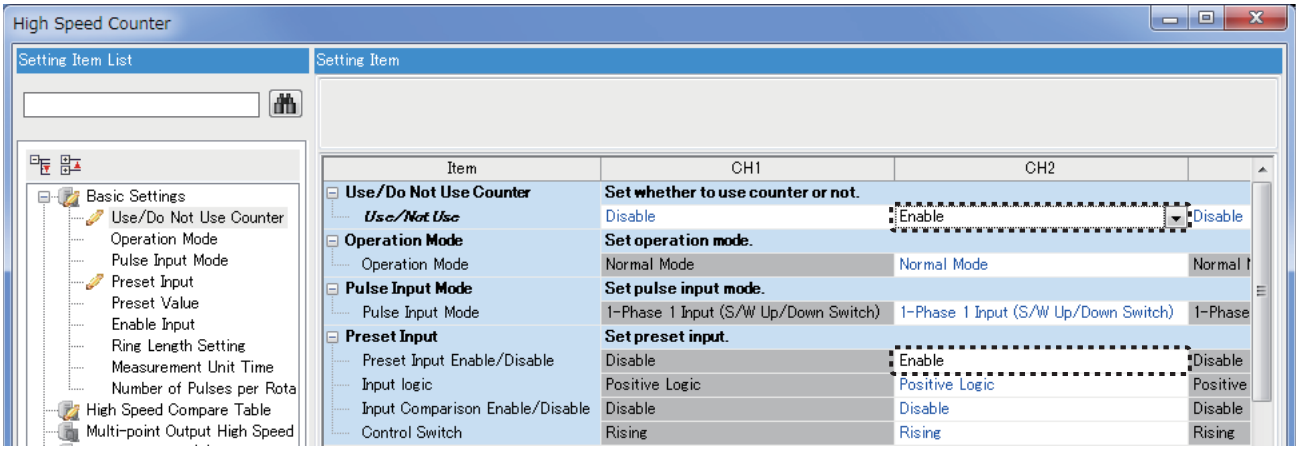
When using the input interrupt (detection of rising of X0), the high-speed comparison match interrupt (the high-speed counter CH2 reaches 10) and the internal timer interrupt (20 ms cycle), set interrupt and the high-speed counter by the parameter.
[Input interrupt]

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [High Speed I/O] ⇨ "Input Function" ⇨ "General/Interrupt/Pulse Catch" ⇨ "Detailed Setting"

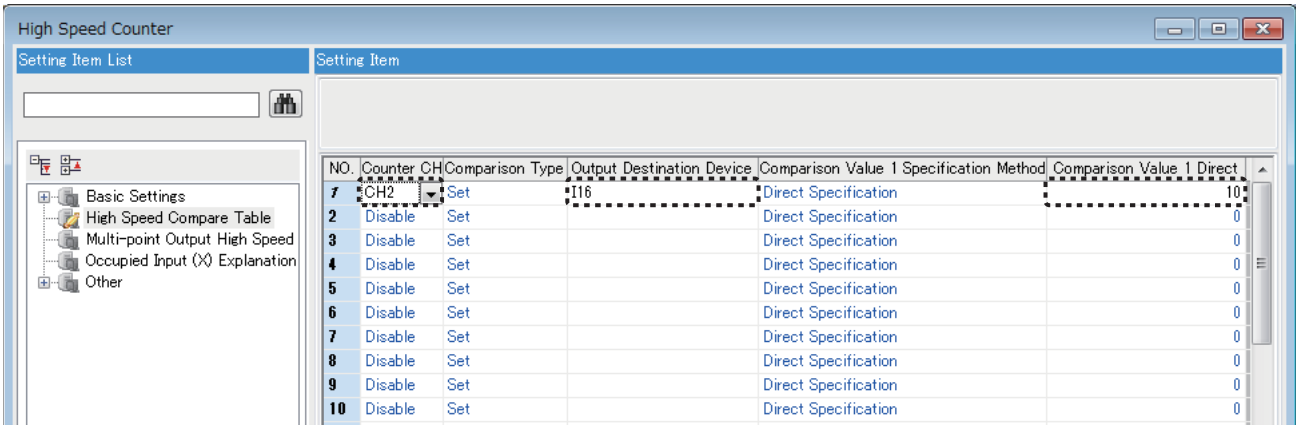


[high-speed comparison match interrupt]

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [High Speed I/O] ⇨ "Input Function" ⇨ "High Speed Counter" ⇨ "Detailed Setting" ⇨ "Basic Settings"



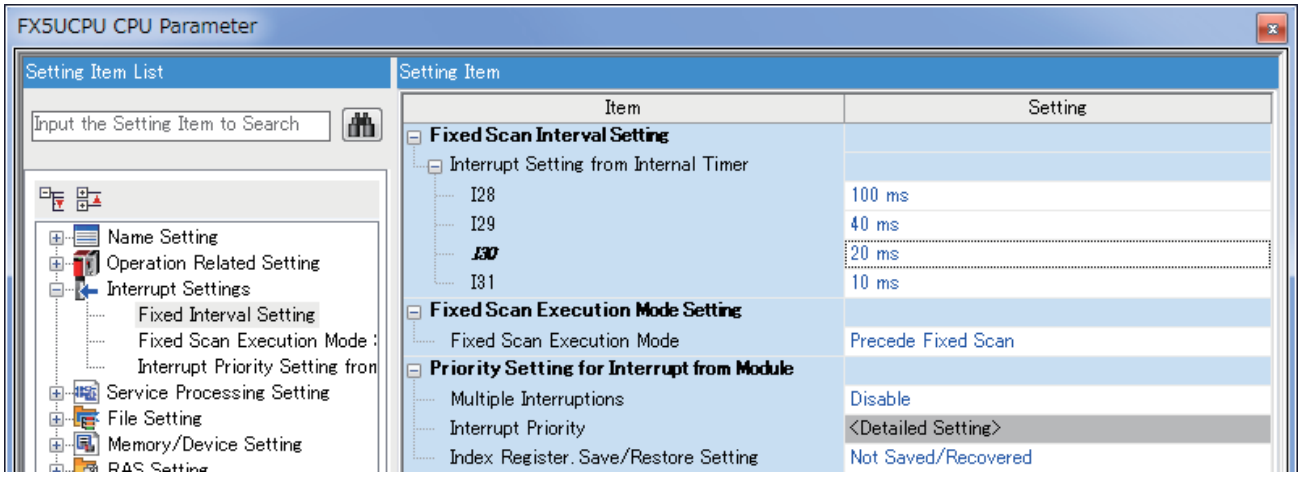
Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [High Speed I/O] ⇨ "Input Function" ⇨ "High Speed Counter" ⇨ "Detailed Setting" ⇨ "High Speed Compare Table"



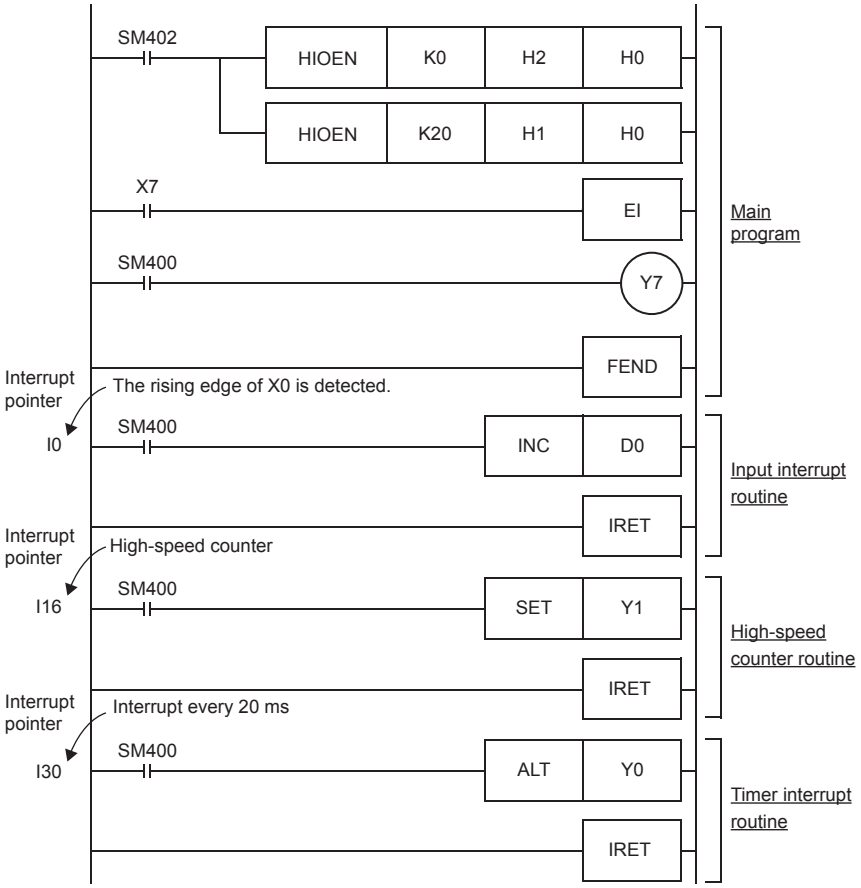
[Interrupt Settings from Internal Timer]

The parameter is used in default setting. To change the parameter, set it as follows.

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [CPU Parameter] ⇨ "Interrupt Settings" ⇨ "Fixed Scan Interval Setting" ⇨ "Interrupt Setting from Internal Timer"



• Program



Interrupts are usually disabled in CPU modules.
Turn ON X7 and use EI instruction to enable interrupts.
When X0 turns ON while the main program is executed, instructions after the input interrupt routine pointer I0 are executed, and the program execution returns to the original main program by IRET instruction.

Make sure to program an interrupt pointer (I***) as a label after FEND instruction.

The high-speed counter interrupt of the pointer I16 is executed when the current value of a high-speed counter becomes equivalent to a value specified by the parameter. The program execution returns to the original main program by IRET instruction.

The timer interrupt of the pointer I30 is executed every timer time of 20 ms, and the program execution is returned to the original main program by IRET instruction each time.

Operation error

Error code (SD0/SD8067)	Description
33E6H	The IRET instruction is executed in the main program.

Resetting the watchdog timer

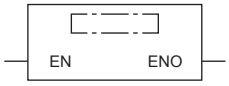
WDT(P)

FX5S FX5UJ FX5U FX5UC

These instructions reset the watchdog timer in a program.

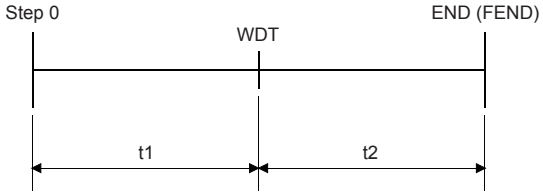
Ladder diagram	Structured text
	ENO:=WDT(EN); ENO:=WDTP(EN);

FBD/LD



Processing details

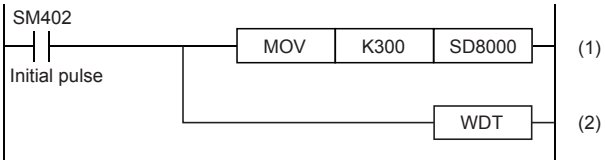
- These instructions reset the watchdog timer in a program.
- These instructions are used when the scan time exceeds the value set for the watchdog timer depending on the condition. If the scan time exceeds the value set for the watchdog timer every scan, change the setting of the watchdog timer in the parameter setting of the engineering tool.
- Design a program so that t1 from the step 0 to the WDT(P) instruction and t2 from the WDT(P) instruction to the END (FEND) instruction do not exceed the setting value of the watchdog timer.



- The WDT(P) instruction can be used more than once in one scan. However, note that turning off the output takes some time if an error occurs.

Precautions

- The time of the watchdog timer can be changed in the [RAS] tab of [CPU Parameter]. The default value is 200 ms.
- By overwriting the contents of SD8000 (watchdog timer time), the watchdog timer detection time can be changed using a program. When the program shown below is input, the sequence program will be monitored with the new watchdog timer time.



(1): Watchdog timer time: 300 ms

(2): Refreshes the watchdog timer.
If the WDT instruction is not programmed, the value of SD8000 is valid during END processing.

Operation error

There is no operation error.

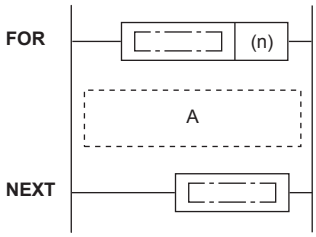
8.4 Structuring Instruction

FOR to NEXT

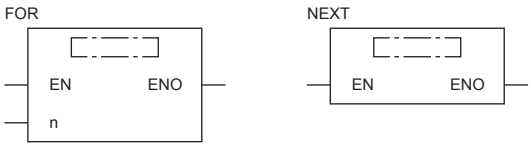
FOR, NEXT

FX5S FX5UJ FX5U FX5UC

When the processing between the FOR and NEXT instructions is executed (n) times without any condition, the processing of the step following the NEXT instruction is executed.

Ladder diagram	Structured text
 A: Repetition program	Not supported

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(n)	Number of repetitions of the loop between FOR and NEXT instructions	1 to 32767	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

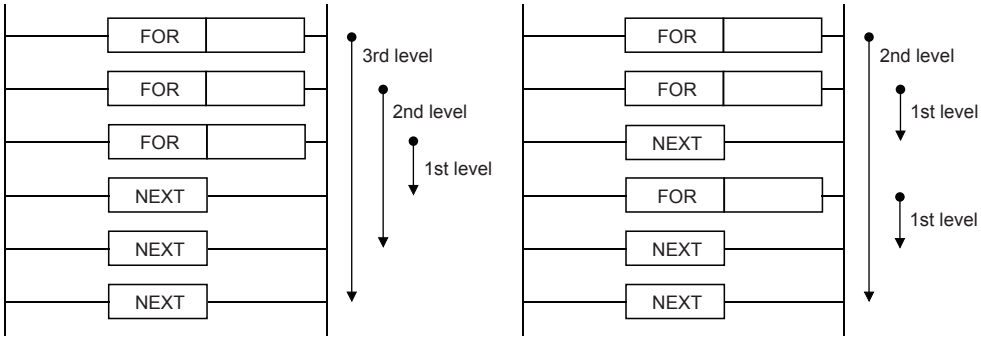
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- When the processing between the FOR and NEXT instructions is executed (n) times without any condition, the processing of the step following the NEXT instruction is executed.
- In (n), any of 1 to 32767 can be specified. If any of -32768 to 0 is specified, the processing of (n)=1 is applied.
- To skip the processing between the FOR and NEXT instructions, jump the program execution with the CJ instruction.
- Up to 16 FOR instructions can be nested.

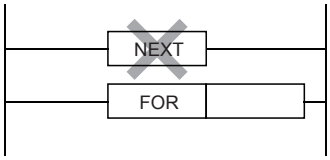
Precautions

- The FOR-NEXT loop can be nested up to 16 levels.

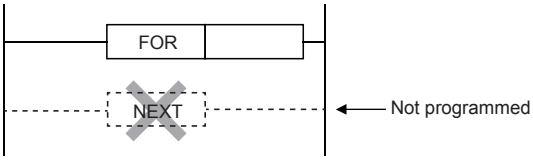


- The FOR-NEXT loop cannot be interrupted by the I, IRET, SRET, RET, FEND, or END instruction.
- When FOR-NEXT loop is repeated many times, the operation cycle is too long, and a watchdog timer error may occur. In such a case, change the watchdog timer time or reset the watchdog timer.
- The following programs are regarded as errors.

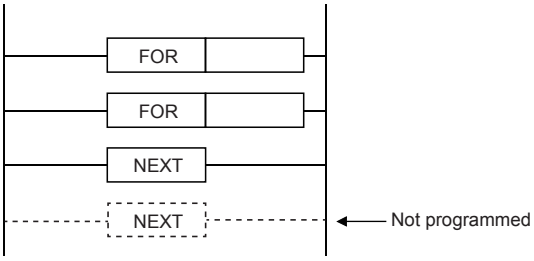
When the NEXT instruction is located before FOR



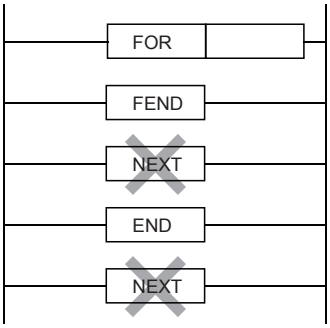
No NEXT instruction



When the number of FOR instructions is not equivalent to the number of NEXT instructions

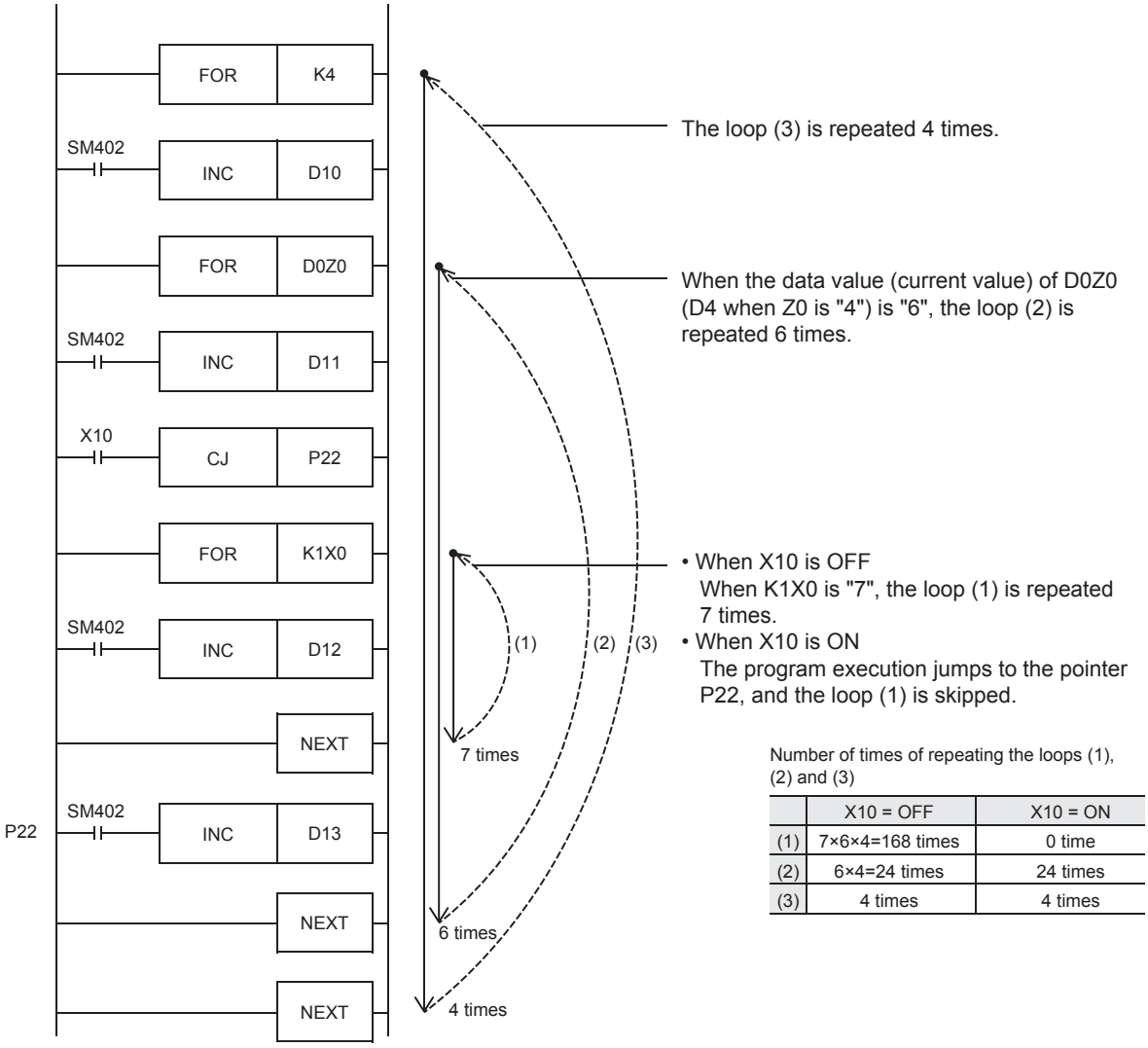


When the NEXT instruction is located after the FEND or END instruction



Program example

- Program with three FOR-NEXT loops



Number of times of repeating the loops (1), (2) and (3)

	X10 = OFF	X10 = ON
(1)	7×6×4=168 times	0 time
(2)	6×4=24 times	24 times
(3)	4 times	4 times

Operation error

Error code (SD0/SD8067)	Description
3340H	After the FOR instruction is executed, the END or GOEND instruction is executed before the NEXT instruction is executed.
3361H	When the FOR instruction is nested, the 17th level is executed.
33E3H	A program is written where the nesting of the FOR to NEXT instructions exceeds 16 levels.

Point

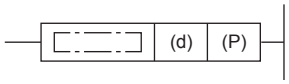
- To terminate the FOR to NEXT instruction loop halfway, use the BREAK instruction. (☞ Page 458 Forcibly terminating the FOR to NEXT instruction loop)

Forcibly terminating the FOR to NEXT instruction loop

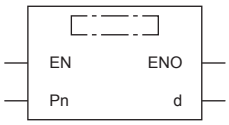
BREAK(P)

FX5S FX5UJ FX5U FX5UC

This instruction forcibly terminates the FOR to NEXT instruction loop and shifts the program execution to the pointer specified by (P).

Ladder diagram	Structured text
	Not supported

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Device number storing the number of remaining loops	—	16-bit signed binary	ANY16
(P)*1	Pointer number of the branch destination when the loop is forcibly terminated	—	Device name	POINTER
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

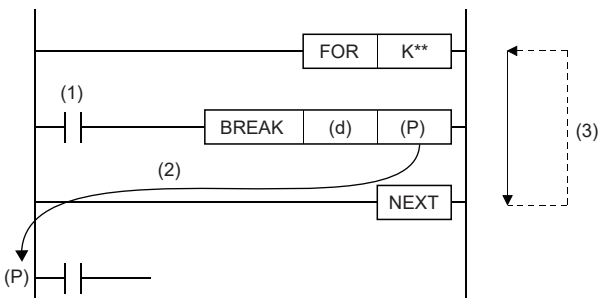
*1 In the case of the FBD/LD language, P displays as Pn.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—
(P)	—	—	—	—	—	—	—	—	—	—	○

Processing details

- This instruction forcibly terminates the FOR to NEXT instruction loop and shifts the program execution to the pointer specified by (P). Only the pointer numbers within the same program file can be specified in (P). If a pointer in another program is specified in (P), an operation error occurs.



(1) Forced termination condition
(2) When the forced termination condition is satisfied
(3) When the BREAK instruction is not executed, the program execution returns to the FOR instruction as many times as specified by the FOR instruction.

- In (d), the number of remaining FOR to NEXT instruction loops at the forced termination is stored. Note that the number includes the loop when the BREAK(P) instructions are executed.
- The BREAK(P) instructions can be used only between the FOR and NEXT instructions.

- The BREAK(P) instructions can be used for only one nesting level. To forcibly terminate multiple nesting levels, execute as many BREAK(P) instructions as the number of nesting levels.

Precautions

- If the branch pointer number of the BREAK instruction outside two nesting levels or more is specified, an operation error occurs and the program execution stops when the BREAK instruction is executed.

Operation error

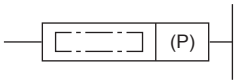
Error code (SD0/SD8067)	Description
3340H	The branch pointer number outside two nesting levels or more is specified.
3342H	The BREAK(P) instructions are used other than between the FOR and NEXT instructions.
3380H	The destination pointer specified by (P) does not exist.
	A pointer in other program file is specified in (P).

Calling a subroutine program

CALL(P)

FX5S FX5UJ FX5U FX5UC

This instruction executes the subroutine program specified by (P).

Ladder diagram	Structured text
	Not supported

FBD/LD
Not supported.

Setting data

■Descriptions, ranges, and data types

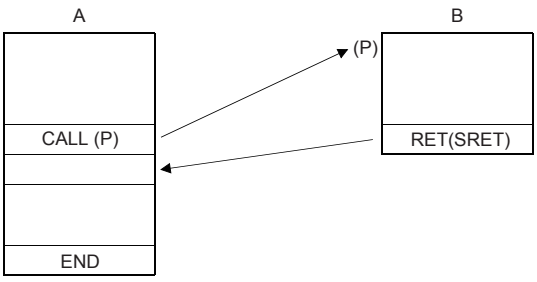
Operand	Description	Range	Data type	Data type (label)
(P)	Start pointer number of the subroutine program	—	Device name	POINTER

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(P)	—	—	—	—	—	—	—	—	—	—	○

Processing details

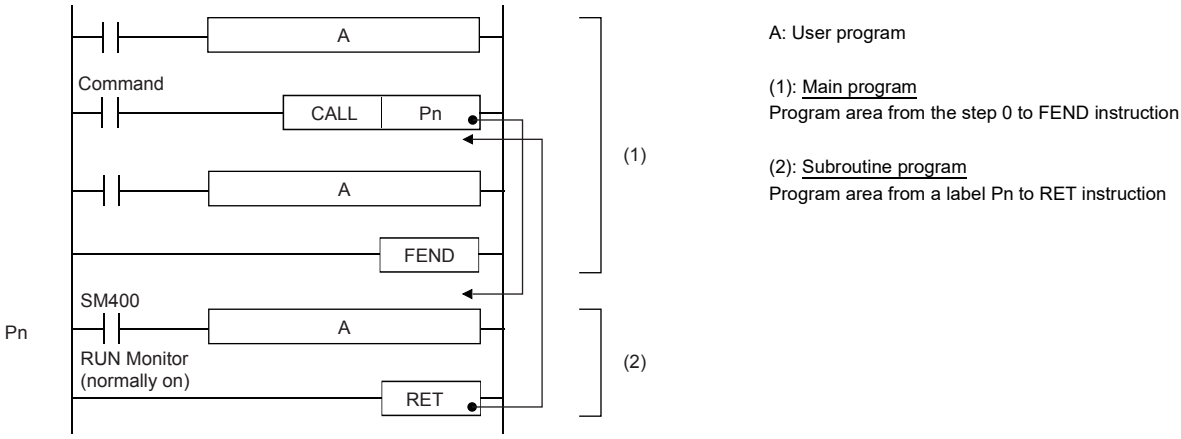
- When the CALL(P) instructions are executed, the subroutine program specified by the pointer (P) is executed. The CALL(P) instructions can execute a subroutine program specified by a pointer in the same program file or by a common pointer.



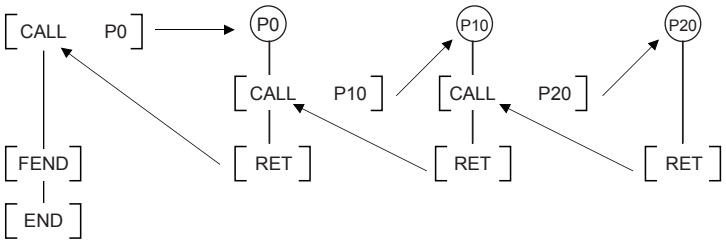
A: Main routine program

B: Subroutine program

- While the command input is ON, the CALL instruction is executed and the program execution jumps to a step with a label (Pn). Then, a subroutine program with the label (Pn) is executed. When the RET (SRET) instruction is executed, the program execution returns to the step following the CALL instruction. At the end of the main program, put FEND instruction. Put a label (Pn) for the CALL instruction after the FEND instruction.

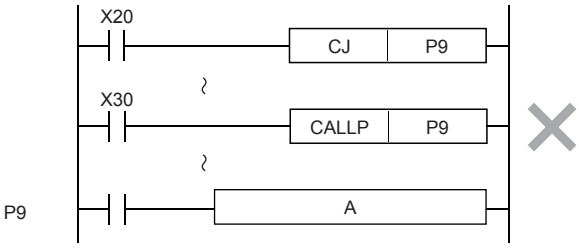


- The CALL(P) instructions can be nested up to 16 levels. However, the 16 levels are the total of the CALL(P) and XCALL instructions.



Precautions

- In the CALL instruction, the same number can be used two or more times in operands (P). However, do not use a label (P) and number used in another instruction (CJ instruction).



A: User program

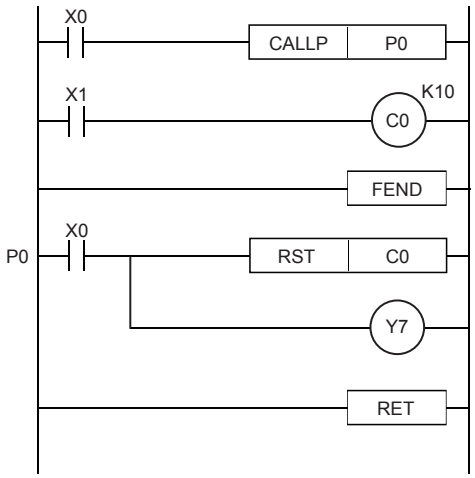
- In a subroutine (or interrupt routine), use timers for routine programs. These timers count when a coil instruction or END instruction is executed. After a timer reaches the set value, the output contact is activated when the coil instruction or END instruction is executed. Because general timers count only when the coil instruction is executed, they do not count if they are used in subroutines in which the coil instruction is executed only under some conditions.
- If a retentive type 1 ms timer is used in a subroutine (interrupt routine), note that the output contact is activated when the first coil instruction (or subroutine) is executed after the timer reaches its set value.

- Devices which were set to ON in a subroutine (or interrupt routine) are latched in the ON status even after the subroutine is finished. (Refer to the program example shown below). When the RST instruction for a timer or counter is executed, the reset status of the timer or counter is latched also. For turning OFF such a device latched in the ON status or for canceling such a timer or counter latched in the reset status, reset such a device in the main program after the subroutine is finished, or program a sequence for resetting such a device or for deactivating the RST instruction in the subroutine. (Refer to the program example shown below).

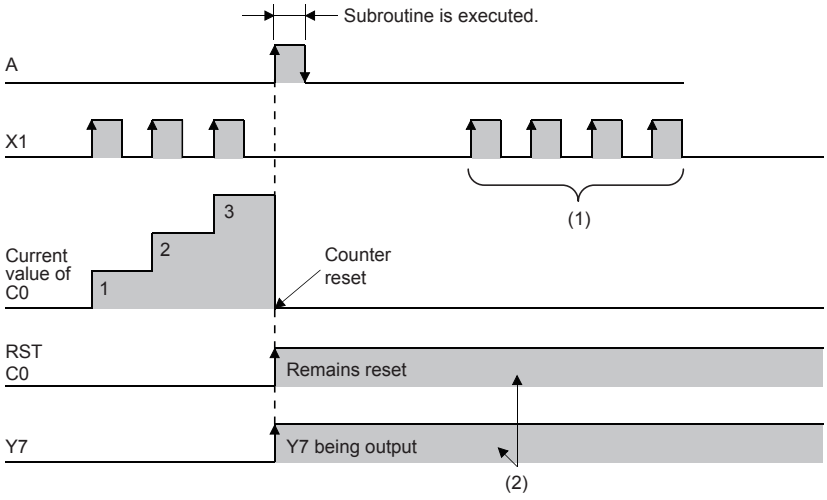
Example in which outputs are latched

In the following program example, the counter C0 is provided to count X1. When X0 is input, the subroutine P0 is executed only in one scan, and then the counter is reset and Y7 is output.

[Program example]



[Timing chart]



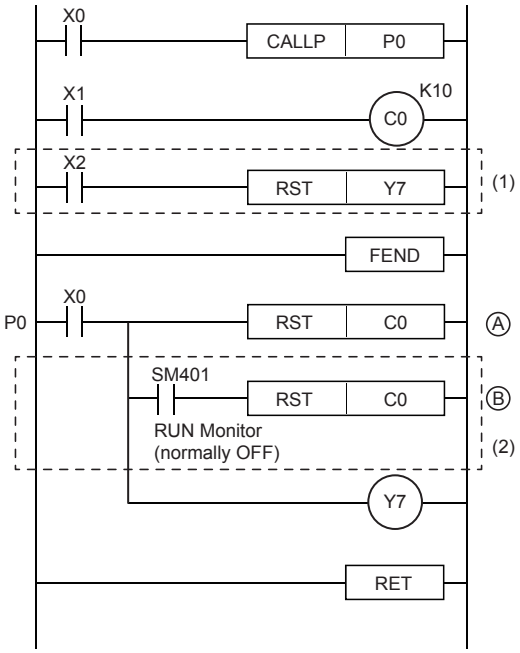
A: Execution of subroutine P0 triggered by X0

(1): Because the C0 reset instruction is valid, the current value of C0 remains unchanged even if a pulses are input.

(2): Outputs are latched.

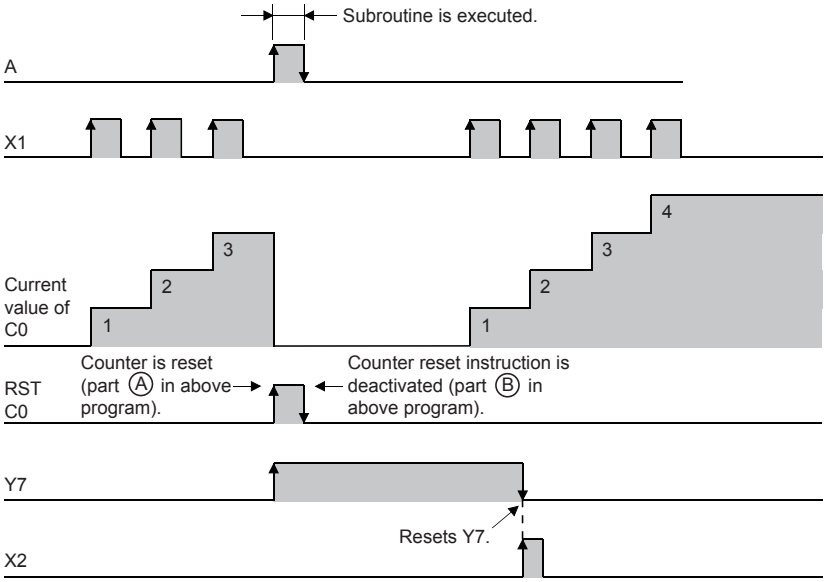
Example in which latched outputs are reset (countermeasures)

[Program example]



(1): Program to reset Y7 at an arbitrary timing
(2): Program to reset the preceding C0 reset instruction inside the subroutine

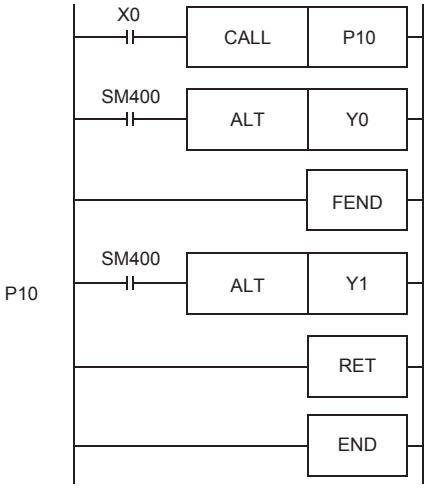
[Timing chart]



A: Execution of subroutine P0 triggered by X0

Program example

• Example of fundamental use (no nesting)

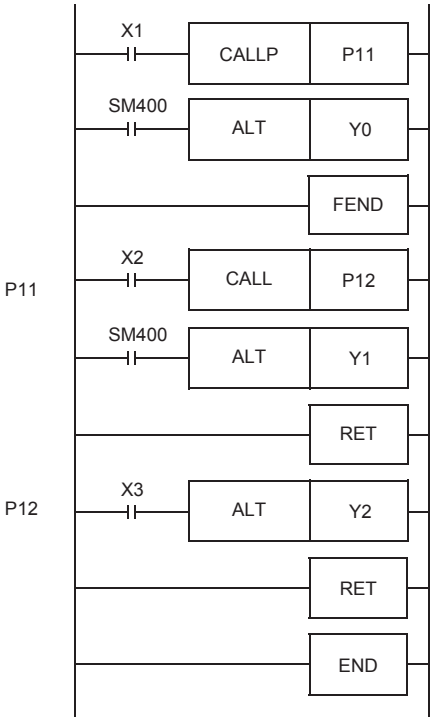


Main program
While X0 is ON, the program execution jumps to a step with the label P10.

Subroutine program
When RET instruction is executed after the subroutine program has executed, the program execution returns to the original step +1.

• Example of multiple CALL instructions in subroutines (multiple nesting)

CALL instruction can be used up to 4 times in subroutine programs. Nesting of up to sixteen layers is allowed.



Main program
When X1 turns from OFF to ON, the program execution jumps to the label P11 only once.

Subroutine program 1
When RET instruction is executed, the program execution returns to the main program.
If X2 is ON while the subroutine program 1 is executed, the program execution jumps to the step with the label P12.

Subroutine program 2
The subroutine program with P12 is executed, and then the program execution returns to the subroutine program with P11 by RET instruction.

Operation error

Error code (SD0/SD8067)	Description
3360H	The 17th level of the nesting is executed.
3380H	The subroutine program specified by the pointer in the CALL(P) instructions do not exist.
3381H	After the CALL(P) instructions are executed, the END, FEND, GOEND, or STOP instruction is executed before the RET (SRET) instruction is executed.
3382H	The RET (SRET) instruction is executed before the CALL(P) instructions are executed.

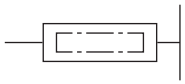
Returning from the subroutine program

RET/SRET

FX5S FX5UJ FX5U FX5UC

These instructions indicate an end of a subroutine program.

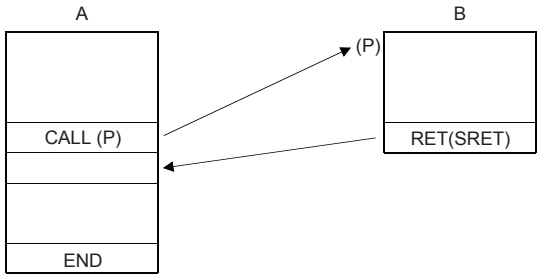
The RET instruction can be used as SRET.

Ladder diagram	Structured text
	Not supported

FBD/LD
Not supported.

Processing details

- These instructions indicate an end of a subroutine program.
- When the RET instruction is executed, the program execution returns to the step following the CALL(P) or XCALL instruction that called the subroutine program.



A: Main routine program

B: Subroutine program

Precautions

- If the RET (SRET) instruction is executed in a user interrupt program (I-IRET), a compiling error occurs.

Operation error

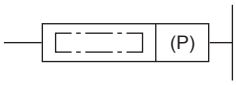
Error code (SD0/SD8067)	Description
3381H	The END, FEND, GOEND, or STOP instruction is executed before the RET instruction is executed.
3382H	While the number of nesting levels is decreased by the return instruction, the result becomes negative. (The number of RET (SRET) instructions is larger than that of the CALL instructions.)
33E3H	The RET instruction is programmed and written between the FOR and NEXT instructions.
33E5H	The RET instruction is programmed and written between the STL and RETSTL instructions.
33E6H	The program for executing the RET instruction is written before the CALL or XCALL instruction is executed.
33E7H	The RET instruction is programmed and written between the MC and MCR instructions. A program is written with no (P) before the RET instruction. The program for executing the RET instruction is written between the I and IRET instructions.

Calling a subroutine program

XCALL

FX5S FX5UJ FX5U FX5UC

This instruction executes CALL for (turns on and executes) the subroutine program specified by (P) when the execution condition is established. When the condition is turned off, this instruction executes turns off and terminates for the subroutine program.

Ladder diagram	Structured text
	Not supported

FBD/LD
Not supported.

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(P)	Start pointer number of the subroutine program	—	Device name	POINTER

■Applicable devices

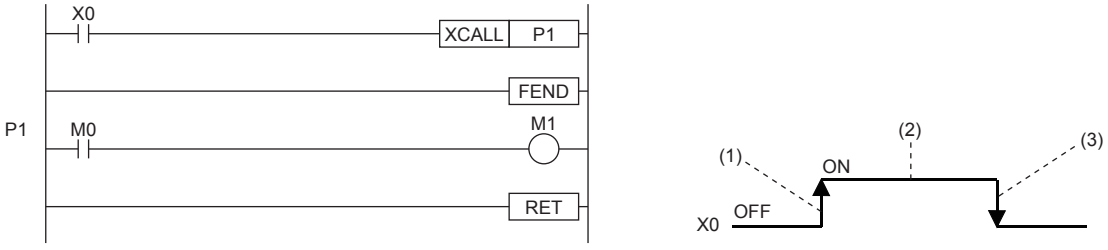
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(P)	—	—	—	—	—	—	—	—	—	—	○

Processing details

- The XCALL instruction controls the execution and non-execution processing of subroutine programs.
  - In the execution of subroutine programs, each coil instruction is operated according to the ON/OFF status of the condition contact.
  - In the non-execution processing of subroutine programs, each coil instruction is operated with the OFF status of the condition contact applied.
- The following table lists the operation result of each coil instruction after the non-execution processing. Regardless of the status of the condition contact, the following result is applied.

Device used for operation	Operation result (device status)
1 ms timer, 10 ms timer, 100 ms timer	0
1 ms retentive timer, 10 ms retentive timer, 100 ms retentive timer, counter	The current status is held.
Device in the OUT instruction	Forcibly turned off.
Device in the SET, RST, or SFT(P) instruction or basic/applied instruction	The current status is held.
PLS instruction, pulse instruction (□P)	Same as when the condition contact is off

- The following shows the operation of the XCALL instruction.

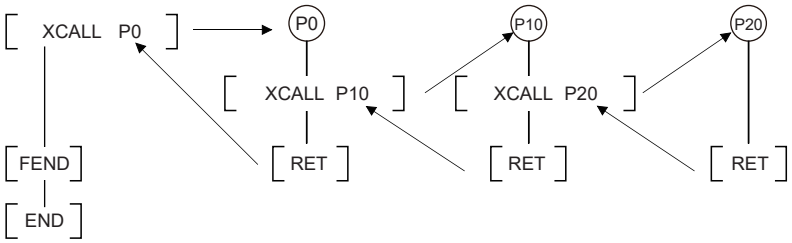


(1) Rising edge of X0 (OFF → ON): The subroutine program of P1 is executed.

(2) While X0 is on: The subroutine program of P1 is executed. (The rising edge of X0 is not included.)

(3) Falling edge of X0 (ON → OFF): The non-execution processing of the subroutine program of P1 is executed.

- The XCALL instruction can be nested up to 16 levels. However, the 16 levels are the total of the CALL(P) and XCALL instructions.



Operation error

Error code (SD0/SD8067)	Description
3360H	The 17th level of the nesting is executed.
3380H	The subroutine program specified by the pointer in the XCALL instruction does not exist.
3381H	After the XCALL instruction is executed, the END, FEND, GOEND, or STOP instruction is executed before the RET instruction is executed.
3382H	The RET (SRET) instruction is executed before the XCALL instruction is executed.

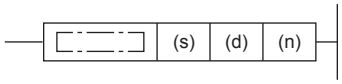
8.5 Data Table Operation Instruction

Reading the oldest data from the data table

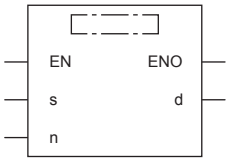
SFRD(P)

FX5S FX5UJ FX5U FX5UC

These instructions read data for first-in first-out control.

Ladder diagram	Structured text
	ENO:=SFRD(EN,s,n,d); ENO:=SFRDP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

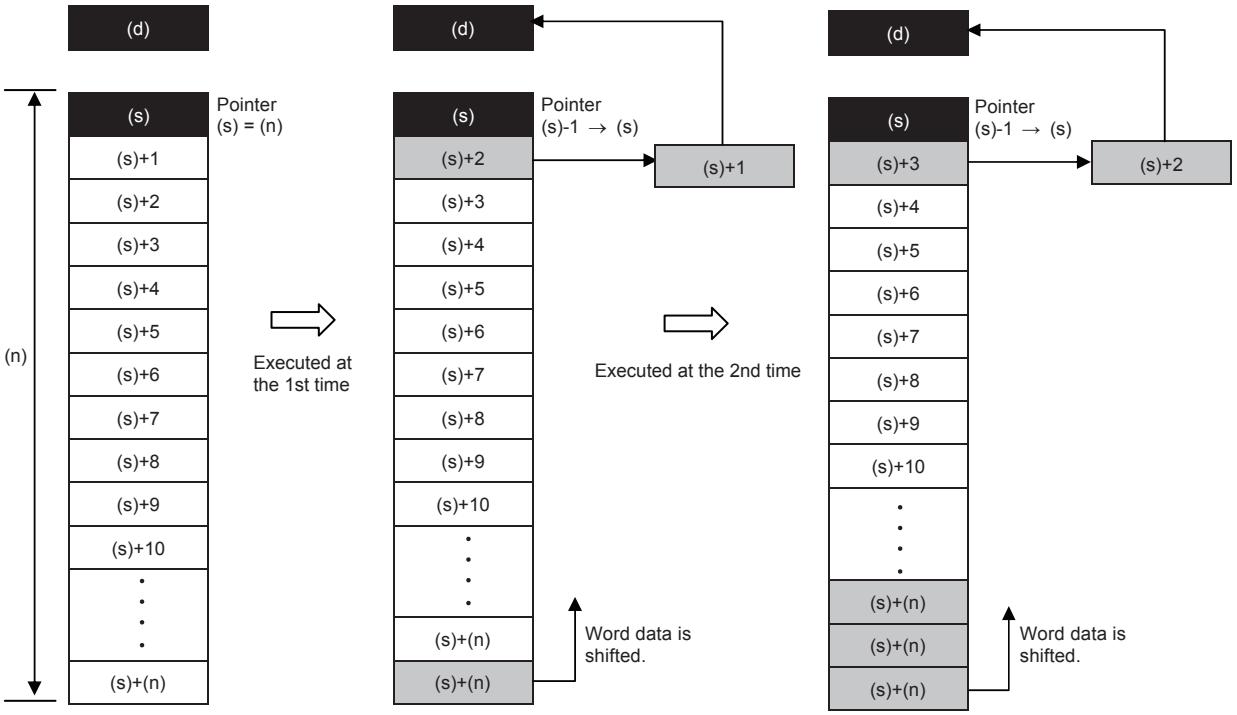
Operand	Description	Range	Data type	Data type (label)
(s)	Start number of the word device storing the data (The start is a pointer. The data is stored starting from (s)+1.)	—	16-bit signed binary	ANY16
(d)	Word device number storing data taken out first	—	16-bit signed binary	ANY16
(n)	Number of stored points plus "1". "+1" is required for the pointer.	2 to 32768	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	—	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions transfer (read) (s)+1, which was sequentially written by the SFWR instruction, to (d), and shift the word data of (n)-1 points starting from (s)+1 upward by 1 word. Then, these instructions decrease the number of data points stored in (s) by 1.

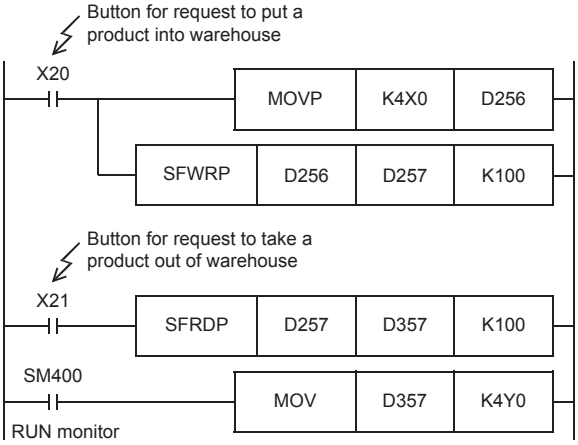


- The data of (s)+1 is transferred (read) to (d). Accompanied by this transfer, the contents of the pointer (s) decrease, and the data is shifted upward by 1 word. (When the continuous operation type SFRD instruction is used, the contents are stored in turn in each operation cycle. Use the pulse operation type SFRDP instruction in programming.)

Precautions

- The contents of (s)+(n) do not change by reading.
- When the continuous operation type (SFRD) instruction is used, data is read in turn in each scan time (operation cycle), but the contents of (s)+(n) do not change.
- When 0 is set in the pointer (s), no processing is executed and the contents of (d) do not change.

Program example



The product number is input from X0 to X17, and transferred to D256.

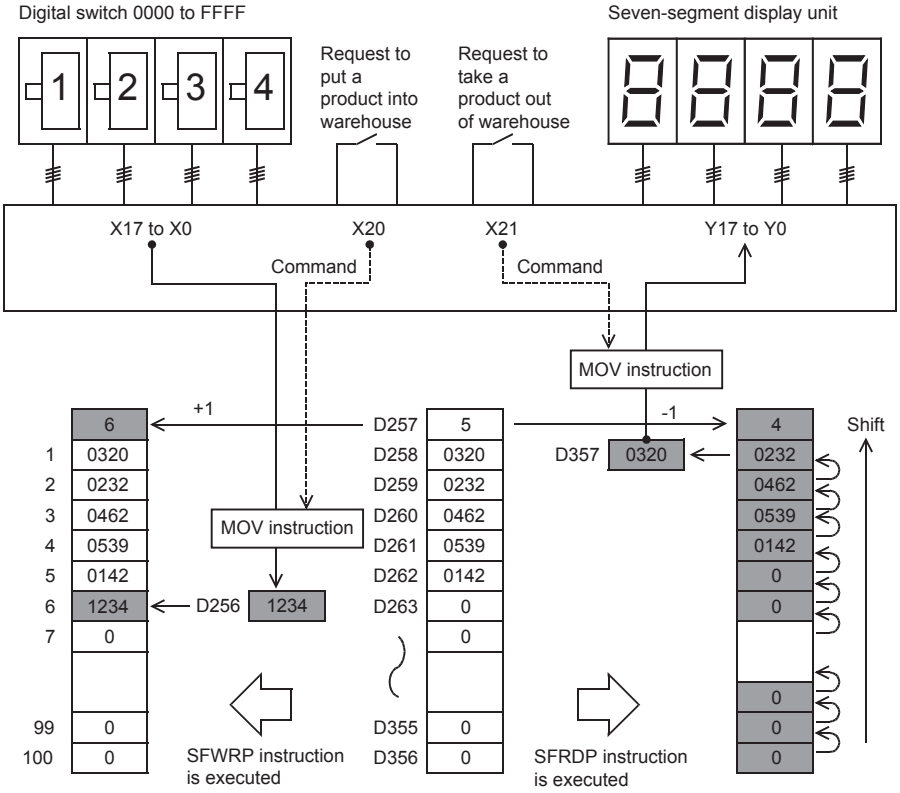
Pointer

D257: Data register for storing the product number

D258 to D356 (99 points)

The product number of a product put into first is output to D357 in response to the request to put a product out of the warehouse.

The product number to be taken out is output to Y0 to Y17 in a four-digit hexadecimal number.



Operation error

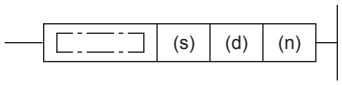
Error code (SD0/SD8067)	Description
2820H	The number of device points (n) from (s) exceed the device range.
3405H	The value set in (n) is other than the following. 2≤(n)≤32768
	In (s), a negative value is specified.

Reading the newest data from the data table

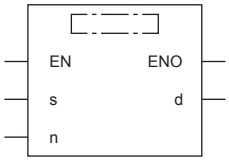
POP(P)

FX5S FX5UJ FX5U FX5UC

These instructions read the latest data written by a shift write (SFWR) instruction for FIFO/FILO control.

Ladder diagram	Structured text
	ENO:=POP(EN,s,n,d); ENO:=POPP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number storing the first-in data (including pointer data) (start number of the word device storing the data)	—	16-bit signed binary	ANY16
(d)	Device number storing last-out data	—	16-bit signed binary	ANY16
(n)	Length of data array (Add "1" because pointer data is also included.)	2 to 32768	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

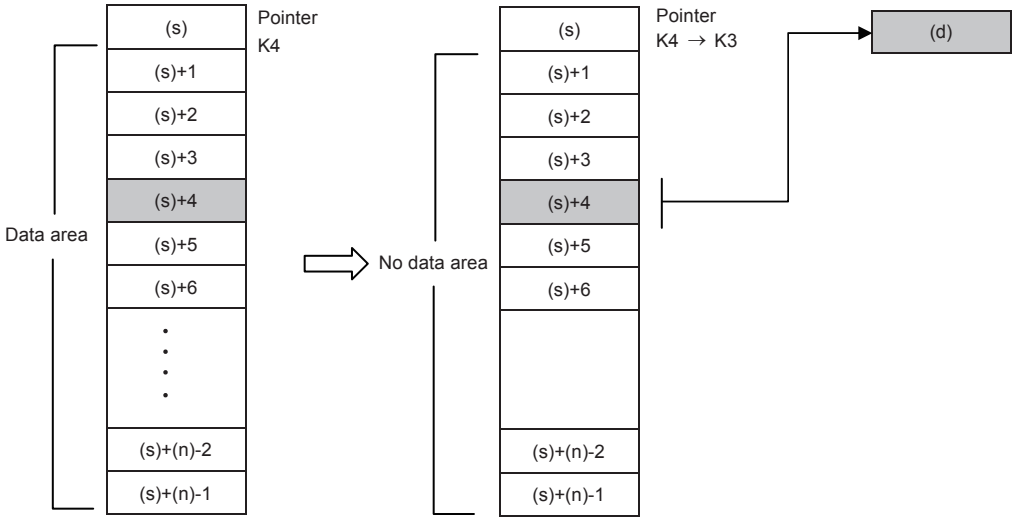
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	—	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- Every time the instruction is executed for the word devices (s) to (s)+(n)-1, a device "(s) + Pointer data (s)" is read to (d). (The last data entry written by the shift write (SFWR) instruction for first-in first-out control is read to (d).) Specify any value between 2 and 32767 for (n).
- Subtract "1" from the value of the pointer data (s).

Data for FILO control

	Description
(s)	Pointer data (amount of data stored)
(s)+1	Data area (First-in data written by shift write (SFWR) instruction)
(s)+2	
(s)+3	
⋮	
(s)+(n)-3	
(s)+(n)-2	
(s)+(n)-1	

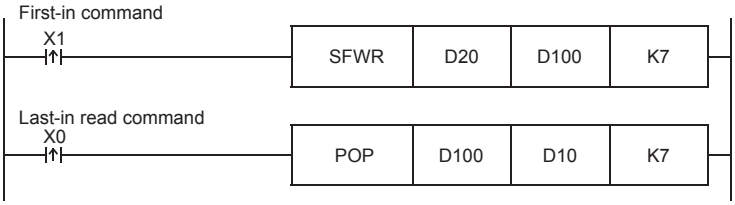


Precautions

- If programed in the continuous operation type, the POP(P) instructions are executed in every operation cycle. As a result, expected operation may not be achieved. Usually, program the POP(P) instructions in the "pulse operation type", or let them be executed by a "pulsed command contact".
- When the current value of the pointer (s) is "0", the zero flag SM8020 turns ON and the POP(P) instructions are not executed.
- When the current value of the pointer (s) is "1", "0" is written to (s) and the zero flag SM8020 turns ON.

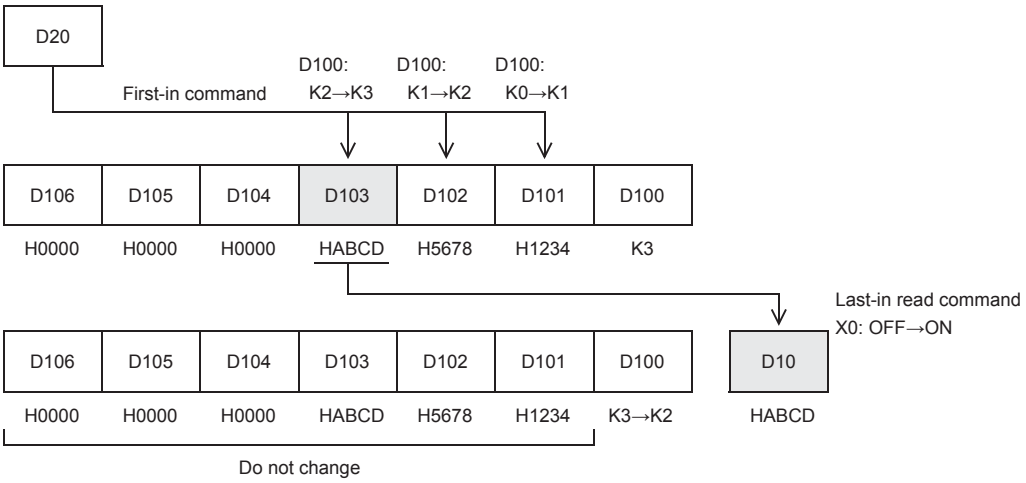
Program example

Among values stored in D20 input first to D101 to D106, the last value input is stored to D10, and “1” is subtracted from the number of stored data (pointer D100) every time X0 turns ON.



When the first-in data is as shown in the table below

Pointer	Data					
D100	D101	D102	D103	D104	D105	D106
K3	H1234	H5678	HABCD	H0000	H0000	H0000



Operation error

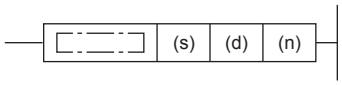
Error code (SD0/SD8067)	Description
2820H	The device range (s)+(n)-1 exceeds the device.
3405H	(s) is larger than (n)-1.
	(s) is smaller than 0.
	The value set in (n) is other than the following. 2≤(n)≤32768

Writing data to the data table

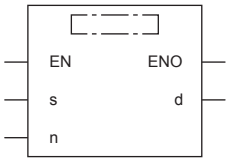
SFWR(P)

FX5S FX5UJ FX5U FX5UC

These instructions write data for first-in first-out (FIFO) and last-in first-out (LIFO) control.

Ladder diagram	Structured text
	ENO:=SFWR(EN,s,n,d); ENO:=SFWRP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

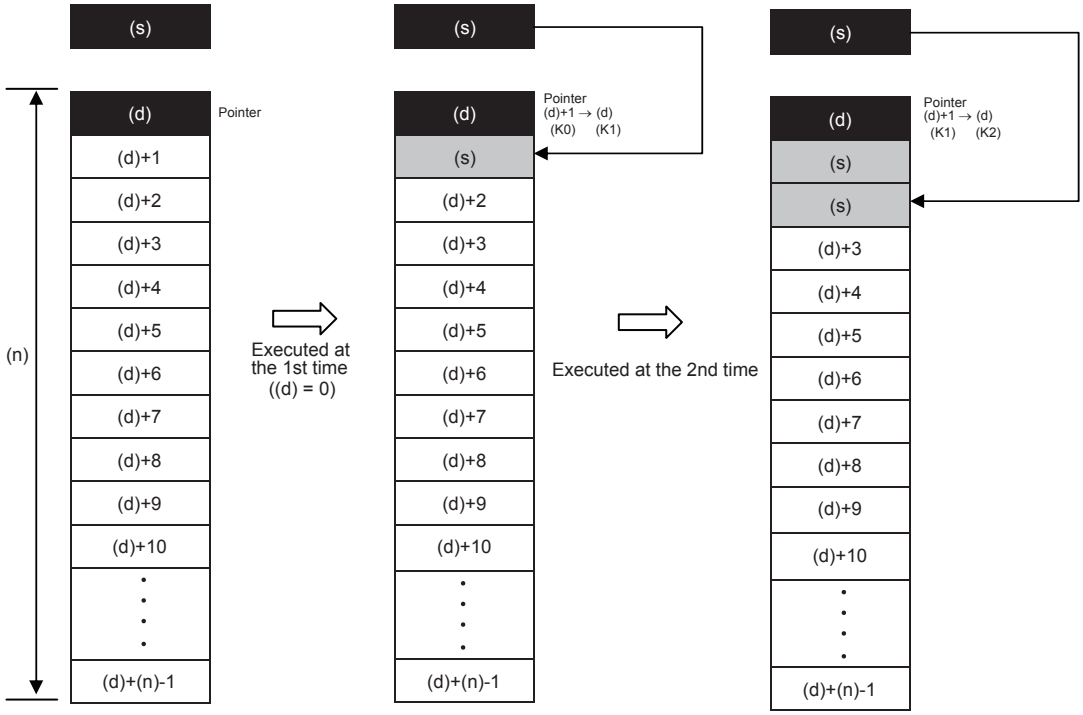
Operand	Description	Range	Data type	Data type (label)
(s)	Word device number storing data to be put in first	—	16-bit signed binary	ANY16
(d)	Start word device number storing and shifting data (The start is a pointer. The data is stored starting from (d)+1.)	—	16-bit signed binary	ANY16
(n)	Number of stored points plus "1".	2 to 32768	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- The contents of (s) are written to "(n)-1" devices from (d)+1, and "1" is added to the number of data stored in (d). For example, for (d)=0, the contents are written to (d)+1, and for (d)=1, to (d)+2.



- At the first execution, the contents of (s) are stored in (d)+1.
- When the contents of (s) are changed and then the instruction is executed again, the new contents of (s) are stored to (d)+2. So the contents of +2 become equivalent to (s). (When the continuous operation type SFWR instruction is used, the contents are stored in each operation cycle. Use the pulse operation type SFWRP instruction in programming.) Data is stored from the right end in the same way, and the number of stored data is specified by the contents of the pointer (d).

Precautions

- In the case of the continuous operation type instruction (SFWR), note that data is stored (overwritten) in every scan time (operation cycle).

Program example

For a program example, refer to ☞ Page 468 Reading the oldest data from the data table.

Operation error

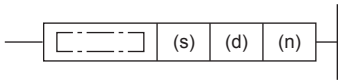
Error code (SD0/SD8067)	Description
2820H	The number of device points (n) from (d) exceeds the device range.
3405H	The value set in (n) is other than the following. 2≤(n)≤32768
	In (d), a negative value is specified.

Inserting data to the data table

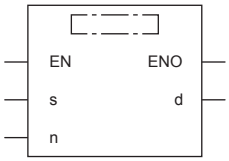
FINS(P)

FX5S FX5UJ FX5U FX5UC

These instructions insert 16-bit data specified by (s) to the data table specified by (d) as the (n)th data.
After these instructions are executed, the data after the (n)th data in the data table is moved down by one data point.

Ladder diagram	Structured text
	ENO:=FINS(EN,s,n,d); ENO:=FINSP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

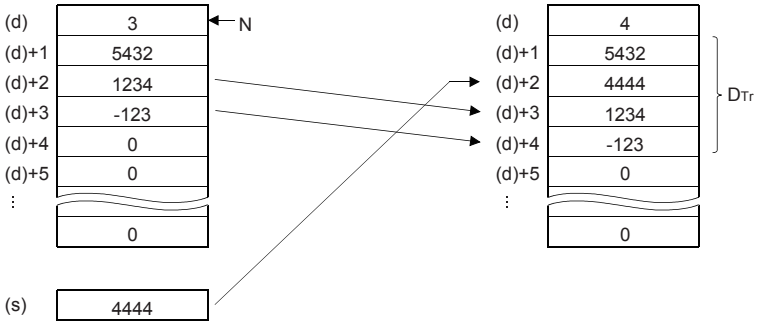
Operand	Description	Range	Data type	Data type (label)
(s)	Head device number where the insertion-target data is stored	—	16-bit signed binary	ANY16
(d)	Start number of the table	—	Word	ANY16
(n)	Data insertion position in the table	1 to 32767	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions insert 16-bit binary data specified by (s) to the data table specified by (d) as the (n)th data. After these instructions are executed, the data after the (n)th data in the data table is moved down by one data point.



N: Number of stored data
D_{Tr}: Data table range

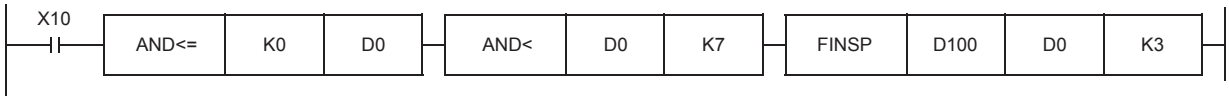
When 2 is set in (n), the data is inserted in (d)+2.

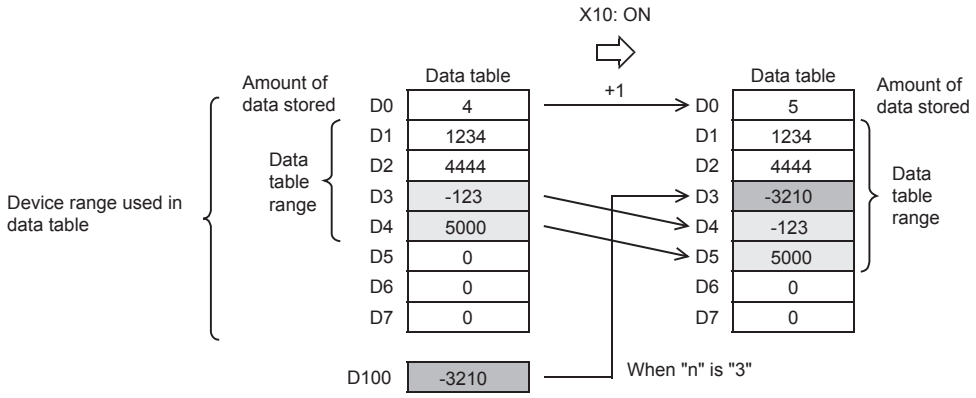
Precautions

- The device range used in a data table should be controlled by the user.
- The data table has (d) number of stored data starting from ((d)+1).

Program example

In the program shown below, data stored in D100 is inserted into the 3rd position of the data table stored in D0 to D4 when X10 is set to ON.
However, when the amount of data stored exceeds "7", the FINSP instruction is not executed.
(The device range used in the data table is D0 to D7).





Operation error

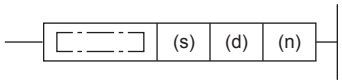
Error code (SD0/SD8067)	Description
2820H	When the FINS(P) instructions are executed, the data table range exceeds the corresponding device range.
3405H	When the FINS(P) instructions are executed, the value (n) exceeds the corresponding device range of the table (d).
	When the FINS(P) instructions are executed, the table position (n) where the data is inserted exceeds "the number of stored data points + 1".
	The value set in (n) is other than the following. 2≤(n)≤32767

Deleting data from the data table

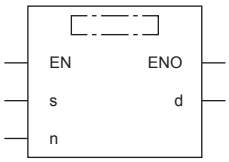
FDEL(P)

FX5S FX5UJ FX5U FX5UC

These instructions remove the (n)th data in the data table specified by (d) and store the data in the device specified by (s). After these instructions are executed, the data after the (n)+1th data in the data table is moved up by one data point.

Ladder diagram	Structured text
	ENO:=FDEL(EN,s,n,d); ENO:=FDELP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

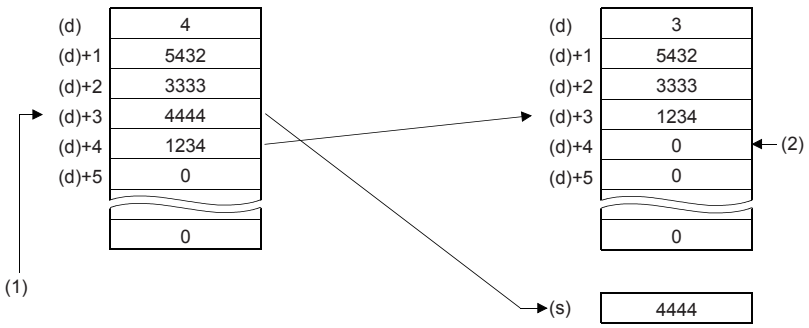
Operand	Description	Range	Data type	Data type (label)
(s)	Head device number for storing the data to be deleted	—	16-bit signed binary	ANY16
(d)	Start number of the table	—	Word	ANY16
(n)	Position of the data to be deleted in the table	1 to 32767	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions remove the (n)th data in the data table specified by (d) and store the data in the device specified by (s). After these instructions are executed, the data after the (n)+1th data in the data table is moved up by one data point.

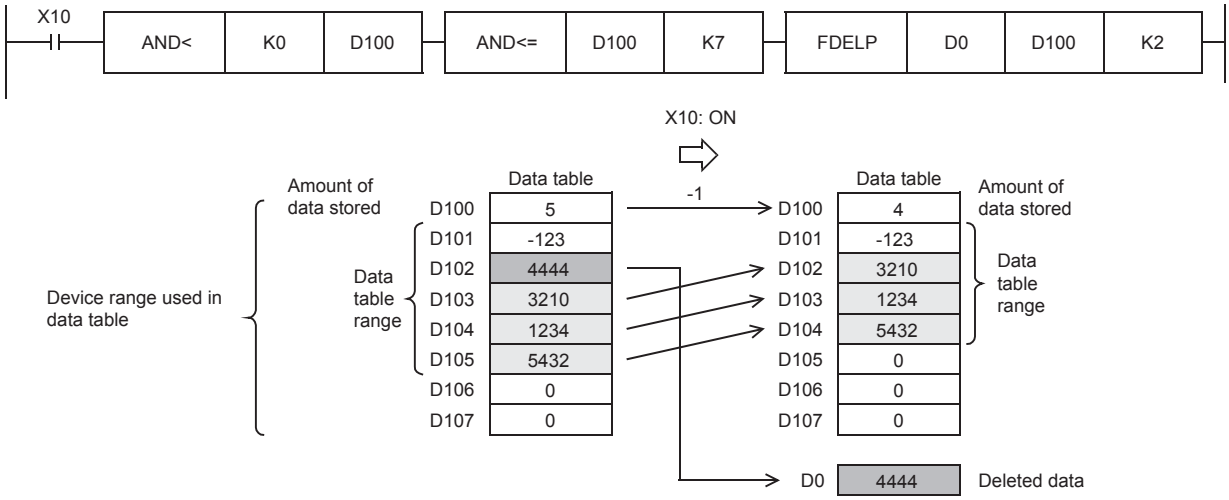


Precautions

- The device range used in a data table should be controlled by the user.
- The data table has (d) number of stored data starting from ((d)+1).

Program example

In the program shown below, the 2nd data value entry is deleted from the data table stored in D100 to D105 when X10 is set to ON, and the deleted data is stored in D0.
However, when the amount of data stored is "0", the FDELP instruction is not executed.
(The device range used in the data table is D100 to D107).



Operation error

Error code (SD0/SD8067)	Description
2820H	When the FDEL(P) instructions are executed, the data table range exceeds the corresponding device range.
3405H	When the FDEL(P) instructions are executed, the value (n) exceeds the corresponding device range of the table (d).
	When 0 is set in (d), and the FDEL(P) instructions are executed.
	When the FDEL(P) instructions are executed, the table position (n) where the data to be deleted is stored exceeds the number of stored data points.
	The value set in (n) is other than the following. 2≤(n)≤32767

8.6 Reading/writing Data Instructions

Reading/Writing data to data memory

The data write instruction is an instruction to write arbitrary device data to data memory.

Writing the fixed values used for operation and operation results to data memory can prevent data loss when the battery is low.

The data that has been written to data memory can be read at any time using the data memory read instruction.

■Execution method

Use the SP.DEVST instruction to write device data to data memory.

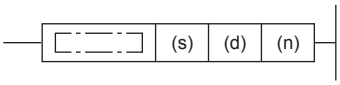
Use the S(P).DEVLD instruction to read device data from data memory to any specified device.

Reading data from the data memory

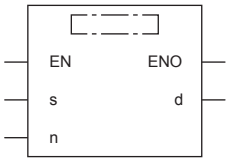
S(P).DEVLD

~~FX5S~~ FX5UJ FX5U FX5UC

Reads data from the device data storage file in data memory.

Ladder diagram	Structured text
	ENO:=S_DEVLD(EN,s,n,d); ENO:=SP_DEVLD(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Read offset of device data storage file (specified in units of 16-bit words)	0 to 63	32-bit unsigned binary	ANY32
(d)	Head device for storing the data that has been read	—	Word	ANY16*1
(n)	Number of read points	1 to 64	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

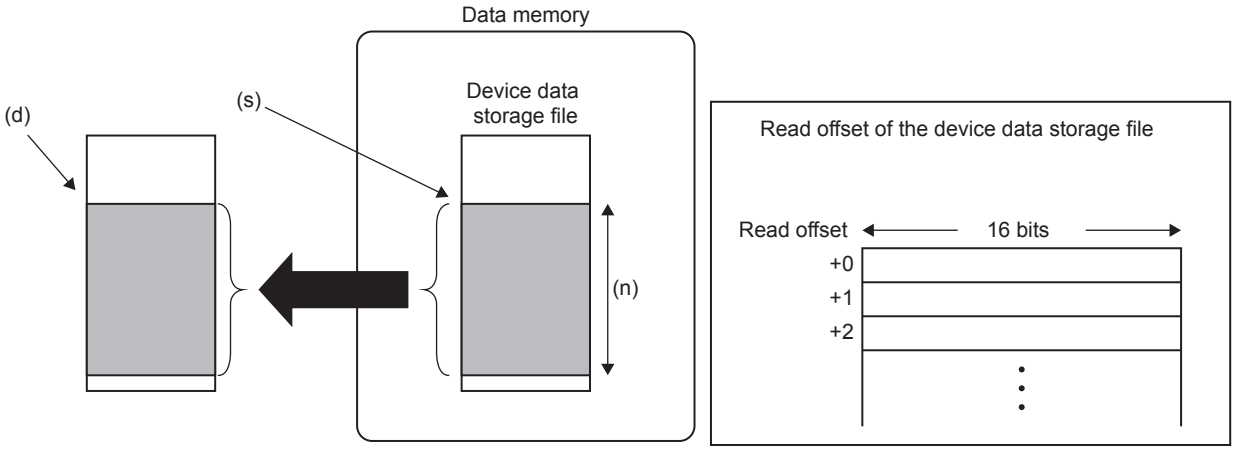
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K,H	E	$	
(s)	—	○	—	—	○	—	○	○	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○*1	○	—	—	—	—	○	○	—	—	—

*1 S cannot be used.

Processing details

- These instructions read device data by the number of points specified by (n) from the read offset specified by (s) in the device data storage file in data memory, and store it in the device specified by (d). (s) indicates the offset from the start of the device data storage file and can be specified by word offsets (incremented by 1 every 16 bits).



- Use the SP.DEVST instruction to write device data to the device data storage file. (☞ Page 483 SP.DEVST)

Operation error

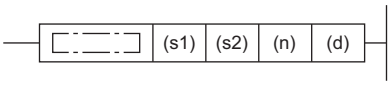
Error code (SD0/SD8067)	Description
2820H	• The value specified by (s), and (s)+(n) are out of range of the device data storage file. • The device range specified by (d) exceeds the corresponding device range.
2822H	Device that cannot be specified is specified.
3405H	The value stored in a device specified by (n) is 0.

Writing data to the data memory

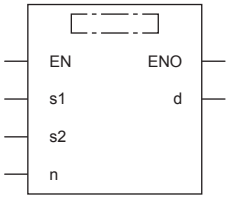
SP.DEVST

~~FX5S~~ FX5UJ FX5U FX5UC

This instruction writes the specified number of points of data to the device data storage file in data memory.

Ladder diagram	Structured text
	ENO:=SP_DEVST(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Write offset of device data storage file (specified in units of 16-bit words)	0 to 63	32-bit unsigned binary	ANY32
(s2)	Start device to which data is to be written	—	Word	ANY16*1
(n)	Number of write points	1 to 64	16-bit unsigned binary	ANY16
(d)	(d): Completion device, (d)+1: Error completion device	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

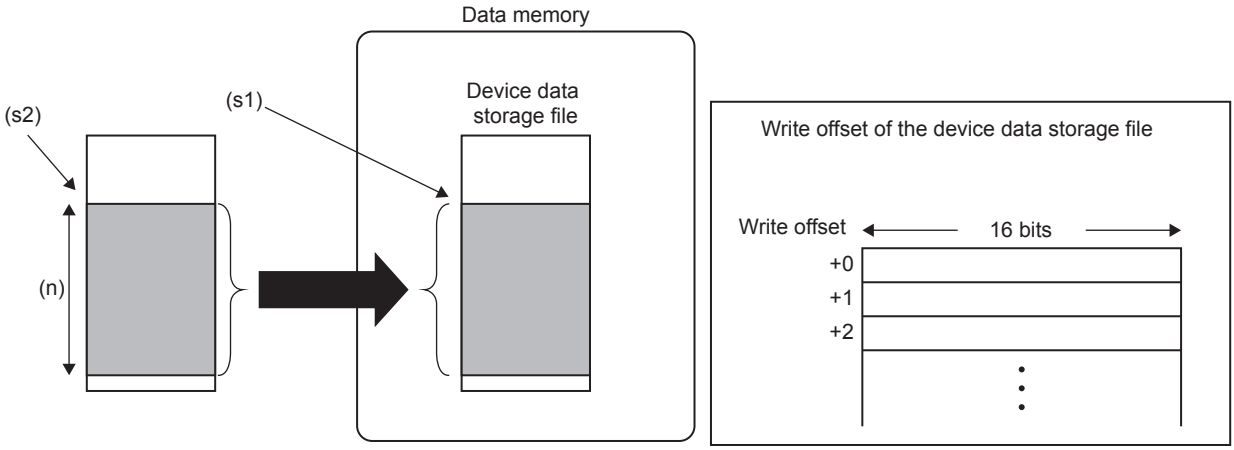
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K,H	E	$	
(s1)	—	○	—	—	○	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(n)	○*1	○	—	—	—	—	○	○	—	—	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.
*2 T, ST, and C cannot be used.

Processing details

- These instructions retrieves the specified number of points of data specified by (n) from the device specified by (s2) and write it to the write offset in the device specified by (s1) in the device data storage file in data memory. (s1) indicates the offset from the start of the device data storage file and can be specified by word offsets (incremented by 1 every 16 bits).



- The completion device specified by (d) automatically turns on upon execution of the END instruction following the detection of processing completion of the SP.DEVST instruction and turns off upon execution of the END instruction in the next scan, so it is used as the execution complete flag of the SP.DEVST instruction.
- If the SP.DEVST instruction completes with an error, the error completion device specified by (d)+1 turns on or off at the same time as the completion device specified by (d). Therefore, the device is used as the error completion flag of the SP.DEVST instruction.
- SM753 (File being accessed) turns on while the SP.DEVST instruction is executed. If SM753 has already been on, the SP.DEVST instruction cannot be executed. (If executed, no processing is performed.)
- If an error is detected during execution of the SP.DEVST instruction, the completion device (d), error completion device (d)+1, and SM753 do not turn on.
- Use the S(P).DEVLD instruction to read device data from the device data storage file to any specified device. (☞ Page 481 S(P).DEVLD)

Precautions

- The value written to data memory is the one at execution of the SP.DEVST instruction.
- Execution of the SP.DEVST instruction increases SD634 and SD635. The number of writes to the data memory of the CPU module is limited. If the data memory write count index exceeds twenty thousands, an error occurs with error code 1080H.
- To prevent the data memory write count from being increased by careless instruction execution, SD771 can be set to limit the write count per day. The maximum number of writes is 8 by default. Change the maximum number of writes by using SD771 as needed. If the specified write count is exceeded, an error occurs with error code 3421H. The number of executions of the instruction to write to data memory per day is initialized to 0 at the following timing.

  - When power off→on, or when reset→reset canceled.
  - The date (year, month, day) in clock data is changed by time advancement.
  - CPU module internal clock data (year, month, day) is changed by the clock data change function.
- Data is written to the device data storage file when the END instruction is executed. Data is written to the device data storage file when the END instruction is executed immediately after the SP.DEVST instruction is executed. Thus, depending on the number of write points, writing to the device data storage file may involve multiple scans. Check the completion device to see whether the writing is completed.
- When executing SP.DEVST instruction during backup by the backup restore function, the writing is completed abnormally. Do not execute the SP.DEVST instruction during backup. Executing the SP.DEVST instruction after confirming that the backup executing flag (SM1350) is off is recommended.

Operation error

Error code (SD0/SD8067)	Description
2820H	• The value specified by (s1), and (s1)+(n) are out of range of the device data storage file. • Any of the device area ranges specified in (s2) and (d) exceed the corresponding device range.
2822H	Device that cannot be specified is specified.
3405H	The value stored in a device specified by (n) is 0.
3421H	When the SP.DEVST instruction is executed, the write count of the day exceeds the value specified in SD771.
	When the SP.DEVST instruction is executed, a value out of the range (1 to 32767) is set in SD771.

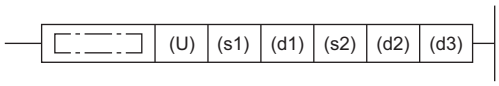
8.7 File Operation Instructions

Reading data from the specified file

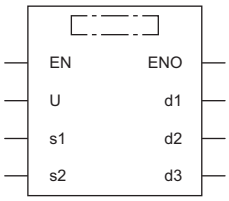
SP.FREAD

FX5S FX5UJ FX5U FX5UC

This instruction reads device data from the specified file in an SD memory card.

Ladder	ST
	ENO:=SP_FREAD(EN,U,s1,s2,d1,d2,d3);

FBD/LD



Setting data

■Description, range, data type

Operand	Description	Range	Data type	Data type (label)
(U)	Dummy	■FX5S CPU module U1 ■FX5UJ CPU module U1 to U8 ■FX5U/FX5UC CPU module U1 to U10	Device name*2	ANY16
(s1)	Drive specification	2 (fixed)*1	16-bit signed binary	ANY16
(d1)	Start device where the control data is stored	☞ Page 487 Control data (d1)	Word	ANY16_ARRAY (Number of elements: 8)
(s2)	Start device where the file name is stored	☞ Page 488 File name (s2)	Unicode string	ANYSTRING_DOUBLE
(d2)	Start device for storing the data that has been read	☞ Page 489 Read data (d2)	Word	ANY16*3
(d3)	Bit device that turns on upon completion of the processing (In the case of an error completion, the device specified by (d3)+1 also turns on.)	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Only drive 2 (for the SD memory card) can be set.
*2 Specification with a label is not allowed.
*3 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	○	—	—	—	—	○	○	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	○	—
(d2)	○*1	○	—	—	—	—	○	—	—	—	—
(d3)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 When the bit device digit is specified in (d2), only multiples of 16 (0, 16, 32, 64…) can be specified as the device number. Only K4 can be specified as the number of digits.
*2 S cannot be used.
*3 T, ST, and C cannot be used.
Only bit specification of word device is applicable.

■Control data (d1)

For the combination of processing units, refer to ☞ Page 490 Combination of processing units .

Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Execution/completion type	Specify the execution type. ■00**H: Reading binary data • 0000H: 16-bit binary data • 0001H: 32-bit binary data ■01**H: Reading data after converted to CSV format • 0100H: Decimal (16-bit data) • 0110H: Decimal (32-bit data) • 0120H: Hexadecimal (16-bit data) • 0121H: Hexadecimal (32-bit data) • 0130H: String (ASCII data) • 0140H: Floating point real number (single-precision real number)	0000H 0001H 0100H 0110H 0120H 0121H 0130H 0140H	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0000H: Completed successfully • Other than 0000H: Completed with an error (error code) (☞ Page 579 Error codes generated for file operation instructions)	—	System
+2	Number of read-target data	Specify the number of data to be read. The processing unit and setting range vary depending on the execution/completion type in (d1)+0 and data type specification in (d1)+7. ■When "Reading binary data" is specified by (d1)+0 • When 16-bit binary data is specified: In units of words (1 to 65535)*1*2*3 • When 32-bit binary data is specified: In units of double words (1 to 32767) ■When "Reading data after converted to CSV format" is specified by (d1)+0 • When decimal (16-bit data) is specified: Number of elements (1 to 65535)*1*2*3 • When decimal (32-bit data) is specified: Number of elements (1 to 32767) • When hexadecimal (16-bit data) is specified: Number of elements (1 to 65535)*3 • When hexadecimal (32-bit data) is specified: Number of elements (1 to 32767) • When a string (ASCII data) is specified: Number of elements (1 to 1023) • When a floating point real number (single-precision real number) is specified: Number of elements (1 to 32767)	1 to 65535	User
+3	Maximum number of read data	■When "0130H: String (ASCII data)" is specified by (d1)+0 • Total size of the characters in the element (in units of words) ■When data other than "0130H: String (ASCII data)" is specified by (d1)+0 • Fixed to 0	0, 1 to 65535	User
+4 +5	File position	■When "Reading binary data" is specified by (d1)+0 • 00000000H: From the beginning of the file • 00000001H to FFFFFFFEH: From the specified position (The data unit is determined by (d1)+7.) • FFFFFFFFH: Cannot be specified. ■When "Reading data after converted to CSV format" is specified by (d1)+0 • 00000000H: From the beginning of the file • 00000001H to FFFFFFFEH: From the specified row • FFFFFFFFH: From the position where the previous reading ends	00000000H to FFFFFFFFH	User

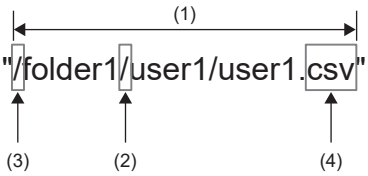
Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+6	Number of columns	When "Reading binary data" is specified by (d1)+0, set 0. When "Reading data after converted to CSV format" is specified by (d1)+0, set the number of read-target columns. • 0: No specification of number of columns. In the CSV format, the data from the first column to the last column are read ignoring line separators. (The number of read-target data is read as one row.) • Other than 0: Number of specified columns	0000H to FFFFH (0 to 65535)	User
+7	Data type specification	• 0: Word • 1: Even number of bytes*1 • 2: Unit of the data type specified by (d1)+0 • 3: Odd number of bytes*1*2 "0: Word", "1: Even number of bytes", and "3: Odd number of bytes" can be specified only when "0000H: 16-bit binary data" or "0100H: Decimal (16-bit data)" is specified by (d1)+0.	0, 1, 2, 3	User

*1 When "1: Even number of bytes" or "3: Odd number of bytes" is specified by (d1)+7 (Data type specification), the setting range of (d1)+2 (Number of read-target data) is 1 to 32767.

*2 Even when "3: Odd number of bytes" is specified by (d1)+7 (Data type specification), specify the number in units of words. Add one byte to the odd number of bytes to be read and set the number in (d1)+2 (Number of read-target data).

*3 Up to 32767 points (when the file register (R) is used) when only devices are used.

■File name (s2)

Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +☐	File name character string	Specify the folder path where the files are stored and the file name. • The folder path and file name (including an extension) must be within 253 characters in total. • The folder path must be within 244 characters. (Delimiters are not included.) • The number of folder path hierarchies must be within 10 levels. • When omitting an extension in the file name, omit the ". (period)" as well. • The file name must be within 60 characters (a period and extension excluded). If 61 or more characters are used, the extension is ignored and replaced with ".BIN" or ".CSV". • Do not specify a half-width space at the end of the character string or just before each delimiter.  (1): Up to 253 characters (2): Use "/" or "\" as delimiters for the folder path and file. (3): Can be omitted. When it is omitted, (1) is up to 252 characters. (4): If the extension is omitted, ".BIN" or ".CSV" will be automatically added by the system.	Unicode string	User

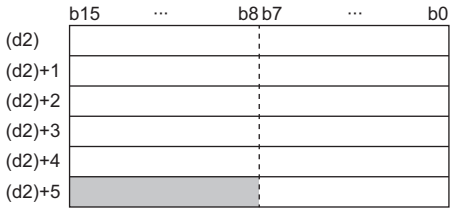
■Read data (d2)

For the combination of processing units, refer to ☞ Page 490 Combination of processing units.

Operand: (d2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Number of data actually read	The number of data actually read is set. The data unit is determined by (d1)+7 (Data type specification).	—	System
+1 to +□	Read data	The read data is stored.	—	System

Processing details

- This instruction reads data from the specified file. Set the execution/completion type in the control data to specify the file read-target format.
- The read target is the data in the SD memory card only.
- The processing completion bit device (d3) automatically turns on at the execution of the END instruction in the scan in which the completion of processing of the SP.FREAD instruction is detected. The bit device (d3) turns off at the execution of the END instruction in the next scan. If the processing completion bit device (d3) is ON, it will automatically turn OFF when the SP.FREAD instruction is executed.
- If the SP.FREAD instruction completes with an error, the error completion device (d3)+1 turns on or off in synchronization with (d3).
- SM753 (File being accessed) turns on while the SP.FREAD instruction is being executed. While SM753 is on, the SP.FREAD instruction cannot be executed. (If the instruction is executed, no processing is performed.)
- If an error is detected during the execution of the instruction, Processing Complete (d3), Error Completion (d3)+1, and SM753 do not turn on.
- When "Odd number of bytes" is specified by (d1)+7, the read data is stored as shown below. (In the shaded area shown below, the data before execution of the instruction is kept.)



■Combination of processing units

Specify data in (d1)+2, (d1)+4, (d1)+5, and (d2)+0 depending on the combination of (d1)+0 and (d1)+7.

Execution/completion type (d1)+0		Data type specification (d1)+7	Processing unit and setting range		
			Number of read-target data (d1)+2	File position (d1)+4, (d1)+5	Number of data actually read (d2)+0
Reading binary data	0000H: 16-bit binary data	0: Word	Word (1 to 65535)*1	Word (00000000H to 7FFFFFFFH)	Word
		1: Even number of bytes	Word (1 to 32767)	Byte (00000000H to FFFFFFFEH)	Byte
		2: Unit of the data type specified by the execution/completion type	Word (1 to 65535)*1	Word (00000000H to 7FFFFFFFH)	Word
		3: Odd number of bytes	Word (1 to 32767)*2	Byte (00000000H to FFFFFFFEH)	Byte
	0001H: 32-bit binary data	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Double word (1 to 32767)	Double word (00000000H to 3FFFFFFFH)	Double word
		3: Odd number of bytes	(Cannot be specified)		

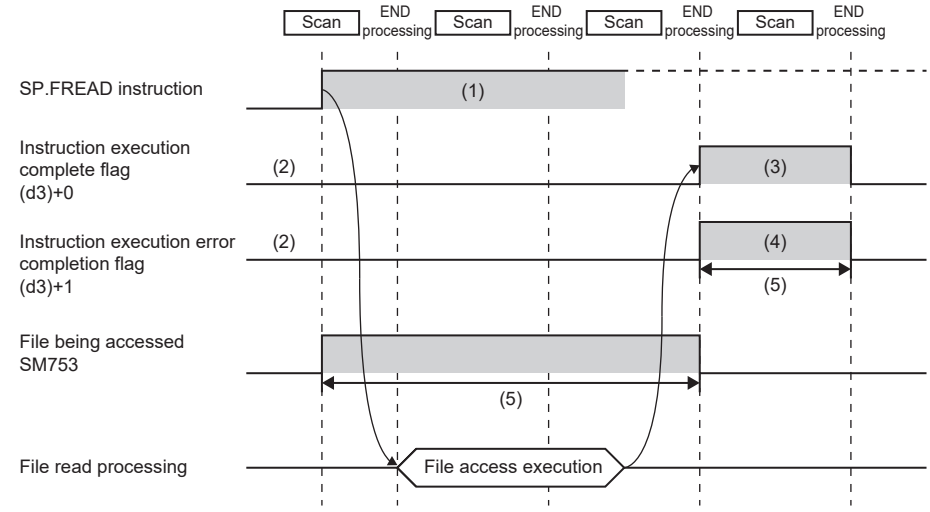
Execution/completion type (d1)+0		Data type specification (d1)+7	Processing unit and setting range		
			Number of read-target data (d1)+2	File position (d1)+4, (d1)+5	Number of data actually read (d2)+0
Reading data after converted to CSV format	0100H: Decimal (16-bit data)	0: Word	Number of elements (1 to 65535)*1	Number of lines	Word
		1: Even number of bytes	Number of elements (1 to 32767)	Number of lines	Byte
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 65535)*1	Number of lines	Word
		3: Odd number of bytes	Number of elements (1 to 32767)*2	Number of lines	Byte
	0110H: Decimal (signed 32-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 32767)	Number of lines	Double word
		3: Odd number of bytes	(Cannot be specified)		
	0120H: Hexadecimal (16-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 65535)	Number of lines	Word
		3: Odd number of bytes	(Cannot be specified)		
	0121H: Hexadecimal (32-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 32767)	Number of lines	Double word
		3: Odd number of bytes	(Cannot be specified)		
	0130H: String (ASCII data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 1023)	Number of lines	Number of elements
		3: Odd number of bytes	(Cannot be specified)		
	0140H: Floating point real number (single-precision real number)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 32767)	Number of lines	Double word
		3: Odd number of bytes	(Cannot be specified)		

*1 Up to 32767 points (when the file register (R) is used) when only devices are used.

*2 Specify the number of read-target data in word units adding 1 to the number of odd byte data to be read.

■Timing chart

Below is shown the timing chart (flag updating timing) from the execution of the SP.FREAD instruction to the completion.



(1) No processing during scanning except when the instruction is turned OFF and ON
(2) Updating to OFF when the instruction is turned OFF and ON
(3) Turning ON upon normal/error completion*1
(4) Turning OFF upon normal completion and turning ON upon error completion*1
(5) Turning ON only during 1 scan after instruction execution
(6) Turning OFF when instruction execution complete flag is turned ON

*1 The complete flag is not turned ON when an error is detected during instruction execution.

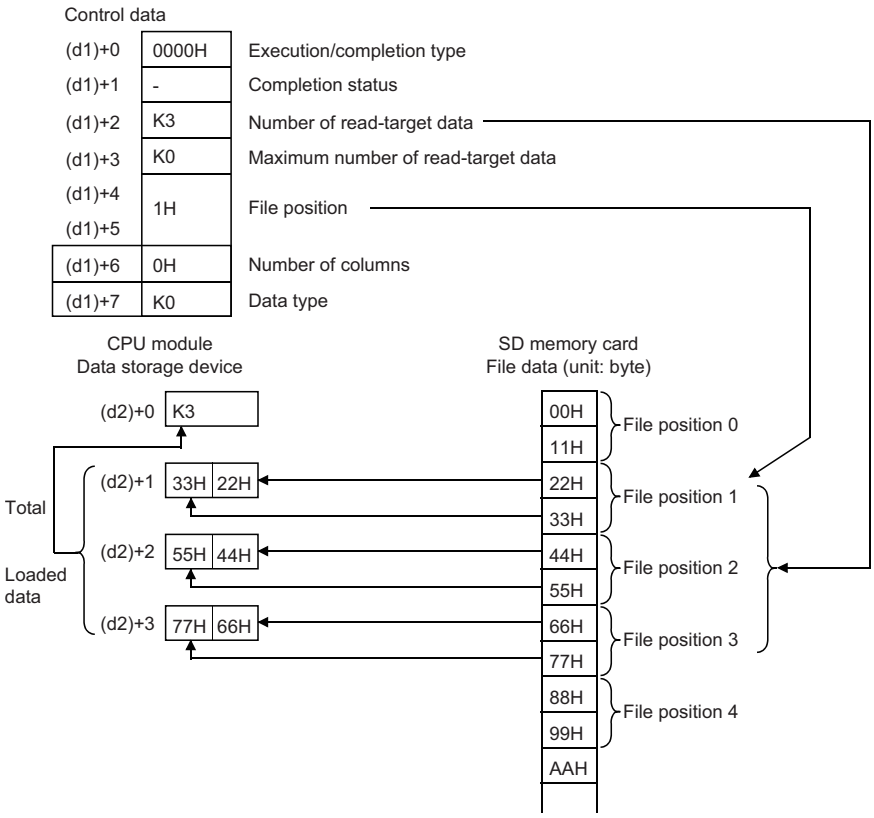
■When reading binary data

- If the extension of the target file is omitted, the extension will be ".BIN".
- If the specified file does not exist, an error (8002H) occurs.
- If the position is specified exceeding the existing file size, 0 point of data is read and the processing completes successfully.

The following figure shows an example of reading binary data.

Ex.

Example of reading binary data (16-bit binary data)



■When reading data after converted to CSV format

- Elements in the CSV format file (cells in Excel®) are read by each row and stored in the device.
- If the extension of the target file is omitted, the extension will be ".CSV".
- If the specified file does not exist, an error (8002H) occurs.
- If the position is specified exceeding the existing file size, 0 point of data is read and the processing completes successfully.
- The number of data specified by (d1)+2 is read from the beginning of the file. If the last data in the file is read before reaching to the number specified, only the available number of data is read.
- When the number of columns is set to 0, the data are read by ignoring the rows (CR+LF, only CR, only LF) in the CSV format file.
- The character code of the specified file is read as a shift JIS code when the specified execution/completion type is decimal, hexadecimal or single-precision real number.
- The string data in the CSV file and the value stored in the device after it is read are determined by the execution/completion type.

Execution/ completion type	Data type specification	Data (one element) in the CSV file	Value stored in the device	Remarks
0100H: Decimal (16-bit data)	0: Word 2: Unit of the data type specified by (d1)+0	-32768 to -1	-32768 to -1 (32768 to 65535)	The value should be within the range of -32768 to -1 as signed 16-bit data and within the range of 32768 to 65535 as unsigned 16-bit data. The same value is stored in the device.
		0 to 32767	0 to 32767	—
		32768 to 65535	-32768 to -1 (32768 to 65535)	The value should be within the range of -32768 to -1 as signed 16-bit data and within the range of 32768 to 65535 as unsigned 16-bit data. The same value is stored in the device.
		• Numeric values other than above • String containing alphabets and symbols	0	Filled with 0 since it cannot be converted.
	1: Even number of bytes 3: Odd number of bytes	-128 to -1	-128 to -1 (128 to 255)	The value should be within the range of -128 to -1 as signed 8-bit data and within the range of 128 to 255 as unsigned 8- bit data. The same value is stored in the device.
		0 to 127	0 to 127	—
		128 to 255	-128 to -1 (128 to 255)	The value should be within the range of -128 to -1 as signed 8-bit data and within the range of 128 to 255 as unsigned 8- bit data. The same value is stored in the device.
		• Numeric values other than above • String containing alphabets and symbols	0	Filled with 0 since it cannot be converted.
0110H: Decimal (32-bit data)	—	-2147483648 to -1	-2147483648 to -1 (2147483648 to 4294967295)	The value should be within the range of -2147483648 to -1 as signed 32-bit data and within the range of 2147483648 to 4294967295 as unsigned 32-bit data. The same value is stored in the device.
		0 to 2147483647	0 to 2147483647	—
		2147483648 to 4294967295	-2147483648 to -1 (2147483648 to 4294967295)	The value should be within the range of -2147483648 to -1 as signed 32-bit data and within the range of 2147483648 to 4294967295 as unsigned 32-bit data. The same value is stored in the device.
		• Numeric values other than above • String containing alphabets and symbols	0	Filled with 0 since it cannot be converted.

Execution/ completion type	Data type specification	Data (one element) in the CSV file	Value stored in the device	Remarks
0120H: Hexadecimal (16-bit data)	—	0H to FFFFH	0H to FFFFH	—
		• Numeric values other than above • String containing alphabets other than A to F and symbols	0000H	Filled with 0 since it cannot be converted.
0121H: Hexadecimal (32-bit data)	—	0H to FFFFFFFFH	0H to FFFFFFFFH	—
		• Numeric values other than above • String containing alphabets other than A to F and symbols	00000000H	Filled with 0 since it cannot be converted.
0130H: String (ASCII data)	—	String (up to 1999 characters)	String (up to 1999 characters)	When the number of bytes of the string in the CSV file is odd, NULL (00H) is added to the end of the string. When the number of bytes of the string in the CSV file is even, 0000H is stored in the next one word. When the string in the CSV file contains 00H, it is ignored.
		String (2000 characters or more)		If the number of characters in one element exceeds 1999, characters until the 1999th character are read as one element. The 2000th character and after are not read and the next element is read.
0140H: Floating point real number (single-precision real number)	—	Values within the range of: $-2^{128} < data \leq -2^{-126}$, 0, $2^{-126} \leq data < 2^{128}$	As given on the left	The value is converted in either decimal point or exponential format.
		Numeric values other than above	0	Filled with 0 since it cannot be converted.

Ex.

When "Reading data after converted to CSV format" (String (ASCII data)) is specified

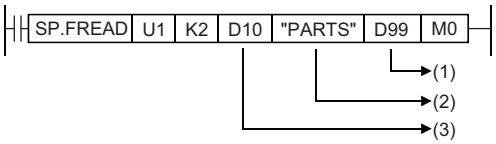
[Data stored in CSV format]

PARTS.CSV

No.	,	Name	,	Value1	,	Value2	CR	LF
AA_0001	,	Prts_A	,	100	,	200	CR	LF
BB_0002	,	Prts_B	,	300	,	400	CR	LF

Enclosed values as shown to the left are read.
(Total of six elements, three columns from two lines of PARTS.CSV, are read.)

[Data to be loaded to the device]



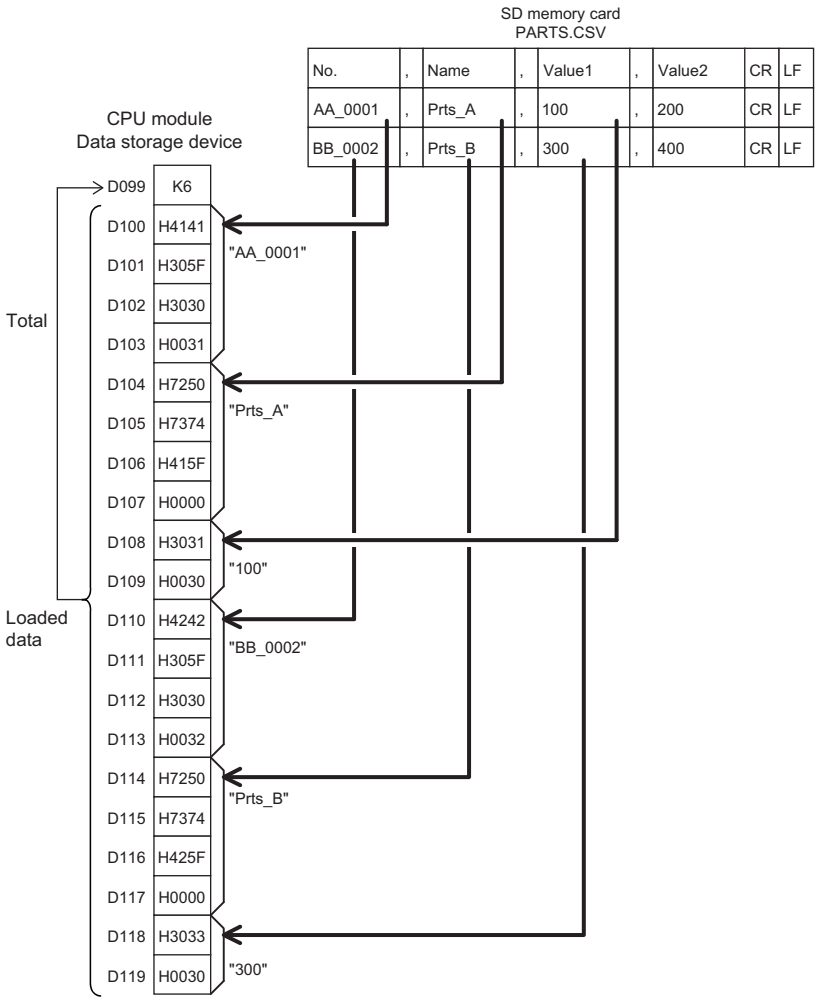
(1) Start device for storing the data that has been read
(2) File name
(3) Control data

[Control data]

D10	H0130
D11	H0000
D12	K6
D13	K100
D14	2H
D15	
D16	3H
D17	K2

D10: Execution/completion type
D11: Completion status
D12: Number of read-target data
D13: Maximum number of read data
D14, D15: File position
D16: Number of columns
D17: Data type specification

[Loaded data]



D99: Number of data actually read
D100 to D103: String in the 1st column of the 2nd line
D104 to D107: String in the 2nd column of the 2nd line
D108, D109: String in the 3rd column of the 2nd line
D110 to D113: String in the 1st column of the 3rd line
D114 to D117: String in the 2nd column of the 3rd line
D118, D119: String in the 3rd column of the 3rd line

Point

When String (ASCII data) is specified, it is necessary to set the total size (in word units) for the maximum number of read data ((d1)+3).

Example: When a string with 100 100-character elements is read from a CSV file, the size is as shown below.

• (100 (character) + 2 (NULL)) × 100 (element) = 10200 byte = 5100 word

In the above case, an area for 5101 words must be secured for the file name (s2).

Ex.

When "Reading data after converted to CSV format" (Decimal (16-bit data)) is specified and the number of columns is set to 0

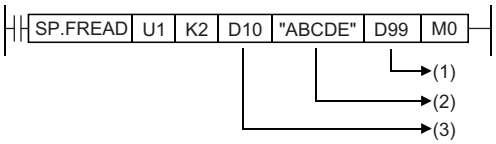
[Data created in Excel]

	A	B	C
1	Main/sub item		Measured value
2	Length	1	3
3	Temperature	-21	
4			

[Data saved in CSV format]
ABCD.CSV

Main/sub item	,	,	Measured value				CR	LF	
Length	,	1	,	3	CR	LF			
Temperature	,	-21	,		CR	LF			

[Data to be loaded to the device]



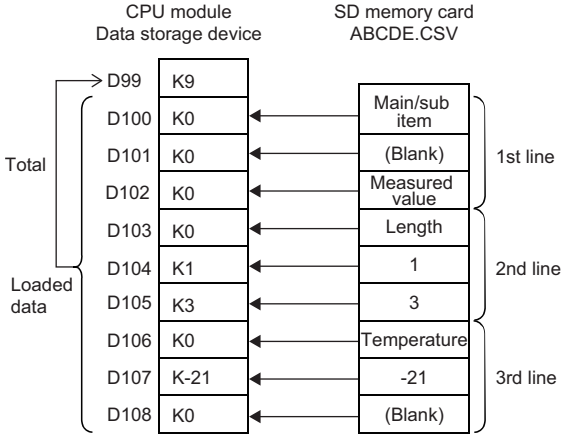
(1) Start device for storing the data that has been read
(2) File name
(3) Control data

[Control data]

D10	0100H
D11	-
D12	K9
D13	K0
D14	0H
D15	
D16	0H
D17	K0

D10: Execution/completion type
D11: Completion status
D12: Number of read-target data
D13: Maximum number of read data
D14, D15: File position
D16: Number of columns
D17: Data type specification

[Loaded data]



D99: Number of data actually read
D100: Since "Main/sub item" is not a numerical value, the conversion data, 0, is stored.
D101: Since " " is not a numerical value, the conversion data, 0, is stored.
D102: Since "Measured value" is not a numerical value, the conversion data, 0, is stored.
D103: Since "Length" is not a numerical value, the conversion data, 0, is stored.
D104: Since "1" is a numerical value, it is converted to a binary value.
D105: Since "3" is a numerical value, it is converted to a binary value.
D106: Since "Temperature" is not a numerical value, the conversion data, 0, is stored.
D107: Since "-21" is a numerical value, it is converted to a binary value.
D108: Since " " is not a numerical value, the conversion data, 0, is stored.

• When the number of columns differs in each row, the data are also read by ignoring the rows.

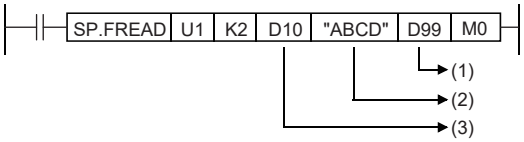
Ex.

When the number of columns differs in each row

[Data saved in CSV format]
ABCD.CSV

Main/sub item	,	,	Measured value	,	Excess	CR	LF	
Length	CR	LF						
Temperature	,	-21	,	CR	LF			

[Data to be loaded to the device]



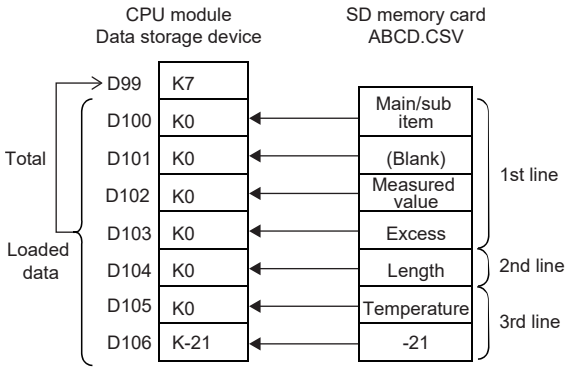
(1) Start device for storing the data that has been read
(2) File name
(3) Control data

[Control data]

D10	0100H
D11	-
D12	K7
D13	K0
D14	0H
D15	
D16	0H
D17	K0

D10: Execution/completion type
D11: Completion status
D12: Number of read-target data
D13: Maximum number of read data
D14, D15: File position
D16: Number of columns
D17: Data type specification

[Loaded data]



D99: Number of data actually read
D100: Since "Main/sub item" is not a numerical value, the conversion data, 0, is stored.
D101: Since " " is not a numerical value, the conversion data, 0, is stored.
D102: Since "Measured value" is not a numerical value, the conversion data, 0, is stored.
D103: Since "Excess" is not a numerical value, the conversion data, 0, is stored.
D104: Since "Length" is not a numerical value, the conversion data, 0, is stored.
D105: Since "Temperature" is not a numerical value, the conversion data, 0, is stored.
D106: Since "-21" is a numerical value, it is converted to a binary value.

Point

This type of file in which the number of columns vary with individual rows cannot be created by Excel. It is created when the CSV file is modified by a user.

- When the specified number of columns is set to a value other than 0, a CSV format file is read as the table with the specified number of columns. The elements outside the specified number of columns are ignored.

Ex.

When "Reading data after converted to CSV format" (Decimal (16-bit data)) is specified and the number of columns is set to a value other than 0 ((d1)+6 is set to 2)

[Data created in Excel]

	A	B	C
1	Main/sub item		Measured value
2	Length	1	3
3	Temperature	-21	
4			

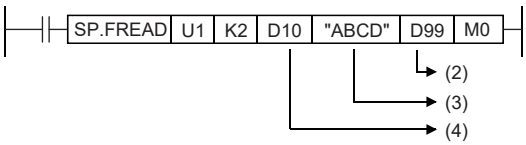
[Data saved in CSV format]
ABCD.CSV

Main/sub item	,	,	Measured value			CR	LF		
Length	,	1	,	3		CR	LF		
Temperature	,	-21				CR	LF		

(1)

(1) Columns out of the specified range are ignored.

[Data to be loaded to the device]



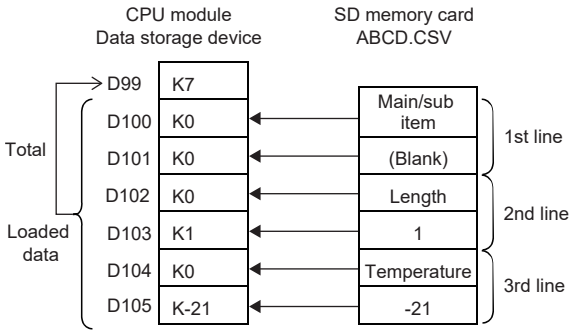
(2) Start device for storing the data that has been read
(3) File name
(4) Control data

[Control data]

D10	0100H
D11	-
D12	K6
D13	K0
D14	0H
D15	
D16	2H
D17	K0

D10: Execution/completion type
D11: Completion status
D12: Number of read-target data
D13: Maximum number of read data
D14, D15: File position
D16: Number of columns
D17: Data type specification

[Loaded data]



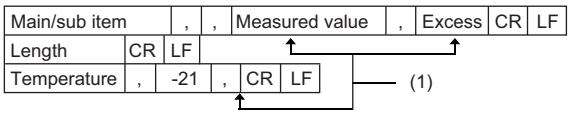
D99: Number of data actually read
D100: Since "Main/sub item" is not a numerical value, the conversion data, 0, is stored.
D101: Since " " is not a numerical value, the conversion data, 0, is stored.
D102: Since "Length" is not a numerical value, the conversion data, 0, is stored.
D103: Since "1" is a numerical value, it is converted to a binary value.
D104: Since "Temperature" is not a numerical value, the conversion data, 0, is stored.
D105: Since "-21" is a numerical value, it is converted to a binary value.

• When the number of columns differs in each row, the elements outside the specified number of columns are ignored and 0 is added to the cells where no element exists.

Ex.

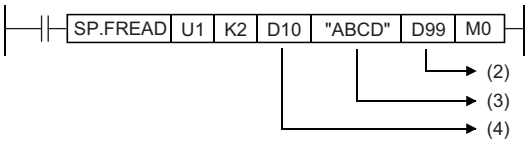
When the number of columns differs in each row

[Data saved in CSV format]
ABCD.CSV



(1) Columns out of the specified range are ignored.

[Data to be read into devices]



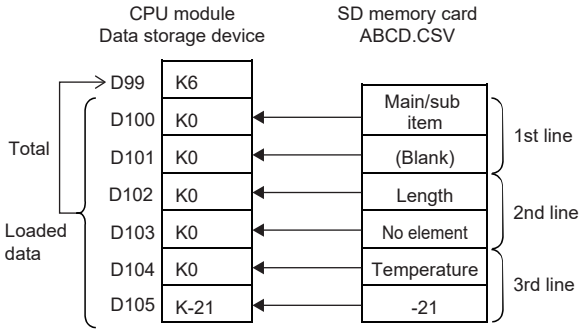
(2) Start device for storing the data that has been read
(3) File name
(4) Control data

[Control data]

D10	0100H
D11	-
D12	K6
D13	K0
D14	0H
D15	
D16	2H
D17	K0

D10: Execution/completion type
D11: Completion status
D12: Number of read-target data
D13: Maximum number of read data
D14, D15: File position
D16: Number of columns
D17: Data type specification

[Loaded data]

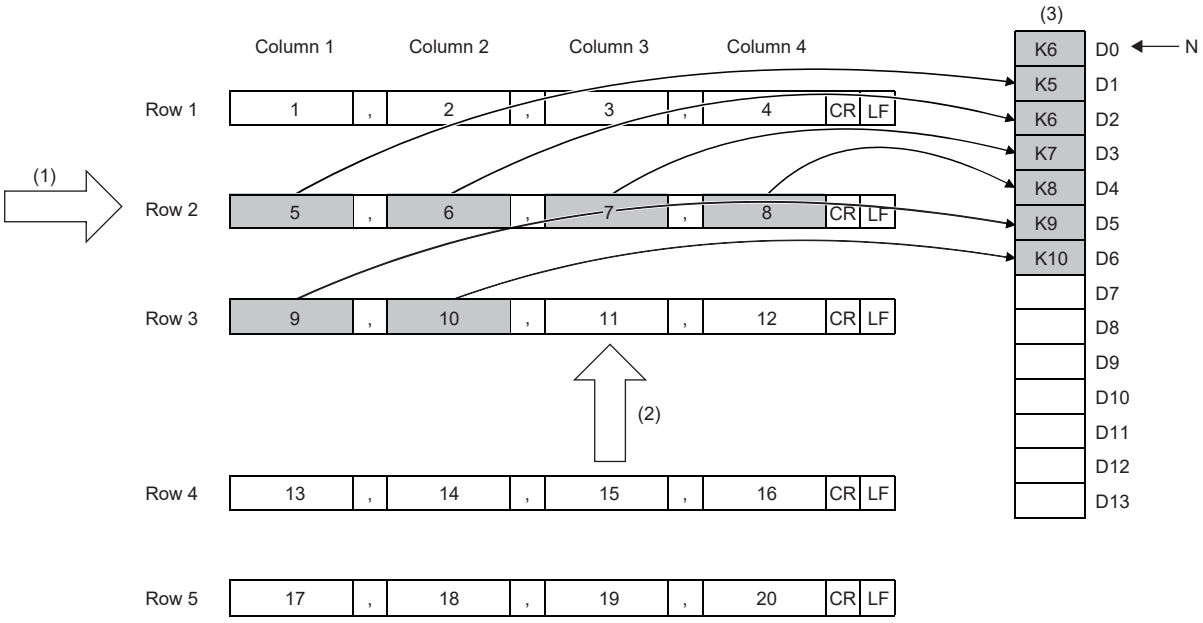


D99: Number of data actually read
D100: Since "Main/sub item" is not a numerical value, the conversion data, 0, is stored.
D101: Since " " is not a numerical value, the conversion data, 0, is stored.
D102: Since "Length" is not a numerical value, the conversion data, 0, is stored.
D103: Since no element exists, the conversion data, 0, is added.
D104: Since "Temperature" is not a numerical value, the conversion data, 0, is stored.
D105: Since "-21" is a numerical value, it is converted to a binary value.

• When "Reading data after converted to CSV format" is specified, data can be divided and read.

[Specify the row to start reading]

- Execution/completion type: Reading data after converted to CSV format (Decimal (16-bit data))
- Number of read-target data: K6
- File position: 2H
- Number of columns specification: 4H
- Data type specification: Words
- Read start device: D0
- Number of data actually read: 6H



(1) Starting row

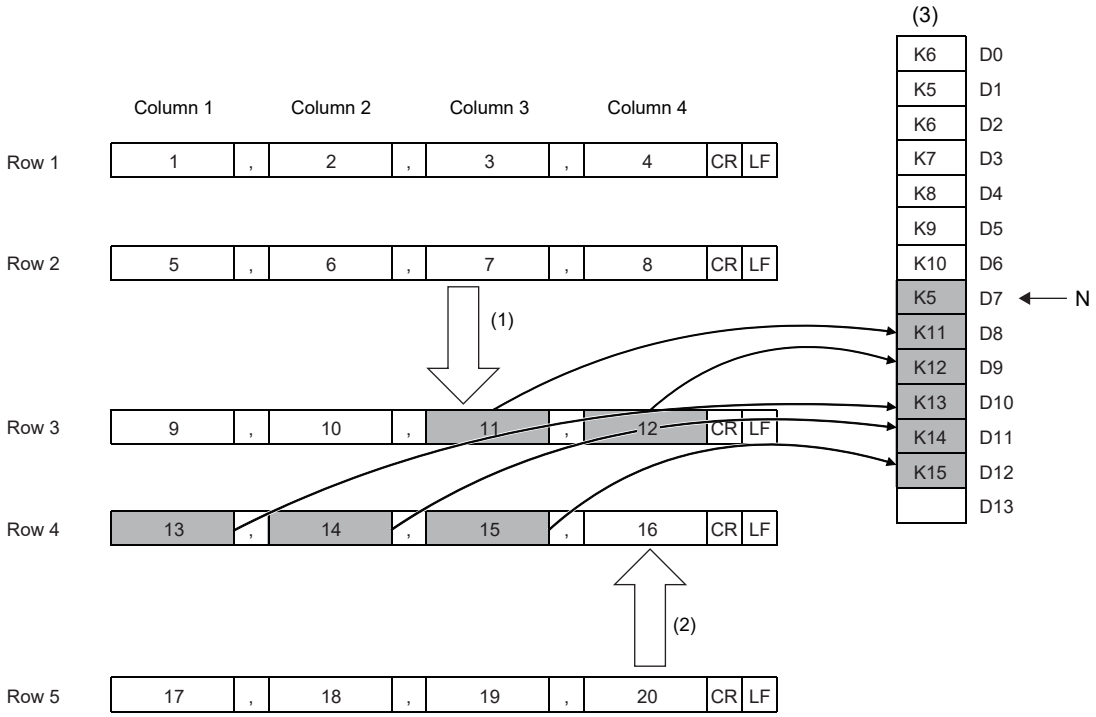
(2) Next starting position

(3) Data in the device (loaded data)

N: Number of data

[Continuing from the last read position]

- Execution/completion type: Reading data after converted to CSV format (Decimal (16-bit data))
- Number of read-target data: K5
- File position: FFFFFFFFH (continuing from the position where the previous reading ends)
- Number of columns specification: 4H
- Data type specification: Words
- Read start device: D7
- Number of data actually read: 5H



(1) Starting row

(2) Next starting position

(3) Data in the device (loaded data)

N: Number of data

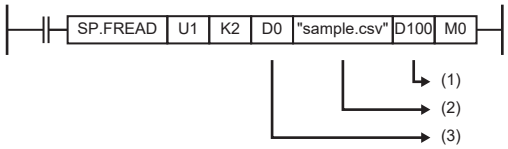
Point

- When reading data from the position where the previous reading ends, specify the same values for "Execution/End type", "Number of columns", and "Data type specification". If not, data cannot be added correctly from the position where the previous reading ends.
- While reading data from the position where the previous reading ends, if the SP.FREAD instruction with different settings or the SP.FWRITE instruction is executed, data cannot be added correctly from the position where the previous reading ends.
- If another file is specified when data is read continuously from the last read position, the data cannot be normally read from the last read position.
- If data is read continuously from the last read position at the time of initial file reading (for example after the power is turned on the CPU module), the data will be read from the beginning of the file position.
- When reading data after converted to CSV format (string (ASCII data)) has been specified, if the number of read data exceeds the maximum number of read data, the file position will be initialized to the beginning of file even if the reading from the last read position is continued.

Ex.

When out-of-range value or string data is read

[Data to be read into devices]



(1): Start device for storing the data that has been read
- D100: Number of data actually read
- D101 to D120: Read data

(2): File name
- "sample.csv"

(3): Control data

[Control data]

D0	0100H
D1	-
D2	K20
D3	K0
D4	K2
D5	
D6	K5
D7	K2

D0: Execution/completion type
D1: Completion status
D2: Number of read-target data
D3: Maximum number of read data
D4, D5: File position
D6: Number of columns
D7: Data type specification

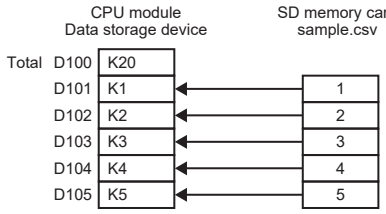
[SD memory card]
sample.csv

	A	B	C	D	E	F
1	1	2	3	4	5	6
2	1	2	3	4	5	6
3	A	B	C	D	E	F
4	-32769	-32768	65535	65536	32768	-32769
5	A	B	C	D	1.11E+16	F
6	G	H	I	J	2.21E+16	L

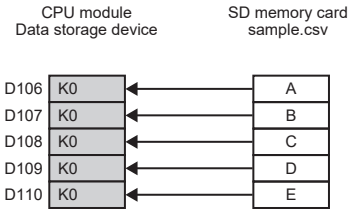
The data in the five columns from the second row are transferred to D101 to D120.

[Loaded data]

■sample.csv 2nd line

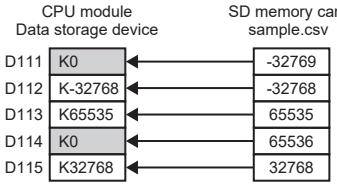


■sample.csv 3rd line



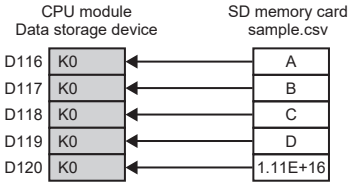
0 is stored in D106 to D110 because all are strings.

■sample.csv 4th line



0 is stored in D111 and D114 because -32769 and 65536 are out of the range.

■sample.csv 5th line



0 is stored in D116 to D120 because all are strings or out of the range.

Point

If values that cannot be expressed in word units in decimal (values out of the range of -32768 to 65535, strings, etc.) are set in the elements of the CSV file, all are converted to 0 and stored in the corresponding devices. Even if 0 is stored due to out-of-range values, they are included in the number of data actually read.

Program example

An example of a program for reading binary data/CSV data from the SD memory card by using the SP.FREAD instruction is shown below.

■Reading binary data

When X0 is turned ON, 100 data starting from the 16th word from the beginning of the “sample.bin” file stored in the SD memory card are stored in D101 to D200.

[Program operation]

1. Control data is created during RUN.
2. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
3. The SP.FREAD instruction is executed.*1
4. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*1 The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.
- SM606 (Memory card disable request)
- SM753 (File being accessed)

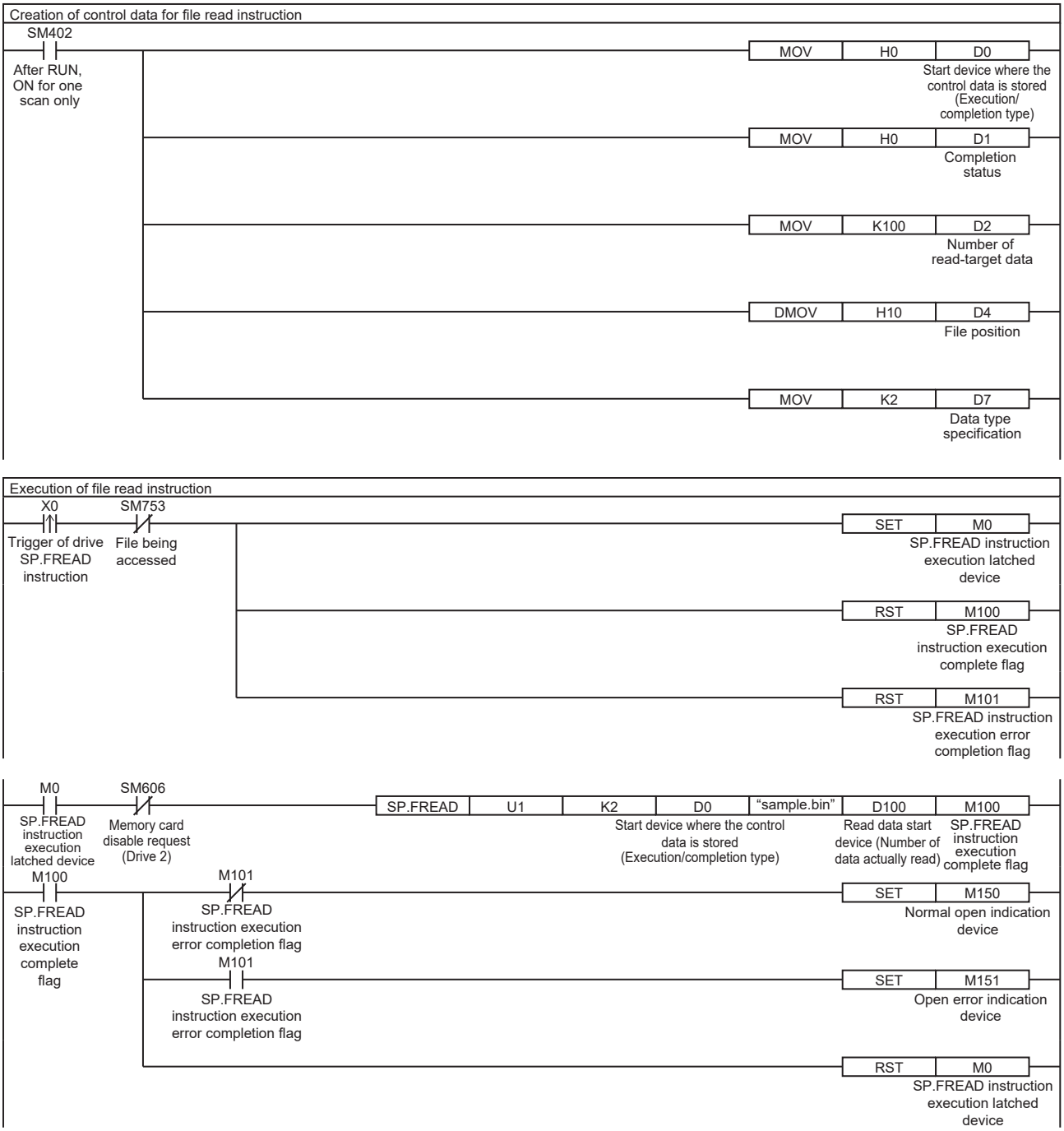
[Devices used]

Device	Description
X0	Trigger of drive SP.FREAD instruction
D0	Start device where the control data is stored • D0: Execution/completion type • D1: Completion status • D2: Number of read-target data • D4, D5: File position • D7: Data type specification
D100	Start device for storing the data that has been read • D100: Number of data actually read • D101, D200: Read data
M0	SP.FREAD instruction execution latched device
M100	SP.FREAD instruction execution complete flag
M101	SP.FREAD instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FREAD instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(s1)	Drive specification	K2 (SD memory card)
(d1)	Start device where the control data is stored	D0: 0H (16-bit binary data) D1: 0H (Completed successfully) D2: K100 (100 words are read.) D4, D5: 10H (Data after the 16th word from the beginning are read.) D7: K2 (Unit of data type specified in D0)
(s2)	Start device where the file name is stored	“sample.bin”
(d2)	Start device for storing the data that has been read	D100
(d3)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]



[ST program]

```
//(1) Creation of control data for file read instruction
IF SM402 THEN
D0 := H0; //Execution/completion type (Control data start device)
D1 := H0; //Completion status
D2 := 100; //Number of read-target data
D4:UD := H10; //File position
D7 := 2; //Data type specification
END_IF;

//(2) Processing to start up the drive contact (X0)
IF LDP(TRUE, X0) THEN
//Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE, M0); //Holds drive contact
RST(TRUE, M100); //Initialize instruction execution complete flag
RST(TRUE, M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(3) Execution of file read instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "sample.bin"(Start device where the file name is stored)
//D1 = D0 (Start device where the control data is stored)
//D2 = D100 (Start device for storing the data that has been read)
//D3 = M100 (Bit device that turns on upon completion of the processing)
SP_FREAD(TRUE, U1, 2, "sample.bin", D0, D100, M100);
END_IF;
END_IF;

//(4) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```

[Execution result]

When the target binary data is read by using the program example, the results are as shown below.

[Read target (sample.bin)]

sample.bin

ADRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	ZZZZZZZZZZZZZZZZ
00000010	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	ZZZZZZZZZZZZZZZZ
00000020	E9	03	EA	03	EB	03	EC	03	ED	03	EE	03	EF	03	F0	03	. . . . .
00000030	F1	03	F2	03	F3	03	F4	03	F5	03	F6	03	F7	03	F8	03	. . . . .
00000040	F9	03	FA	03	FB	03	FC	03	FD	03	FE	03	FF	03	00	04	. . . . .
00000050	01	04	02	04	03	04	04	04	05	04	06	04	07	04	08	04	. . . . .
00000060	09	04	0A	04	0B	04	0C	04	0D	04	0E	04	0F	04	10	04	. . . . .
00000070	11	04	12	04	13	04	14	04	15	04	16	04	17	04	18	04	. . . . .
00000080	19	04	1A	04	1B	04	1C	04	1D	04	1E	04	1F	04	20	04	. . . . .
00000090	21	04	22	04	23	04	24	04	25	04	26	04	27	04	28	04	! . " . # . $. % . & . ' . (.
000000A0	29	04	2A	04	2B	04	2C	04	2D	04	2E	04	2F	04	30	04	) . * . + . , . - . . . / . 0.
000000B0	31	04	32	04	33	04	34	04	35	04	36	04	37	04	38	04	1. 2. 3. 4. 5. 6. 7. 8.
000000C0	39	04	3A	04	3B	04	3C	04	3D	04	3E	04	3F	04	40	04	9. : . ; . <. =. >. ?. @.
000000D0	41	04	42	04	43	04	44	04	45	04	46	04	47	04	48	04	A. B. C. D. E. F. G. H.
000000E0	49	04	4A	04	4B	04	4C	04									I . J . K. L.

(1)

[Read result]*1

Device Name	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Current Value	String
D99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	..
D100	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0064	d.
D101	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	1	03E9	)064
D102	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	0	03EA	
D103	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	1	03EB	
D104	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0	03EC	
D105	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	1	03ED	
																		(2)
D196	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0048	H.
D197	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0049	I.
D198	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	004A	J.
D199	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	044B	K.
D200	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	044C	L.
D201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	..
D202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	..

(1) 100 words from 03E9 to 044C are described. In the program example, the data is read starting from the 16th word, and so 03E9 (boxed) is stored in D101.

(2) In the program example, 100 words of data are read, and so the number of data actually read is stored in D100, and the read data are stored in D101 to D200.

*1 Device/Buffer Memory Batch Monitor (hexadecimal display) of engineering tool

■Reading data after converted to CSV format

When X0 is turned ON, 20 words of data in the five columns in the second to fifth rows in the “sample.csv” file stored in the SD memory card are stored in D101 to D120.

[Program operation]

1. Control data is created during RUN.
2. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
3. The SP.FREAD instruction is executed.*1
4. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*1 The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.
- SM606 (Memory card disable request)
- SM753 (File being accessed)

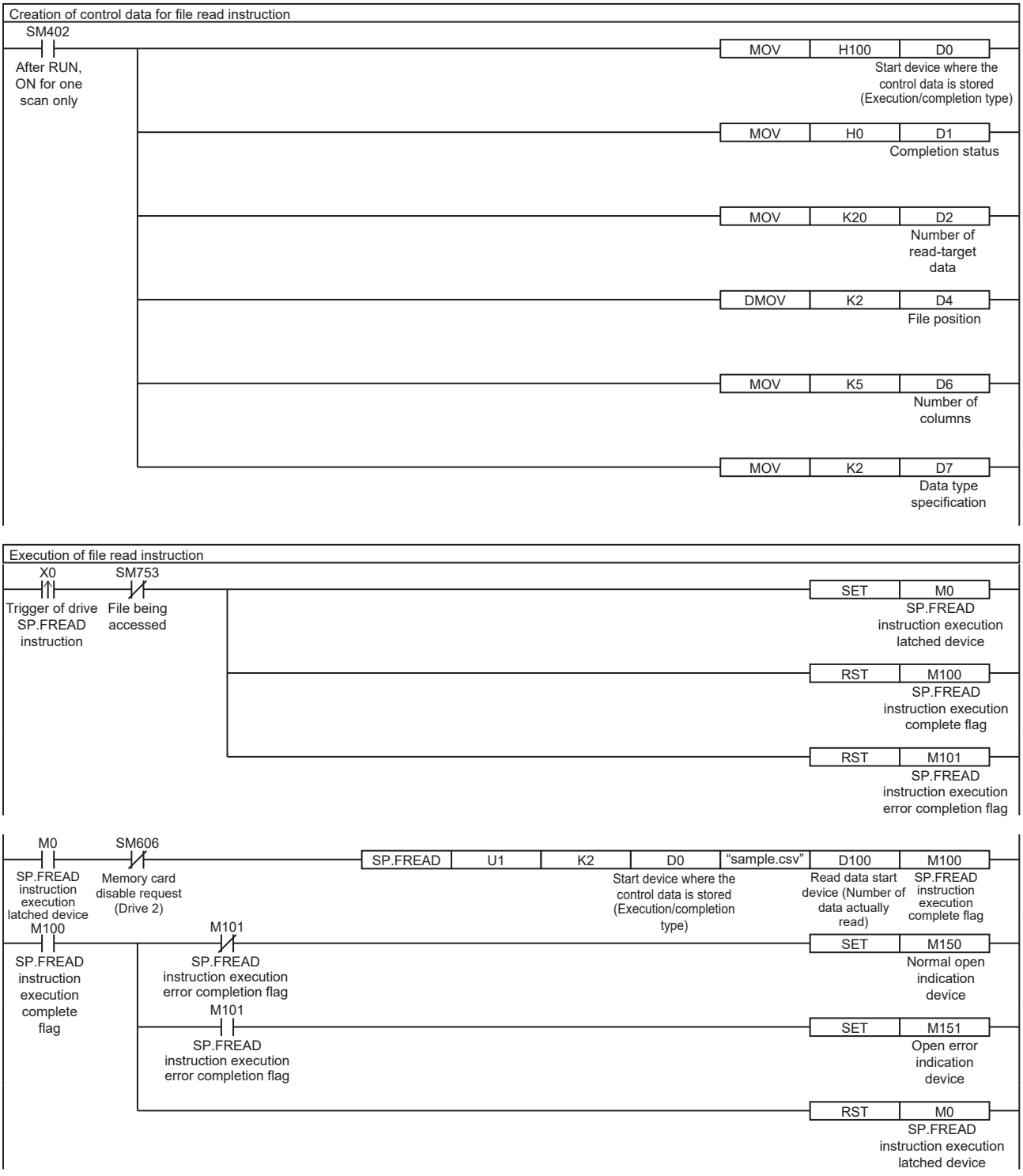
[Devices used]

Device	Description
X0	Trigger of drive SP.FREAD instruction
D0	Start device where the control data is stored • D0: Execution/completion type • D1: Completion status • D2: Number of read-target data • D4,D5: File position • D6: Number of columns • D7: Data type specification
D100	Start device for storing the data that has been read • D100: Number of data actually read • D101 to D120: Read data
M0	SP.FREAD instruction execution latched device
M100	SP.FREAD instruction execution complete flag
M101	SP.FREAD instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FREAD instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(s1)	Drive specification	K2 (SD memory card)
(d1)	Start device where the control data is stored	D0: 0H (16-bit binary data) D1: 0H (Completed successfully) D2: K20 (20 words are read.) D4,D5: K2 (Reading from the second row) D6: K5 (Reading the data in 5 columns) D7: K2 (Unit of data type specified in D0)
(s2)	Start device where the file name is stored	“sample.csv”
(d2)	Start device for storing the data that has been read	D100
(d3)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]



[ST program]

```
//(1) Creation of control data for file read instruction
IF SM402 THEN
D0 := H100; //Execution/completion type (Control data start device)
D1 := H0; //Completion status
D2 := 20; //Number of read-target data
D4:UD := 2; //File position
D6 := 5; //Number of columns
D7 := 2; //Data type specification
END_IF;

//(2) Processing to start up the drive contact (X0)
IF LDP(TRUE, X0) THEN
// Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE, M0); //Holds drive contact
RST(TRUE, M100); //Initialize instruction execution complete flag
RST(TRUE, M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(3) Execution of file read instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "sample.csv" (Start device where the file name is stored)
//D1 = D0 (Start device where the control data is stored)
//D2 = D100 (Start device for storing the data that has been read)
//D3 = M100 (Bit device that turns on upon completion of the processing)
SP_FREAD(TRUE, U1, 2, "sample.csv", D0, D100, M100);
END_IF;
END_IF;

//(4) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```

[[Execution result]

When the target CSV data is read by using the program example, the results are as shown below.

[Read target (sample.bin)]

	A	B	C	D	E	F
1	1	2	3	4	5	6
2	1	2	3	4	5	(1) 6
3	A	B	C	D	E	F
4	-32769	-32768	65535	65536	32768	-32769
5	G	H	I	J	1.11E+16	
6	M	N	O	P	2.22E+16	R

[Read result]*1

Device Name	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Current Value	String
D100	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0		20	
D101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		1	
D102	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0		2	
D103	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1		3	
D104	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0		4	
D105	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1		5	
D106	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
D107	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
D108	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
D109	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	(2)
D110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
D111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
D112	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		-32768	
D113	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		-1	□□
D114	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
D115	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		-32768	
D116	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
D117	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
D118	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
D119	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
D120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	

(1) In the program example, five columns in each row from the second row are read, and so 1 (boxed) is stored in D101.

(2) In the program example, 20 words of data are read, and so the number of data actually read is stored in D100, and the read data are stored in D101 to D120. For values/strings that cannot be read in the execution/completion type format, the stored device values of the read data are 0. Even if the device values are 0, they are included in the number of read data.

*1 Device/Buffer Memory Batch Monitor (decimal display) of engineering tool

Precautions

- The SP.FREAD instruction cannot be executed in user interrupt programs. An error (3582H) will occur.
- The SP.FREAD instruction cannot be executed while SM606 (Memory card disable request) is ON. When SM606 is turned ON during execution of the instruction, the program will terminate abnormally. (The data that has been read before SM606 is turned ON will be stored in the devices.)
- When reading multiple elements at reading data after converted to CSV format, secure devices sufficient for the total size of the elements in the read data area before executing the instruction. Since read data is stored from (d2)+1, the number of words required to be secured as (d2) is ((total number of words for each element) + 1) words.
- The files in the system folder ($MELPRJ$) cannot be handled by the SP.FREAD instruction. If an access to any file in the system folder ($MELPRJ$) is requested, a calculation error (3405H) will occur.
- If the storage device exceeds the end of the device range, an error (2820H) will occur.
- The SP.FREAD instruction cannot be executed simultaneously with the SP.DEVST instruction, the SP.FTPPUT instruction, and the SP.FTPGET.
- Do not disconnect the power or remove the SD memory card during execution of the SP.FREAD instruction. (The file may be damaged, or an error may occur.)

Point

When 16-bit data (0000H, 0100H or 0120H) has been specified for Execution/completion type (d1)+0, the setting range of the number of read-target data (d1)+2 is as shown below.

- Up to 64512 points when only label is used.
- Up to 32767 points (when the file register (R) is used) when only device is used.

Operation error

Error code (SD0/SD8067)	Description
2820H	The storage device of the control data (d1) exceeds the end of the device range.
	The bit label digits specified in (s2) and (d2) are unacceptable settings (the number of digits is not K4).
	The size of the read data exceeds the read device size.
3405H	The drive specified by (s1) is not the one for the SD memory card.
	Any value that is set in the device specified by (d1) and later as control data is out of the range.
	The file name string specified by (s2) cannot be read. • The number of characters of the string in the file name specified exceeds the range. • An inhibited value is set. • The specified file name string ends with a delimiter. • The system folder ($MELPRJ$) directly under the route folder is specified. • A half-width period is specified at the end of the specified file name string or immediately before each delimiter.
3427H	An invalid combination of (d1)+0 (Execution/completion type) and (d1)+7 (Data type specification) is specified.
3582H	The SP.FREAD instruction is executed in an interrupt program.

If the SP.FREAD instruction completes with an error, an error code is stored in the device specified by (d1)+1. (Note that an error code is not stored if the instruction results in an operation error.)

For the error code stored in (d1)+1, refer to the following.

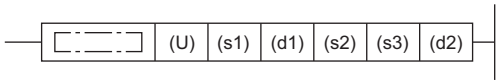
☞ Page 579 Error codes generated for file operation instructions

Writing data to the specified file

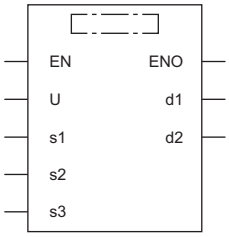
SP.FWRITE

FX5S FX5UJ FX5U FX5UC

This instruction writes device data to the specified file in an SD memory card.

Ladder	ST
	ENO:=SP_FWRITE(EN,U,s1,s2,s3,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)	Dummy	■FX5S CPU module U1 ■FX5UJ CPU module U1 to U8 ■FX5U/FX5UC CPU module U1 to U10	Device name*2	ANY16
(s1)	Drive specification	2 (fixed)*1	16-bit signed binary	ANY16
(d1)	Start device where the control data is stored	☞ Page 513 Control data (d1)	Word	ANY16_ARRAY (Number of elements: 8)
(s2)	Start device where the file name is stored	☞ Page 514 File name (s2)	Unicode string	ANYSTRING_DOUBLE
(s3)	Head device where the write-target data is stored	☞ Page 515 Write data (s3)	Word	ANY16*3
(d2)	Bit device that turns on upon completion of the processing	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

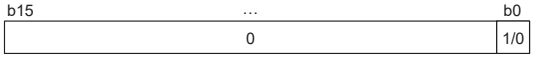
*1 Only drive 2 (for the SD memory card) can be set.
*2 Specification with a label is not allowed.
*3 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	○	—	—	—	—	○	○	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	○	—
(s3)	○*1	○	—	—	—	—	○	—	—	—	—
(d2)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 If the bit device digit is specified in (s3), only multiples of 16 (0, 16, 32, 64…) can be specified as the device number. Only K4 can be specified as the number of digits.
*2 S cannot be used.
*3 T, ST, and C cannot be used.
Only bit specification of word device is applicable.

■Control data (d1)

Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Execution/completion type	Specify the execution type. ■00**H: Writing binary data • 0000H: 16-bit binary data • 0001H: 32-bit binary data ■01**H: Writing data after converted to CSV format • 0100H: Decimal (signed 16-bit data) • 0101H: Decimal (unsigned 16-bit data) • 0110H: Decimal (signed 32-bit data) • 0111H: Decimal (unsigned 32-bit data) • 0120H: Hexadecimal (16-bit data) • 0121H: Hexadecimal (32-bit data) • 0130H: String (ASCII data) • 0140H: Floating point real number (single-precision real number)	0000H 0001H 0100H 0101H 0110H 0111H 0120H 0121H 0130H 0140H	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0000H: Completed successfully • Other than 0000H: Completed with an error (error code) (☞ Page 579 Error codes generated for file operation instructions)	—	System
+2	Number of data actually written	The number of actually written data for the data specified in (s3) is stored. The unit of the value conforms to the data type specification in (d1)+7.	—	System
+3	Application setting area	 b0: Write start position setting Specify the write start position.*1 • 0: Add to the end of the file • 1: Add converting the last line feed code of file to a comma (write continuing the last line)	—	—
+4 +5	File position	■When "Writing binary data" is specified by (d1)+0 • 00000000H: From the beginning of the file • 00000001H to FFFFFFFEH: From the specified position (The unit conforms to the data type specification in (d1)+7.) • FFFFFFFFH: Added to the end of the file. ■When "Converting and writing data in CSV format" is specified by (d1)+0 • 00000000H to FFFFFFFEH: From the beginning of the file • FFFFFFFFH: Added to the end of the file.	00000000H to FFFFFFFFH	User
+6	Number of columns	When "Writing binary data" is specified by (d1)+0, set 0. When "Converting and writing data in CSV format" is specified by (d1)+0, specify the number of columns to write. • 0: No specification of number of columns. Only one row is written. • Other than 0: Number of specified columns	0000H to FFFFH (0 to 65535)	User

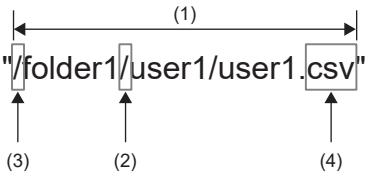
Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+7	Data type specification	• 0: Word • 1: Even number of bytes*2 • 2: Unit of the data type specified by (d1)+0 • 3: Odd number of bytes*2*3 "0: Word", "1: Even number of bytes", and "3: Odd number of bytes" can be specified only when "0000H: 16-bit binary data" or "0100H: Decimal (signed 16-bit data)" is specified by (d1)+0.	0, 1, 2, 3	User

*1 The setting is enabled only when “01**H: Writing data after converted to CSV format” has been selected for execution/completion type (d1)+0 and “FFFFFFFFH: Added to the end of the file” has been specified for file position (d1)+4 and (D1)+5. When the setting is disabled, specify “0.”

*2 When "1: Even number of bytes" or "3: Odd number of bytes" is specified by (d1)+7 (Data type specification), the setting range of (s3)+0 (Number of request write data) is 1 to 32767.

*3 Even when "3: Odd number of bytes" is specified by (d1)+7 (Data type specification), specify the number in units of words. Add one byte to the odd number of bytes to be written and set the number in (s3)+0 (Number of request write data).

■File name (s2)

Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +☐	File name character string	Specify the folder path where the files are stored and the file name. • The folder path and file name (including an extension) must be within 253 characters in total. • The folder path must be within 244 characters. (Delimiters are not included.) • The number of folder path hierarchies must be within 10 levels. • When omitting an extension in the file name, omit the ". (period)" as well. • The file name must be within 60 characters (a period and extension excluded). If 61 or more characters are used, the extension is ignored and replaced with ".BIN" or ".CSV". • Do not specify a half-width space at the end of the character string or just before each delimiter.  (1): Up to 253 characters (2): Use "/" or "\" as delimiters for the folder path and file. (3): Can be omitted. When it is omitted, (1) is up to 252 characters. (4): If the extension is omitted, “.BIN” or “.CSV” will be automatically added by the system.	Unicode string	User

■Write data (s3)

Operand: (s3)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Number of request write data	Specify the number of data to be requested to write. The processing unit and setting range vary depending on Execution/completion type set in (d1)+0 and data type specification in (d1)+7. ■When "Writing binary data" is specified by (d1)+0 • When 16-bit binary data is specified: In units of words (1 to 65535)*1*2 • When 32-bit binary data is specified: In units of double words (1 to 32767) ■When "Converting and writing data in CSV format" is specified by (d1)+0: Number of elements • When decimal (signed 16-bit data) is specified: In units of words (1 to 65535)*1*2 • When decimal (unsigned 16-bit data) is specified: In units of words (1 to 65535) • When decimal (signed 32-bit data) is specified: In units of double words (1 to 32767) • When decimal (unsigned 32-bit data) is specified: In units of double words (1 to 32767) • When hexadecimal (16-bit data) is specified: In units of words (1 to 65535) • When hexadecimal (32-bit data) is specified: In units of double words (1 to 32767) • When a string (ASCII data) is specified: Number of elements (1 to 1023) • When a floating point real number (single-precision real number) is specified: In units of double words (1 to 32767) ■When 0 is specified • If the specified file exists, the data will be written to 0 point, and the program will terminate normally. • If the specified file does not exist, only a folder will be created, and the program will terminate without creating a file.	1 to 65535, 0	User
+1 to +□	Write data	The number of data to be requested to write is stored.	0000H to FFFFH	User

*1 When "1: Even number of bytes" or "3: Odd number of bytes" is specified by (d1)+7 (Data type specification), the setting range of (s3)+0 (Number of request write data) is 1 to 32767.

*2 Even when "3: Odd number of bytes" is specified by (d1)+7 (Data type specification), specify the number in units of words. Add one byte to the odd number of bytes to be written and set the number in (s3)+0 (Number of request write data).

Processing details

- This instruction writes the specified number of data to the specified file. Set the execution/completion type in the control data to specify the file write-target format.
- The write target is the SD memory card only.
- The processing completion bit device (d2) automatically turns on at the execution of the END instruction in the scan in which the completion of processing of the SP.FWRITE instruction is detected. The bit device (d2) turns off at the execution of the END instruction in the next scan. If the processing completion bit device (d2) is ON, it will automatically turn OFF when the SP.FWRITE instruction is executed.
- If the SP.FWRITE instruction completes with an error, the error completion device (d2)+1 turns on or off in synchronization with (d2).
- SM753 (File being accessed) turns on while the SP.FWRITE instruction is being executed. While SM753 is on, the SP.FWRITE instruction cannot be executed. (If the instruction is executed, no processing is performed.)
- Even if an error is detected during the execution of the instruction, the bit devices (d2) and (d2)+1 and SM753 do not turn on.
- Specify data in (s3)+0, (d1)+4, (d1)+5, and (d1)+2 depending on the combination of (d1)+0 and (d1)+7.

Execution/completion type (d1)+0		**Data type specification (d1)+7**	**Processing unit and setting range**		
			Number of request write data (s3)+0	**File position (d1)+4, (d1)+5**	**Number of data actually written (d1)+2**
Writing binary data	0000H: 16-bit binary data	0: Word	Word (0[*1], 1 to 65535)	Word (00000000H to 7FFFFFFFH, FFFFFFFFH)	Word
		1: Even number of bytes	Word (0[*1], 1 to 32767)	Byte (00000000H to FFFFFFFFH)	Byte
		2: Unit of the data type specified by the execution/completion type	Word (0[*1], 1 to 65535)	Word (00000000H to 7FFFFFFFH, FFFFFFFFH)	Word
		3: Odd number of bytes	Word (0[*1], 1 to 32767)[*2]	Byte (00000000H to FFFFFFFFH)	Byte
	0001H: 32-bit binary data	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Double word (0[*1], 1 to 32767)	Double word (00000000H to 3FFFFFFFH, FFFFFFFFH)	Double word
		3: Odd number of bytes	(Cannot be specified)		

Execution/completion type (d1)+0		Data type specification (d1)+7	Processing unit and setting range		
			Number of request write data (s3)+0	File position (d1)+4, (d1)+5	Number of data actually written (d1)+2
Writing data after converted to CSV format	0100H: Decimal (signed 16-bit data)	0: Word	Word (0[*1], 1 to 65535)	Head/end[*3]	Word
		1: Even number of bytes	Word (0[*1], 1 to 32767)	Head/end[*3]	Byte
		2: Unit of the data type specified by the execution/completion type	Word (0[*1], 1 to 65535)	Head/end[*3]	Word
		3: Odd number of bytes	Word (0[*1], 1 to 32767)	Head/end[*3]	Byte
	0101H: Decimal (unsigned 16-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Word (0[*1], 1 to 65535)	Head/end[*3]	Word
		3: Odd number of bytes	(Cannot be specified)		
	0110H: Decimal (signed 32-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Double word (0[*1], 1 to 32767)	Head/end[*3]	Double word
		3: Odd number of bytes	(Cannot be specified)		
	0111H: Decimal (unsigned 32-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Double word (0[*1], 1 to 32767)	Head/end[*3]	Double word
		3: Odd number of bytes	(Cannot be specified)		
	0120H: Hexadecimal (16-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Word (0[*1], 1 to 65535)	Head/end[*3]	Word
		3: Odd number of bytes	(Cannot be specified)		
	0121H: Hexadecimal (32-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Double word (0[*1], 1 to 32767)	Head/end[*3]	Double word
		3: Odd number of bytes	(Cannot be specified)		
	0130H: String (ASCII data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (0[*1], 1 to 1023)	Head/end[*3]	Number of elements
		3: Odd number of bytes	(Cannot be specified)		
	0140H: Floating point real number (single-precision real number)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Double word (0[*1], 1 to 32767)	Head/end[*3]	Double word
		3: Odd number of bytes	(Cannot be specified)		

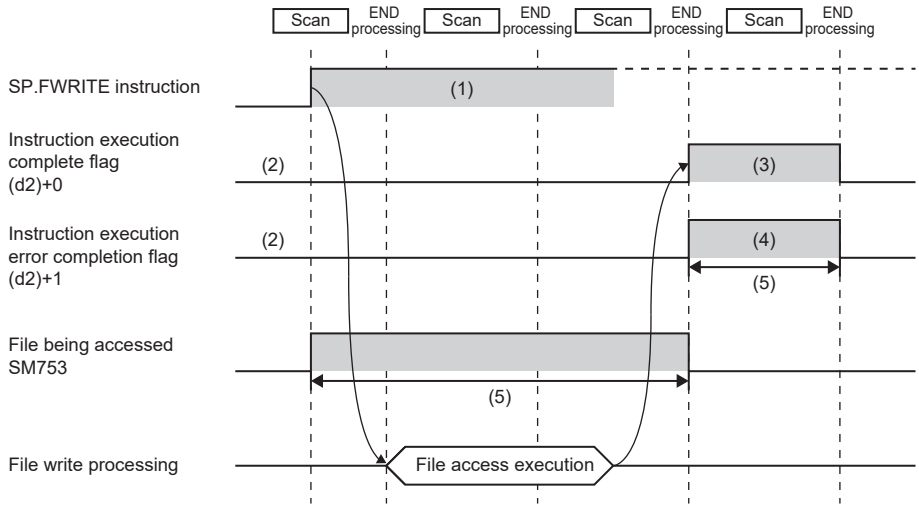
*1 If the specified file does not exist and the number of write target data is 0, a folder to up to the specified path will be created, and the program will terminate normally without creating a file.
If the specified file exists and the number of request write data is 0, the program will terminate normally with the number of actually written data to 0 point

*2 To specify the number of request write data, add 1 byte to the number of odd byte data to be written, and specify the sum in word units.

*3 The set value is from 00000000H to FFFFFFFEH (from the beginning of the file) or FFFFFFFFH (added to the end of the file).

■Timing chart

Below is shown the timing chart (flag updating timing) from the execution of the SP.FWRITE instruction to the completion.

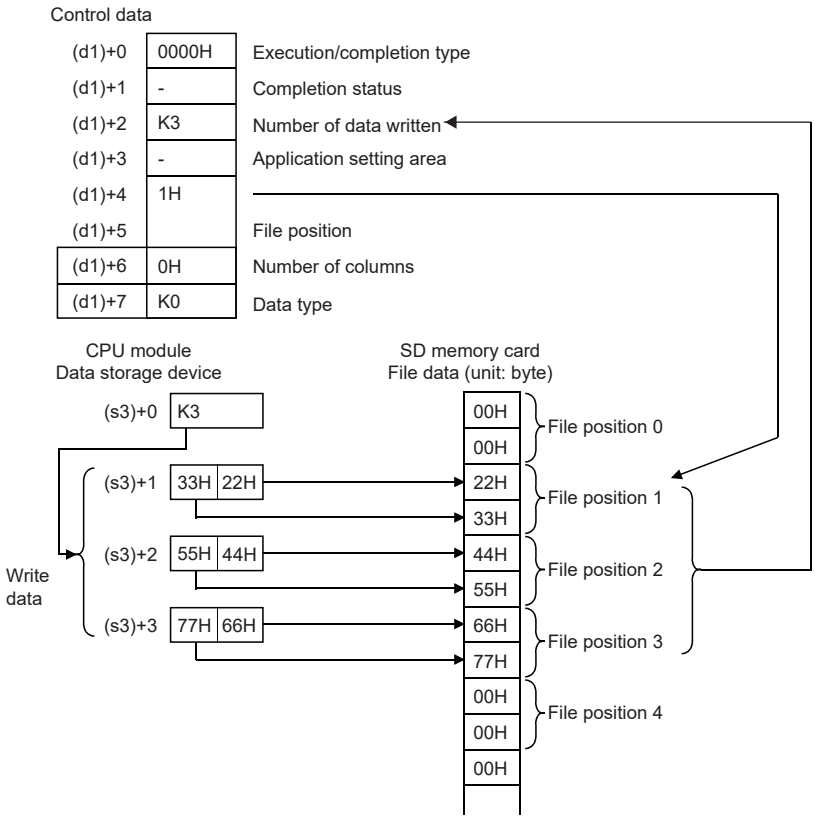


(1) No processing during scanning except when the instruction is turned OFF and ON
(2) Updating to OFF when the instruction is turned OFF and ON
(3) Turning ON upon normal/error completion*1
(4) Turning OFF upon normal completion and turning ON upon error completion*1
(5) Turning ON only during 1 scan after instruction execution
(6) Turning OFF when instruction execution complete flag is turned ON

*1 The complete flag is not turned ON when an error is detected during instruction execution.

■Writing binary data

- If the extension of the target file is omitted, the extension will be ".BIN".
- If a file that does not exist is specified, it will be created and data will be saved to the beginning of the file. The newly created file has the archive attribute.
- When an existing file is specified, data will be saved to the beginning of the file. If the size of data exceeds the size of the existing area in the file during writing, the excess data is additionally stored.
- If the specified position exceeds the existing file size, 0 point of data is written and the processing completes successfully.
- If the media runs out of free space during additional saving of data, an error occurs. In this case, the successfully written and additionally saved data is kept in the written state, and the program will terminate abnormally after additionally saving only the part that can be additionally saved. (8002H is stored in (d1)+1.)
- In 16-bit binary data write mode, when the number of request write data and file position are specified, data is written as follows.



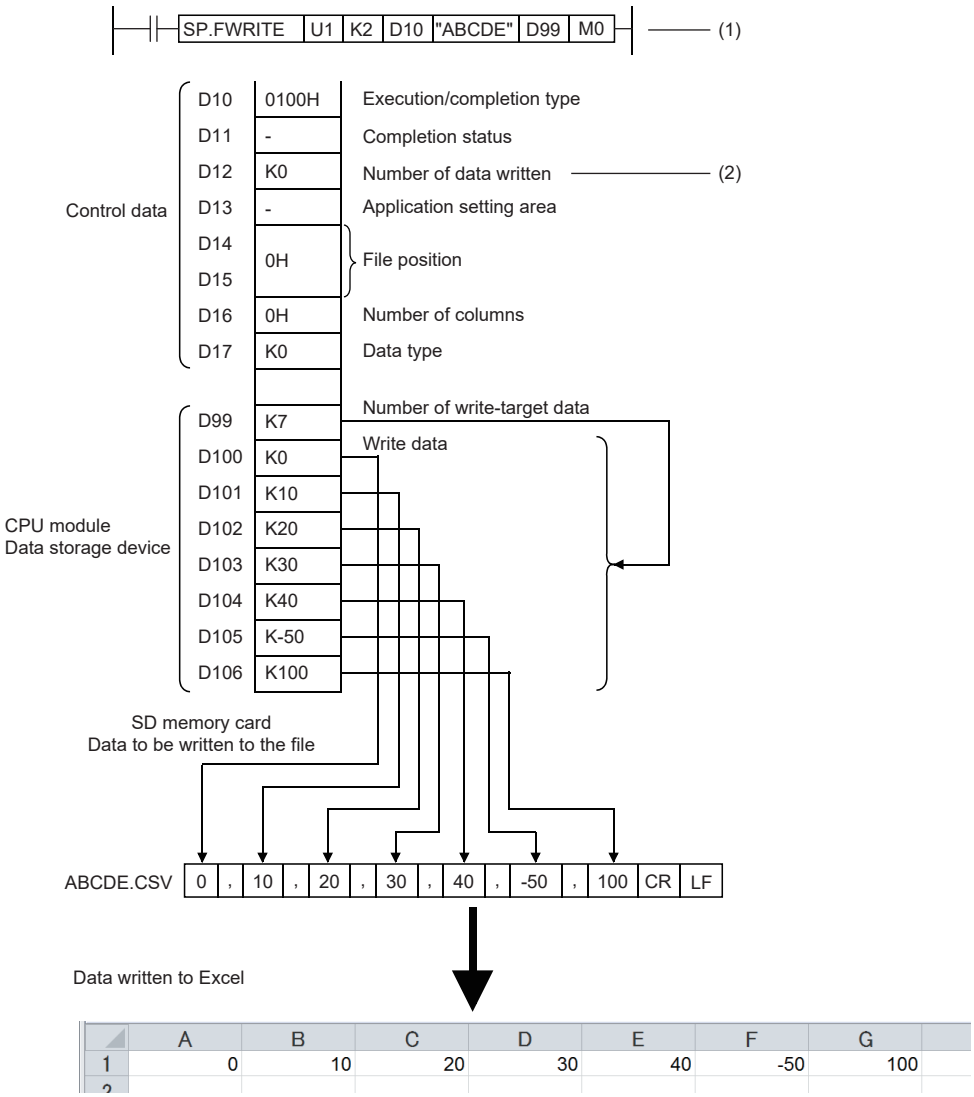
■Converting and writing data in CSV format

- If the extension is omitted, the extension will be ".CSV".
- When an existing file is specified, the following occurs.
  - When a value specified by (d1)+4 and (d1)+5 is other than FFFFFFFFH, data is saved to the file after deleting all the existing data in the file.
  - When a value specified by (d1)+4 and (d1)+5 is FFFFFFFFH, data is added and saved to the end of the file.*1
- When an existing file has been specified and “Add converting the last line feed code of file to a comma” has been specified in (d1)+3.b0, the data will be saved after the last line feed code of the file is converted to a comma. However, the last character of the file is not a line feed code, the character will not be converted to a comma, and the data will be saved to the end of the file.
- If a file that does not exist is specified, it will be created and data will be saved to the beginning of the file. The newly created file has the archive attribute.
- If the media runs out of free space during additional saving of data, an error occurs. In this case, the successfully saved data is kept in the written state, and the program will terminate abnormally after additionally saving only the part that can be additionally saved. (8002H is stored in (d1)+1.)
- When the number of columns is set to 0, data is read as a single-row data in a CSV format file.

*1 When “Added to the end of the file” has been specified in (d1)+3.b0, the data will be saved by staring a new line under the last line of the file.
When “Add converting the last line feed code of file to a comma (write continuing the last line)” has been specified, the data will be saved continuing the last line of the file.

Ex.

The number of columns is set to 0 when writing data after conversion to the CSV format.



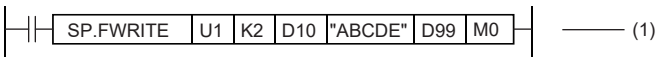
(1) Specified in units of words

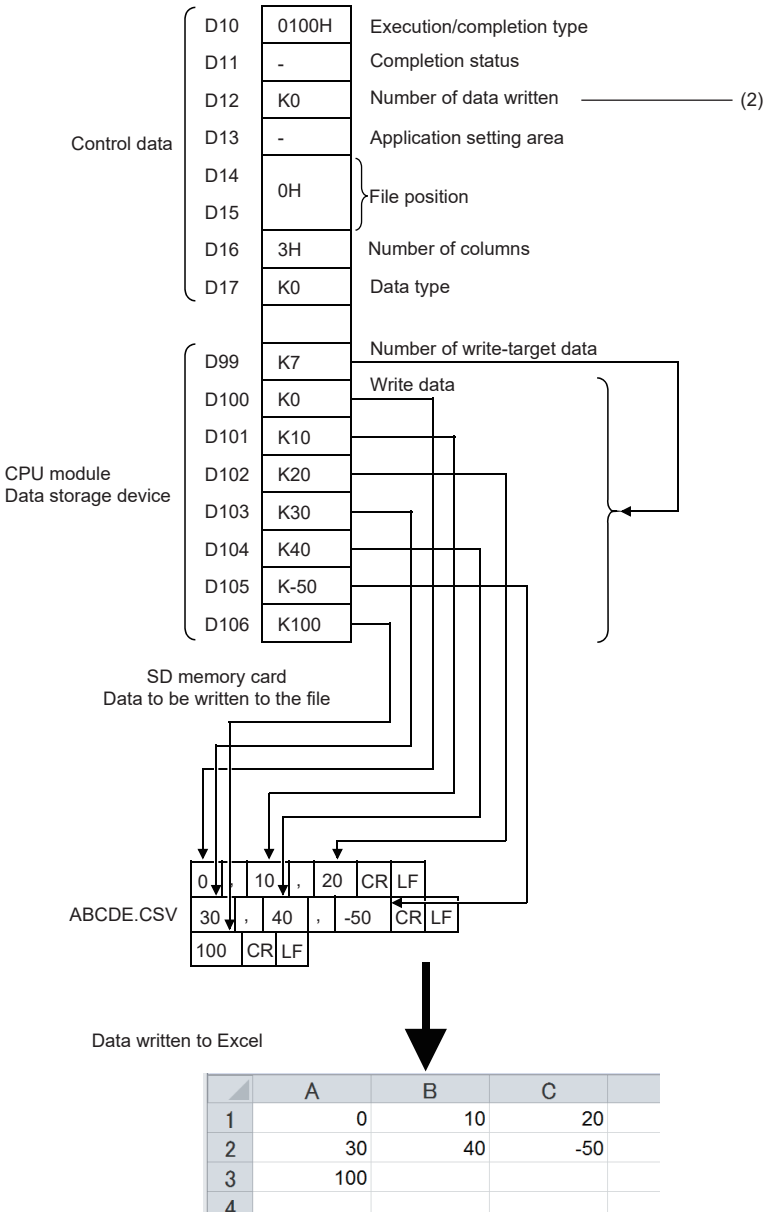
(2) Same as the number of write data if case of normal completion

- When the specified number of columns is set to a value other than 0, a CSV format file is stored as the table with the specified number of columns.

Ex.

The number of columns is set to a value other than 0 when writing data after conversion to the CSV format.





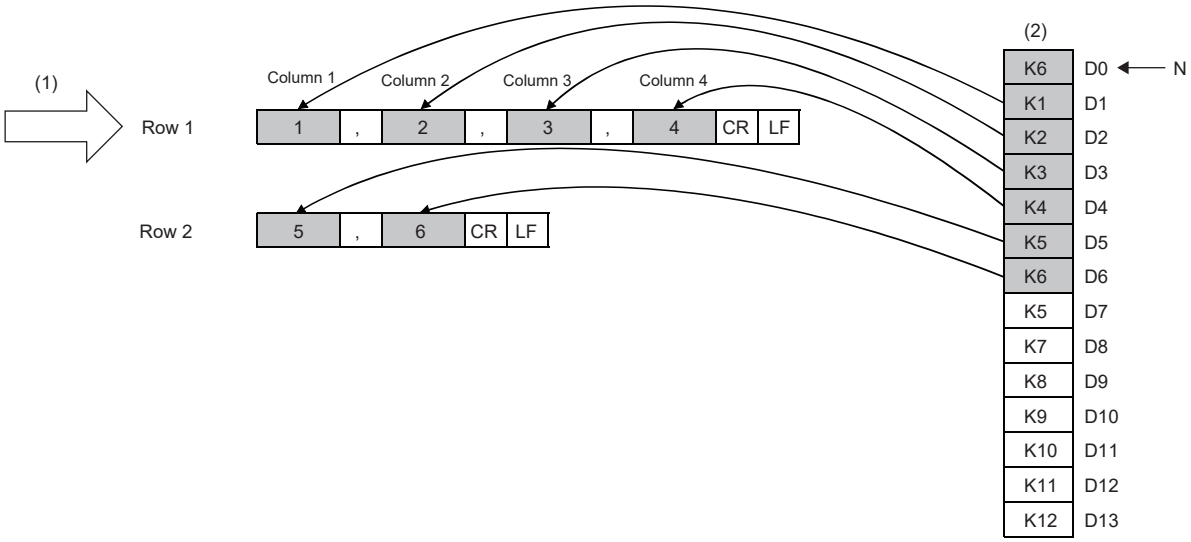
(1) Specified in units of words

(2) Same as the number of write data if case of normal completion

• The following figure shows how data is added.

[Specify the file to which data will be written.] (Even if the file exists, it is deleted and re-created.)

- Execution/completion type: Writing data after converted to CSV format (Decimal (signed 16-bit data))
- Number of request write data: 6H
- File position: 00000000H (from the beginning of the file)
- Number of columns specification: 4H
- Data type specification: Words
- Write start device: D0



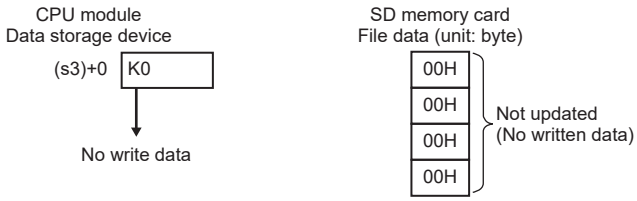
(1) Starting row

(2) Data in the device (write data)

N: Number of data

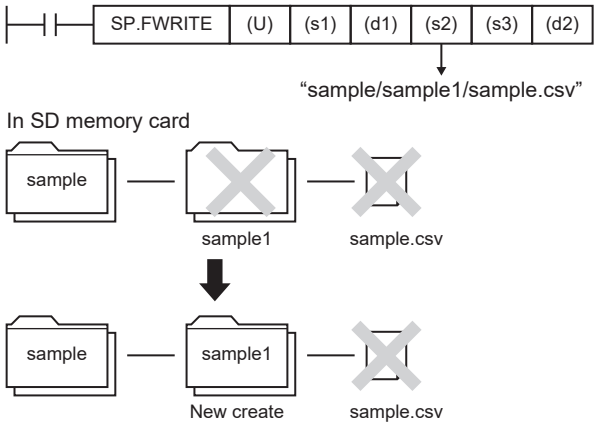
• When the number of request write data is 0, the processing varies depending on the existence of the specified file.

[When the specified file exists]



Since data is written to 0 point, the specified file will not be updated.
The program is finished normally.

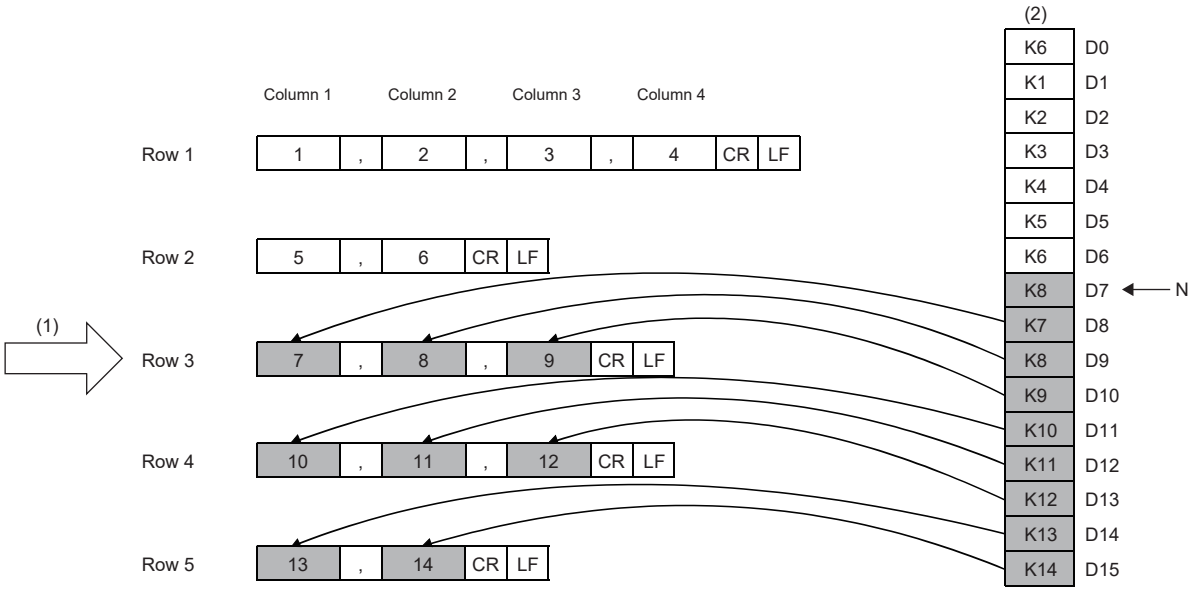
[When the specified file does not exist]



The specified folder path will be created, and the processing will be terminated. When the folder exists, no processing will be performed.
The specified file will not be created.

[Added to the end of the file]

- Execution/completion type: Writing data after converted to CSV format (Decimal (signed 16-bit data))
- Number of request write data: 8H
- File position: FFFFFFFFH (added to the end of the file)
- Number of columns specification: 3H
- Data type specification: Words
- Write start device: D7



(1) Starting position

(2) Data in the device (write data)

N: Number of data

Point

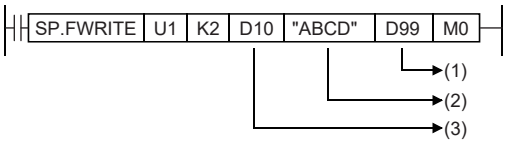
- To justify the written data columns, set “Number of request write data” to an integral multiple of “Specified number of columns.” If the setting is not an integral multiple, the lines will vary in the number of columns.
- When data is added to the end of a file, columns are shifted if "Number of columns" is changed from the previous writing.
- When “String (ASCII data)” has been specified for execution/completion type, set 00H (NULL) at the end of the string of one element. When the string has an even number of bytes, set 0000H (NULL for 2 bytes) for the next 1 word.
- When “String (ASCII data)” has been specified for execution/completion type, the maximum number of characters in 1 element is 1999. If the number of characters exceeds 1999 and 00H (NULL) is not stored, the 2000th and following characters will not be written, and the next element will be written.
- When “String (ASCII data)” has been specified for execution/completion type, up to 1023 elements can be written by executing the instruction once.

• The following figures show an example of specifying "String (ASCII data)" for the execution/completion type.

Ex.

Writing data after converted to CSV format (string (ASCII data))

[Data to be written to a file]



(1) Data to be written
(2) File name
(3) Control data

[Control data]

D10	H0130
D11	H0000
D12	-
D13	K0
D14	0H
D15	
D16	0H
D17	K2

D10: Execution/completion type: String (ASCII data)
D11: Completion status
D12: Number of data actually written
D13: Application setting area
D14, D15: File position
D16: Number of columns
D17: Data type specification

CPU module
Data storage device

D099	K3
D100	H4241
D101	H4443
D102	H0000
D103	H3231
D104	H0033
D105	H3534
D106	H0036

"ABCD"
"123"
"456"

D99: Number of request write data
D100 to D102: String to be written to the 1st column of the 1st line
D103, D104: String to be written to the 2nd column of the 1st line
D105, D106: String to be written to the 3rd column of the 1st line

SD memory card
Data to be written to the file

ABCD.CSV

ABCD	,	123	,	456	CR	LF

Data written to Excel

	A	B	C	D
1	ABCD	123	456	
2				

Point

- Set 00H (NULL) in the end of the string in an element. When the number of bytes of the string is even, set 0000H (two bytes of NULL) in the next one word.
- The maximum number of characters in one element is 1999. If this maximum number is exceeded and 00H (NULL) is not stored, characters of 2000th character and after are not written and the write processing shifts to the next element.
- A maximum of 1023 elements can be written in a single instruction execution.

• The following table lists values to set in (s3)+1 and later and data to be written in a CSV file when "0140H: Floating point real number (single-precision real number)" is set to (d1)+0.

Execution/completion type ((d1)+0)	Value to set in the write data ((s3)+1 and later)	Data to be written in a CSV file
0140H: Floating point real number (single-precision real number)	Values within the range of: $-2^{128} < \text{data} \leq -2^{-126}$, 0, $2^{-126} \leq \text{data} < 2^{128}$	A value (0 to 7 digits in the decimal part) as given on the left is written in exponential format.
	Values other than above	0 is written. (Data cannot be converted.)

• The following shows how the file size (total number of bytes) is calculated when a CSV format file is written to the SD memory card.

[Total number of bytes] = [Total number of bytes excluding the last row] + [Number of bytes of the last row]

([Number of bytes of each row] = [Number of columns[*1]] +1+ [total number of bytes of all data values per line][*2])

*2 The specified number of columns applies to rows other than the last row. The number of columns of the last row is calculated as shown below because it may differ from the specified number of rows depending on the number of write data.

· The number of rows excluding the last row is calculated. (Number of rows excluding the last row = number of requested write data÷number of columns (remainders rounded down)

· The number of columns of the last row is calculated. (Number of columns of the last row = number of requested write data - (number of rows excluding the last row × number of columns))

*3 The following shows how the number of bytes of each data value is calculated.

Sign of data value	Number of bytes of each data value	Range of bytes	Example
Positive	Number of digits	1 to 5 (word specification) 1 to 3 (byte specification)	• 12345: 5 bytes • 67: 2 bytes
Negative	Number of digits + 1	2 to 6 (word specification) 2 to 4 (byte specification)	• -12345: 6 bytes • -67: 3 bytes

Program example

An example of a program for writing binary data/CSV data to the SD memory card by using the SP.FWRITE instruction is shown below.

■Writing binary data

When X0 is turned ON, the data in D101 to D200 are written starting from the 16th word from the beginning of the "sample.bin" file stored in the SD memory card.

[Program operation]

1. Control data is created during RUN.
2. Set the number of request write data.*1
3. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
4. The SP.FWRITE instruction is executed*2
5. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*1 The data to be written must have been set although this step is not contained in the program example.

*2 The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.
- SM606 (Memory card disable request)
- SM753 (File being accessed)

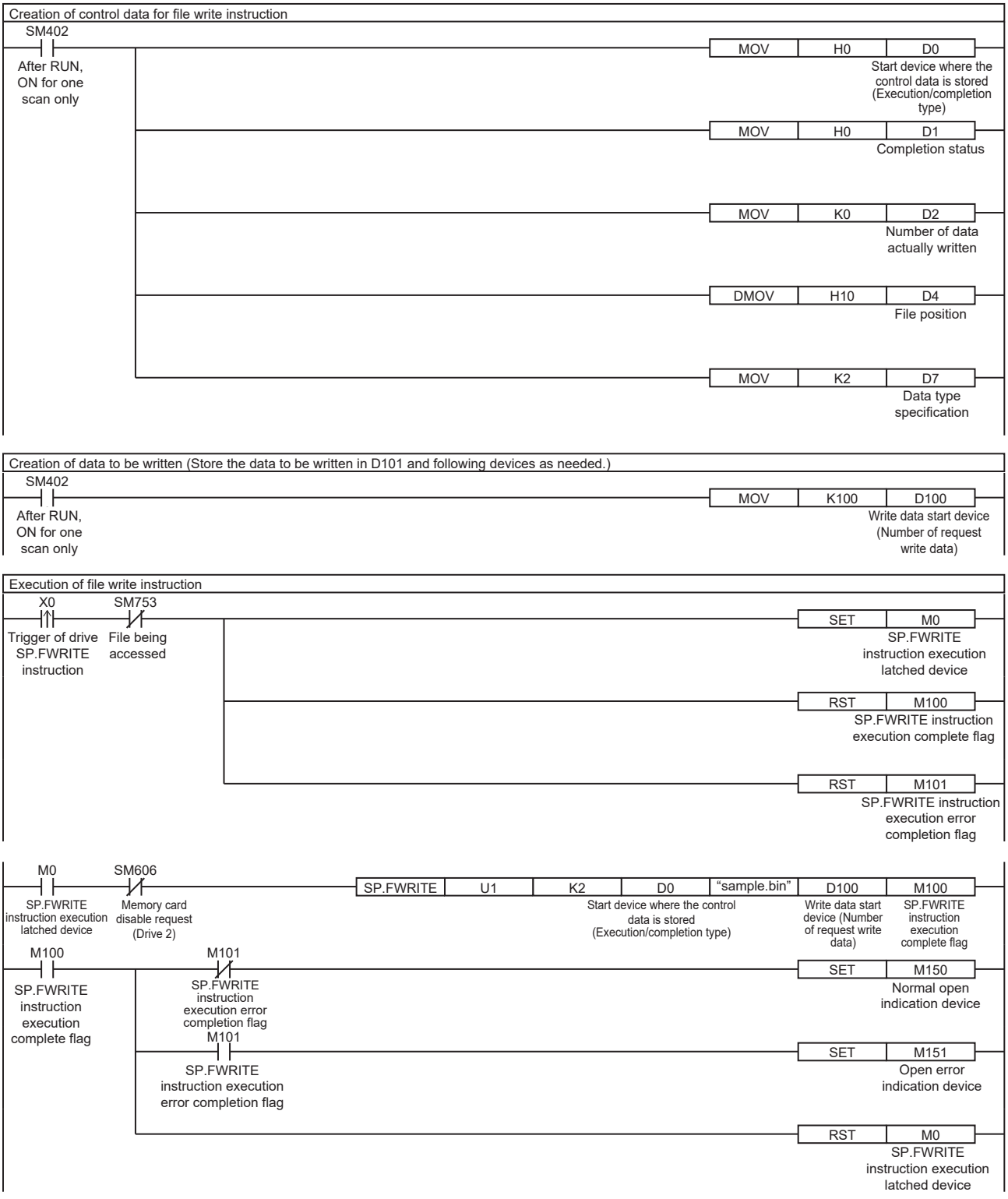
[Devices used]

Device	Description
X0	Trigger of drive SP.FWRITE instruction
D0	Start device where the control data is stored • D0: Execution/completion type • D1: Completion status • D2: Number of data actually written • D4,D5: File position • D7: Data type specification
D100	Head device where the write-target data is stored • D100: Number of request write data • D101 to D200: Write data
M0	SP.FWRITE instruction execution latched device
M100	SP.FWRITE instruction execution complete flag
M101	SP.FWRITE instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FWRITE instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(s1)	Drive specification	K2 (SD memory card)
(d1)	Start device where the control data is stored	D0: 0H (16-bit binary data) D1: 0H(Completed successfully) D2: K0 D4,D5: 10H(Writing starting from the 16th word from the beginning) D7: K2 (Unit of data type specified in D0)
(s2)	Start device where the file name is stored	"sample.bin"
(s3)	Head device where the write-target data is stored	D100: K100 (Writing 100 words) D101 to D200: Write data
(d2)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]



[ST program]

```
//(1) Creation of control data for file write instruction
IF SM402 THEN
D0 := H0; //Execution/completion type (Control data start device)
D1 := H0; //Completion status
D2 := 0; //Number of data actually written
D4:UD := H10; //File position
D7 := 2; //Data type specification
END_IF;

//(2) Creation of data to be written
IF SM402 THEN
D100 := 100; //Number of request write data
//Store the data to be written in D101 and following devices as needed.
END_IF;

//(3) Processing to start up the drive contact (X0)
IF LDP(TRUE, X0) THEN
//Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE, M0); //Holds drive contact
RST(TRUE, M100); //Initialize instruction execution complete flag
RST(TRUE, M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(4) Execution of file write instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "sample.bin" (Start device where the file name is stored)
//S3 = D100(Head device where the write-target data is stored)
//D1 = D0 (Start device where the control data is stored)
//D2 = M100 (Bit device that turns on upon completion of the processing)
SP_FWRITE(TRUE, U1, 2, "sample.bin", D100, D0, M100);
END_IF;
END_IF;

//(5) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```

[Execution result]

When the target device data is written by using the program example, the results are as shown below.

[Write target]*1

Device Name	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Current Value	String
D99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	..
D100	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0064	d.
D101	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	1	03E9	é.d
D102	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	0	03EA	
D103	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	1	03EB	
D104	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0	03EC	
D105	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	1	03ED	
																		(1)
D196	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0048	H.
D197	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0049	I.
D198	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	004A	J.
D199	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	044B	K.
D200	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	044C	L.
D201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	..
D202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	..

[Write result (sample.bin)]

sample.bin

ADRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	ZZZZZZZZZZZZZZZZ
00000010	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	5A	ZZZZZZZZZZZZZZZZ
00000020	E9	03											EF	03	F0	03	.....
00000030	F1	03	F2	03	F3	03	F4	03	F5	03	F6	03	F7	03	F8	03	.....
00000040	F9	03	FA	03	FB	03	FC	03	FD	03	FE	03	FF	03	00	04	.....
00000050	01	04	02	04	03	04	04	04	05	04	06	04	07	04	08	04	.....
00000060	09	04	0A	04	0B	04	0C	04	0D	04	0E	04	0F	04	10	04	.....
00000070	11	04	12	04	13	04	14	04	15	04	16	04	17	04	18	04	.....
00000080	19	04	1A	04	1B	04	1C	04	1D	04	1E	04	1F	04	20	04	.....
00000090	21	04	22	04	23	04	24	04	25	04	26	04	27	04	28	04	!.".#.$.%.&.'.(.
000000A0	29	04	2A	04	2B	04	2C	04	2D	04	2E	04	2F	04	30	04	).*.+.,.-.../.0.
000000B0	31	04	32	04	33	04	34	04	35	04	36	04	37	04	38	04	1.2.3.4.5.6.7.8.
000000C0	39	04	3A	04	3B	04	3C	04	3D	04	3E	04	3F	04	40	04	9.:.;.<.=.>.?.@.
000000D0	41	04	42	04	43	04	44	04	45	04	46	04	47	04	48	04	A.B.C.D.E.F.G.H.
000000E0	49	04	4A	04	4B	04	4C	04									I.J.K.L.

(2)

(1) In the program example, 100 words of data are written. Accordingly, the number of request write data is stored in D100, and the data to be written are stored in D101 to D200.

(2) 100 words from 03E9 to 044C (D101 to D200) are written. In the program example, the data is written starting from the 16th word, and therefore 03E9 (device value of D101) is stored in the 16th word.

*1 Device/Buffer Memory Batch Monitor (hexadecimal display) of engineering tool

■Converting and writing data in CSV format

When X0 is turned on, the data in D101 to D115 are written to the end of the “sample.csv” file stored in the SD memory card.

[Program operation]

1. Control data is created during RUN.
2. Set the number of request write data.*1
3. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
4. The SP.FWRITE instruction is executed.*2
5. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*1 The data to be written must have been set although this step is not contained in the program example.

*2 The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.

· SM606 (Memory card disable request)

· SM753 (File being accessed)

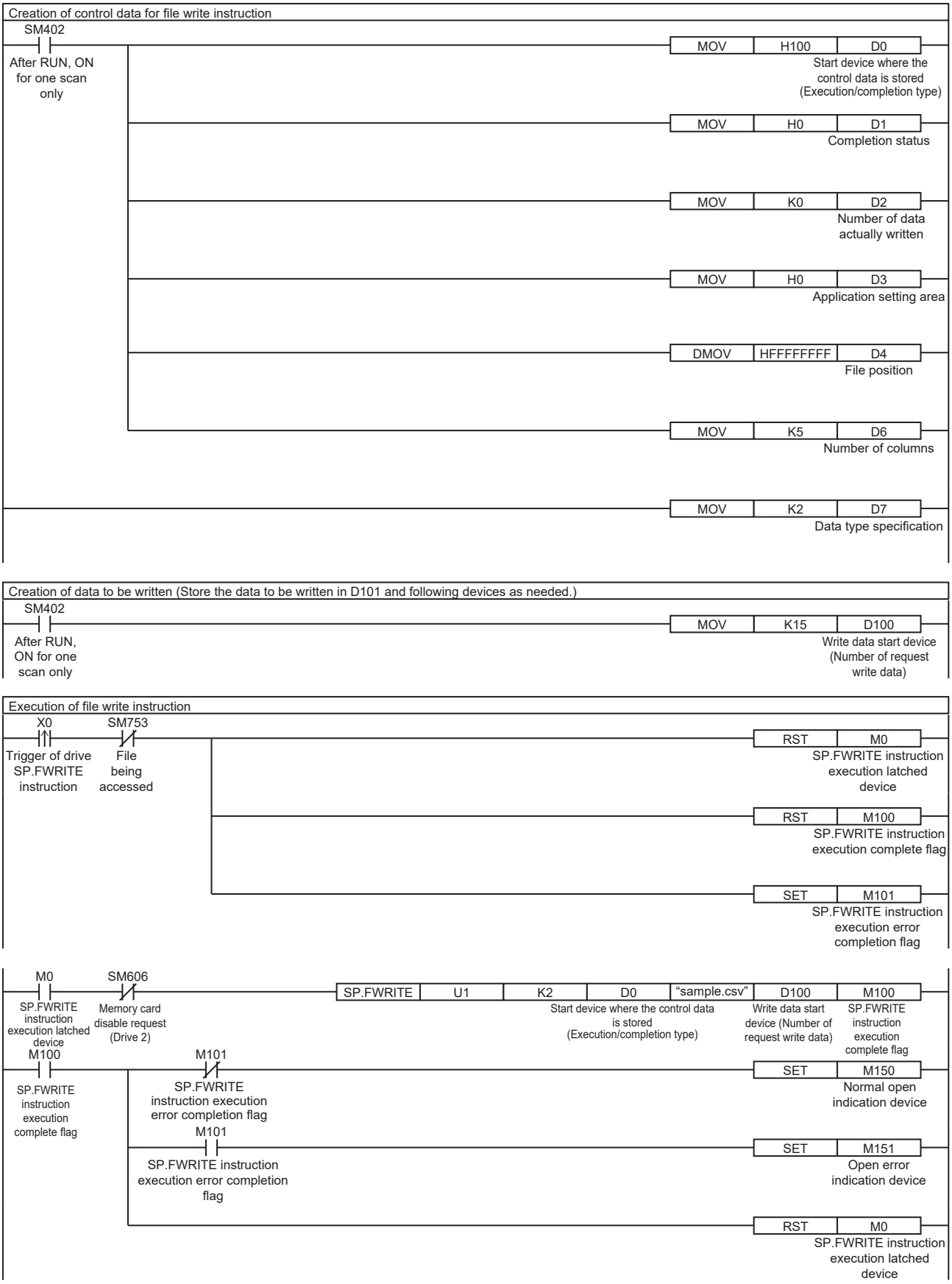
[Devices used]

Device	Description
X0	Trigger of drive SP.FWRITE instruction
D0	Start device where the control data is stored • D0: Execution/completion type • D1: Completion status • D2: Number of data actually written • D3: Application setting area • D4,D5: File position • D6: Number of columns • D7: Data type specification
D100	Head device where the write-target data is stored • D100: Number of request write data • D101 to D115: Write data
M0	SP.FWRITE instruction execution latched device
M100	SP.FWRITE instruction execution complete flag
M101	SP.FWRITE instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FWRITE instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(s1)	Drive specification	K2 (SD memory card)
(d1)	Start device where the control data is stored	D0: 100H (Decimal (signed 16-bit data)) D1: 0H (Completed successfully) D2: K0 D3: H0 (Add to the end of the file) D4,D5: FFFFFFFFH (Added to the end of the file.) D6: K5 (Writing by 5 columns) D7: K2 (Unit of data type specified in D0)
(s2)	Start device where the file name is stored	“sample.csv”
(s3)	Head device where the write-target data is stored	D100: K15 (Writing 15 words) D101 to D115: Write data
(d2)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]



[ST program]

```
//(1) Creation of control data for file write instruction
IF SM402 THEN
D0 := H100; //Execution/completion type (Control data start device)
D1 := H0; //Completion status
D2 := 0; //Number of data actually written
D3 := H0; //Application setting area
D4:UD := HFFFFFFFF; //File position
D6 := 5; //Number of columns
D7 := 2; //Data type specification
END_IF;

//(2) Creation of data to be written
IF SM402 THEN
D100 := 15; //Number of request write data
//Store the data to be written in D101 and following devices as needed.
END_IF;

//(3) Processing to start up the drive contact (X0)
IF LDP(TRUE, X0) THEN
//Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE, M0); //Holds drive contact
RST(TRUE, M100); //Initialize instruction execution complete flag
RST(TRUE, M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(4) Execution of file write instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "sample.csv" (Start device where the file name is stored)
//S3 = D100 (Head device for storing the data to be written)
//D1 = D0 (Start device where the control data is stored)
//D2 = M100 (Bit device that turns on upon completion of the processing)
SP_FWRITE(TRUE, U1, 2, "sample.csv", D100, D0, M100);
END_IF;
END_IF;

//(5) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```

[Execution result]

When the target device data is written by using the program example, the results are as shown below.

[Write target]*1

Device Name	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Current Value	String
D100	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15	
D101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	
D102	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15	
D103	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	16	
D104	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	255	].
D105	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	256	
D106	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	4095	].
D107	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4096	
D108	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	32767	
D109	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-32768	
D110	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-32767	
D111	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1	]□
D112	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D113	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	4369	
D114	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	21845	JU
D115	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	-21846	I

[Write result (sample.csv)] (1)

	A	B	C	D	E	F
1	1	2	3	4	5	6
2	1	2	3	4	5	6
3	A	B		D	E	F
4	1	15	16	255	256	
5	4095	4096	32767	-32768	-32767	(2)
6	-1	0	4369	21845	-21846	
7						

(1) In the program example, 15 words of data are written. Accordingly, the number of request write data is stored in D100, and the data to be written are stored in D101 to D115.

(2) The 15 words from 1 to -21846 (D101 to D115) are written. In the program example, the data are written by 5 columns to the end of the file, and therefore the data are stored after the existing data (that exists before writing).

*1 Device/Buffer Memory Batch Monitor (decimal display) of engineering tool)

Precautions

- The SP.FWRITE instruction cannot be executed in user interrupt programs. An error (3582H) will occur.
- When “01**H” has been specified in Execution/completion type (d1)+0 and an option other than “FFFFFFFFH” has been specified in File position (d1)+4 and (d1)+5, if “Add converting the last line feed code of file to a comma (write continuing the last line)” is specified in (d1)+3.b0, an error will occur.
- The SP.FWRITE instruction cannot be executed while SM606 (Memory card disable request) is ON. When SM606 is turned ON during execution of the instruction, the program will terminate abnormally. (The data that has been written before SM606 is turned ON will be stored in the SD memory card.)
- The files in the system folder ($MELPRJ$) cannot be handled by the SP.FWRITE instruction. If an access to any file in the system folder ($MELPRJ$) is requested, a calculation error (3405H) will occur.
- The SP.FWRITE instruction cannot be executed simultaneously with the SP.DEVST instruction, the SP.FTPPUT instruction, and the SP.FTPGET.
- Do not disconnect the power or remove the SD memory card during execution of the SP.FWRITE instruction. (The file may be damaged, or an error may occur.)

Operation error

Error code (SD0/SD8067)	Description
2820H	The bit label digits specified in (s2) and (s3) are unacceptable settings (the number of digits is not K4).
	The storage device of the control data (d1) exceeds the end of the device range.
	The value in the device specified by (s3)+0 is out of the range (1 to 65535), or exceeds the setting area specified by (s3)+1 and later in the device/label memory.
3405H	The drive specified by (s1) is not the one for the SD memory card.
	Any value that is set in the device specified by (d1) and later as control data is out of the range.
	The file name string specified by (s2) cannot be read. • The number of characters of the string in the file name specified exceeds the range. • An inhibited value is set. • The specified file name string ends with a delimiter. • The system folder ($MELPRJ$) directly under the route folder is specified. • A half-width period is specified at the end of the specified file name string or immediately before each delimiter.
3427H	The combination of the execution/completion type specified in (d1)+0, the write start position set in (d1)+3 and the file position specified in (d1)+4 is not allowed.
	An invalid combination of (d1)+0 (Execution/completion type) and (d1)+7 (Data type specification) is specified.
3582H	The SP.FWRITE instruction is executed in an interrupt program.

If the SP.FWRITE instruction completes with an error, an error code is stored in the device specified by (d1)+1. (Note that an error code is not stored if the instruction results in an operation error.)

For the error code stored in (d1)+1, refer to the following.

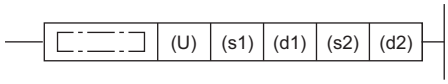
☞ Page 579 Error codes generated for file operation instructions

Deleting the specified file

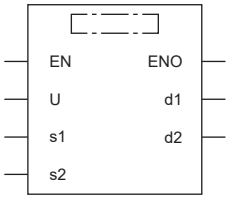
SP.FDELETE

FX5S FX5UJ FX5U FX5UC

This instruction deletes the specified file or folder in an SD memory card.

Ladder	ST
	ENO:=SP_FDELETE(EN,U,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)	Dummy	■FX5S CPU module U1 ■FX5UJ CPU module U1 to U8 ■FX5U/FX5UC CPU module U1 to U10	Device name	ANY16
(s1)	Drive specification	2 (fixed)*1	16-bit signed binary	ANY16
(d1)	Start device where the control data is stored	☞ Page 536 Control data (d1)	Word	ANY16_ARRAY (Number of elements: 2)
(s2)	Start device where the file name or folder name is stored	☞ Page 536 File name/folder name (s2)	Unicode string	ANYSTRING_DOUBLE
(d2)	Bit device that turns on upon completion of the processing	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Only drive 2 (for the SD memory card) can be set.

■Applicable devices

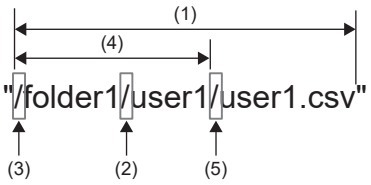
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	○	—	—	—	—	○	○	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	○	—
(d2)	○*1	○*2	—	—	—	—	—	—	—	—	—

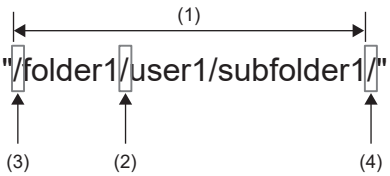
*1 S cannot be used.
*2 T, ST, and C cannot be used.
Only bit specification of word device is applicable.

■Control data (d1)

Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Application setting area	b15 … b1 b0 0 1/0 1/0 b0: Target type setting Specify the type of data (file or folder) to delete. • 0: File • 1: Folder b1: Empty folder deletion setting Specify whether to delete only empty folders when deleting folders. (This setting is enabled when b0 is 1 (Folder).) • 0: Delete the folder even when it contains data. • 1: Delete the folder only when it is empty.	Refer to the "Description" column.	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0000H: Completed successfully • Other than 0000H: Completed with an error (error code) (☞ Page 579 Error codes generated for file operation instructions)	—	System

■File name/folder name (s2)

Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +□	File name character string	When specifying a file, specify the folder path where the file to be deleted is stored and the file name. • When the file name has an extension, specify the name without omitting the extension. • The folder path and file name (including an extension) must be within 253 characters in total. • The folder path must be within 244 characters. (Delimiters are not included.) • Specify one character or more for the file name or folder path in addition to a delimiter. • Do not add a delimiter at the end of a string. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify any file in the system folder ($MELPRJ$). • The number of folder path hierarchies must be within 10 levels.  (1): Up to 253 characters (2): Use "/" or "\" as delimiters for the folder path and file. (3): Can be omitted. When it is omitted, (1) is up to 252 characters. (4): The folder path is up to 244 characters long (243 characters long when (3) is omitted). (5): The delimiter between the folder path and the file name is not included in the number of characters of the folder path.	Unicode string	User

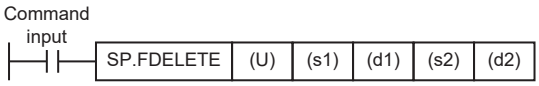
Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +☐	Folder name character string	When specifying a folder, specify the folder path of the folder to be deleted. • The folder path must be within 244 characters. (Delimiters at the end of the folder path are not included.) • Specify one character or more for the folder path in addition to a delimiter. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in it. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 244 characters (2): Use "/" or "\" as delimiters for the folder path. (3): Can be omitted. When it is omitted, (1) is up to 243 characters. (4): Can be omitted.	Unicode string	User

Processing details

- This instruction deletes the file or folder specified by (s2) in the drive specified by (s1).
- SM753 (File being accessed) turns on while the SP.FDELETE instruction is being executed. While SM753 is on, the SP.FDELETE instruction cannot be executed. (If the instruction is executed, no processing is performed.)
- The processing completion bit device (d2) automatically turns on at the execution of the END instruction in the scan in which the completion of processing of the SP.FDELETE instruction is detected. The bit device (d2) turns off at the execution of the END instruction in the next scan. If the processing completion bit device (d2) is ON, it will automatically turn OFF when the SP.FDELETE instruction is executed.
- If the SP.FDELETE instruction completes with an error, the error completion device (d2)+1 turns on or off in synchronization with (d2).
- If an operation error is detected during the execution of the instruction, (d2) and (d2)+1 do not turn on.

■Timing chart

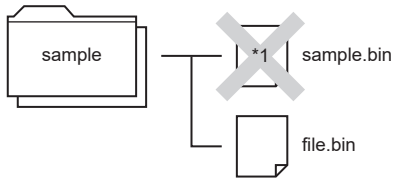
The operation specifications of the file delete function are shown below.



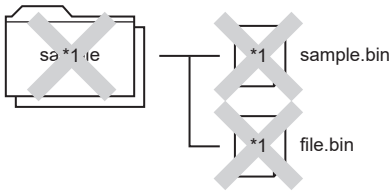
Application setting area (d1)+0	File name/folder name (s2)
0H, 1H, 3H	[When a file is specified] sample/sample.bin [When a folder is specified] sample

[In SD memory card]

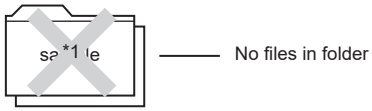
■When application setting area is 0H (File)



■When application setting area is 1H ("Folder" + "Delete the folder even when it contains data.")

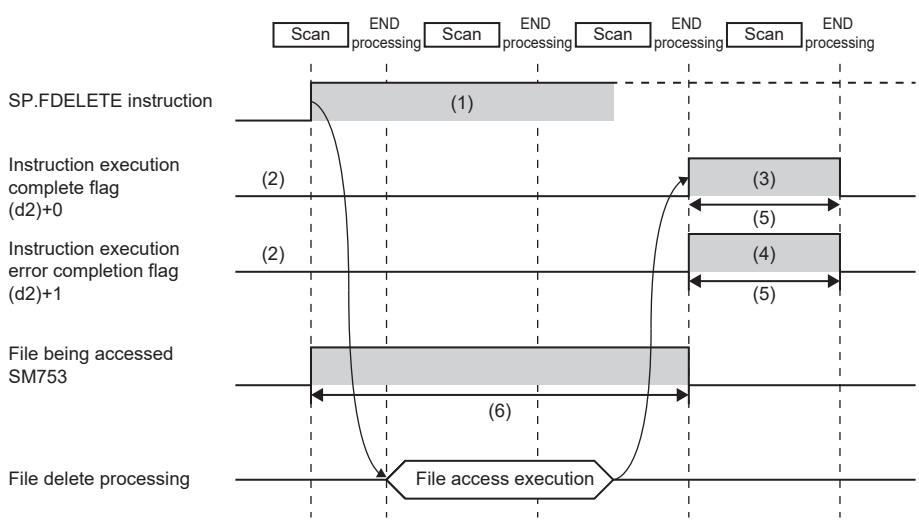


■When application setting area is 3H ("Folder" + "Delete the folder only when it is empty.")



*1 The file or folder is deleted.

Below is shown the timing chart (flag updating timing) from the execution of the SP.FDELETE instruction to the completion.



(1) No processing during scanning except when the instruction is turned OFF and ON
(2) Updating to OFF when the instruction is turned OFF and ON
(3) Turning ON upon normal/error completion*2
(4) Turning OFF upon normal completion and turning ON upon error completion*2
(5) Turning ON only during 1 scan after instruction execution
(6) Turning OFF when instruction execution complete flag is turned ON

*2 The complete flag is not turned ON when an error is detected during instruction execution.

Program example

When X0 is turned ON, the "\sample\sample.csv" file stored in the SD memory card is deleted.
[Program operation]

1. Control data is created during RUN.
2. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
3. The SP.FDELETE instruction is executed.*1
4. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*1 The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.
· SM606 (Memory card disable request)
· SM753 (File being accessed)

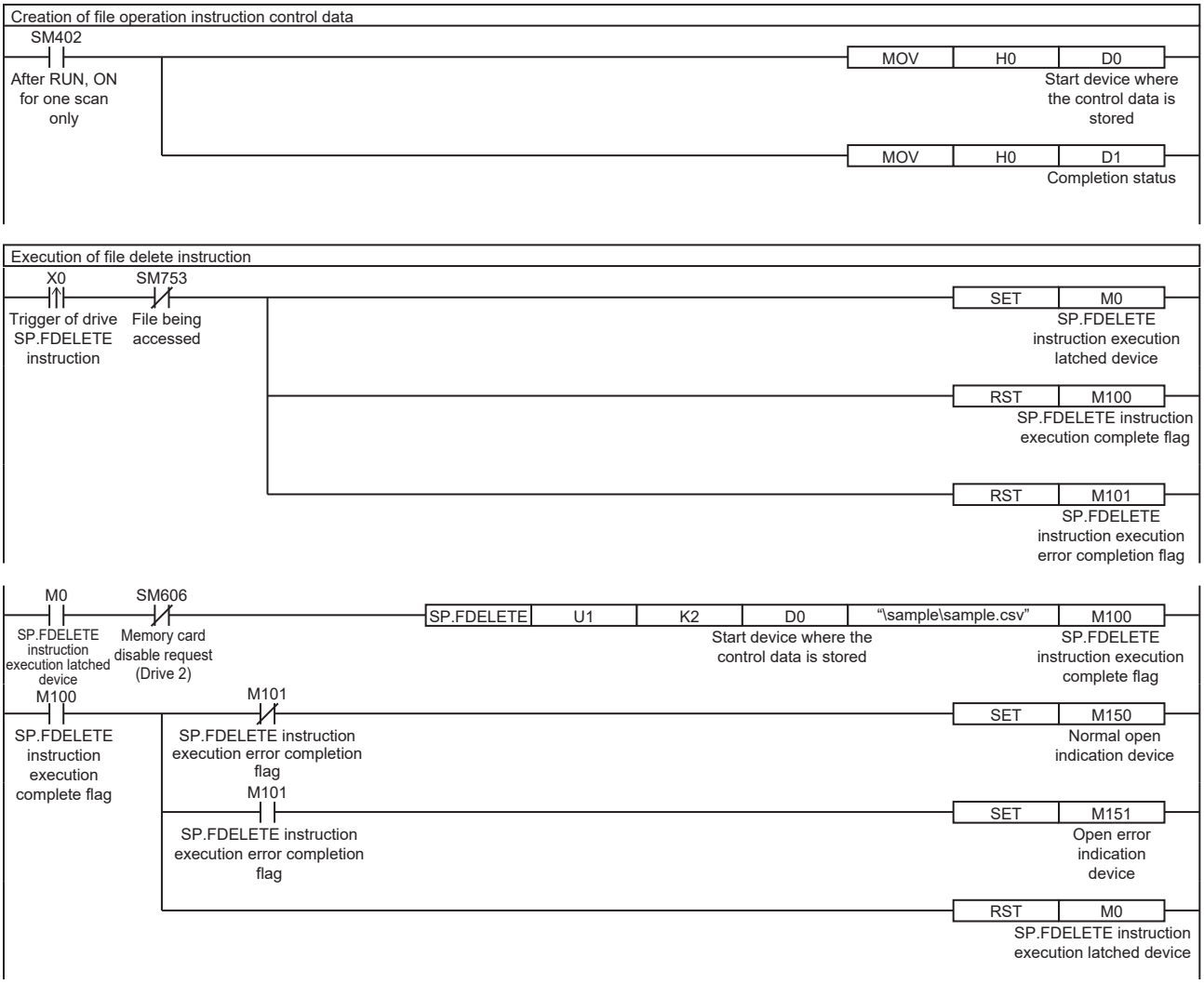
[Devices used]

Device	Description
X0	Trigger of drive SP.FDELETE instruction
D0	Start device where the control data is stored • D0: Application setting area • D1: Completion status
M0	SP.FDELETE instruction execution latched device
M100	SP.FDELETE instruction execution complete flag
M101	SP.FDELETE instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FDELETE instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(s1)	Drive specification	K2 (SD memory card)
(d1)	Start device where the control data is stored	D0: 0H(File) D1: 0H(Completed successfully)
(s2)	Start device where the file name is stored	"\sample\sample.csv"
(d2)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]



[ST program]

```
//(1) Creation of file delete instruction control data
IF SM402 THEN
D0:=H0; //Application setting area (File)
D1:=H0; //Completion status
END_IF;

//(2) Processing to start up the drive contact (X0)
IF LDP(TRUE,X0) THEN;
//Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE,M0); //Holds drive contact
RST(TRUE,M100); //Initialize instruction execution complete flag
RST(TRUE,M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(3) Execution of file delete instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN;
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "\sample\sample.csv" (Start device where the file name is stored)
//D1 = D0 (Start device where the control data is stored)
//D2 = M100 (Bit device that turns on upon completion of the processing)
SP_FDELETE(TRUE, U1, 2, "\sample\sample.csv", D0, M100);
END_IF;
END_IF;

//(4) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```

Precautions

- The SP.FDELETE instruction cannot be performed in user interrupt programs. An error (3582H) will occur.
- The SP.FDELETE instruction cannot be executed while SM606 (Memory card disable request) is ON. If SM606 is turned ON after the instruction is executed, the program will terminate abnormally. (The data deleted before SM606 is turned ON will not be restored.)
- The SP.FDELETE instruction specifying the system folder ($MELPRJ$) or a file in the system folder cannot be executed. If the SP.FDELETE instruction is executed, a calculation error (3405H) will occur.
- Even though the operating status of the CPU module is switched from RUN to STOP during instruction execution, the CPU module continues the processing of the instruction.
- If the instruction is completed with an error during processing, the file or folder already deleted is not restored.
- When the size of a file to be deleted or the number of files to be deleted becomes larger, the instruction will take more time to complete.
- Do not access the file being processed by the SP.FDELETE instruction from other functions. (The file may be corrupt or an error may occur.)
- Do not operate files or folders being accessed from other functions.
- The SP.FDELETE instruction cannot be executed simultaneously with the SP.DEVST instruction, the SP.FTPPUT instruction, and the SP.FTPGET.
- When the application setting area of the SP.FDELETE instruction is 1H (“Folder” + “Delete the folder even when it contains data”), if the folder specified in (s2) contains a folder whose path exceeds 244 characters or a file exceeding 253 characters, the 8006H error will occur.
- When the application setting area of the SP.FDELETE instruction is 3H (“Folder” + “Delete the folder only when it is empty”), if the folder specified in (s2) contains any file or folder, the 8004H error will occur.
- Do not disconnect the power or remove the SD memory card during execution of the SP.FDELETE instruction. (The file may be damaged, or an error may occur.)

Operation error

Error code (SD0/SD8067)	Description
2820H	The storage device of the control data (d1) exceeds the end of the device range.
	The number of bit label digits specified in (s2) is an unallowable setting (the number of digits is not K4).
	The size of the read data exceeds the read device size.
3405H	The drive specified by (s1) is not the one for the SD memory card.
	The file name/folder name string specified by (s2) cannot be read. • The specified file name string contains no character. • The specified file name string contains 254 characters or more. • The specified folder path contains 245 characters or more. • The specified folder path hierarchies contains 11 levels or more. • The specified file name string has a period (one-byte) at its end or directly before each delimiter. • When a file is specified, the file name string ends with a delimiter.
	A system folder ($MELPRJ$) which is directly under the root folder is in the folder path specified by (s2).
3582H	The SP.FDELETE instruction is executed in an interrupt program.

If the SP.FDELETE instruction completes with an error, an error code is stored in the device specified by (d1)+1. (Note that an error code is not stored if the instruction results in an operation error.)

For the error code stored in (d1)+1, refer to the following.

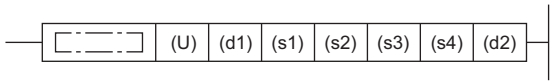
☞ Page 579 Error codes generated for file operation instructions

Copying the specified file

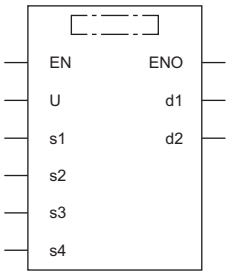
SP.FCOPY

FX5S FX5UJ FX5U FX5UC

This instruction copies the specified file or folder in an SD memory card. When a folder is specified, the specified folder is copied in its entirety, or all the files and subfolders in the specified folder are copied.

Ladder	ST
	ENO:=SP_FCOPY(EN,U,s1,s2,s3,s4,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)	Dummy	■FX5S CPU module U1 ■FX5UJ CPU module U1 to U8 ■FX5U/FX5UC CPU module U1 to U10	Device name	ANY16
(d1)	Start device where the control data is stored	☞ Page 544 Control data (d1)	Word	ANY16_ARRAY (Number of elements: 2)
(s1)	Copy source drive	2 (fixed)*1	16-bit signed binary	ANY16
(s2)	Start device where the file name or folder name of the copy source is stored	☞ Page 546 Copy source file name/folder name (s2)	Unicode string	ANYSTRING_DOUBLE
(s3)	Copy destination drive	2 (fixed)*1	16-bit signed binary	ANY16
(s4)	Start device for storing the copy destination folder path	☞ Page 547 Copy destination folder path (s4)	Unicode string	ANYSTRING_DOUBLE
(d2)	Bit device that turns on upon completion of the processing	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Only drive 2 (for the SD memory card) can be set.

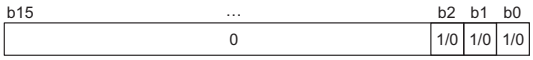
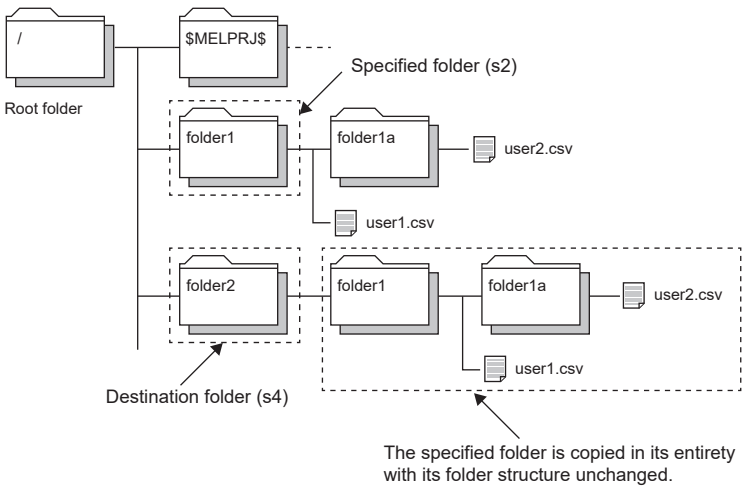
■Applicable devices

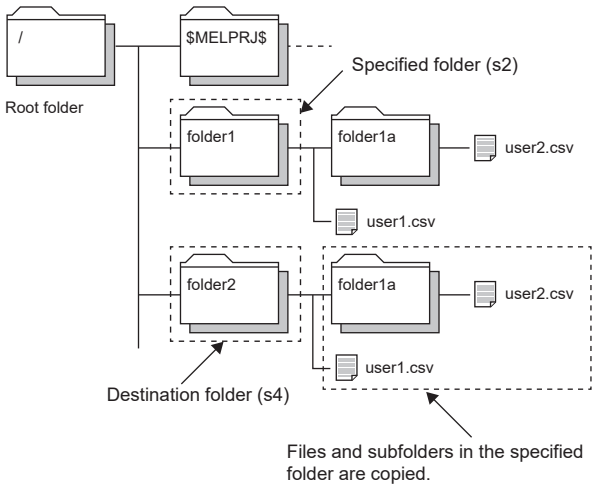
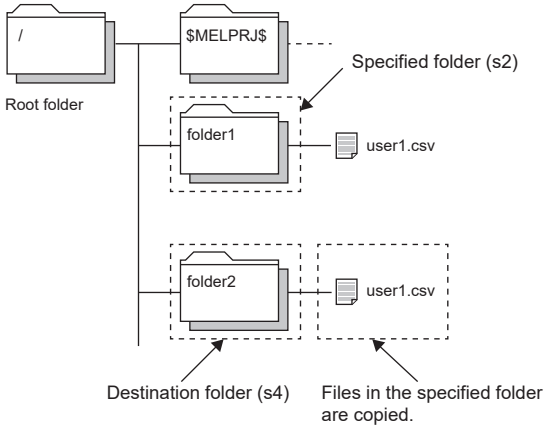
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	—	○
(d1)	—	○	—	—	—	—	○	—	—	—	—
(s1)	○	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	○	—
(s3)	○	○	—	—	—	—	○	○	—	—	—
(s4)	—	○	—	—	—	—	○	—	—	○	—
(d2)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

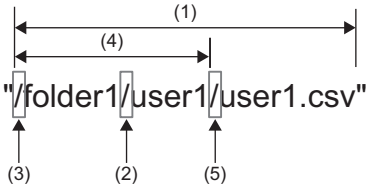
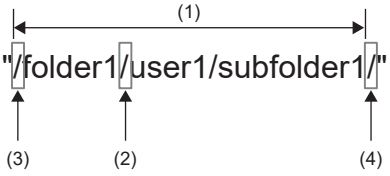
*2 T, ST, and C cannot be used.
Only bit specification of word device is applicable.

■Control data (d1)

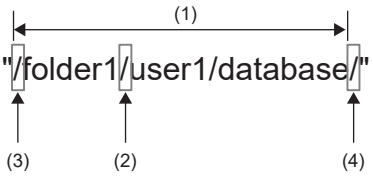
Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Application setting area	 ■b0, b2: Target type setting b0: Specify the type of data (file or folder) to copy. • 0: File • 1: Folder b2: Specify how the data is copied. (This bit is valid when b0 is set to 1 (Folder).) • 0: Copy the folder in its entirety. • 1: Copy all the files and subfolders in the folder. [Operation when b2 is set to 0 (Copy the folder in its entirety)] The specified folder (including all files and subfolders in it) is copied in its entirety. 	Refer to the "Description" column.	User

Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Application setting area	[Operation when b2 is set to 1 (Copy all the files and subfolders in the folder), and the specified folder includes files and a subfolder] All the files and subfolders in the specified folder are copied.  [Operation when b2 is set to 1 (Copy all the files and subfolders in the folder), and the specified folder includes only files (no subfolder)] All the files in the specified folder are copied.  ■b1: Overwrite setting Specify whether to overwrite a file or a folder if one with the same name as the copy source already exists in the copy destination. • 0: Do not overwrite the file or folder. • 1: Overwrite the file or folder. When b0 is set to 1 (Folder) or b1 is set to 0 (Do not overwrite the file or folder), the instruction skips copying a file or a folder having the same name with the one in the copy destination. (The instruction is completed and no error occurs.) However, even if the folder copy is skipped, the instruction does not skip copying a file/subfolder existing directly under the folder having the same name.	Refer to the "Description" column.	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0000H: Completed successfully • Other than 0000H: Completed with an error (error code) (☞ Page 579 Error codes generated for file operation instructions)	—	System

■Copy source file name/folder name (s2)

Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +□	File name character string	When specifying a file, specify the folder path where the copy source file is stored and the file name. • When the file name has an extension, specify the name without omitting the extension. • The folder path and file name (including an extension) must be within 253 characters in total. • The folder path must be within 244 characters. (Delimiters are not included.) • Specify one character or more for the file name or folder path in addition to a delimiter. • Do not add a delimiter at the end of a string. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in the system folder. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 253 characters (2): Use "/" or "\" as delimiters for the folder path and file. (3): Can be omitted. When it is omitted, (1) is up to 252 characters. (4): The folder path is up to 244 characters long (243 characters long when (3) is omitted). (5): The delimiter between the folder path and the file name is not included in the number of characters of the folder path.	Unicode string	User
	Folder name character string	When specifying a folder, specify the copy source folder path. • The folder path must be within 244 characters. (Delimiters at the end of the folder path are not included.) • Specify one character or more for the folder path in addition to a delimiter. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in the system folder. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 244 characters (2): Use "/" or "\" as delimiters for the folder path. (3): Can be omitted. When it is omitted, (1) is up to 243 characters. (4): Can be omitted.		

■Copy destination folder path (s4)

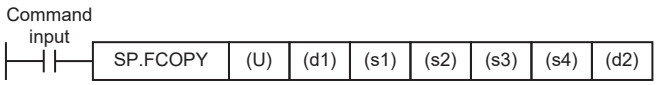
Operand: (s4)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +☐	Folder path	Specify the copy destination folder path. • The folder path must be within 244 characters. (Delimiters at the end of the folder path are not included.) • Specify one character or more for the folder path in addition to a delimiter. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in the system folder. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 244 characters (2): Use "/" or "\" as delimiters for the folder path. (3): Can be omitted. When it is omitted, (1) is up to 243 characters. (4): Can be omitted.	Unicode string	User

Processing details

- This instruction copies the file or folder specified by (s2) in the drive specified by (s1) to the folder specified by (s4) in the drive specified by (s3). When a folder is specified by (s2), a file and subfolder with the same names may exist in the copy destination. In this case, the instruction is not completed with an error even if the value of bit 1 (overwrite setting) of (d1) is 0. (Copying is skipped.)
- If the folder specified by (s4) does not exist in the copy destination, the folder is created automatically.
- SM753 (File being accessed) turns on while the SP.FCOPY instruction is being executed. While SM753 is on, the SP.FCOPY instruction cannot be executed. (If the instruction is executed, no processing is performed.)
- The processing completion bit device (d2) automatically turns on at the execution of the END instruction in the scan in which the completion of processing of the SP.FCOPY instruction is detected. The bit device (d2) turns off at the execution of the END instruction in the next scan. If the SP.FCOPY instruction completes with an error, the error completion device (d2)+1 turns on or off in synchronization with (d2). If the processing completion bit device (d2) is ON, it will automatically turn OFF when the SP.FCOPY instruction is executed.
- If an operation error is detected during the execution of the instruction, (d2) and (d2)+1 do not turn on.

■Timing chart

The operation specifications of the file copy function are shown below.

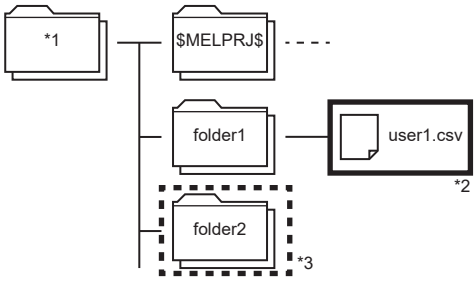


Application setting area (d1)+0	File name/folder name of the copy source (s2)	Copy destination folder path (s4)
0H, 1H, 5H, 7H	[When a file is specified] user1.csv [When a folder is specified] folder1	folder2

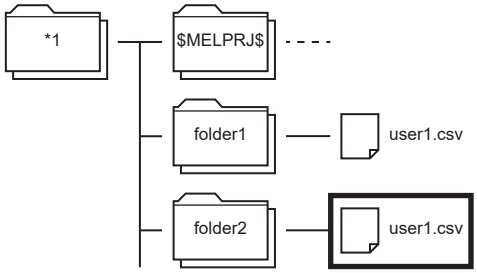
[In SD memory card]

■When application setting area is 0H (File)

Before execution of instruction



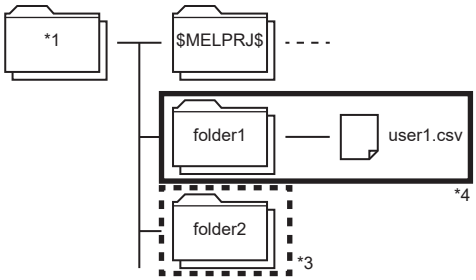
After execution of instruction



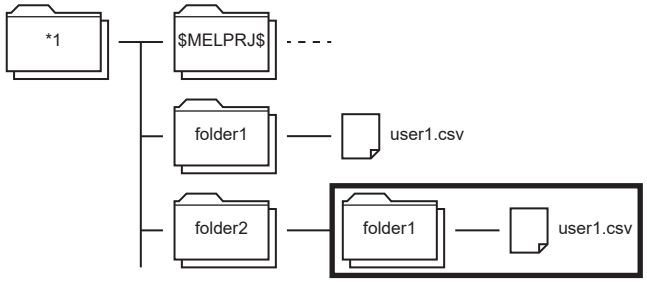
The folder1/user1.csv file is copied into folder2.

■When application setting area is 1H ("Folder", "Copy the folder in its entirety.")

Before execution of instruction



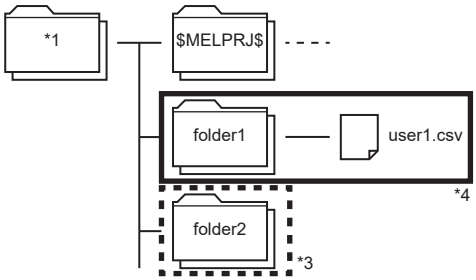
After execution of instruction



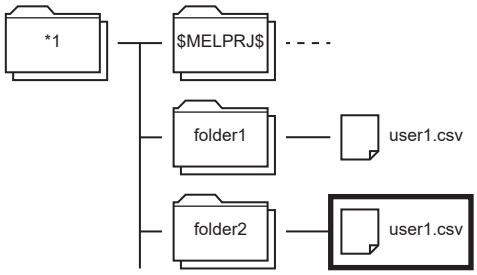
The folder configuration of the copy source is copied into folder2.

■When application setting area is 5H ("Folder", "Copy all the files and subfolders in the folder.")

Before execution of instruction



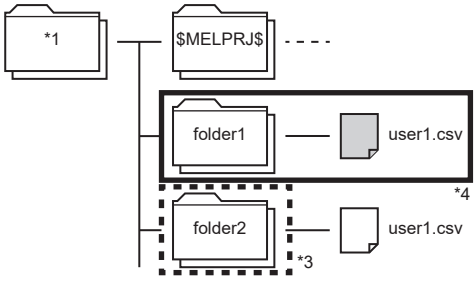
After execution of instruction



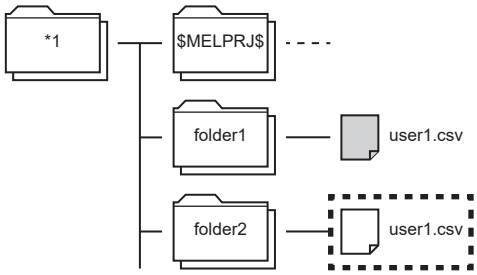
Only the file in the copy source folder is copied into folder2.

■When application setting area is 5H ("Folder", "Copy all the files and subfolders in the folder.", "Do not overwrite the file or folder.")

Before execution of instruction



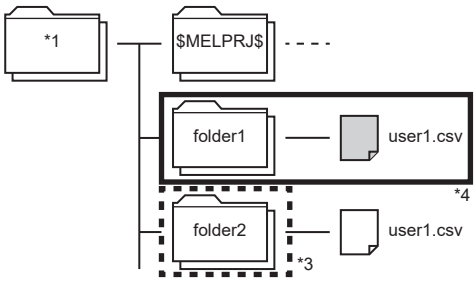
After execution of instruction



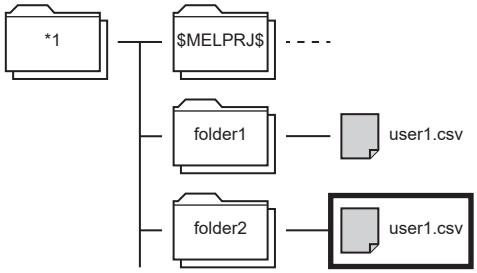
Since the copy destination contains a file with the same name, the file is not copied. The SP.FCOPY instruction is completed normally.
When the copy destination does not contain a file with the same name, the file is copied.

■When application setting area is 7H ("Folder", "Copy all the files and subfolders in the folder.", "Overwrite the file or folder.")

Before execution of instruction



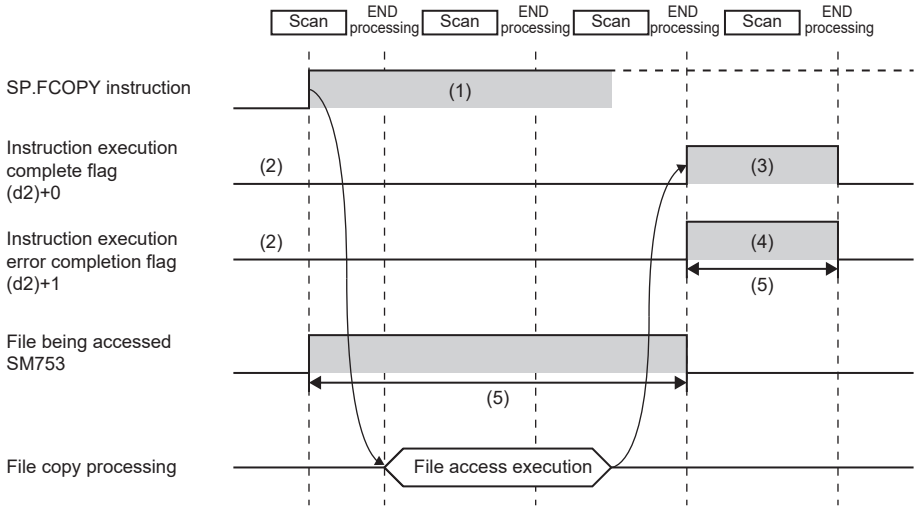
After execution of instruction



folder2/user1.csv is copied overwriting the folder1/user1.csv file.

*1 Root folder
*2 File name of the copy source
*3 Copy destination folder
*4 Folder name of the copy source

Below is shown the timing chart (flag updating timing) from the execution of the SP.FCOPY instruction to the completion.



(1) No processing during scanning except when the instruction is turned OFF and ON
(2) Updating to OFF when the instruction is turned OFF and ON
(3) Turning ON upon normal/error completion*5
(4) Turning OFF upon normal completion and turning ON upon error completion*5
(5) Turning ON only during 1 scan after instruction execution
(6) Turning OFF when instruction execution complete flag is turned ON

*5 The complete flag is not turned ON when an error is detected during instruction execution.

Program example

When X0 is turned ON, the “sample.bin” file stored in the SD memory card is copied into the “sample1” folder.

[Program operation]

1. Control data is created during RUN.
2. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
3. The SP.FCOPY instruction is executed.*1
4. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*1 The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.
- SM606 (Memory card disable request)
- SM753 (File being accessed)

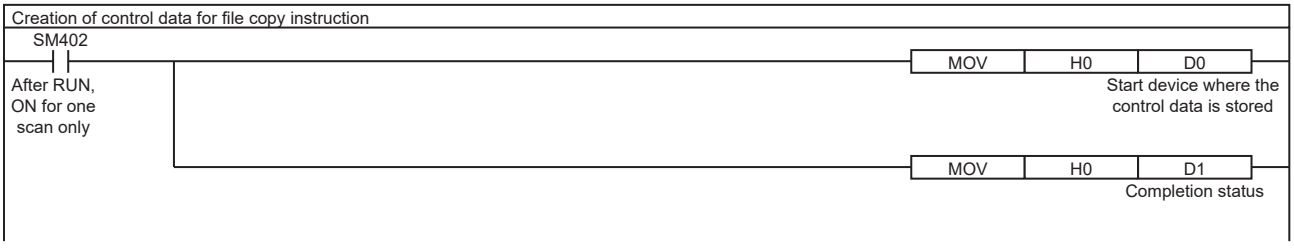
[Devices used]

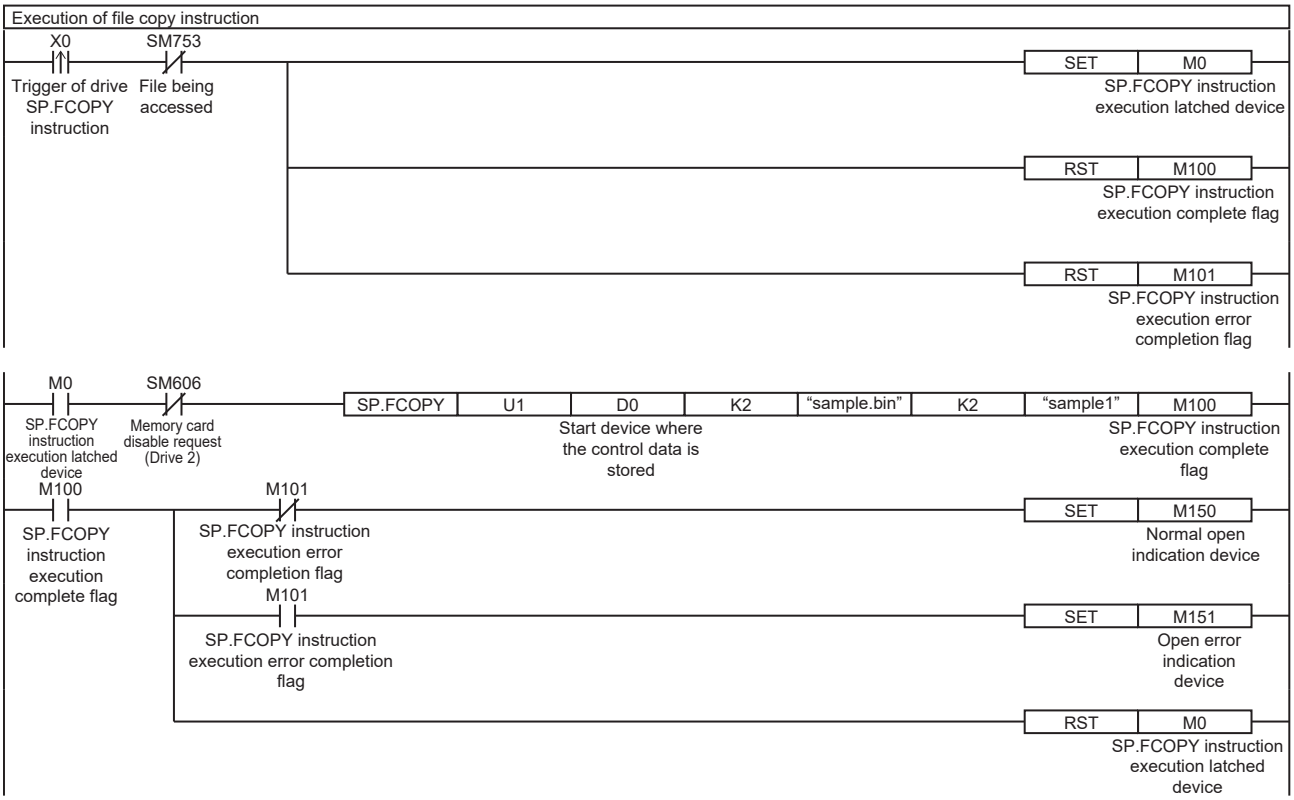
Device	Description
X0	Trigger of drive SP.FCOPY instruction
D0	Start device where the control data is stored • D0: Application setting area • D1: Completion status
M0	SP.FCOPY instruction execution latched device
M100	SP.FCOPY instruction execution complete flag
M101	SP.FCOPY instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FCOPY instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(d1)	Start device where the control data is stored	D0: 0H (File) D1: 0H (Completed successfully)
(s1)	Copy source drive	K2 (SD memory card)
(s2)	Start device where the file name of the copy source is stored	“sample.bin”
(s3)	Copy destination drive	K2 (SD memory card)
(s4)	Start device for storing the copy destination folder path	“sample1”
(d2)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]





[ST program]

```
//(1) Creation of control data for file copy instruction
IF SM402 THEN
D0:=H0; //Application setting area (File)
D1:=H0; //Completion status
END_IF;

//(2) Processing to start up the drive contact (X0)
IF LDP(TRUE,X0) THEN;
//Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE,M0); //Holds drive contact
RST(TRUE,M100); //Initialize instruction execution complete flag
RST(TRUE,M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(3) Execution of file copy instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN;
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "sample.bin" (Start device where the file name of the copy source is stored)
//S3 = 2 (Drive specification, 2 fixed)
//S4 = "sample1" (Start device for storing the copy destination folder path)
//D1 = D0 (Start device where the control data is stored)
//D2 = M100 (Bit device that turns on upon completion of the processing)
SP_FCOPY (TRUE, U1, 2, "sample.bin", 2, "sample1", D0, M100);
END_IF;
END_IF;

//(4) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```

Precautions

- Do not execute the SP.FCOPY instruction in an interrupt program. Doing so may cause malfunction of the module.
- The SP.FCOPY instruction cannot be executed while SM606 (Memory card disable request) is ON. If SM606 is turned ON after the completion of the initial END processing during execution of the instruction, the program will terminate abnormally upon detection of turning ON. (The data copied before SM606 is turned ON are copied.)
- The SP.FCOPY instruction specifying the system folder ($MELPRJ$) or a file in the system folder cannot be executed. If the SP.FCOPY instruction is executed, a calculation error (3405H) will occur.
- Even though the operating status of the CPU module is switched from RUN to STOP during instruction execution, the CPU module continues the processing of the instruction.
- Ensure that the number of characters in a folder path or in the total of a folder path and a file name after copying does not exceed its limit. (The file may be inaccessible or an error may occur.)
- If the instruction is completed with an error during processing, the file or folder may remain under processing.
- When the size of a file to be copied or the number of files to be copied becomes larger, the instruction will take more time to complete.
- Do not access the file being processed by the SP.FCOPY instruction from other functions. (The file may be corrupt or an error may occur.)
- Do not operate files or folders being accessed from other functions. (8001H is stored in (d1)+1.)
- The SP.FCOPY instruction cannot be executed simultaneously with the SP.DEVST instruction, the SP.FTPPUT instruction, and the SP.FTPGET.
- Do not disconnect the power or remove the SD memory card during execution of the SP.FCOPY instruction. (The file may be damaged, or an error may occur.)

Operation error

Error code (SD0/SD8067)	Description
2820H	The storage device of the control data (d1) exceeds the end of the device range.
	The bit label digits specified in (s2) and (s4) are unacceptable settings (the number of digits is not K4).
	The size of the read data exceeds the read device size.
3405H	The drive specified by (s1) or (s3) is not the one for the SD memory card.
	The file name/folder name string specified by (s2) cannot be read. • The specified file name string contains no character. • The specified file name string contains 254 characters or more. • The specified folder path contains 245 characters or more. • The set value is unusable. • The specified folder path hierarchies contains 11 levels or more. • When a file is specified, the file name string ends with a delimiter. • The specified file name string has a period (one-byte) at its end or directly before each delimiter.
	The folder path string specified by (s4) cannot be read. • The specified file path string contains no character. • The specified folder path string contains 245 characters or more. • The specified folder path hierarchies contains 11 levels or more. • The specified folder path string has a period (one-byte) at its end or directly before each delimiter.
	A system folder ($MELPRJ$) which is directly under the root folder is in the folder path specified by (s2) or (s4).
3582H	The SP.FCOPY instruction is executed in an interrupt program.

If the SP.FCOPY instruction completes with an error, an error code is stored in the device specified by (d1)+1. (Note that an error code is not stored if the instruction results in an operation error.)

For the error code stored in (d1)+1, refer to the following.

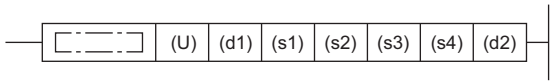
☞ Page 579 Error codes generated for file operation instructions

Moving the specified file

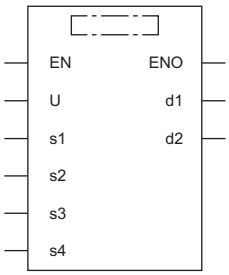
SP.FMOVE

FX5S FX5UJ FX5U FX5UC

This instruction moves the specified file or folder in an SD memory card. When a folder is specified, the specified folder is moved in its entirety, or all the files and subfolders in the specified folder are moved.

Ladder	ST
	ENO:=SP_FMOVE(EN,U,s1,s2,s3,s4,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)	Dummy	■FX5S CPU module U1 ■FX5UJ CPU module U1 to U8 ■FX5U/FX5UC CPU module U1 to U10	Device name	ANY16
(d1)	Start device where the control data is stored	☞ Page 554 Control data (d1)	Word	ANY16_ARRAY (Number of elements: 2)
(s1)	Source drive	2 (fixed)*1	16-bit signed binary	ANY16
(s2)	Start device where the file name or folder name to move is stored	☞ Page 556 File name/folder name (s2)	Unicode string	ANYSTRING_DOUBLE
(s3)	Destination drive	2 (fixed)*1	16-bit signed binary	ANY16
(s4)	Device for storing the destination folder path	☞ Page 557 Destination folder path (s4)	Unicode string	ANYSTRING_DOUBLE
(d2)	Bit device that turns on upon completion of the processing	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Only drive 2 (for the SD memory card) can be set.

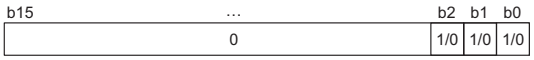
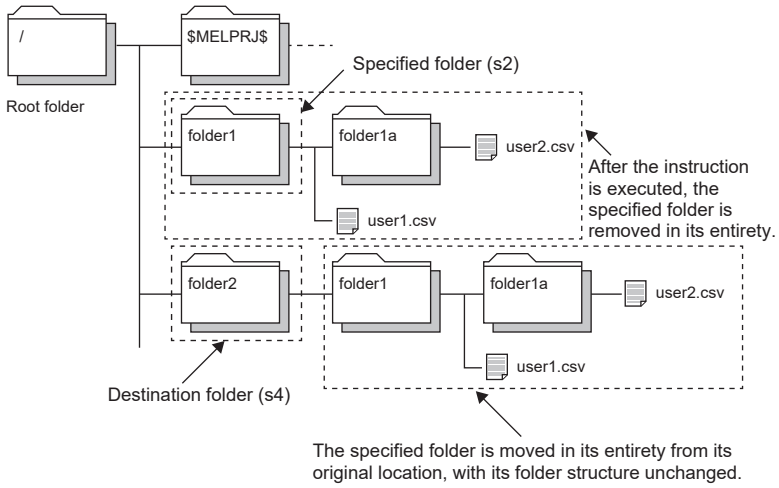
■Applicable devices

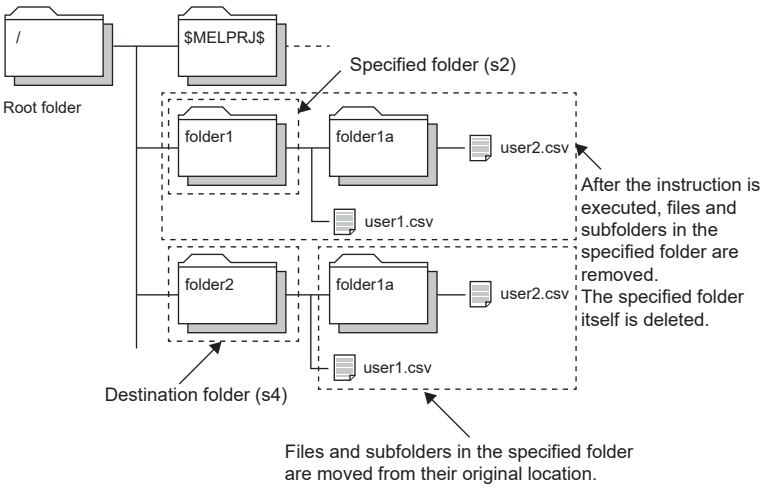
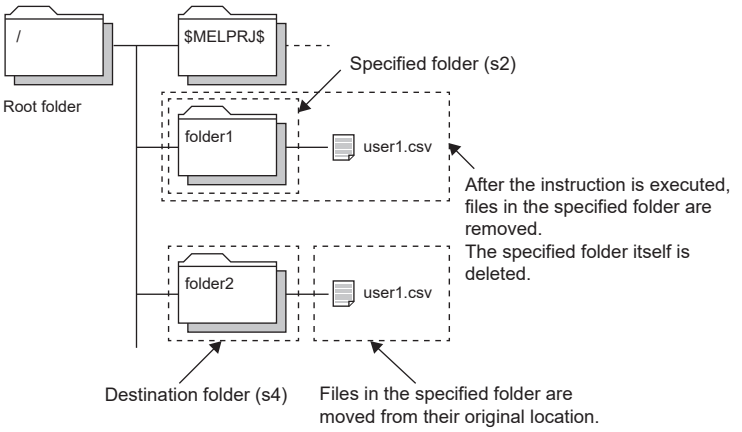
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	—	○
(d1)	—	○	—	—	—	—	○	—	—	—	—
(s1)	○	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	○	—
(s3)	○	○	—	—	—	—	○	○	—	—	—
(s4)	—	○	—	—	—	—	○	—	—	○	—
(d2)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

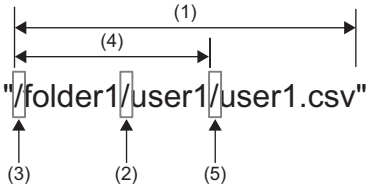
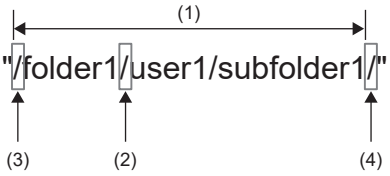
*2 T, ST, and C cannot be used.
Only bit specification of word device is applicable.

■Control data (d1)

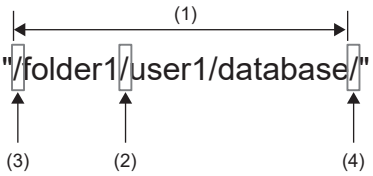
Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Application setting area	 ■b0, b2: Target type setting b0: Specify the type of data (file or folder) to move. • 0: File • 1: Folder b2: Specify how the data is moved. (This bit is valid when b0 is set to 1 (Folder).) • 0: Move the folder in its entirety. • 1: Move all the files and subfolders in the folder. [Operation when b2 is set to 0 (Move the folder in its entirety)] The specified folder (including all files and subfolders in it) is moved in its entirety. 	Refer to the "Description" column.	User

Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Application setting area	[Operation when b2 is set to 1 (Move all the files and subfolders in the folder), and the specified folder includes files and a subfolder] All the files and subfolders in the specified folder are moved.  [Operation when b2 is set to 1 (Move all the files and subfolders in the folder), and the specified folder includes only files (no subfolder)] All the files in the specified folder are moved.  ■b1: Overwrite setting Specify whether to overwrite a file or a folder if one with the same name as the source already exists in the destination. • 0: Do not overwrite the file or folder. • 1: Overwrite the file or folder. When 1 (Folder) and 0 (Do not overwrite the file or folder.) have been specified in b0 and b1, respectively, the movement of the file or folder having the same name as one in the destination is skipped. (The program does not terminate abnormally.) However, even if the folder movement is skipped, the instruction does not skip moving the file/subfolder existing directly under the folder having the same name.	Refer to the "Description" column.	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0000H: Completed successfully • Other than 0000H: Completed with an error (error code) (☞ Page 579 Error codes generated for file operation instructions)	—	System

■File name/folder name (s2)

Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +□	File name character string	When specifying a file, specify the folder path where the file to move is stored and the file name. • When the file name has an extension, specify the name without omitting the extension. • The folder path and file name (including an extension) must be within 253 characters in total. • The folder path must be within 244 characters. (Delimiters are not included.) • Specify one character or more for the file name or folder path in addition to a delimiter. • Do not add a delimiter at the end of a string. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in the system folder. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 253 characters (2): Use "/" or "\" as delimiters for the folder path and file. (3): Can be omitted. When it is omitted, (1) is up to 252 characters. (4): The folder path is up to 244 characters long (243 characters long when (3) is omitted). (5): The delimiter between the folder path and the file name is not included in the number of characters of the folder path.	Unicode string	User
	Folder name character string	When specifying a folder, specify the folder path of the folder to move. • The folder path must be within 244 characters. (Delimiters at the end of the folder path are not included.) • Specify one character or more for the folder path in addition to a delimiter. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in the system folder. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 244 characters (2): Use "/" or "\" as delimiters for the folder path. (3): Can be omitted. When it is omitted, (1) is up to 243 characters. (4): Can be omitted.		

■Destination folder path (s4)

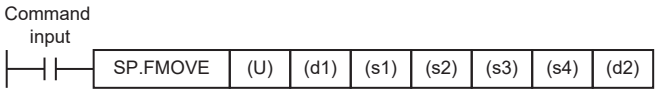
Operand: (s4)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +☐	Folder path	Specify the destination folder path. • The folder path must be within 244 characters. (Delimiters at the end of the folder path are not included.) • Specify one character or more for the folder path in addition to a delimiter. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in the system folder. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 244 characters (2): Use "/" or "\" as delimiters for the folder path and file. (3): Can be omitted. When it is omitted, (1) is up to 243 characters. (4): Can be omitted.	Unicode string	User

Processing details

- This instruction moves the file or folder specified by (s2) in the drive specified by (s1) to the folder specified by (s4) in the drive specified by (s3). When a folder has been specified in (s2) and a file or a folder having the same name exists in the destination, the program does not terminate abnormally even if the value of b1 (overwrite setting) of (d1) is 0. (The movement is skipped.)
- If the folder specified by (s4) does not exist in the destination, the folder is created automatically.
- SM753 (File being accessed) turns on while the SP.FMOVE instruction is being executed. While SM753 is on, the SP.FMOVE instruction cannot be executed. (If the instruction is executed, no processing is performed.)
- The processing completion bit device (d2) automatically turns on at the execution of the END instruction in the scan in which the completion of processing of the SP.FMOVE instruction is detected. The bit device (d2) turns off at the execution of the END instruction in the next scan. If the SP.FMOVE instruction completes with an error, the error completion device (d2)+1 turns on or off in synchronization with (d2). If the processing completion bit device (d2) is ON, it will automatically turn OFF when the SP.FMOVE instruction is executed.
- If an operation error is detected during the execution of the instruction, (d2) and (d2)+1 do not turn on.

■Timing chart

The operation specifications of the file move function are shown below.

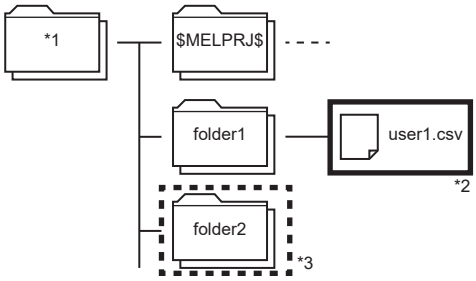


Application setting area (d1)+0	File name/folder name (s2)	Destination folder path (s4)
0H, 1H, 5H, 7H	[When a file is specified] user1.csv [When a folder is specified] folder1	folder2

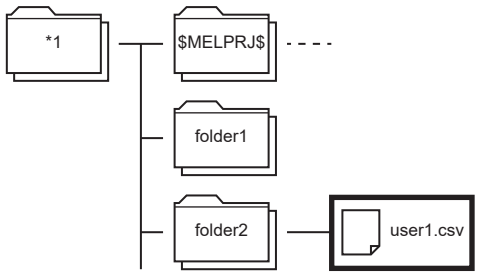
[In SD memory card]

■When application setting area is 0H (File)

Before execution of instruction



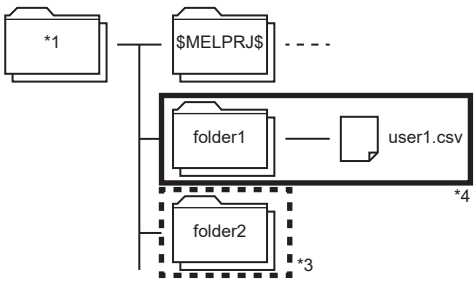
After execution of instruction



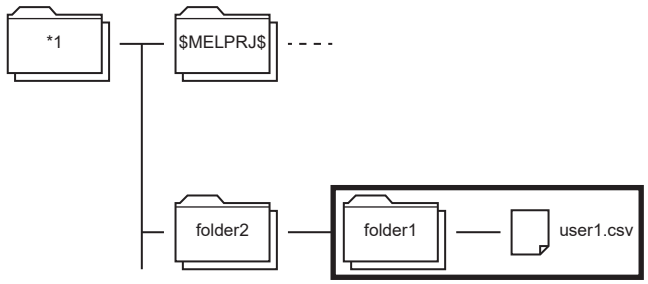
The folder1/user1.csv file is moved into folder2.

■When application setting area is 1H ("Folder", "Move the folder in its entirety.")

Before execution of instruction



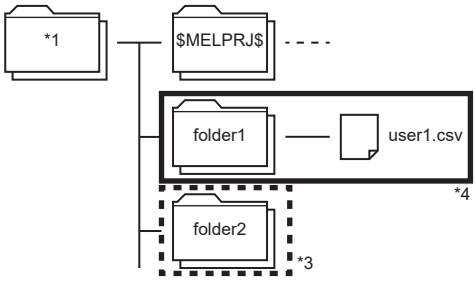
After execution of instruction



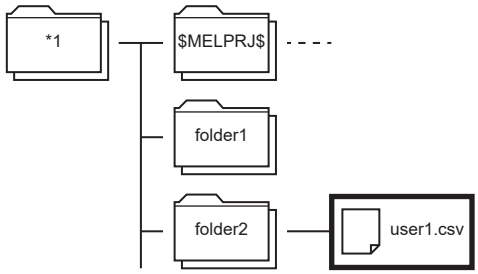
The folder configuration of the source is moved into folder2.

■When application setting area is 5H ("Folder", "Move all the files and subfolders in the folder.")

Before execution of instruction



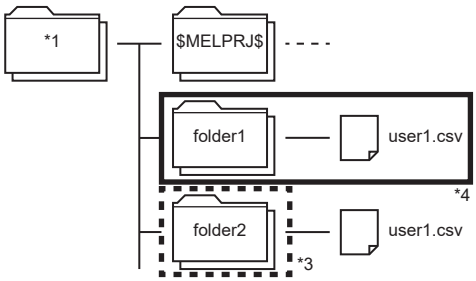
After execution of instruction



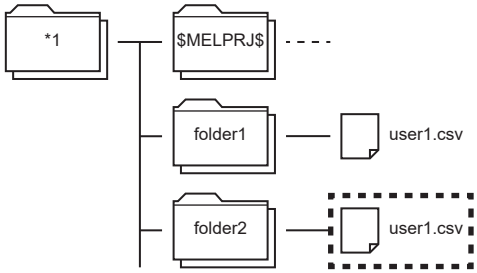
Only the file in the source folder is moved into folder2.

■When application setting area is 5H ("Folder", "Move all the files and subfolders in the folder.", "Do not overwrite the file or folder.")

Before execution of instruction



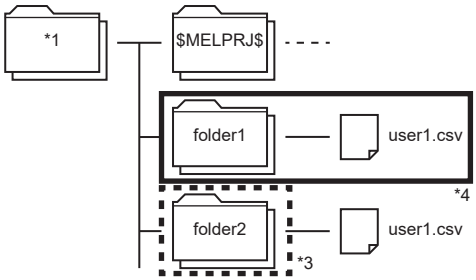
After execution of instruction



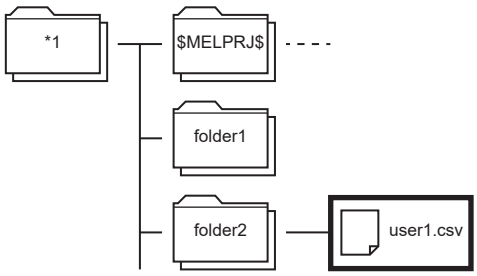
The file is not moved because a file with the same name exists in the destination. The SP.FCOPY instruction is finished normally.
When there is not a file with the same name in the destination, the file is moved.

■When application setting area is 7H ("Folder", "Move all the files and subfolders in the folder.", "Overwrite the file or folder.")

Before execution of instruction



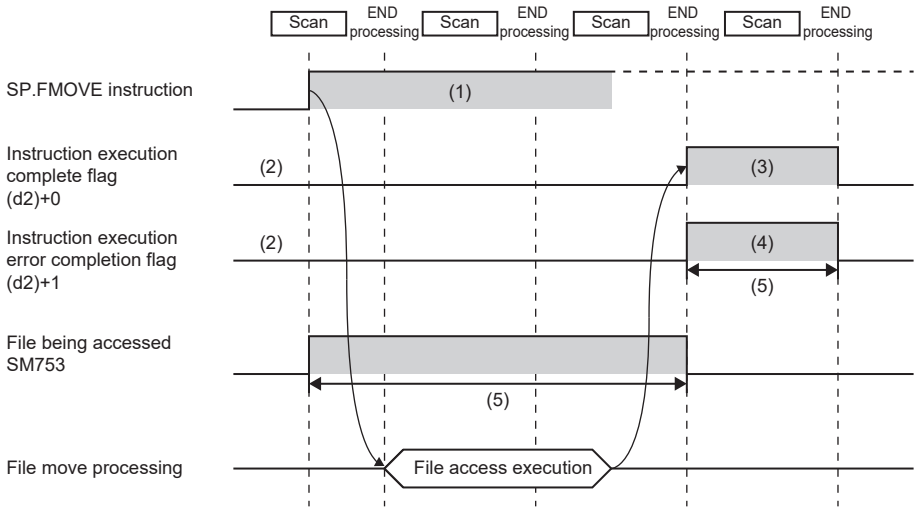
After execution of instruction



folder2/user1.csv is moved overwriting the folder1/user1.csv file.

*1 Root folder
*2 Source file
*3 Destination folder
*4 Source folder

Below is shown the timing chart (flag updating timing) from the execution of the SP.FMOVE instruction to the completion.



(1) No processing during scanning except when the instruction is turned OFF and ON
(2) Updating to OFF when the instruction is turned OFF and ON
(3) Turning ON upon normal/error completion*5
(4) Turning OFF upon normal completion and turning ON upon error completion*5
(5) Turning ON only during 1 scan after instruction execution
(6) Turning OFF when instruction execution complete flag is turned ON

*5 The complete flag is not turned ON when an error is detected during instruction execution.

Program example

When X0 is turned ON, the “sample” folder stored in the SD memory card is moved to the “sample1” folder.

[Program operation]

1. Control data is created during RUN.
2. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
3. The SP.FMOVE instruction is executed.*1
4. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*1 The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.
- SM606 (Memory card disable request)
- SM753 (File being accessed)

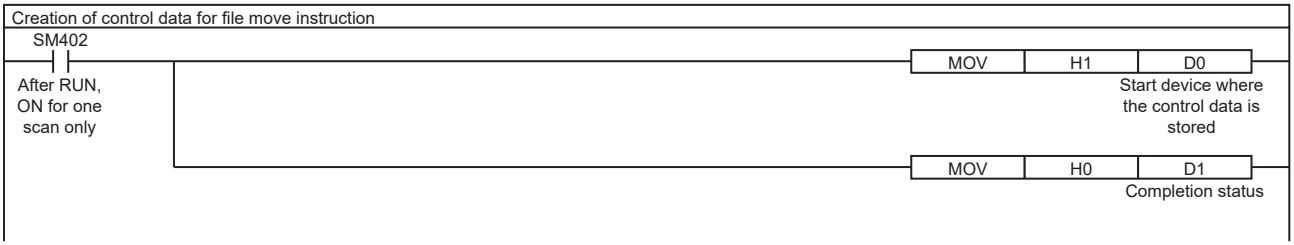
[Devices used]

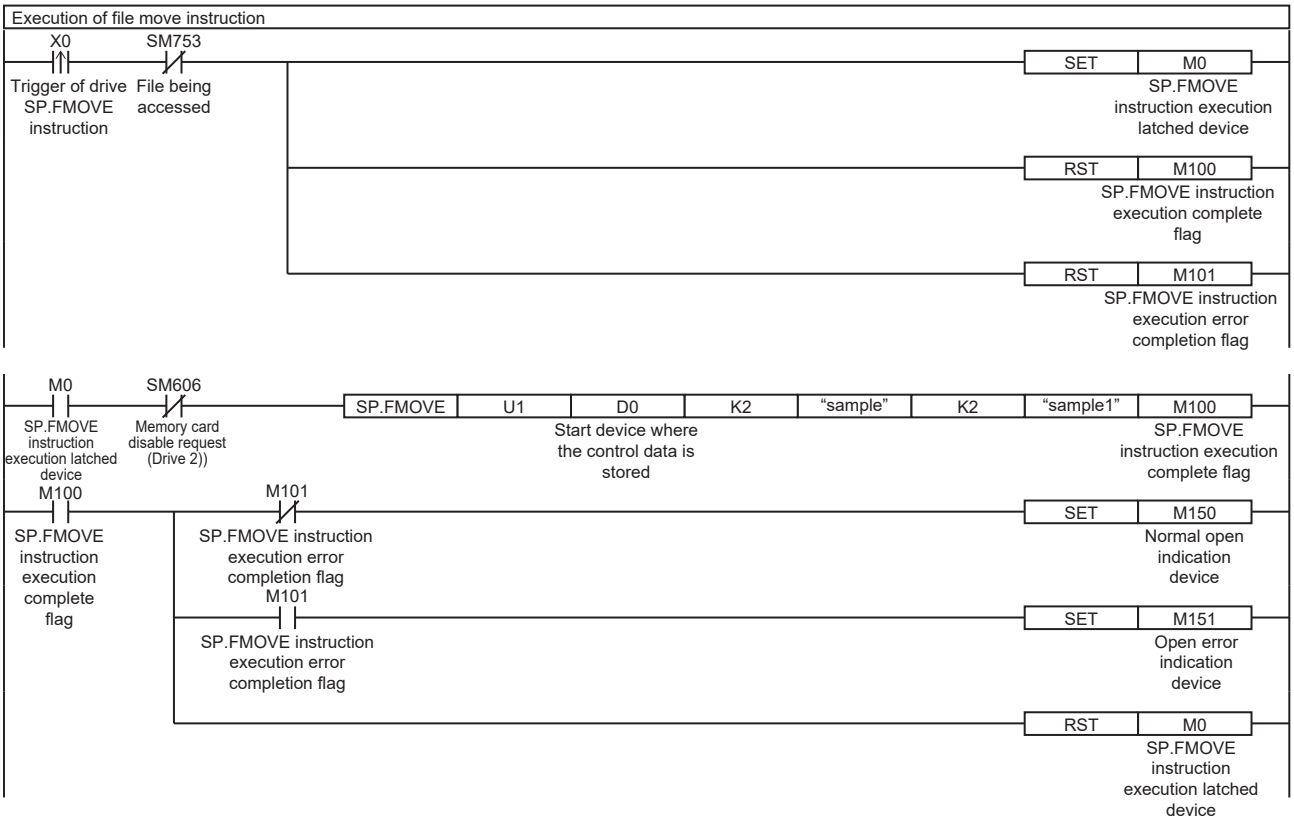
Device	Description
X0	Trigger of drive SP.FMOVE instruction
D0	Start device where the control data is stored • D0: Application setting area • D1: Completion status
M0	SP.FMOVE instruction execution latched device
M100	SP.FMOVE instruction execution complete flag
M101	SP.FMOVE instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FMOVE instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(d1)	Start device where the control data is stored	D0: 1H (File) D1: 0H (Completed successfully)
(s1)	Source drive	K2 (SD memory card)
(s2)	Start device where the file name to move is stored	“sample”
(s3)	Destination drive	K2 (SD memory card)
(s4)	Device for storing the destination folder path	“sample1”
(d2)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]





[ST program]

```
//(1) Creation of control data for file move instruction
IF SM402 THEN
D0:=H1; //Application setting area(Folder, move the folder in its entirety)
D1:=H0; //Completion status
END_IF;

//(2) Processing to start up the drive contact (X0)
IF LDP(TRUE,X0) THEN;
//Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE,M0); //Holds drive contact
RST(TRUE,M100); //Initialize instruction execution complete flag
RST(TRUE,M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(3) Execution of file move instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN;
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "sample" (Start device where the file name to move is stored)
//S3 = 2 (Drive specification, 2 fixed)
//S4 = "sample1" (Device for storing the destination folder path)
//D1 = D0 (Start device where the control data is stored)
//D2 = M100 (Bit device that turns on upon completion of the processing)
SP_FMOVE(TRUE, U1, 2, "sample", 2, "sample1", D0, M100);
END_IF;
END_IF;

//(4) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```

Precautions

- Do not execute the SP.FMOVE instruction in an interrupt program. Doing so may cause malfunction of the module.
- The SP.FMOVE instruction cannot be executed while SM606 (Memory card disable request) is ON. If SM606 is turned ON after the completion of the initial END processing during execution of the instruction, the program will terminate abnormally upon detection of turning ON.
- The SP.FMOVE instruction specifying the system folder ($MELPRJ$) or a file in the system folder cannot be executed.
- Even though the operating status of the CPU module is switched from RUN to STOP during instruction execution, the CPU module continues the processing of the instruction.
- Ensure that the number of characters in a folder path or in the total of a folder path and a file name after moving does not exceed its limit. (The file may be corrupt or an error may occur.)
- If overwriting occurs, the time to complete the instruction may be extended.
- When overwriting occurs, ensure a free space of the same size as the operation target file.
- If the instruction is completed with an error during processing, the file or folder may remain under processing.
- If the size of a file to be moved or the number of files to be moved becomes larger due to overwriting, the instruction will take more time to complete.
- Do not access the file being processed by the SP.FMOVE instruction from other functions. (The file may be inaccessible or an error may occur.)
- Do not operate files or folders being accessed from other functions.
- The SP.FMOVE instruction cannot be executed simultaneously with the SP.DEVST instruction, the SP.FTPPUT instruction, and the SP.FTPGET.
- Do not disconnect the power or remove the SD memory card during execution of the SP.FMOVE instruction. (The file may be damaged, or an error may occur.)

Operation error

Error code (SD0/SD8067)	Description
2820H	The storage device of the control data (d1) exceeds the end of the device range.
	The bit label digits specified in (s2) and (s4) are unacceptable settings (the number of digits is not K4).
	The size of the read data exceeds the read device size.
3405H	The drive specified by (s1), (s3) is not the one for the SD memory card.
	The file name/folder name string specified by (s2) cannot be read. • The specified file name string contains no character. • The specified file name string contains 254 characters or more. • The specified folder path contains 245 characters or more. • The set value is unusable. • The specified folder path hierarchies contains 11 levels or more. • When a file is specified, the file name string ends with a delimiter. • The specified file name string has a period (one-byte) at its end or directly before each delimiter.
	The folder path string specified by (s4) cannot be read. • The specified file path string contains no character. • The specified folder path string contains 245 characters or more. • The specified folder path hierarchies contains 11 levels or more. • The specified folder path string has a period (one-byte) at its end or directly before each delimiter.
	A system folder ($MELPRJ$) which is directly under the root folder is in the folder path specified by (s2) or (s4).
3582H	The SP.FMOVE instruction is executed in an interrupt program.

If the SP.FMOVE instruction completes with an error, an error code is stored in the device specified by (d1)+1. (Note that an error code is not stored if the instruction results in an operation error.)

For the error code stored in (d1)+1, refer to the following.

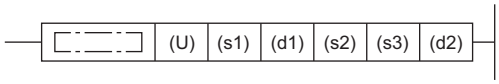
☞ Page 579 Error codes generated for file operation instructions

Renaming the specified file

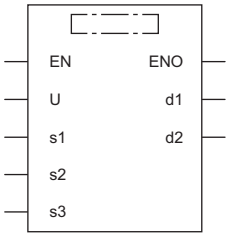
SP.FRENAME

FX5S FX5UJ FX5U FX5UC

This instruction renames the specified file or folder in an SD memory card.

Ladder	ST
	ENO:=SP_FRENAME(EN,U,s1,s2,s3,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)	Dummy	■FX5S CPU module U1 ■FX5UJ CPU module U1 to U8 ■FX5U/FX5UC CPU module U1 to U10	Device name	ANY16
(s1)	Drive specification	2 (fixed)*1	16-bit signed binary	ANY16
(d1)	Start device where the control data is stored	☞ Page 564 Control data (d1)	Word	ANY16_ARRAY (Number of elements: 2)
(s2)	Start device where the file name or folder name to be changed is stored	☞ Page 565 File name/folder name to be changed (s2)	Unicode string	ANYSTRING_DOUBLE
(s3)	Start device where the new file name/folder name after the change is stored	☞ Page 566 File name/folder name after the change (s3)	Unicode string	ANYSTRING_DOUBLE
(d2)	Bit device that turns on upon completion of the processing	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Only drive 2 (for the SD memory card) can be set.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	○	—	—	—	—	○	○	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	○	—
(s3)	—	○	—	—	—	—	○	—	—	○	—
(d2)	○*1	○*2	—	—	—	—	—	—	—	—	—

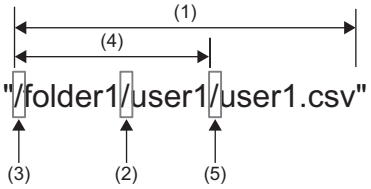
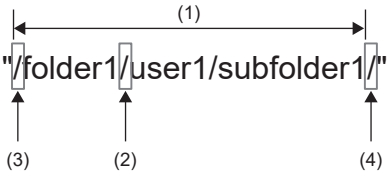
*1 S cannot be used.

*2 T, ST, and C cannot be used.
Only bit specification of word device is applicable.

■Control data (d1)

Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Application setting area	b15 … b0 0 | 1/0 b0: Target type setting Specify the type of data (file or folder) to rename. • 0: File • 1: Folder	Refer to the "Description" column.	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0000H: Completed successfully • Other than 0000H: Completed with an error (error code) (☞ Page 579 Error codes generated for file operation instructions)	—	System

■File name/folder name to be changed (s2)

Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +☐	File name character string	When specifying a file, specify the folder path where the file to be renamed is stored and the file name. • When the file name has an extension, specify the name without omitting the extension. • The folder path and file name (including an extension) must be within 253 characters in total. • The folder path must be within 244 characters. (Delimiters are not included.) • Specify one character or more for the file name or folder path in addition to a delimiter. • Do not add a delimiter at the end of a string. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in the system folder. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 253 characters (2): Use "/" or "\" as delimiters for the folder path and file. (3): Can be omitted. When it is omitted, (1) is up to 252 characters. (4): The folder path is up to 244 characters long (243 characters long when (3) is omitted). (5): The delimiter between the folder path and the file name is not included in the number of characters of the folder path.	Unicode string	User
	Folder name character string	When specifying a folder, specify the folder path of the folder to be renamed. • The folder path must be within 244 characters. (Delimiters at the end of the folder path are not included.) • Specify one character or more for the folder path in addition to a delimiter. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in the system folder. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 244 characters (2): Use "/" or "\" as delimiters for the folder path. (3): Can be omitted. When it is omitted, (1) is up to 243 characters. (4): Can be omitted.		

■File name/folder name after the change (s3)

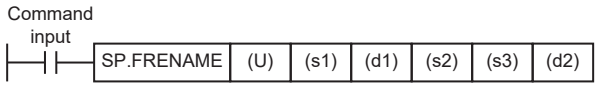
Operand: (s3)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +□	File name/folder name	Specify the string of the file name/folder name after the change. • When the file name has an extension, specify the name without omitting the extension. • Do not specify a folder path. • The file name (including an extension) must be within 252 characters. • The folder name must be within 243 characters. • Specify one character or more for the file name or folder name. • Do not specify a delimiter. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string. • Do not specify the system folder ($MELPRJ$) or any file in the system folder.	Unicode string	User

Processing details

- This instruction rename the file or folder specified by (s2) in the drive specified by (s1) to the file name or folder name specified by (s3). If the file name or folder name specified by (s3) already exists, the instruction is completed with an error.
- SM753 (File being accessed) turns on while the SP.FRENAME instruction is being executed. While SM753 is on, the SP.FRENAME instruction cannot be executed. (If the instruction is executed, no processing is performed.)
- The processing completion bit device (d2) automatically turns on at the execution of the END instruction in the scan in which the completion of processing of the SP.FRENAME instruction is detected. The bit device (d2) turns off at the execution of the END instruction in the next scan. If the SP.FRENAME instruction completes with an error, the error completion device (d2)+1 turns on or off in synchronization with (d2). If the processing completion bit device (d2) is ON, it will automatically turn OFF when the SP.FRENAME instruction is executed.
- If an operation error is detected during the execution of the instruction, (d2) and (d2)+1 do not turn on.
- If the file path + folder name and the folder path exceed the limit number of characters after renaming, the program is finished abnormally. (8006H is stored in (d1)+1.)

■Timing chart

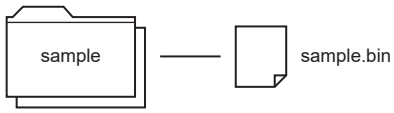
The operation specifications of the file rename function are shown below.



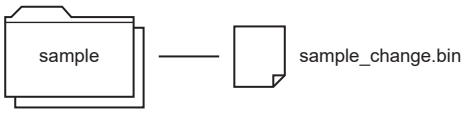
Application setting area (d1)+0	File name to be changed (s2)	File name after the change (s3)
0H	sample/sample.bin	sample_change.bin

[In SD memory card]

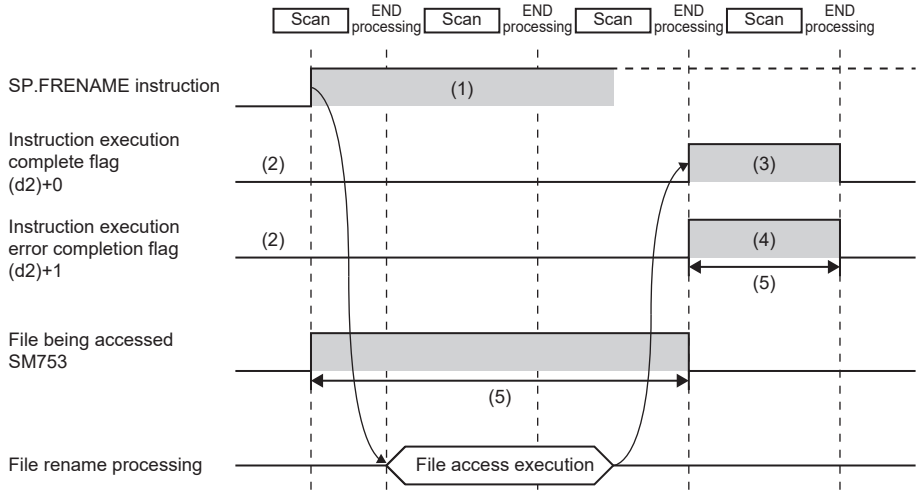
■Before execution of instruction



■After execution of instruction



Below is shown the timing chart (flag updating timing) from the execution of the SP.FRENAME instruction to the completion.



(1) No processing during scanning except when the instruction is turned OFF and ON
(2) Updating to OFF when the instruction is turned OFF and ON
(3) Turning ON upon normal/error completion *1
(4) Turning OFF upon normal completion and turning ON upon error completion*1
(5) Turning ON only during 1 scan after instruction execution
(6) Turning OFF when instruction execution complete flag is turned ON

*1 The complete flag is not turned ON when an error is detected during instruction execution.

Program example

When X0 is turned ON, the file name of “sample.bin” stored in the SD memory card is changed to “sample_change.bin”.
[Program operation]

1. Control data is created during RUN.
2. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
3. The SP.FRENAME instruction is executed.*1
4. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*1 The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.
· SM606 (Memory card disable request)
· SM753 (File being accessed)

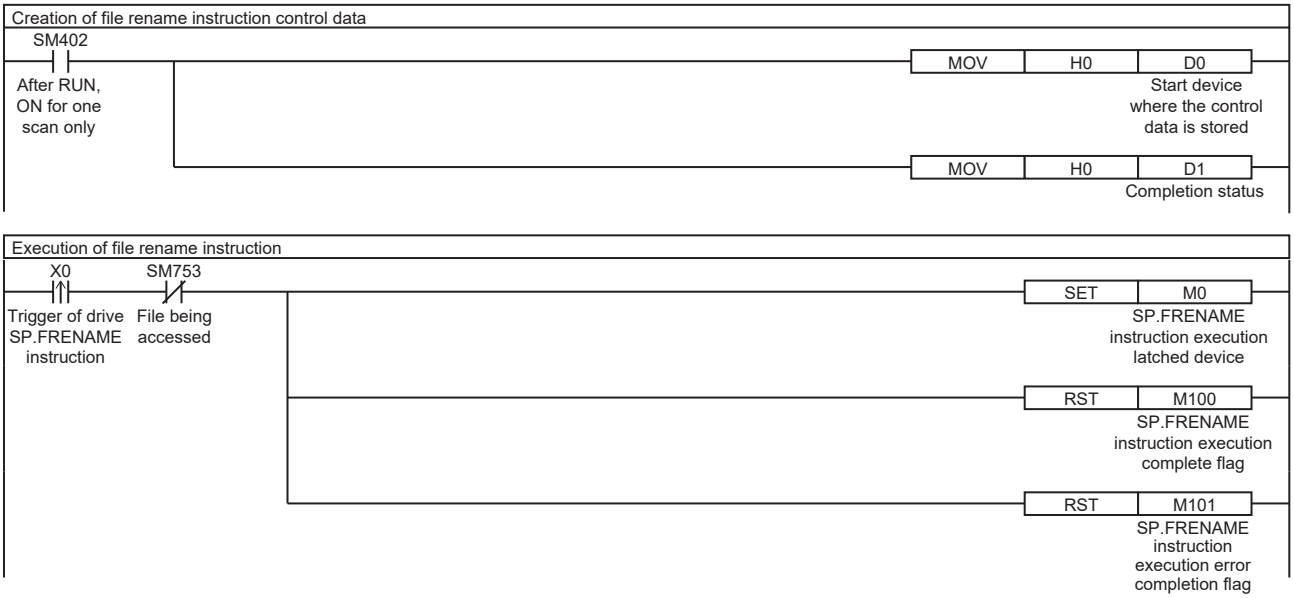
[Devices used]

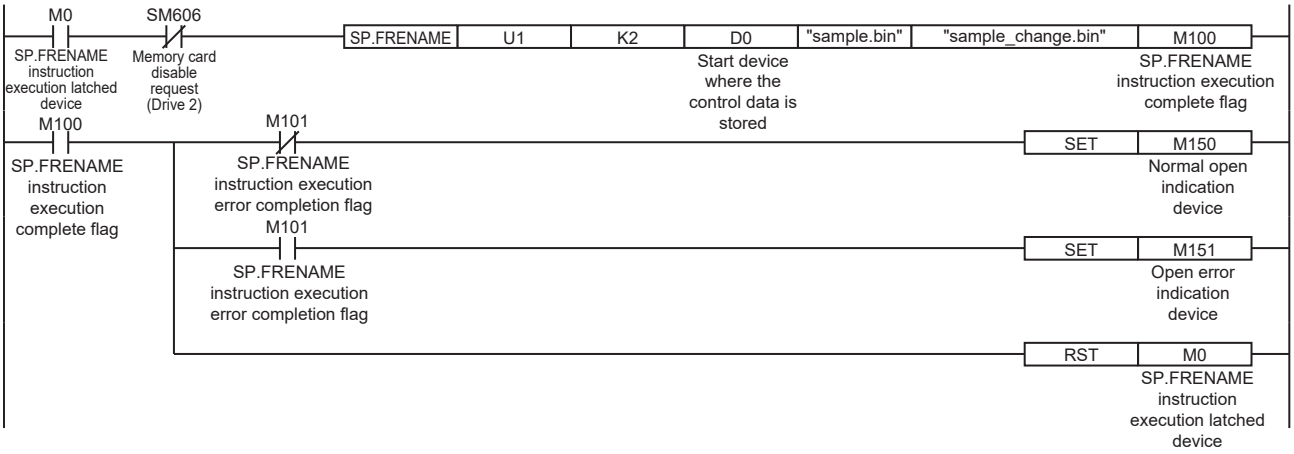
Device	Description
X0	Trigger of drive SP.FRENAME instruction
D0	Start device where the control data is stored • D0: Application setting area • D1: Completion status
M0	SP.FRENAME instruction execution latched device
M100	SP.FRENAME instruction execution complete flag
M101	SP.FRENAME instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FRENAME instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(s1)	Drive specification	K2 (SD memory card)
(d1)	Start device where the control data is stored	D0: 0H (File) D1: 0H (Completed successfully)
(s2)	Start device where the file name to be changed is stored	"sample.bin"
(s3)	Start device where the new file name after the change is stored	"sample_change.bin"
(d2)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]





[ST program]

```
//(1) Creation of control data for file rename instruction
IF SM402 THEN
D0:=H0; //Application setting area (File)
D1:=H0; //Completion status
END_IF;

//(2) Processing to start up the drive contact (X0)
IF LDP(TRUE,X0) THEN;
//Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE,M0); //Holds drive contact
RST(TRUE,M100); //Initialize instruction execution complete flag
RST(TRUE,M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(3) Execution of file rename instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN;
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "sample.bin" (Start device where the file name to be changed is stored)
//S3 = "sample_change.bin" (Start device where the new file name after the change is stored)
//D1 = D0 (Start device where the control data is stored)
//D2 = M100 (Bit device that turns on upon completion of the processing)
SP_FRENAME(TRUE, U1, 2, "sample.bin", "sample_change.bin", D0, M100);
END_IF;
END_IF;

//(4) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```

Precautions

- Do not execute the SP.FRENAME instruction in an interrupt program. Doing so may cause malfunction of the module.
- The SP.FRENAME instruction cannot be executed while SM606 (Memory card disable request) is ON. When SM606 is turned ON during execution of the instruction, the program will terminate abnormally. (If the fine renaming has been completed before SM606 is turned ON, the file name will be changed.)
- The SP.FRENAME instruction specifying the system folder ($MELPRJ$) or a file in the system folder cannot be executed.
- Even though the operating status of the CPU module is switched from RUN to STOP during instruction execution, the CPU module continues the processing of the instruction.
- Ensure that the number of characters in a folder path or in the total of a folder path and a file name after changing does not exceed its limit. (The file may be inaccessible or an error may occur.)
- Do not access the file being processed by the SP.FRENAME instruction from other functions. (The file may be corrupt or an error may occur.)
- Do not operate files or folders being accessed from other functions.
- The SP.FRENAME instruction cannot be executed simultaneously with the SP.DEVST instruction, the SP.FTPPUT instruction, and the SP.FTPGET.
- Do not disconnect the power or remove the SD memory card during execution of the SP.FRENAME instruction. (The file may be damaged, or an error may occur.)

Operation error

Error code (SD0/SD8067)	Description
2820H	The storage device of the control data (d1) exceeds the end of the device range.
	The bit label digits specified in (s2) and (s3) are unacceptable settings (the number of digits is not K4).
3405H	The drive specified by (s1) is not the one for the SD memory card.
	The file name/folder name string specified by (s2) cannot be read. • The specified file name string contains no character. • The specified file name string contains 254 characters or more. • The specified folder path contains 245 characters or more. • The specified folder path hierarchies contains 11 levels or more. • When a file is specified, the file name string ends with a delimiter. • The specified file name string has a period (one-byte) at its end or directly before each delimiter.
	The file name/folder name string specified by (s3) cannot be read. • The specified file name/folder name string contains no character. • The specified file name string contains 253 characters or more. • The specified folder name string contains 244 characters or more. • The specified file name string contains a delimiter. • The specified file name string has a period (one-byte) at its end.
	A system folder ($MELPRJ$) which is directly under the root folder is in the folder path specified by (s2).
	$MELPRJ$ is specified for the file name/folder name specified by (s3).
3582H	The SP.FRENAME instruction is executed in an interrupt program.

If the SP.FRENAME instruction completes with an error, an error code is stored in the device specified by (d1)+1. (Note that an error code is not stored if the instruction results in an operation error.)

For the error code stored in (d1)+1, refer to the following.

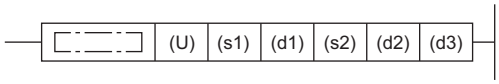
☞ Page 579 Error codes generated for file operation instructions

Acquiring the status of the specified file

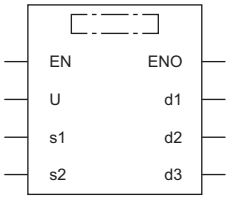
SP.FSTATUS

FX5S FX5UJ FX5U FX5UC

This instruction acquires the status of the specified file or folder in an SD memory card.

Ladder	ST
	ENO:=SP_FSTATUS(EN,U,s1,s2,d1,d2,d3);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)	Dummy	■FX5S CPU module U1 ■FX5UJ CPU module U1 to U8 ■FX5U/FX5UC CPU module U1 to U10	Device name	ANY16
(s1)	Drive specification	2 (fixed)*1	16-bit signed binary	ANY16
(d1)	Start device where the control data is stored	☞ Page 572 Control data (d1)	Word	ANY16_ARRAY (Number of elements: 2)
(s2)	Start device where the file name or folder name is stored	☞ Page 573 File name/folder name (s2)	Unicode string	ANYSTRING_DOUBLE
(d2)	Start device for storing the file status	☞ Page 574 File status (d2)	Word	ANY16_ARRAY (Number of elements: 10)
(d3)	Bit device that turns on upon completion of the processing	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Only drive 2 (for the SD memory card) can be set.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	○	—	—	—	—	○	○	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	○	—
(d2)	○*1	○	—	—	—	—	○	—	—	—	—
(d3)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 When the bit device digit is specified in (d2), only multiples of 16 (0, 16, 32, 64…) can be specified as the device number. Only K4 can be specified as the number of digits.

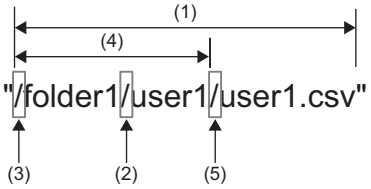
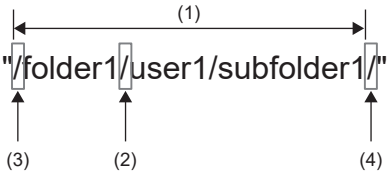
*2 S cannot be used.

*3 T, ST, and C cannot be used.
Only bit specification of word device is applicable.

■Control data (d1)

Operand: (d1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Application setting area	b15 … b0 0 | 1/0 b0: Target type setting Specify the type of data (file or folder) to acquire the status. • 0: File • 1: Folder	Refer to the "Description" column.	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0000H: Completed successfully • Other than 0000H: Completed with an error (error code) (☞ Page 579 Error codes generated for file operation instructions)	—	System

■File name/folder name (s2)

Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0 to +☐	File name character string	When specifying a file, specify the folder path where the file whose status is to be acquired is stored and the file name. • When the file name has an extension, specify the name without omitting the extension. • The folder path and file name (including an extension) must be within 253 characters in total. • The folder path must be within 244 characters. (Delimiters are not included.) • Specify one character or more for the file name or folder path in addition to a delimiter. • Do not add a delimiter at the end of a string. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in the system folder. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 253 characters (2): Use "/" or "\" as delimiters for the folder path and file. (3): Can be omitted. When it is omitted, (1) is up to 252 characters. (4): The folder path is up to 244 characters long (243 characters long when (3) is omitted). (5): The delimiter between the folder path and the file name is not included in the number of characters of the folder path.	Unicode string	User
	Folder name character string	When specifying a folder, specify the folder path of the folder whose status is to be acquired. • The folder path must be within 244 characters. (Delimiters at the end of the folder path are not included.) • Specify one character or more for the folder path in addition to a delimiter. • Do not specify a half-width space at the end of the character string or just before each delimiter. • Do not put a period (one-byte) at the end of a string or directly before each delimiter. • Do not specify the system folder ($MELPRJ$) or any file in the system folder. • The number of folder path hierarchies must be within 10 levels.  (1): Up to 244 characters (2): Use "/" or "\" as delimiters for the folder path. (3): Can be omitted. When it is omitted, (1) is up to 243 characters. (4): Can be omitted.		

■File status (d2)

Operand: (d2)			
Device	**Item**	**Range**	**Set by**
+0	File attribute bit0: Turns on for a read-only file. bit1: Turns on for a hidden file. bit2: Turns on for a system file. bit3: Reserved (fixed to 0) bit4: Turns on for a directory. bit5: Turns on for an archive. bit6 to bit15: Reserved (fixed to 0)	Refer to the "Item" column.	System
+1	Reserved	0	
+2 to +3	File size (in units of bytes)	0 to 4294967294*1	
+4	Last update date/time: Year	0, 1980 to 2079*1	
+5	Last update date/time: Month	0 to 12	
+6	Last update date/time: Day	0 to 31	
+7	Last update date/time: Hour	0 to 23	
+8	Last update date/time: Minute	0 to 59	
+9	Last update date/time: Second		

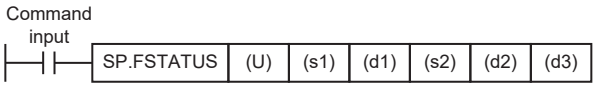
*1 In the case of files/folders accessed in an environment (OS) other than the CPU module, the range of values to be acquired depends on the environment (OS).

Processing details

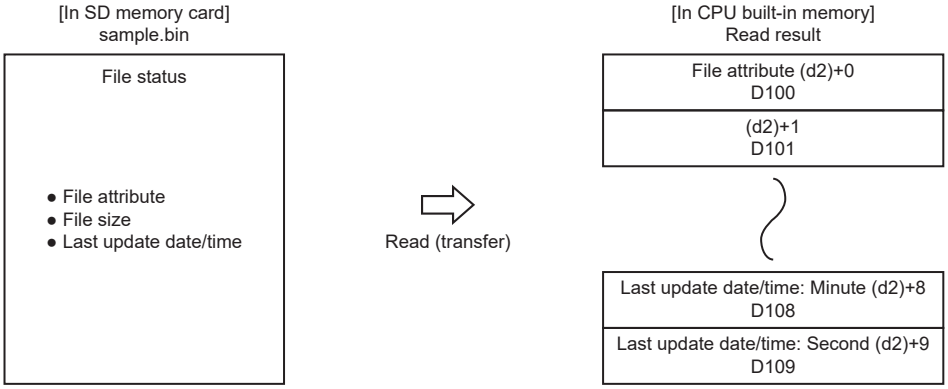
- This function acquires the status of the file or folder specified by (s2) in the drive specified by (s1) and stores it in the device (d2) and after. When a folder is specified by (s2), 0 is stored in (d2)+2 to (d2)+3.
- SM753 (File being accessed) turns on while the SP.FSTATUS instruction is being executed. While SM753 is on, the SP.FSTATUS instruction cannot be executed. (If the instruction is executed, no processing is performed.)
- The processing completion bit device (d3) automatically turns on at the execution of the END instruction in the scan in which the completion of processing of the SP.FSTATUS instruction is detected. The bit device (d3) turns off at the execution of the END instruction in the next scan. If the SP.FSTATUS instruction completes with an error, the error completion device (d3)+1 turns on or off in synchronization with (d3). If the processing completion bit device (d3) is ON, it will automatically turn OFF when the SP.FSTATUS instruction is executed.
- If an operation error is detected during the execution of the instruction, (d2) and (d2)+1 do not turn on.

■Timing chart

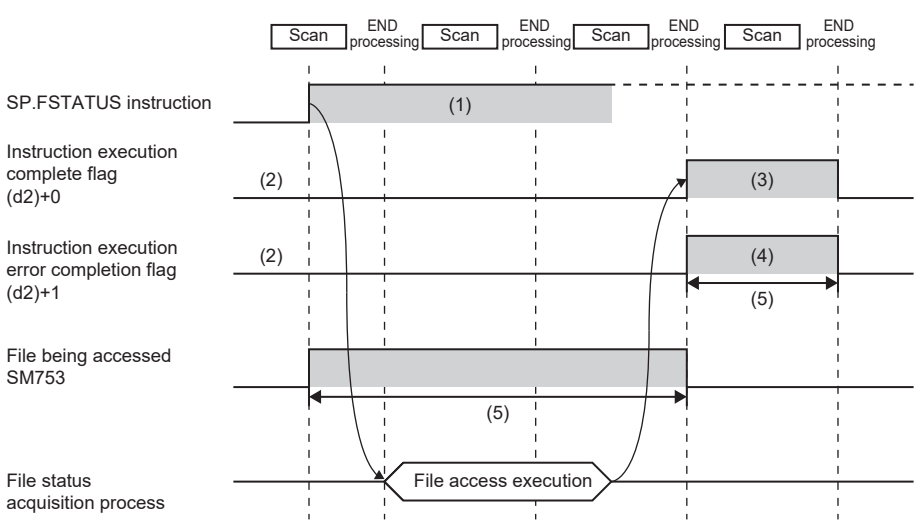
The operation specifications of the file status acquisition function are shown below.



Application setting area (d1)+0	File name (s2)	File status (d2)
0H	sample.bin	D100 to D109



Below is shown the timing chart (flag updating timing) from the execution of the SP.FSTATUS instruction to the completion.



(1) No processing during scanning except when the instruction is turned OFF and ON
(2) Updating to OFF when the instruction is turned OFF and ON
(3) Turning ON upon normal/error completion *1
(4) Turning OFF upon normal completion and turning ON upon error completion*1
(5) Turning ON only during 1 scan after instruction execution
(6) Turning OFF when instruction execution complete flag is turned ON

*1 The complete flag is not turned ON when an error is detected during instruction execution.

Program example

When X0 is turned ON, the status of the “sample.bin” file stored in the SD memory card is read to D100 to D109.

[Program operation]

1. Control data is created during RUN.
2. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
3. The SP.FSTATUS instruction is executed.*1
4. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*1 The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.
- SM606 (Memory card disable request)
- SM753 (File being accessed))

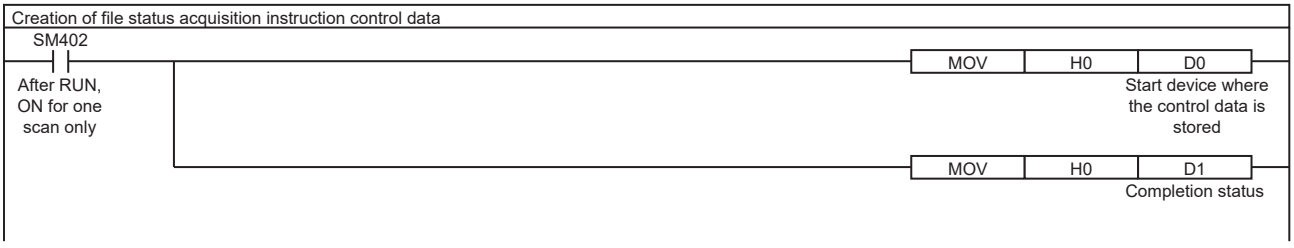
[Devices used]

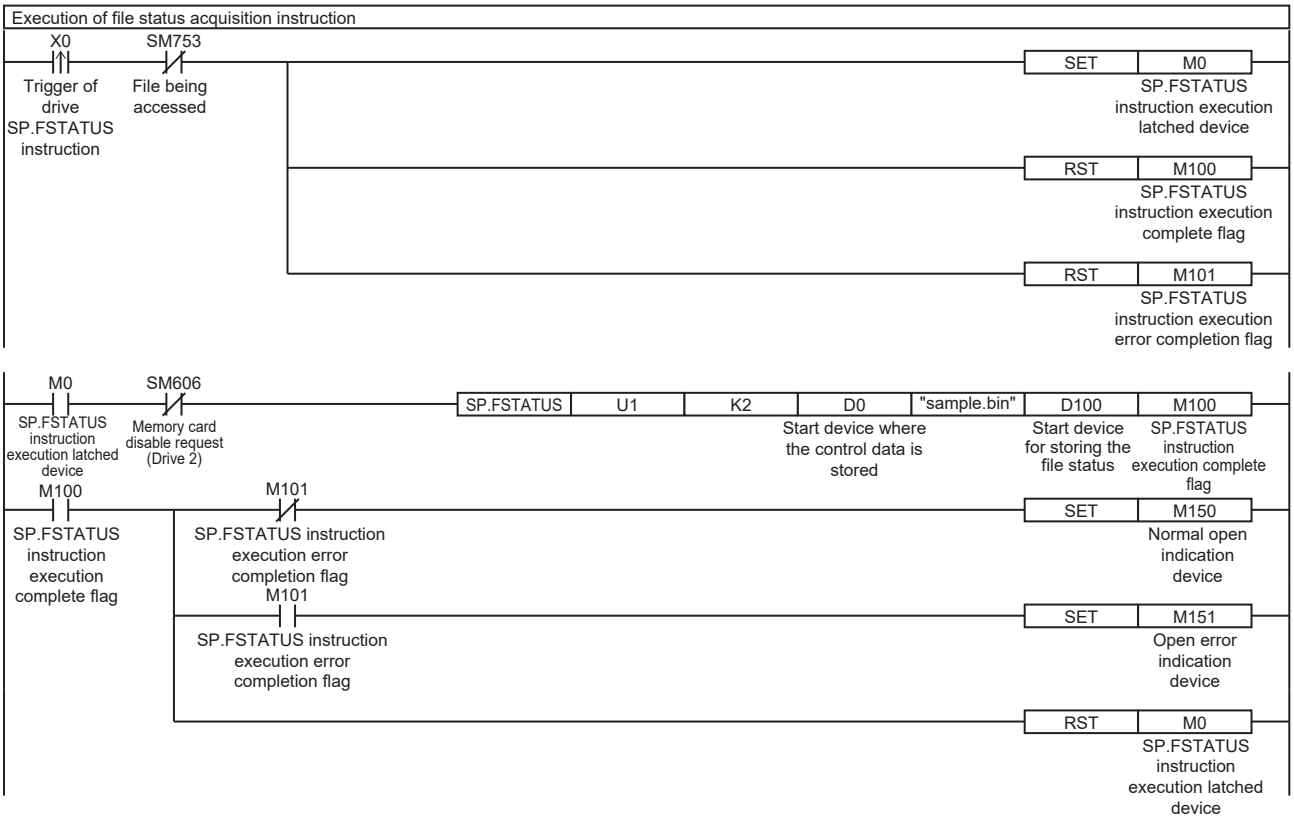
Device	Description
X0	Trigger of drive SP.FSTATUS instruction
D0	Start device where the control data is stored • D0: Application setting area • D1: Completion status
D100	Start device for storing the file status
M0	SP.FSTATUS instruction execution latched device
M100	SP.FSTATUS instruction execution complete flag
M101	SP.FSTATUS instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FSTATUS instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(s1)	Drive specification	K2 (SD memory card)
(d1)	Start device where the control data is stored	D0: 0H (File) D1: 0H (Completed successfully)
(s2)	Start device where the file name is stored	"sample.bin"
(d2)	Start device for storing the file status	D100
(d3)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]





[ST program]

```
//(1) Creation of control data for file status acquisition instruction
IF SM402 THEN
D0:=H0; //Application setting area (File)
D1:=H0; //Completion status
END_IF;

//(2) Processing to start up the drive contact (X0)
IF LDP(TRUE,X0) THEN;
//Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE,M0); //Holds drive contact
RST(TRUE,M100); //Initialize instruction execution complete flag
RST(TRUE,M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(3) Execution of file status acquisition instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN;
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "sample.bin" (Start device where the file name is stored)
//D1 = D0 (Start device where the control data is stored)
//D2 = D100 (Start device for storing the file status)
//D3 = M100 (Bit device that turns on upon completion of the processing)
SP_FSTATUS(TRUE, U1, 2, "sample.bin",  D0, D100, M100);
END_IF;
END_IF;

//(4) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```

Precautions

- Do not execute the SP.FSTATUS instruction in an interrupt program. Doing so may cause malfunction of the module.
- The SP.FSTATUS instruction cannot be executed while SM606 (Memory card disable request) is ON. When SM606 is turned ON during execution of the instruction, the program will terminate abnormally.
- The SP.FSTATUS instruction specifying the system folder ($MELPRJ$) or a file in the system folder cannot be executed.
- Even though the operating status of the CPU module is switched from RUN to STOP during instruction execution, the CPU module continues the processing of the instruction.
- The SP.FSTATUS instruction cannot be executed simultaneously with the SP.DEVST instruction, the SP.FTPPUT instruction, and the SP.FTPGET.
- Do not disconnect the power or remove the SD memory card during execution of the SP.FSTATUS instruction.

Operation error

Error code (SD0/SD8067)	Description
2820H	The storage device of the control data (d1) exceeds the end of the device range.
	The number of bit label digits specified in (s2) is an unallowable setting (the number of digits is not K4).
	The device to store the read file status in (d2) exceeds the end of the device area.
	The number of bit label digits specified in (d2) is an unallowable setting. (The device No. is not a multiple of 16 (0, 16, 32, 64…)/the number of digits is not K4.)
3405H	The drive specified by (s1) is not the one for the SD memory card.
	The file name/folder name string specified by (s2) cannot be read. • The specified file name string contains no character. • The specified file name string contains 254 characters or more. • The specified folder path contains 245 characters or more. • Unusable characters (prohibited characters) are set. • The specified folder path hierarchies contains 11 levels or more. • When a file is specified, the file name string ends with a delimiter. • The specified file name string has a period (one-byte) at its end or directly before each delimiter. • The system folder ($MELPRJ$) directly under the root folder or a file/folder under the system folder is specified.
3582H	The SP.FSTATUS instruction is executed in an interrupt program.

If the SP.FSTATUS instruction completes with an error, an error code is stored in the device specified by (d1)+1. (Note that an error code is not stored if the instruction results in an operation error.)

For the error code stored in (d1)+1, refer to the following.

☞ Page 579 Error codes generated for file operation instructions

Error codes generated for file operation instructions

The following table lists the error codes that could be stored in the completion status of file operation instructions.

Error code	Description	Action
8000H	SM606 (SD memory card forced disable instruction) is on.	Turn off SM606 to cancel the SD memory card forced disable status.
	No SD memory card is inserted.	Insert an SD memory card.
	The SD memory card is not mounted.	Mount the SD memory card.
8001H	An access to the SD memory card has failed.	• Check that the file name string are correctly specified. • Check that no other functions are accessing the file. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the SD memory card. Please consult your local Mitsubishi representative.
	The SD memory card is write-protected.	Unlock the write protect switch of the SD memory card.
	The file is set for read only.	Clear the read-only setting.
	The size of the file to be stored in the SD memory card exceeds the free space or the maximum capacity of the SD memory card.	• Delete unnecessary files in the SD memory card to secure free space. • Adjust the file size so that it is equal to or less than its limit.
8002H	The specified file or folder does not exist.	• Check that the specified file or folder exists. • Check that the specified folder path exists.
	The free space in the SD memory card is insufficient.	Delete unnecessary files in the SD memory card to secure free space.
	The specified folder contains a file/folder that is of a different type (file/folder) but has the same name.	Change the file name or folder name.
8003H	The total number of data read from the file has exceeded (d1)+3 (Maximum number of read data).	Adjust (d1)+2 (Number of read-target data) or (d1)+3 (Maximum number of read data).
	The number of elements of written data is smaller than (s3)+0 (number of request write data).	Adjust (s3)+0 (number of request write data) or (s3)+1 (written data) and the followings.
8004H	The folder to be deleted is not empty.	Check the setting of the control data.
8005H	A file or folder with the same name exists.	Change the name of the file or folder.
8006H	• A folder whose folder path will exceed 244 characters after instruction execution exists. • A file whose file path will exceed 253 characters after instruction execution exists.	Ensure that the folder path or file path after instruction execution will not exceed the maximum number of characters.
	The following folder/subfolder is specified for the copy or move destination. • Same folder/subfolder as the copy source • Same folder/subfolder as the move source	Change the copy or move destination path.

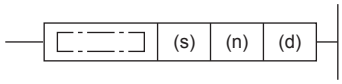
8.8 Extended File Register Operation Instruction

Reading extended file register

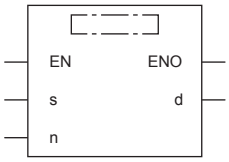
ERREAD

FX5S FX5UJ FX5U FX5UC

Reads the current value of the extended file register (ER) to the file register (R) in the CPU built-in memory.

Ladder diagram	Structured text
	ENO:=ERREAD(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	The device number of file register (R) in the data storage destination (Extended file register (ER) in the data transfer source and (s) is the same number.)	—	Word	ANY16_ARRAY
(n)	Number of read (transfer) points	0 to 32767	16-bit unsigned binary	ANY16_U
(d)	Head device number which turns on when the instruction is completed When the reading is completed abnormally, (d)+1 turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	—	—	—	—	—
(n)	○	○	—	○	—	—	○	○	—	—	—
(d)	○	○*2	—	—	—	—	—	—	—	—	—

*1 Only R can be used.
*2 T, ST, and C cannot be used.

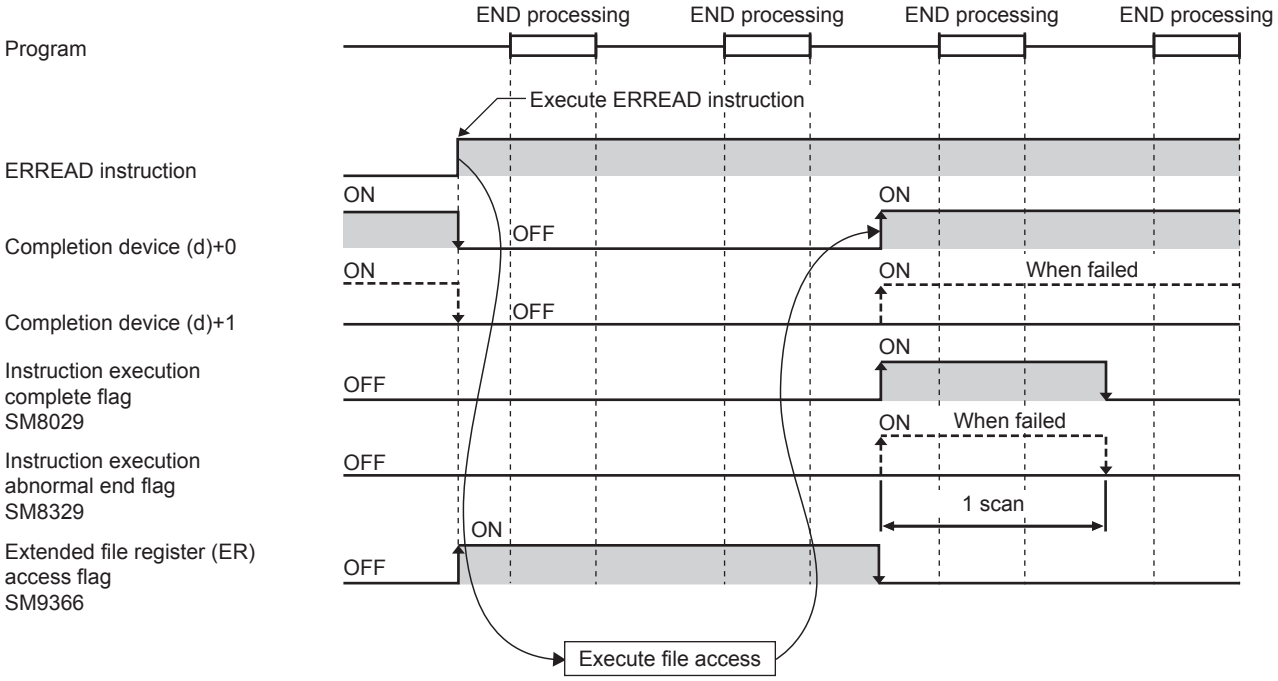
Processing details

The current value of the extended file register (ER) stored into the SD memory card is read for (n) points to the file register (R) in the CPU built-in memory specified with (s).

The normal or abnormal completion of the ERREAD instruction can be confirmed with the instruction completion device (d) specified with the setting data.

- Instruction completion device (d)+0: Turns off when the ERREAD instruction is activated, and turns on during the instruction processing for the scan in which the ERREAD instruction has completed normally.
- Instruction completion device (d)+1: Turns off when the ERREAD instruction is activated, and turns on during the instruction processing for the scan in which the ERREAD instruction has completed abnormally.

Also, when the ERREAD instruction has completed normally/abnormally, instruction execution complete flag (SM8029) and instruction execution abnormal end flag (SM8329) turn on only for one scan at the same timing with the instruction completion device (d). After the contact of the ERREAD instruction turns on, the extended file register (ER) access flag (SM9366) turns on during access to the extended file register.



Point

- When (s)=0 (R0), and (n)=0 (the number of read points: 0) are specified, all the points (32768 points) of the extended file register (ER) are read to the file register (R) in the CPU built-in memory in a batch.
- The following action takes place when (s) = 1 or more (from R1) and (n) = 0 (number of read points: 0) is set.

- FX5S/FX5U/FX5UC CPU module: ER (s) to the range of assigned device points are read out to the file register (R) in the CPU built-in memory.
- FX5UJ CPU module: ER (s) to ER32767 are read out to the file register (R) in the CPU built-in memory.

- Even if the command input of ERREAD instruction is set to OFF while the extended file register is being read, ERREAD instruction is executed until reading is completed.

■Related devices

Device	Name	Description
SM8029	Instruction execution complete	Turns on when an instruction is finished normally.
SM8329	Instruction execution abnormal end	Turns on when an instruction is finished abnormally.
SM9366	Extended file register (ER) access flag	Turns on during access to extended file register (ER).

Point

SM8029 and SM8329 are the flags shared by multiple instructions. Program flag contacts directly under each instruction. For details, refer to ☞ Page 45 Handling general flags.

Precautions

- If the ERREAD instruction is executed, an access to the SD memory card occurs and the scan time is extended.
- This instruction cannot be executed with the extended file register operation instruction simultaneously.
- When executing the following extended file register operation instruction, detect the extended file register (ER) access flag: SM9366 switching from ON to OFF, and turn on the drive contact of the extended file register operation instruction.
- The ERREAD instruction cannot be executed in the interrupt program.
- Do not change the value of the file register (R) during transferring the extended file register (ER).
- Do not turn off the power supply of the CPU module during access to the extended file register (ER).

Operation error

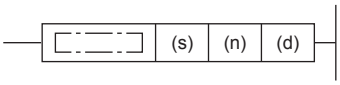
Error code (SD0/SD8067)	Description
2121H	The SD memory card is not formatted.
2820H	The last device number of the device to be transferred (s) exceeds the device points assignment setting of the file register (R).
2821H	The device other than the target device is specified to each operand.
3583H	The extended file register operation instruction is executed in the F/W of the serial number 16X**** or earlier. (FX5U/FX5UC CPU modules only)
3586H	• The SD memory card is not inserted. • The SD memory card is removed or unmounted during transferring the extended file register (ER).
3587H	• The reading to the CPU built-in memory did not complete normally. • The file of extended file register is damaged.
3588H	The file of extended file register (ER) is not in the SD memory card.

Writing extended file register

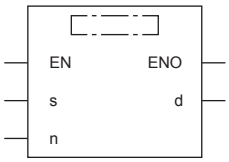
ERWRITE

FX5S FX5UJ FX5U FX5UC

Writes the current value of the file register (R) in the CPU built-in memory to the extended file register (ER).

Ladder diagram	Structured text
	ENO:=ERWRITE(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	The device number of file register (R) in the data storage destination (Extended file register (ER) in the data transfer source and (s) is the same number.)	—	Word	ANY16_ARRAY
(n)	Number of write (transfer) points	0 to 32767	16-bit unsigned binary	ANY16_U
(d)	Head device number which turns on when the instruction is completed When the reading is completed abnormally, (d)+1 turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	—	—	—	—	—
(n)	○	○	—	○	—	—	○	○	—	—	—
(d)	○	○*2	—	—	—	—	—	—	—	—	—

*1 Only R can be used.
*2 T, ST, and C cannot be used.

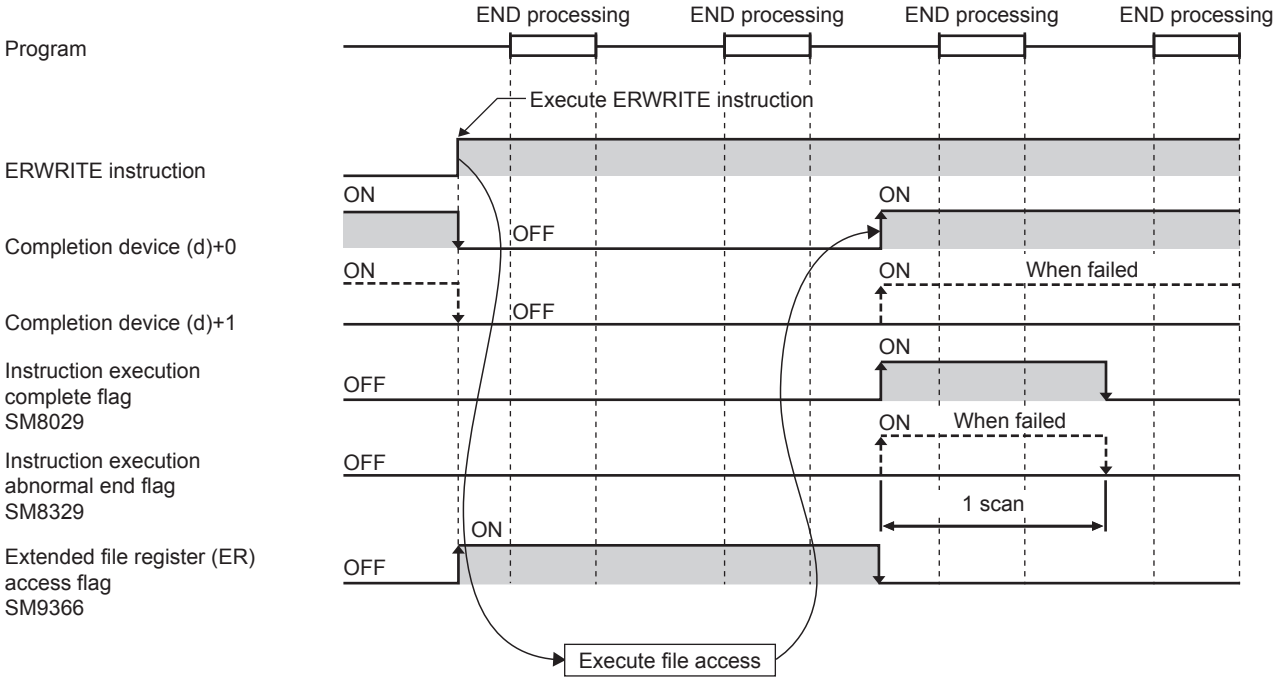
Processing details

The current value of the file register (R) in the CPU built-in memory is written (transferred) for (n) points to the extended file register (ER) in the SD memory card specified with (s).

The normal or abnormal completion of the ERWRITE instruction can be confirmed with the instruction completion device (d) specified with the setting data.

- Instruction completion device (d)+0: Turns off when the ERWRITE instruction is activated, and turns on during the instruction processing for the scan in which the ERWRITE instruction has completed normally.
- Instruction completion device (d)+1: Turns off when the ERWRITE instruction is activated, and turns on during the instruction processing for the scan in which the ERWRITE instruction has completed abnormally.

Also, when the ERWRITE instruction has completed normally/abnormally, instruction execution complete flag (SM8029) and instruction execution abnormal end flag (SM8329) turn on only for one scan at the same timing with the instruction completion device (d). After the contact of the ERWRITE instruction turns on, the extended file register (ER) access flag (SM9366) turns on during access to the extended file register (ER).



Point

- When (s)=0 (R0), and (n)=0 (the number of write points: 0) are specified, all the points (32768 points) of the file register (R) in the CPU built-in memory are written to the extended file register (ER) in a batch.
- The following action takes place when (s) = 1 or more (from R1) and (n) = 0 (number of write points: 0) is set.

- FX5S/FX5U/FX5UC CPU module: R (s) to the range of assigned device points in the CPU built-in memory are written to the extended file register (ER).
- FX5UJ CPU module: R (s) to R32767 in the CPU built-in memory are written to the extended file register (ER).

- Even if the command input of ERWRITE instruction is set to OFF while the extended file register is being written, ERWRITE instruction is executed until writing is completed.

■Related devices

Device	Name	Description
SM8029	Instruction execution complete	Turns on when an instruction is finished normally.
SM8329	Instruction execution abnormal end	Turns on when an instruction is finished abnormally.
SM9366	Extended file register (ER) access flag	Turns on during access to extended file register (ER).

Point

SM8029 and SM8329 are the flags shared by multiple instructions. Program flag contacts directly under each instruction. For details, refer to ☞ Page 45 Handling general flags.

Precautions

- If the ERWRITE instruction is executed, an access to the SD memory card occurs and the scan time is extended.
- This instruction cannot be executed with the extended file register operation instruction simultaneously.
- When executing the following extended file register operation instruction, detect the extended file register (ER) access flag: SM9366 switching from ON to OFF, and turn on the drive contact of the extended file register operation instruction.
- The ERWRITE instruction cannot be executed in the interrupt program.
- Do not change the value of the file register (R) during transferring the extended file register (ER).
- Do not turn off the power supply of the CPU module during access to the extended file register (ER).

Operation error

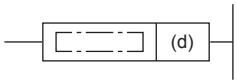
Error code (SD0/SD8067)	Description
2121H	The SD memory card is not formatted.
2820H	The last device number of the device to be transferred (s) exceeds the device points assignment setting of the file register (R).
2821H	The device other than the target device is specified to each operand.
3583H	The extended file register operation instruction is executed in the F/W of the serial number 16X**** or earlier. (FX5U/FX5UC CPU modules only)
3584H	Write protect of the SD memory card is enabled.
3585H	• A storage capacity in the SD memory card for the extended file register "EXFILER.ERD" is insufficient. • A storage capacity in the SD memory card for the copy file "~$EXFILER.ERD" is insufficient.
3586H	• The SD memory card is not inserted. • The SD memory card is removed or unmounted during transferring the extended file register (ER).
3587H	The writing to the SD memory card did not complete normally.

Batch initialization function of extended file register

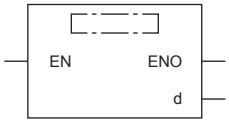
ERINIT

FX5S FX5UJ FX5U FX5UC

Initialize all the points of the extended file register (ER) in a batch.

Ladder diagram	Structured text
	ENO:=ERINIT(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number which turns on when the instruction is completed When the reading is completed abnormally, (d)+1 turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○*1	—	—	—	—	—	—	—	—	—

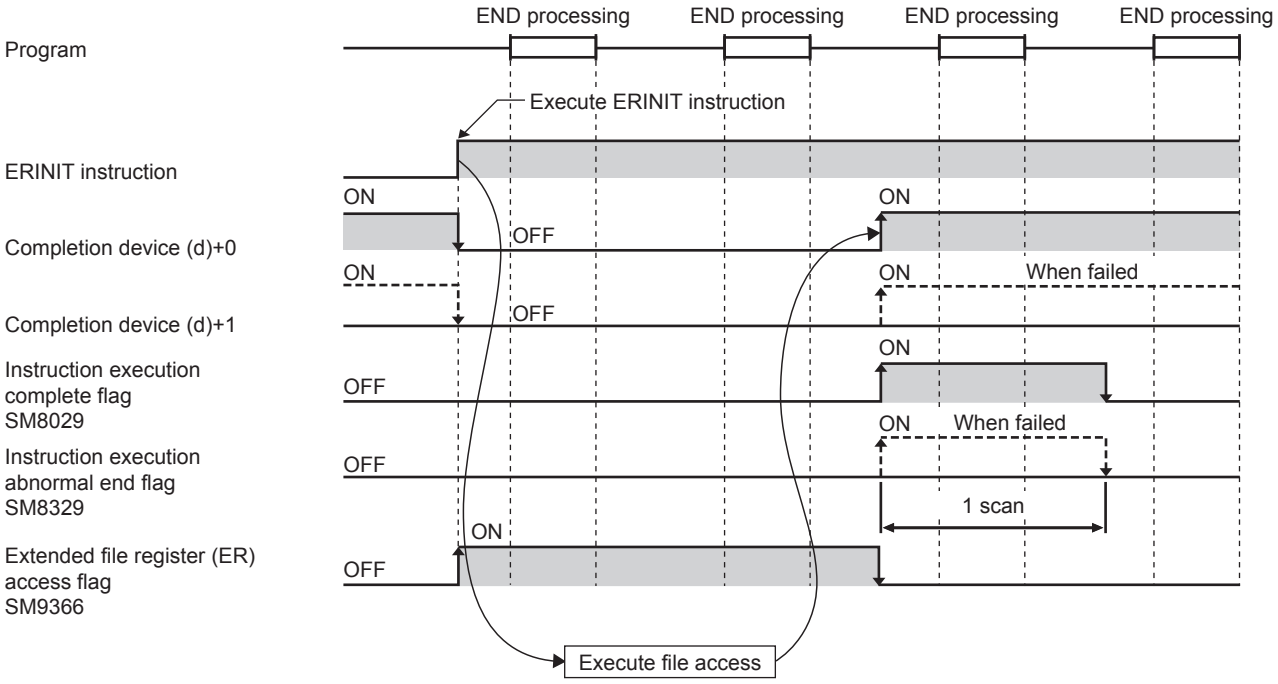
*1 T, ST, and C cannot be used.

Processing details

Initializes all the points of the extended file register (ER) in the SD memory card in a batch.
The normal or abnormal completion of the ERINIT instruction can be confirmed with the instruction completion device (d) specified with the setting data.

- Instruction completion device (d)+0: Turns off when the ERINIT instruction is activated, and turns on during the instruction processing for the scan in which the ERINIT instruction has completed normally.
- Instruction completion device (d)+1: Turns off when the ERINIT instruction is activated, and turns on during the instruction processing for the scan in which the ERINIT instruction has completed abnormally.

Also, when the ERINIT instruction has completed normally/abnormally, instruction execution complete flag (SM8029) and instruction execution abnormal end flag (SM8329) turn on only for one scan at the same timing with the instruction completion device (d). After the contact of the ERINIT instruction turns on, the extended file register (ER) access flag (SM9366) turns on during access to the extended file register (ER).



Point

- The initial value of the extended file register (ER) when initializing is 65535 (FFFFH).
- Even if the command input of ERINIT instruction is set to OFF while the extended file register is initialized in a batch, ERINIT instruction is executed until initialization is completed.

■Related devices

Device	Name	Description
SM8029	Instruction execution complete	Turns on when an instruction is finished normally.
SM8329	Instruction execution abnormal end	Turns on when an instruction is finished abnormally.
SM9366	Extended file register (ER) access flag	Turns on during access to extended file register (ER).

Point

SM8029 and SM8329 are the flags shared by multiple instructions. Program flag contacts directly under each instruction. For details, refer to ☞ Page 45 Handling general flags.

Precautions

- If the ERINIT instruction is executed, an access to the SD memory card occurs and the scan time is extended.
- This instruction cannot be executed with the extended file register operation instruction simultaneously.
- When executing the following extended file register operation instruction, detect the extended file register (ER) access flag: SM9366 switching from ON to OFF, and turn on the drive contact of the extended file register operation instruction.
- The ERINIT instruction cannot be executed in the interrupt program.
- Do not turn off the power supply of the CPU module during access to the extended file register (ER).

Operation error

Error code (SD0/SD8067)	Description
2821H	The device other than the target device is specified to each operand.
3583H	The extended file register operation instruction is executed in the F/W of the serial number 16X**** or earlier. (FX5U/FX5UC CPU modules only)
3584H	Write protect of the SD memory card is enabled.
3585H	A storage capacity in the SD memory card for the extended file register "EXFILER.ERD" is insufficient.
3586H	• The SD memory card is not inserted. • The SD memory card is removed or unmounted during transferring the extended file register (ER).
3587H	Initialization to the SD memory card did not complete normally.

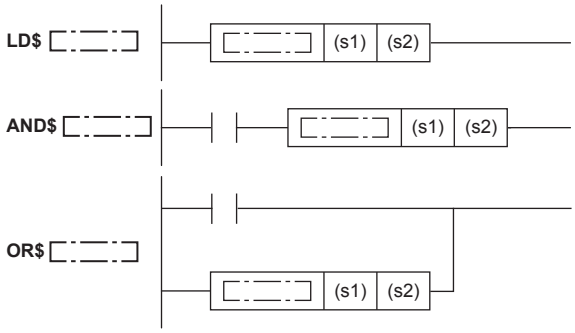
8.9 Character String Operation Instruction

Comparing character strings

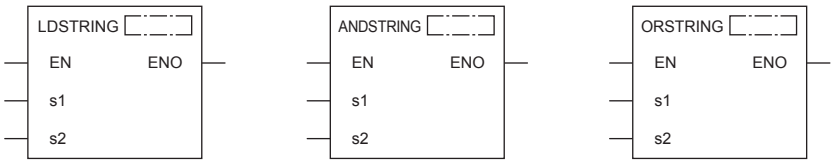
LD$□, AND$□, OR$□

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation between the character string data in the device specified by (s1) and later and the character string data in the device specified by (s2) and later. (Devices are used as a normally open contact.)

Ladder diagram	Structured text*1
 ("=", "<>", ">", "<=", "<", ">=" enters □.)	ENO:=LDSTRING_□(EN,s1,s2); ENO:=ANDSTRING_□(EN,s1,s2); ENO:=ORSTRING_□(EN,s1,s2); ("EQ", "NE", "GT", "LE", "LT", "GE" enters □.)*2

FBD/LD



("_EQ", "_NE", "_GT", "_LE", "_LT", "_GE" enters □.)*2

*1 Supported by engineering tool version "1.035M" and later.
*2 EQ is =, NE is <>, GT is >, LE is <=, LT is <, and GE is >=.

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Comparison data or head device number where the comparison data is stored	—	Character string	ANYSTRING_SINGLE
(s2)	Comparison data or head device number where the comparison data is stored	—	Character string	ANYSTRING_SINGLE
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

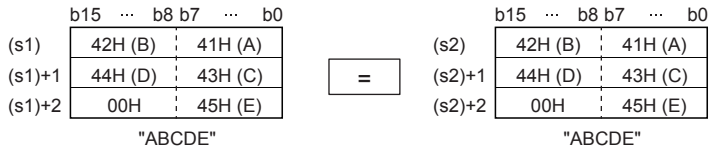
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	—	—	—	—	○	—	—	○	—
(s2)	—	○*1	—	—	—	—	○	—	—	○	—

*1 T, ST, and C cannot be used.

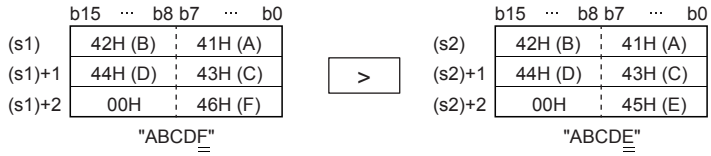
Processing details

- These instructions perform a comparison operation between the character string data specified by (s1) and the character string data specified by (s2). (Devices are used as a normally open contact.)
- In the comparison operation, the ASCII codes of the character strings are compared one by one from the start of the strings.
- Character strings in the devices specified by (s1) and (s2) to a device that stores 00H are compared.
  - When all the character strings match, the comparison is considered as matched.



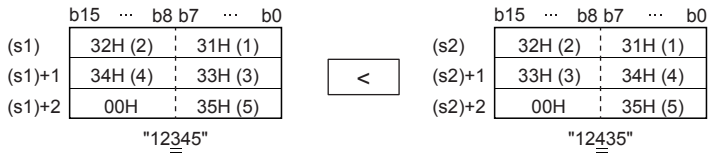
Instruction symbol in ☐	Result
$=	Conductive state
$<>	Non-conductive state
$>	Non-conductive state
$<=	Conductive state
$<	Non-conductive state
$>=	Conductive state

  - When the character strings are different, the string with a large character code is considered as the large one.



Instruction symbol in ☐	Result
$=	Non-conductive state
$<>	Conductive state
$>	Conductive state
$<=	Non-conductive state
$<	Non-conductive state
$>=	Conductive state

  - When the character strings are different, the magnitude relation between them is determined based on the size of the first different character code.



Instruction symbol in ☐	Result
$=	Non-conductive state
$<>	Conductive state
$>	Non-conductive state
$<=	Conductive state
$<	Conductive state
$>=	Non-conductive state

• When the length of the character string data differs for (s1) and (s2), the relative sizes of the character strings are determined based on the size of the first different character code.

	b15 ··· b8	b7 ··· b0
(s1)	32H (2)	31H (1)
(s1)+1	33H (3)	34H (4)
(s1)+2	36H (6)	35H (5)
(s1)+3	00H	00H

"124356"

>

	b15 ··· b8	b7 ··· b0
(s2)	32H (2)	31H (1)
(s2)+1	34H (4)	33H (3)
(s2)+2	36H (6)	35H (5)
(s2)+3	00H	37H (7)

"1234567"

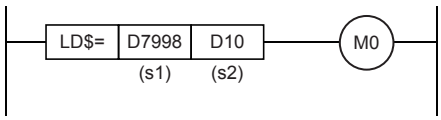
Instruction symbol in ☐	Result
$=	Non-conductive state
$<>	Conductive state
$>	Conductive state
$<=	Non-conductive state
$<	Non-conductive state
$>=	Conductive state

• If the character string specified by (s1) or (s2) has more than 16383 characters, the operation result is the non-conductive state.

Precautions

• In character string comparison operation, if the target device range does not have "00H", the values until the last number of the device are retrieved. Thus, even if the target device range does not have "00H", a comparison operation result is output when a mismatch between the acquired character strings is detected.

Ex.



Data of (s1)

D7998	42H(B)	41H(A)
D7999	44H(D)	43H(C)

Data of (s2)

D10	5A(Z)	41H(A)
D11	00H	43H(C)

• For the data specified by (s1) and (s2) as shown above, the second character is different between them. Thus, the operation result is non-conductive.

Operation error

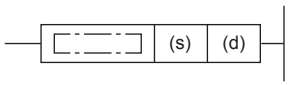
There is no operation error.

Concatenating character strings

$+(P) [For 2 operands]

FX5S FX5UJ FX5U FX5UC

These instructions concatenate the character string data stored in the device specified by (s) and later to the end of the character string data stored in the device specified by (d) and later, and store the concatenated string in the device specified by (d) and later.

Ladder diagram	Structured text
	Not supported (☞ Page 594 $+(P) [For 3 operands])

FBD/LD
Not supported. (☞ Page 594 $+(P) [For 3 operands])

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number storing data to be concatenated or data, or directly specified character string	—	Character string	ANYSTRING_SINGLE
(d)	Head device number storing data to which another data is concatenated	—	Character string	ANYSTRING_SINGLE

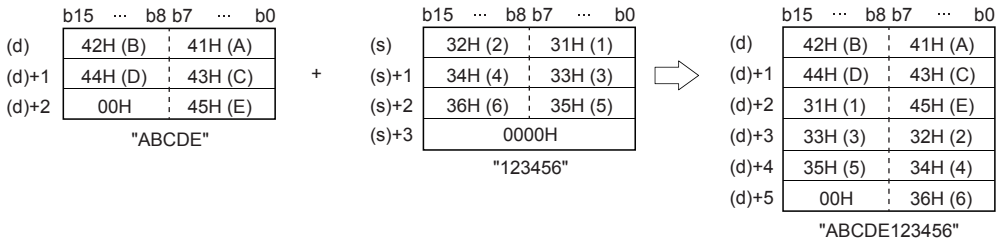
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions concatenate the character string data stored in the device specified by (s) and later to the end of the character string data stored in the device specified by (d) and later, and store the concatenated string in the device specified by (d) and later.



- Character strings in the devices specified by (s) and (d) up to a device that stores 00H are concatenated.
- When character strings are concatenated, 00H indicating an end of the character string specified by (d) is ignored and the character string specified by (s) is concatenated to the last character of (d).

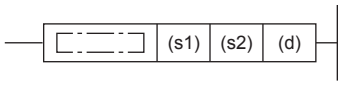
Operation error

Error code (SD0/SD8067)	Description
2820H	In the corresponding device range after the device specified by (s), "00H" does not exist.
	In the corresponding device range after the device specified by (d), "00H" does not exist.
3406H	The whole concatenated character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.
	The number of characters of the character string in the device specified by (s)+(d) exceeds 16383.
3405H	The character string specified by (s) has more than 16383 characters.
	The character string specified by (d) has more than 16383 characters.

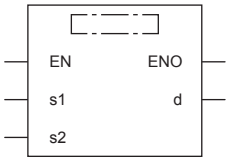
$+(P) [For 3 operands]

FX5S FX5UJ FX5U FX5UC

These instructions concatenate the character string data stored in the device specified by (s2) and later to the end of the character string data stored in the device specified by (s1) and later, and store the concatenated string in the device specified by (d) and later.

Ladder diagram	Structured text
	ENO:=STRINGPLUS(EN,s1,s2,d); ENO:=STRINGPLUSP(EN,s1,s2,d);

FBD/LD



("STRINGPLUS", "STRINGPLUSP" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device number storing data to which another data is concatenated or data, or directly specified character string	—	Character string	ANYSTRING_SINGLE
(s2)	Head device number storing data to be concatenated or data, or directly specified character string	—	Character string	ANYSTRING_SINGLE
(d)	Head device number for storing the concatenated data	—	Character string	ANYSTRING_SINGLE
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

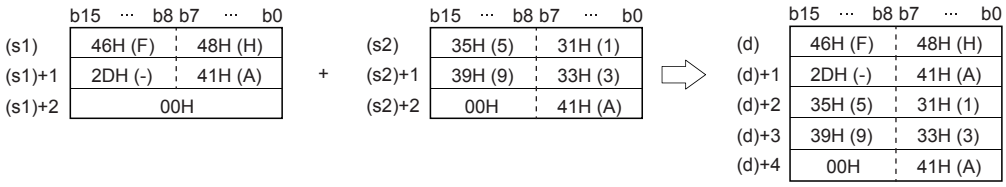
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	—	—	—	—	○	—	—	○	—
(s2)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions concatenate the character string data stored in the device specified by (s2) and later to the end of the character string data stored in the device specified by (s1) and later, and store the concatenated string in the device specified by (d) and later.
- Character strings in the devices specified by (s1) and (s2) up to a device that stores 00H are concatenated.



- When character strings are concatenated, 00H indicating an end of the character string specified by (s1) is ignored and the character string specified by (s2) is concatenated to the last character of (s1).
- After two character strings are connected, "00H" is automatically added at the end. When the number of characters after the concatenation is odd, 00H is stored in the upper byte of the device storing the last character. When the number is even, 0000H is stored in the device after the last character.

Precautions

- For direct specification, up to 255 characters can be specified (input). When word devices are specified in (s1) or (s2), this restriction (up to 32 characters) is not applicable.
- When the values in both (s1) and (s2) start from "00H" (that is, when the number of characters is "0"), "0000H" is stored in (d).

Operation error

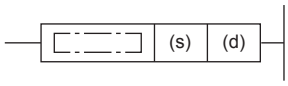
Error code (SD0/SD8067)	Description
2820H	In the corresponding device range after the device specified by (s1), "00H" does not exist.
	In the corresponding device range after the device specified by (s2), "00H" does not exist.
2821H	The numbers of the character string-storing devices specified by (s1), (s2), and (d) overlap.
3405H	The character string specified by (s1) has more than 16383 characters.
	The character string specified by (s2) has more than 16383 characters.
3406H	The character string specified by (d) has more than 16383 characters.
	The whole concatenated character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.

Transferring character strings

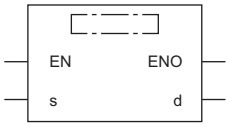
$MOV(P)

FX5S FX5UJ FX5U FX5UC

These instructions transfer the character string data specified by (s) to the device specified by (d) and later.

Ladder diagram	Structured text*1
	ENO:=STRINGMOV(EN,s,d); ENO:=STRINGMOVP(EN,s,d);

FBD/LD



("STRINGMOV", "STRINGMOVP" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Character string to be transferred (up to 255 characters) or head device number storing a character string	—	Character string	ANYSTRING_SINGLE
(d)	Head device number storing transferred character string	—	Character string	ANYSTRING_SINGLE
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

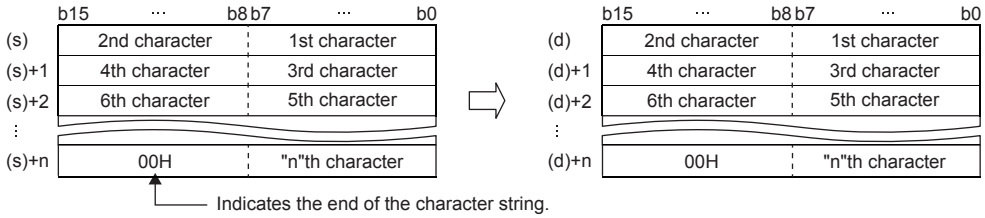
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—

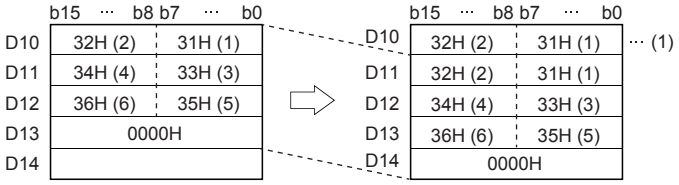
*1 T, ST, and C cannot be used.

Processing details

- These instructions transfer the character string data specified by (s) to the device specified by (d) and later. A character string enclosed with double quotation marks and specified by (s) or stored in the devices from the device specified by (s) to the device storing 00H is transferred in a batch.

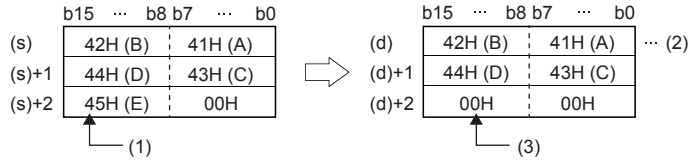


- Even though the device range of the data to be transferred (s) to (s)+n and the device range for storing the transferred data (d) to (d)+n overlap, the processing is performed normally. For example, when a character string stored in D10 to D13 is transferred to D11 to D14, the transfer is executed as shown below:



(1): It is the same as the character string before transfer.

- When "00H" is stored in the lower byte of (s)+n, "00H" is stored to both the upper byte and lower byte of (d)+n.



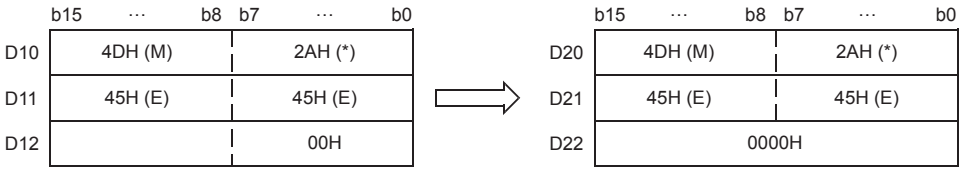
(1): The high-order byte is not transferred.

(2): It is the same as the character string before transfer.

(3): "00H" is automatically stored in the high-order byte.

Program example

In the program example shown below, character string data stored in D10 to D12 is transferred to D20 through D22 when X0 is set to ON.



Operation error

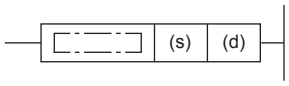
Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s) and later, "00H" does not exist.
3405H	The character string specified by (s) has more than 16383 characters.
3406H	The whole specified character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.

Transferring Unicode string data

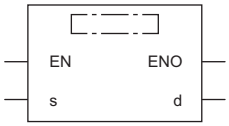
$MOV(P)_WS

FX5S FX5UJ FX5U FX5UC

These instructions transfer Unicode string data to the specified device number and later.

Ladder	ST
	ENO:=STRINGMOV_WS(EN,s,d); ENO:=STRINGMOVP_WS(EN,s,d);

FBD/LD



(☐ is replaced by STRINGMOV_WS or STRINGMOVP_WS.)

Setting data

■Descriptions, ranges, and data types

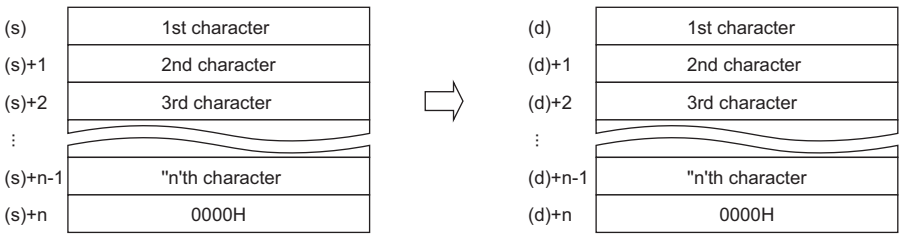
Operand	Description	Range	Data type	Data type (label)
(s)	Unicode character string to be transferred (maximum of 255 characters) or the start device containing the Unicode character string	—	Unicode string	ANYSTRING_DOUBLE
(d)	Start device for storing the transferred Unicode character string	—	Unicode string	ANYSTRING_DOUBLE
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

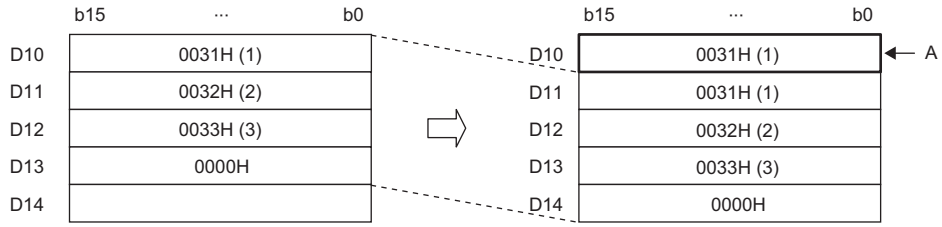
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U☐\G☐	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	○	—	○	—	—	○	—
(d)	—	○	—	—	○	—	○	—	—	—	—

Processing details

- These instructions transfer the Unicode character string data in the device specified by (s) to the device number specified by (d) and later. The Unicode character strings specified by (s) or the Unicode character strings from the device number specified by (s) to the device number containing 0000H are transferred all at once.



- Even when the device range ((s) to (s)+n)) in which the Unicode character string data to be transferred and the device range ((d) to (d)+n) for storing the transferred data are overlapping, the processing is performed normally. For example, the character strings stored in the devices specified by D10 to D13 are transferred to the devices specified by D11 to D14 as shown below.



A: Same as before transfer

Precautions

- The character code of the Unicode string data specified in (s) is not judged. Even if a string other than the Unicode string has been specified in (s), the word data to up to 0000H are transferred to (d).
- When the data length of the Unicode string specified in (s) is 0, only the one character (one word) of termination NULL is written to (d)+0.

Operation error

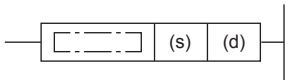
Error code (SD0/SD8067)	Description
2820H	There is no 0000H in the setting area specified by (s) and later in the device/label memory.
3405H	The number of characters in the Unicode string specified by (s) exceeds 512.
3406H	The entire Unicode string cannot be stored in the setting area specified by (d) in the device/label memory. (The number of required points is insufficient.)

Converting 16-bit binary data to decimal ASCII

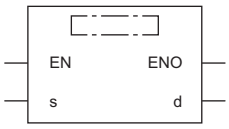
BINDA(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert 16-bit binary data specified by (s) into decimal ASCII codes, and store the converted data in the device specified by (d) and later.

Ladder diagram	Structured text	
	ENO:=BINDA(EN,s,d); ENO:=BINDAP(EN,s,d)	ENO:=BINDA_U(EN,s,d); ENO:=BINDAP_U(EN,s,d)

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)	BINDA(P)	Binary data to be converted into ASCII codes	-32768 to +32767	16-bit signed binary	ANY16_S
	BINDA(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)		Head device number storing conversion result	—	Character string	ANYSTRING_SINGLE
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions convert 16-bit binary data specified by (s) into decimal ASCII codes, and store the converted data in the device specified by (d) and later.
- The format of the decimal ASCII data to be stored in (d) depends on the status of SM705 (Number of conversion digits selection).

Status of SM705*1	Storage format of (d)	Reference
OFF	Data is stored in a fixed number of digits (a sign + 5 digits).	Page 602 Operation of when SM705 (Number of conversion digits selection) is off
ON	Each digit is stored left-justified depending on the value of (s).	Page 602 Operation of when SM705 (Number of conversion digits selection) is on

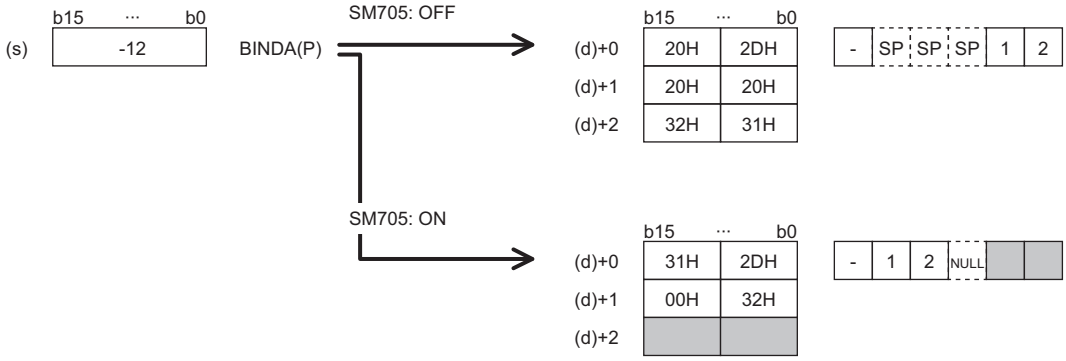
*1 For the firmware version of the CPU module supporting SM705, refer to the following.
MELSEC iQ-F FX5 User's Manual (Application)
A CPU module which does not support SM705 operates in the same way as SM705 is off even if it is turned on.

■Operation overview

The following figure shows the operations when SM705 (Number of conversion digits selection) is off and on.

Ex.

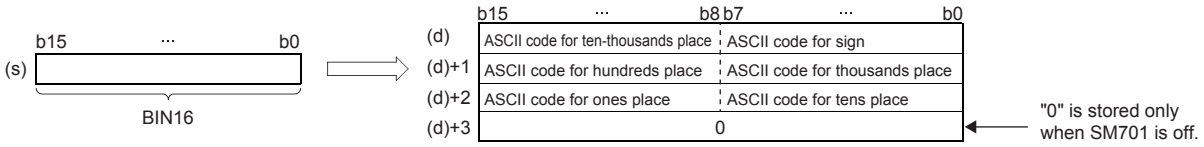
When the BINDA(P) instruction is executed with the numeric value "-12" stored in (s)



- When SM705 is off, the number of digits is fixed. The first character is a sign and it is 2DH(-) in the above example. (If (s) is 0 or positive, the first character is 20H (space).) The numeric part is right-justified. When the length of the numeric part is less than 5 digits, the ASCII code 20H (space) is stored for the ASCII code of the upper digit(s).
- When SM705 is on, data is left-justified. When the length of the numeric part is less than 5 digits, 00H is stored in the end.

■Operation of when SM705 (Number of conversion digits selection) is off

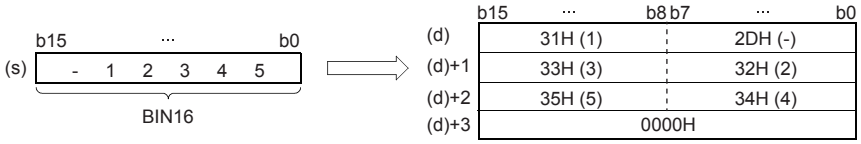
Decimal ASCII data is stored in a fixed number of digits in (d) to (d)+2.



- The following shows the operation result to be stored in (d).
  - As sign data, "20H" is stored if the binary data is positive, and "2DH" is stored if the data is negative.
  - "20H" is stored for "0" on the left side of the valid digits (zero suppression). For "00325", 20H is stored for "00", and the number of digits is 3 based on "325".
  - In the device specified by (d)+3, 0 is stored when SM701 (Output characters selection) is off, and the original data remains when SM701 is on.

Ex.

When -12345 is specified in (s) (when signed data is specified)



■Operation of when SM705 (Number of conversion digits selection) is on

Decimal ASCII data is stored right-justified in (d).

The following figures show an example of a value of (s) and a value stored in (d).

Value of (s)	Data of (d) to (d)+2	Value of (s)	Data of (d) to (d)+2
• 0 • Positive value (1 digit in numeric part)	• The upper byte of (d) is filled with 00H. • Data in (d)+1 and (d)+2 remains unchanged. b15 ··· b8 b7 ··· b0 (d) 00H | ASCII 10^0 (d)+1 (d)+2	• Positive value (2 digits in numeric part) • Negative value (1 digit in numeric part)	• (d)+1 is filled with 00H. • Data in (d)+2 remains unchanged. b15 ··· b8 b7 ··· b0 (d) ASCII 10^0 | ASCII 10^1 / 2DH (-) (d)+1 00H (d)+2
⋮			
• Positive value (5 digits in numeric part) • Negative value (4 digits in numeric part)	• The upper byte of (d)+2 is filled with 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	• Negative value (5 digits in numeric part)	• (1): (d)+3 is filled with 00H only when SM701 (Output characters selection) is off. 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 (1)

ASCII 10^4: ASCII code for the ten-thousands place
ASCII 10^3: ASCII code for the thousands place
ASCII 10^2: ASCII code for the hundreds place
ASCII 10^1: ASCII code for the tens place
ASCII 10^0: ASCII code for ones place

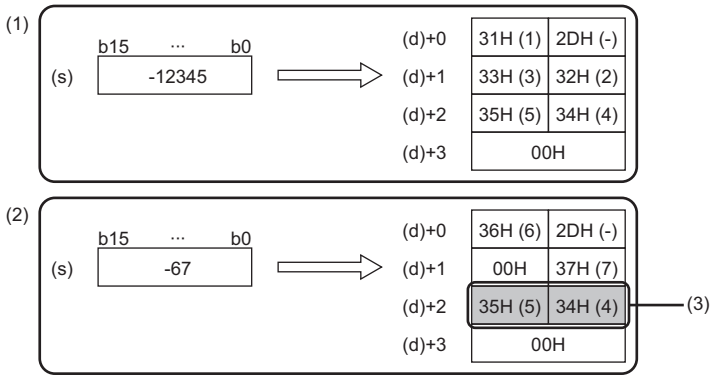
- When the number of operation digits is less than the maximum number of digits (sign + 5 digits), 00H is stored in the end of the string regardless of the status (on/off) of SM701 (Output characters selection). If the end of the string is the lower byte, 00H is also stored in the upper byte.
- When the number of operation digits is equal to the maximum number of digits (a sign + 5 digits), 00H is stored in (d)+3 when SM701 is off. (d)+3 remains unchanged if SM701 is on.

Precautions

- The number of occupied points of (d) is 3 when SM701 is on, and 4 when SM701 is off.
- When SM705 (Number of conversion digits selection) is on, the operation result is stored in (d) for the effective number of digits. Therefore, when the BINDA(P)(_U) instruction is executed successively and the operation result for each execution is stored in the same device, a part of the previous operation result may not be overwritten by the succeeding result and can remain in (d).

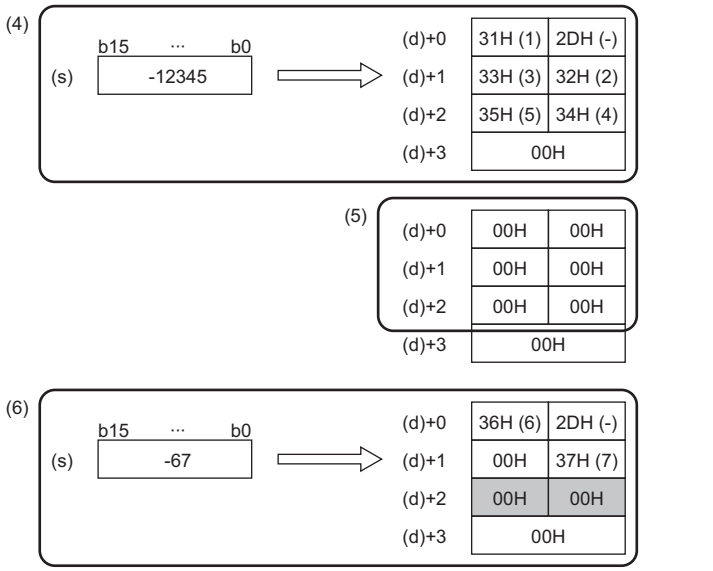
Ex.

Executing the BINDA(P) instruction when (s) is "-12345" and then executing another BINDA(P) instruction when (s) is "-67"



(1) "-12345" is converted into a string.
(2) "-67" is converted into a string.
(3) A part of the previous conversion result remains in (d)+2.

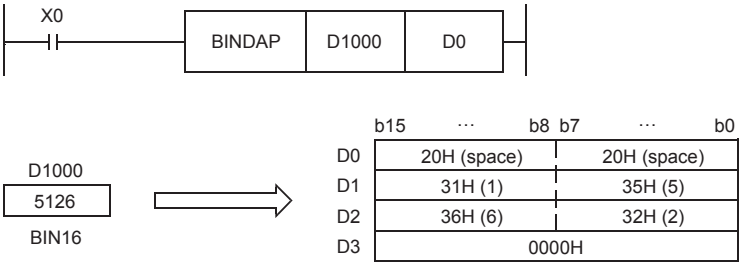
To avoid this, create a program to clear the entire data storage areas (d)+0 to (d)+2 before executing the BINDA(P)(_U) instruction.



(4) "-12345" is converted into a string.
(5) (d)+0 to (d)+2 are cleared.
(6) "-67" is converted into a string.

Program example

In the program below, the value of 16-bit binary data D1000 is converted into decimal ASCII codes when X0 is set to ON.



Operation error

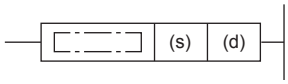
Error code (SD0/SD8067)	Description
2820H	The device specified by (d) exceeds the corresponding device range.

Converting 32-bit binary data to decimal ASCII

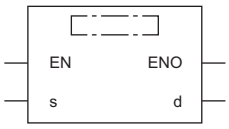
DBINDA(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions convert 32-bit binary data specified by (s) into decimal ASCII codes, and store the converted data in the device specified by (d) and later.

Ladder diagram	Structured text	
	ENO:=DBINDA(EN,s,d); ENO:=DBINDAP(EN,s,d);	ENO:=DBINDA_U(EN,s,d); ENO:=DBINDAP_U(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)	DBINDA(P)	Binary data to be converted into ASCII codes	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DBINDA(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)		Head device number storing conversion result	—	Character string	ANYSTRING_SINGLE
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions convert 32-bit binary data specified by (s) into decimal ASCII codes, and store the converted data in the device specified by (d) and later.
- The format of the decimal ASCII data to be stored in (d) depends on the status of SM705 (Number of conversion digits selection).

Status of SM705*1	Storage format of (d)	Reference
OFF	Data is stored in a fixed number of digits (a sign + 10 digits).	Page 607 Operation of when SM705 (Number of conversion digits selection) is off
ON	Each digit is stored left-justified depending on the value of (s).	Page 608 Operation of when SM705 (Number of conversion digits selection) is on

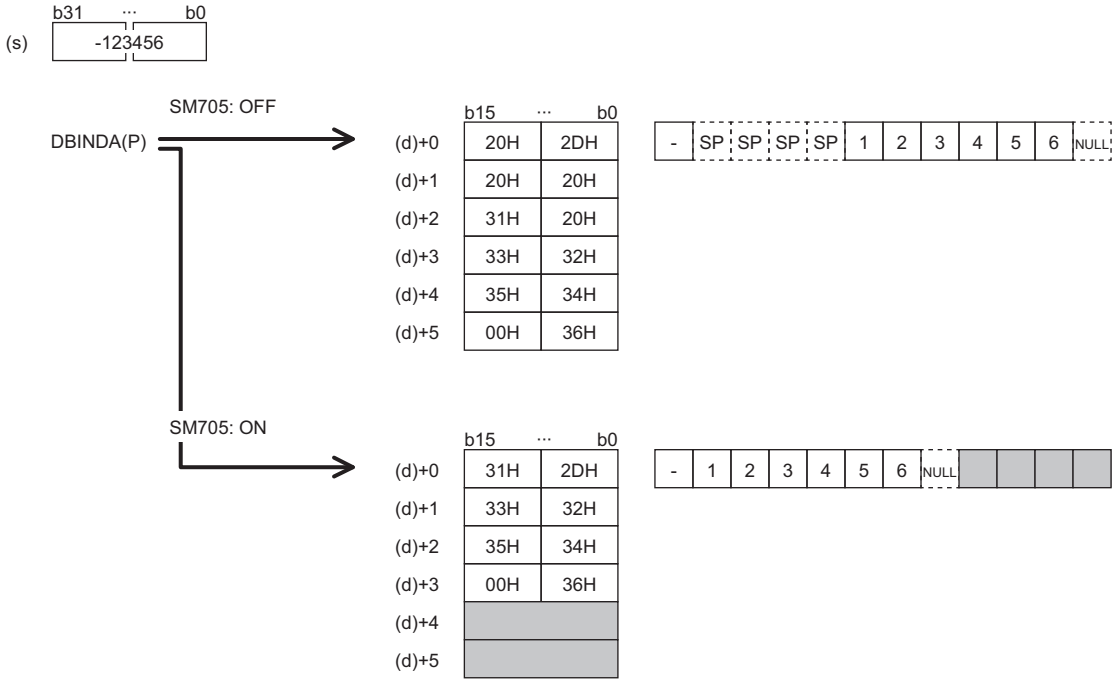
*1 For the firmware version of the CPU module supporting SM705, refer to the following.
MELSEC iQ-F FX5 User's Manual (Application)
A CPU module which does not support SM705 operates in the same way as SM705 is off even if it is turned on.

■Operation overview

The following figure shows the operation when SM705 (Number of conversion digits selection) is off and on.

Ex.

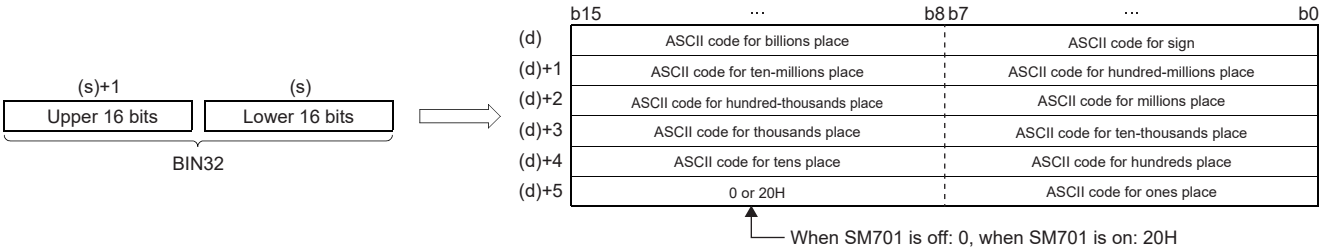
When the DBINDA(P) instruction is executed with a numeric value "-123456" stored in (s)



- When SM705 is off, the number of digits is fixed. The first character is a sign and it is 2DH(-) in the above example. (If (s) is 0 or positive, the first character is 20H (space).) The numeric part is right-justified. When the length of the numeric part is less than 10 digits, the ASCII code 20H (space) is stored for the ASCII code of the upper digit(s).
- When SM705 is on, data is left-justified. When the length of the numeric part is less than 10 digits, 00H is stored in the end.

■Operation of when SM705 (Number of conversion digits selection) is off

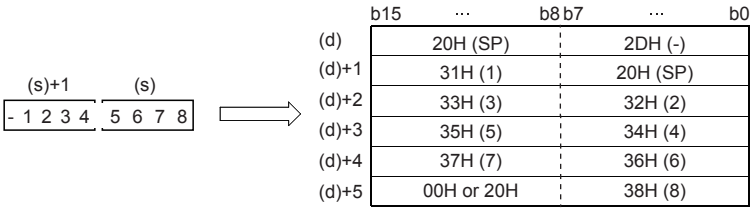
Decimal ASCII data is stored in a fixed number of digits in (d) to (d)+5.



- The following shows the operation result to be stored in (d).
  - As sign data, "20H" is stored if the binary data is positive, and "2DH" is stored if the data is negative.
  - "20H" is stored for "0" on the left side of the valid digits (zero suppression). For "0012034560", 20H is stored for "00", and the number of digits is 8 based on "12034560".
  - In the upper 8 bits of the device specified by (d)+5, 0 is stored when SM701 (Output characters selection) is off, and 20H is stored when SM701 is on.

Ex.

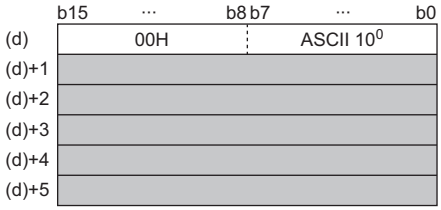
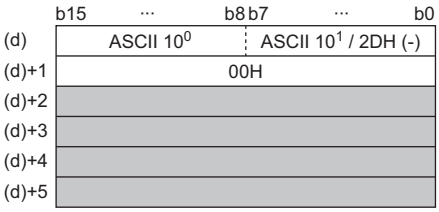
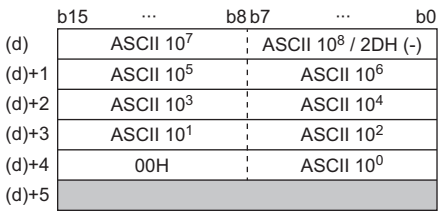
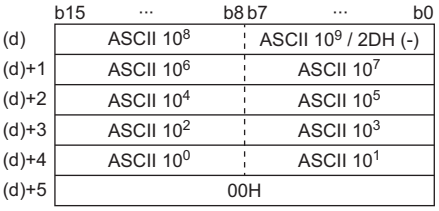
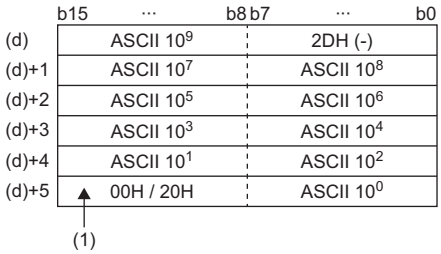
When -12345678 is specified in (s) (when signed data is specified)



■Operation of when SM705 (Number of conversion digits selection) is on

Decimal ASCII data is stored right-justified in (d).

The following figures show an example of a value of (s) and a value stored in (d).

<table>
<tr><th>Value of (s)</th><th>Data of (d) to (d)+5</th><th>Value of (s)</th><th>Data of (d) to (d)+5</th></tr>
<tr><td>• 0
• Positive value (1 digit in numeric part)</td><td>• The upper byte of (d) is filled with 00H.
• (d)+1 and later remain unchanged.
 
</td><td>• Positive value (2 digits in numeric part)
• Negative value (1 digit in numeric part)</td><td>• (d)+1 is filled with 00H.
• (d)+2 and later remain unchanged.
 
</td></tr>
<tr><td colspan="4">⋮</td></tr>
<tr><td>• Positive value (9 digits in numeric part)
• Negative value (8 digits in numeric part)</td><td>• The upper byte of (d)+4 is filled with 00H.
• (d)+5 and later remain unchanged.
 
</td><td>• Positive value (10 digits in numeric part)
• Negative value (9 digits in numeric part)</td><td>• (d)+5 is filled with 00H.
 
</td></tr>
<tr><td>• Negative value (10 digits in numeric part)</td><td>• 00H is stored in the upper byte (1) of (d)+5 when SM701 (Output characters selection) is off, and 20H (space) is stored when it is on.
 
</td><td colspan="2">ASCII 10^9: ASCII code for the billions place
ASCII 10^8: ASCII code for the hundred-millions place
⋮
ASCII 10^1: ASCII code for the tens place
ASCII 10^0: ASCII code for the ones place</td></tr>
</table>

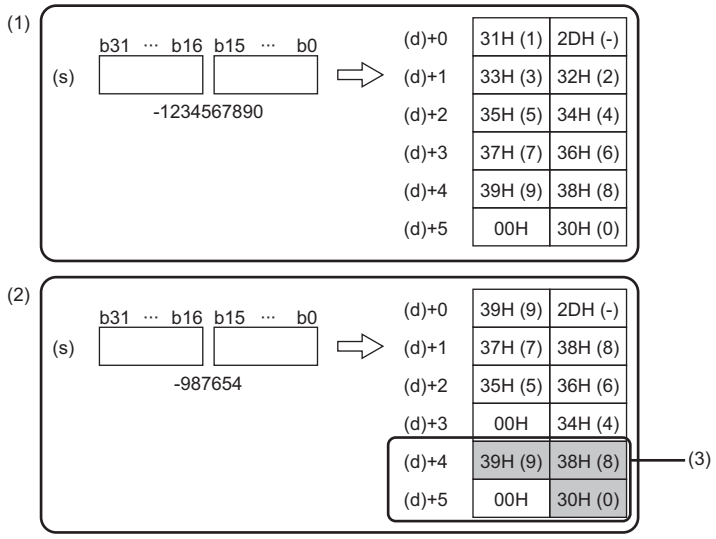
- When the number of operation digits is less than the maximum number of digits (a sign + 10 digits), 00H is stored in the end of the string regardless of the status (on/off) of SM701 (Output characters selection). If the end of the string is the lower byte, 00H is also stored in the upper byte.
- When the number of operation digits is equal to the maximum number of digits (a sign + 10 digits), 00H is stored in the upper byte of (d)+5 if SM701 is off. 20H (space) is stored in the upper byte of (d)+5 if SM701 is on.

Precautions

- (d) occupies six points.
- When SM705 (Number of conversion digits selection) is on, the operation result is stored in (d) for the effective number of digits. Therefore, when the DBINDA(P)(_U) instruction is executed successively and the operation result for each execution is stored in the same device, a part of the previous operation result may not be overwritten by the succeeding result and can remain in (d).

Ex.

Executing the DBINDA(P) instruction when (s) is "-1234567890" and then executing another DBINDA(P) instruction when (s) is "-987654"

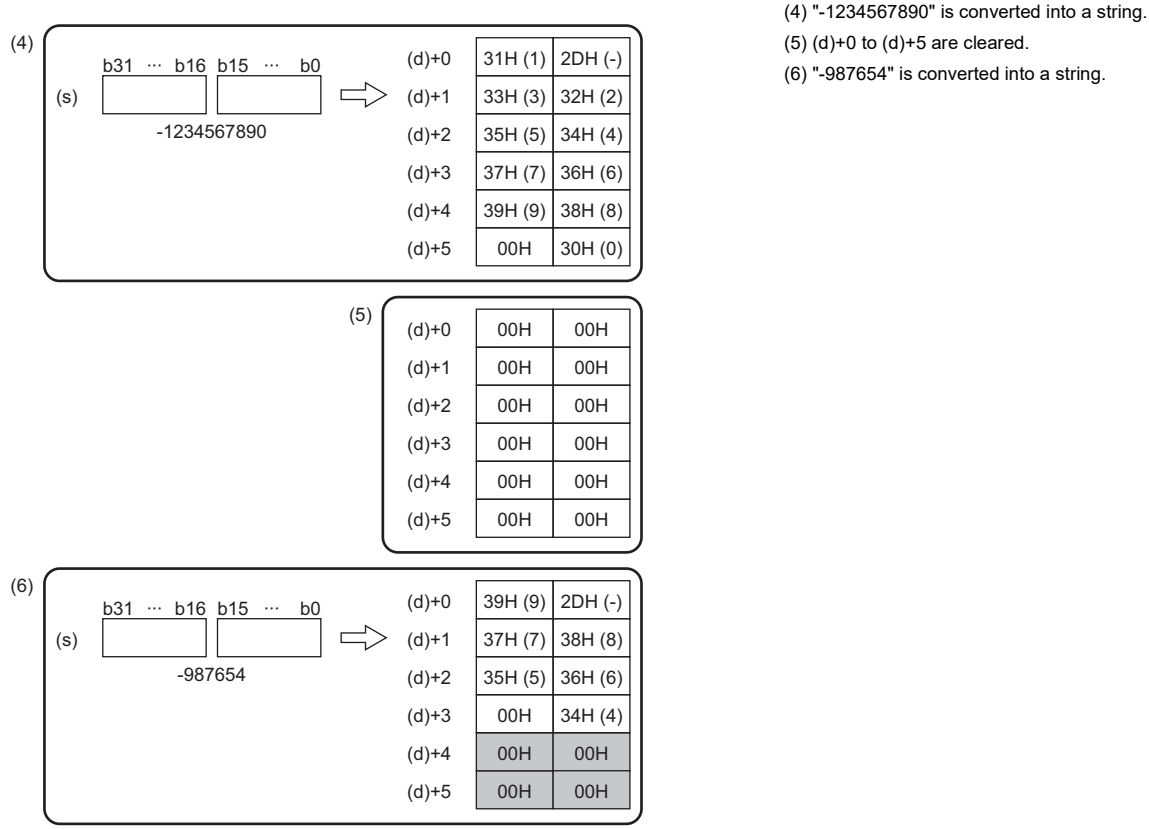


(1) "-1234567890" is converted into a string.

(2) "-987654" is converted into a string.

(3) A part of the previous conversion result remains in (d)+4 and (d)+5.

To avoid this, create a program to clear the entire data storage areas (d)+0 to (d)+5 before executing the DBINDA(P)(_U) instruction.



(4) "-1234567890" is converted into a string.

(5) (d)+0 to (d)+5 are cleared.

(6) "-987654" is converted into a string.

Operation error

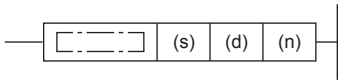
Error code (SD0/SD8067)	Description
2820H	The device specified by (d) exceeds the corresponding device range.

Converting HEX code data to ASCII

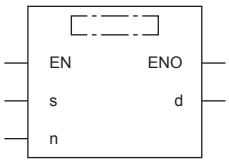
ASCI(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the (n) characters (digits) within the hexadecimal code data specified by (s) to ASCII, and store the converted data in the device specified by (d) and later.

Ladder diagram	Structured text
	ENO:=ASCI(EN,s,n,d); ENO:=ASCIP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number storing hexadecimal code to be converted	—	16-bit signed binary	ANY16
(d)	Head device number storing converted ASCII code	—	Character string	ANYSTRING_SINGLE
(n)	Number of characters (digits) of hexadecimal code to be converted	1 to 32767	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

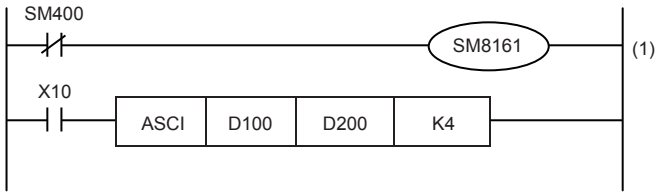
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○*1	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions convert the (n) characters (digits) within the hexadecimal code data specified by (s) to ASCII, and store the converted data in the device specified by (d) and later.
- The 16-bit mode and 8-bit mode options are available for the ASCI(P) instructions. For the operation in each mode, refer to the proceeding pages.
  - 16-bit conversion mode (while SM8161 is OFF)

    Each digit of hexadecimal data stored in the device specified by (s) and later is converted into ASCII code, and transferred to the upper 8 bits and lower 8 bits of each device specified by (d) and later. SM8161 must always be off in the 16-bit conversion mode.
    In the following program, conversion is executed as follows:



Devices specified by (s) and later
(D100)=0ABCH
(D101)=1234H
(D102)=5678H

(1): 16-bit mode

■Number of specified digits (characters) and conversion result

(n) / (d)	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200 lowest-order byte	"C"	"B"	"A"	"0"	"4"	"3"	"2"	"1"	"8"
D200 highest-order byte		"C"	"B"	"A"	"0"	"4"	"3"	"2"	"1"
D201 lowest-order byte			"C"	"B"	"A"	"0"	"4"	"3"	"2"
D201 highest-order byte				"C"	"B"	"A"	"0"	"4"	"3"
D202 lowest-order byte					"C"	"B"	"A"	"0"	"4"
D202 highest-order byte						"C"	"B"	"A"	"0"
D203 lowest-order byte			(1)				"C"	"B"	"A"
D203 highest-order byte								"C"	"B"
D204 lowest-order byte									"C"

(1): Do not change

■Bit configuration when (n) is K4

D100 = 0ABCH

0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0
0				A				B				C			

D200

0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0
"A" → 41H								"0" → 30H							

D201

0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0
"C" → 43H								"B" → 42H							

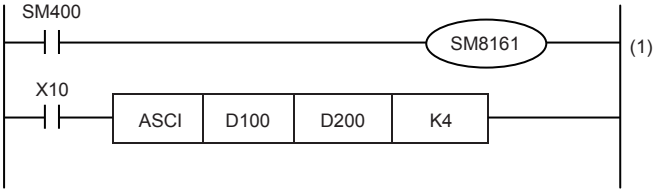
ASCII code

"0" = 30H	"1" = 31H	"5" = 35H
"A" = 41H	"2" = 32H	"6" = 36H
"B" = 42H	"3" = 33H	"7" = 37H
"C" = 43H	"4" = 34H	"8" = 38H

• 8-bit conversion mode (while SM8161 is ON)

Each digit of hexadecimal data stored in the device specified by (s) and later is converted into ASCII code, and transferred to the lower 8 bits of each device specified by (d) and later. SM8161 must always be on in the 8-bit conversion mode.

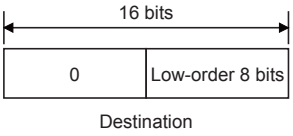
In the following program, conversion is executed as follows:



Devices specified by (s) and later
(D100)=0ABCH
(D101)=1234H
(D102)=5678H

(1): 8-bit mode

When SM8161 is set to on, the 8-bit mode is selected.
The conversion processing is executed as follows.



■Number of specified digits (characters) and conversion result

(n) / (d)	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200	"C"	"B"	"A"	"0"	"4"	"3"	"2"	"1"	"8"
D201		"C"	"B"	"A"	"0"	"4"	"3"	"2"	"1"
D202			"C"	"B"	"A"	"0"	"4"	"3"	"2"
D203				"C"	"B"	"A"	"0"	"4"	"3"
D204					"C"	"B"	"A"	"0"	"4"
D205						"C"	"B"	"A"	"0"
D206				(1)			"C"	"B"	"A"
D207								"C"	"B"
D208									"C"

(1): Do not change

■Bit configuration when (n) is K2

D100 = 0ABCH

0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0
0				A				B				C			

D200 = ASCII code of B = 42H

0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
								4				2			

D201 = ASCII code of C = 34H

0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
								4				3			

ASCII code

"0" = 30H	"1" = 31H	"5" = 35H
"A" = 41H	"2" = 32H	"6" = 36H
"B" = 42H	"3" = 33H	"7" = 37H
"C" = 43H	"4" = 34H	"8" = 38H

Precautions

- When outputting data in the BCD format for a printer, for example, it is necessary to convert binary data into BCD data before executing the ASCI(P) instructions.
- Whether NULL (00H) is stored after the last character or not depends on the ON/OFF status of the output character number selector signal SM701. When SM701 is off, NULL (00H) is stored. When SM701 is on, the original data remains.
- Depending on the ON/OFF status of SM701 and SM8161, the number of devices occupied by (d) differs.

SM701	SM8161	Number of devices occupied by (d)
ON	ON	Number of letters
ON	OFF	Number of letters ÷ 2
OFF	ON	Number of letters ÷ 1
OFF	OFF	(Number of letters ÷ 2) + 1

- When HEXA(P), CRC(P), or CCD(P) is used, the extension flag SM8161 is common to other instructions. When using an instruction described above and the ASCI(P) instructions in the same program, make sure to set SM8161 to ON or OFF just before each instruction so that SM8161 does not apply to another instruction.

Operation error

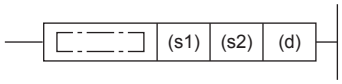
Error code (SD0/SD8067)	Description
2820H	The device specified by (s) or (d) exceeds the corresponding device range.
3405H	The value specified by (s) is other than any of 1 to 32767.

Converting 16-bit binary data to character string

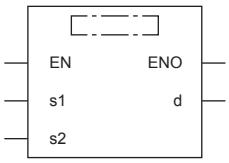
STR(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add a decimal point to the 16-bit binary data in the device specified by (s2) at the location specified by (s1), convert the data to character string data, and store the converted data in the device areas specified by (d) and later.

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	STR(P)	Head device number where the number of digits of the conversion target data is stored	—	16-bit signed binary	ANY16_S_ARRAY (Number of elements: 2)
	STR(P)_U			16-bit unsigned binary	ANY16_U_ARRAY (Number of elements: 2)
(s2)	STR(P)	Conversion target data	-32768 to +32767	16-bit signed binary	ANY16_S
	STR(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)		Head device number for storing the converted data	—	Character string	ANYSTRING_SINGLE
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

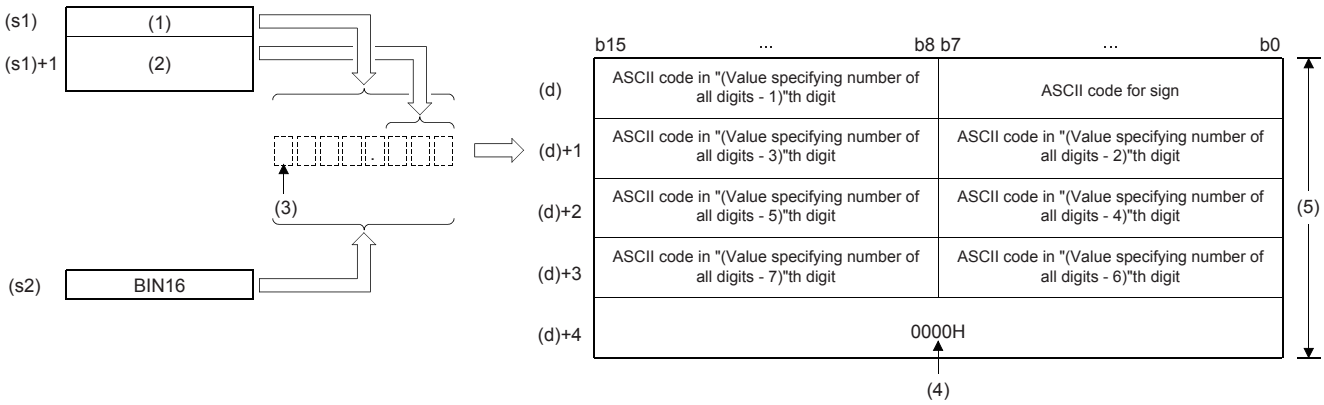
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	—	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

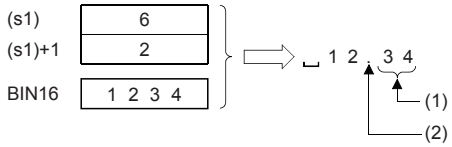
Processing details

- These instructions add a decimal point to the 16-bit binary data in the device specified by (s2) at the location specified by (s1), convert the data to character string data, and store the converted data in the device areas specified by (d) and later.



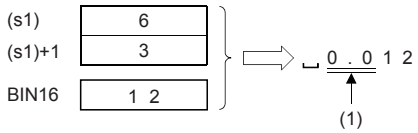
(1): Total number of digits
(2): Number of digits of decimal part
(3): Sign
(4): "0000H" is automatically stored at the end of the character string.
(5): For specifying number of all digits

- The total number of digits that can be specified by (s1) is 2 to 8.
- The number of digits in the decimal part that can be specified by (s1)+1 is 0 to 5. Note that the number of digits in the decimal part must be smaller than or equal to the total number of digits minus 3.
- The converted character string data are stored in the device areas specified by (d) and later as shown below.
  - As sign data, "20H" (space) is stored if the 16-bit binary data is positive, and "2DH" (-) is stored if the data is negative.
  - If the number of digits in the decimal part is set to other than 0, "2EH" (.) is automatically stored at the position before the specified number of digits. If the number of digits in the decimal part is 0, "2EH" (.) is not stored.



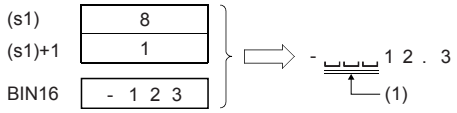
(1): Number of digits of decimal part
(2): Added automatically

  - If the specified number of digits in the decimal part is greater than the number of digits of the 16-bit binary data, 0(s) is automatically added and the data is regarded as "0.□□□□".



(1): Added automatically

  - If the total number of digits excluding the sign and the decimal point is greater than the number of digits of the 16-bit binary data, "20H" (space) is stored between the sign and the numeric value. If the number of digits of the 16-bit binary data is greater, an error occurs.

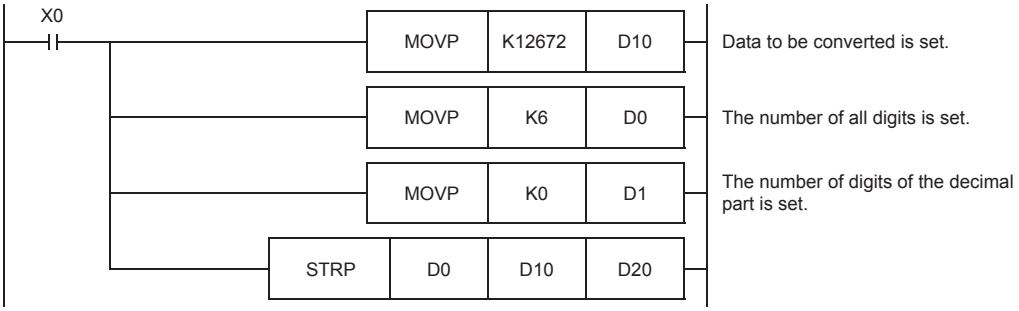


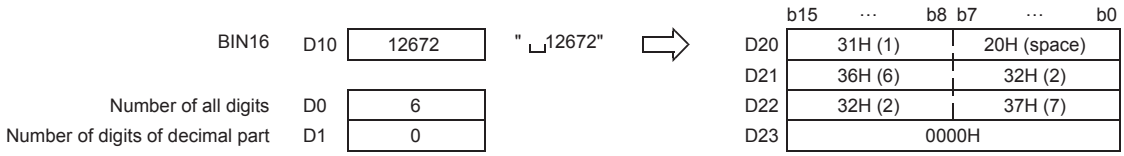
(1): "20H (SP)" is stored.

  - The value "00H" is automatically stored at the end of the converted character string.
  - When the number of all digits is even, "0000H" is stored in the device after the last character. When the number of all digits is odd, "00H" is stored in the upper byte (8 bits) of the device storing the final character.

Program example

In the program below, the 16-bit binary data stored in D10 is converted into a character string in accordance with the digit specification by D0 and D1 when X0 is set to ON, and then stored in D20 to D23.





Operation error

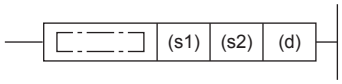
Error code (SD0/SD8067)	Description
3401H	The number of digits specified by (s1) is smaller than the number of digits plus 2 of the 16-bit binary data in the device specified by (s2). (The additional 2 digits indicate the sign (+/-) and the decimal point.)
	The total number of digits specified by (s1) is out of the valid range (2 to 8).
	The number of digits in the decimal part specified by (s1)+1 is out of the valid range (0 to 5).
	The relationship between the total number of digits specified by (s1) and the number of digits in the decimal part specified by (s1)+1 does not satisfy the following. (Total number of digits)-3 ≥ Number of digits in the decimal part
3406H	The device areas storing the character string specified by (d) exceed the corresponding device range.
2820H	The device range specified by (s1) exceeds the corresponding device range.

Converting 32-bit binary data to character string

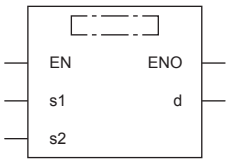
DSTR(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add a decimal point to the 32-bit binary data in the device specified by (s2) at the location specified by (s1), convert the data to character string data, and store the converted data in the device areas specified by (d) and later.

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DSTR(P)	Head device number where the number of digits of the conversion target data is stored	—	16-bit signed binary	ANY16_S_ARRAY (Number of elements: 2)
	DSTR(P)_U			16-bit unsigned binary	ANY16_U_ARRAY (Number of elements: 2)
(s2)	DSTR(P)	Conversion target data	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DSTR(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)		Head device number for storing the converted data	—	Character string	ANYSTRING_SINGLE
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

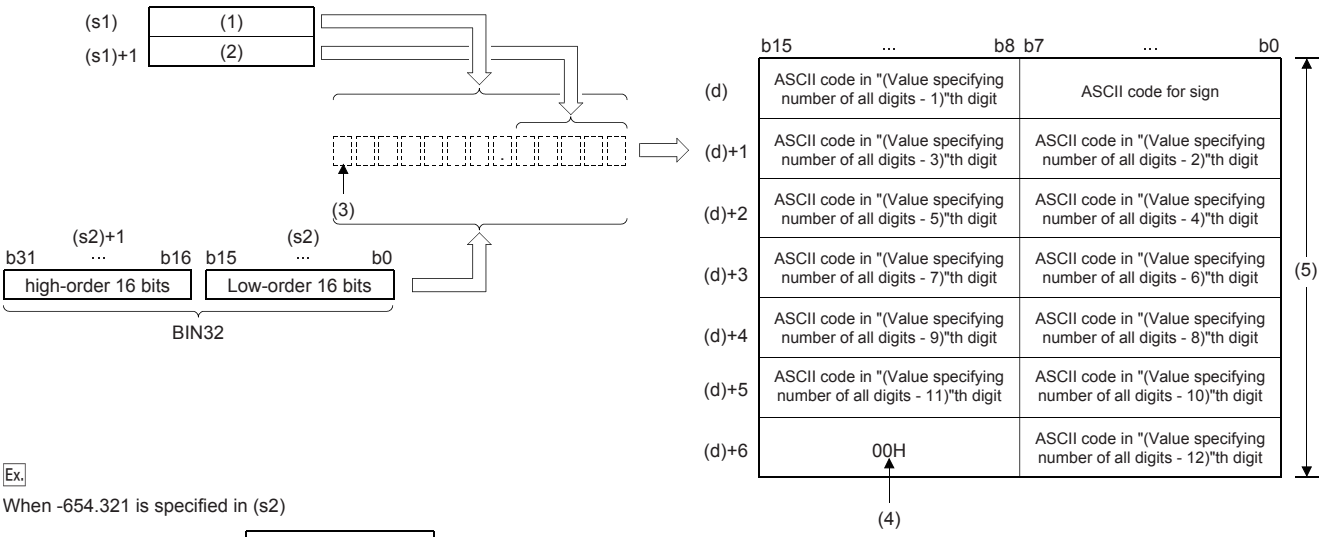
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

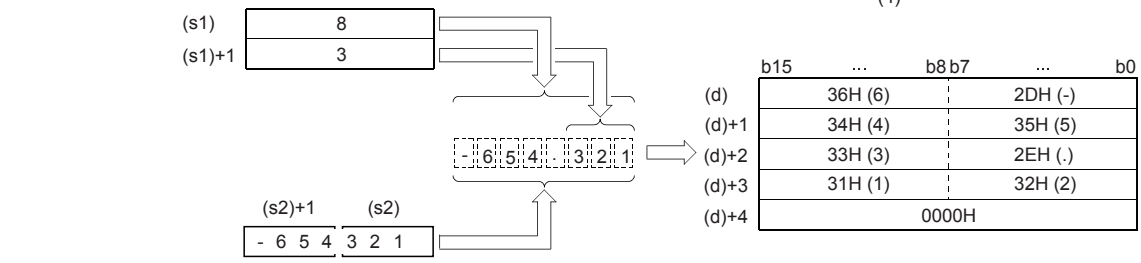
Processing details

- These instructions add a decimal point to the 32-bit binary data in the device specified by (s2) at the location specified by (s1), convert the data to character string data, and store the converted data in the device areas specified by (d) and later.



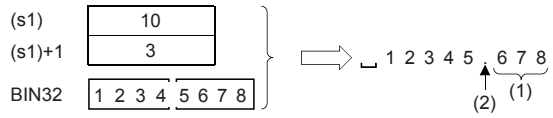
Ex.

When -654.321 is specified in (s2)



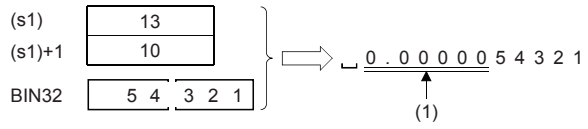
(1): Total number of digits
(2): Number of digits of decimal part
(3): Sign
(4): "00H" is automatically stored at the end of the character string.
(5): For specifying number of all digits

- The total number of digits that can be specified by (s1) is 2 to 13.
- The number of digits in the decimal part that can be specified by (s1)+1 is 0 to 10. Note that the number of digits in the decimal part must be smaller than or equal to the total number of digits minus 3.
- The converted character string data are stored in the device areas specified by (d) and later as shown below.
  - As sign data, "20H" (space) is stored if the 32-bit binary data is positive, and "2DH" (-) is stored if the data is negative.
  - If the number of digits in the decimal part is set to other than 0, "2EH" (.) is automatically stored at the position before the specified number of digits. If the number of digits in the decimal part is 0, "2EH" (.) is not stored.



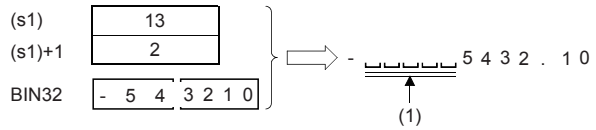
(1): Number of digits of decimal part
(2): Added automatically

- If the specified number of digits in the decimal part is greater than the number of digits of the 32-bit binary data, 0(s) is automatically added and the data is regarded as "0.□□□□".



(1): Added automatically

- If the total number of digits excluding the sign and the decimal point is greater than the number of digits of the 32-bit binary data, "20H" (space) is stored between the sign and the numeric value. If the number of digits of the 32-bit binary data is greater, an error occurs.



(1): "20H (SP)" is stored.

- The value "00H" is automatically stored at the end of the converted character string.
- When the number of all digits is even, "0000H" is stored in the device after the last character. When the number of all digits is odd, "00H" is stored in the upper byte (8 bits) of the device storing the final character.

Operation error

Error code (SD0/SD8067)	Description
3401H	The number of digits specified by (s1) is smaller than the number of digits plus 2 of the 16-bit binary data in the device specified by (s2). (The additional 2 digits indicate the sign (+/-) and the decimal point.)
	The total number of digits specified by (s1) is out of the valid range (2 to 13).
	The number of digits in the decimal part specified by (s1)+1 is out of the valid range (0 to 10).
	The relationship between the total number of digits specified by (s1) and the number of digits in the decimal part specified by (s1)+1 does not satisfy the following. (Total number of digits)-3 ≥ Number of digits in the decimal part
3406H	The device areas storing the character string specified by (d) exceed the corresponding device range.
2820H	The device range specified by (s1) exceeds the corresponding device range.

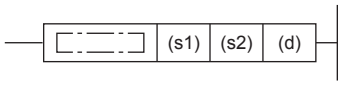
Converting single-precision real number to character string

ESTR(P)/DESTR(P)

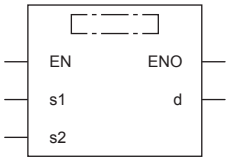
FX5S FX5UJ FX5U FX5UC

These instructions convert the single-precision real number data stored in the device specified by (s1) into a character string according to the display specification stored in the device specified by (s2) and later, and store the string in the device specified by (d) and later.

The ESTR(P) instructions can also be used as DESTR(P).

Ladder diagram	Structured text
	ENO:=ESTR(EN,s1,s2,d); ENO:=ESTRP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Single-precision real number data to be converted or the start number of the device where data is stored	0, 2^{-126}<|(s1)|<2^{128}	Single-precision real number	ANYREAL_32
(s2)	Head device number storing the display specification of a numeric value to be converted The device specified in (s2) and following 2 devices are used.	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(d)	Head device number for storing the converted data	—	Character string	ANYSTRING_SINGLE
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

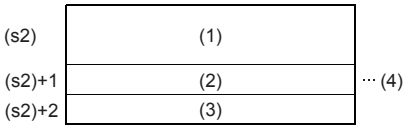
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	—	○	—	—
(s2)	○	○	○	—	—	—	○	—	—	—	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

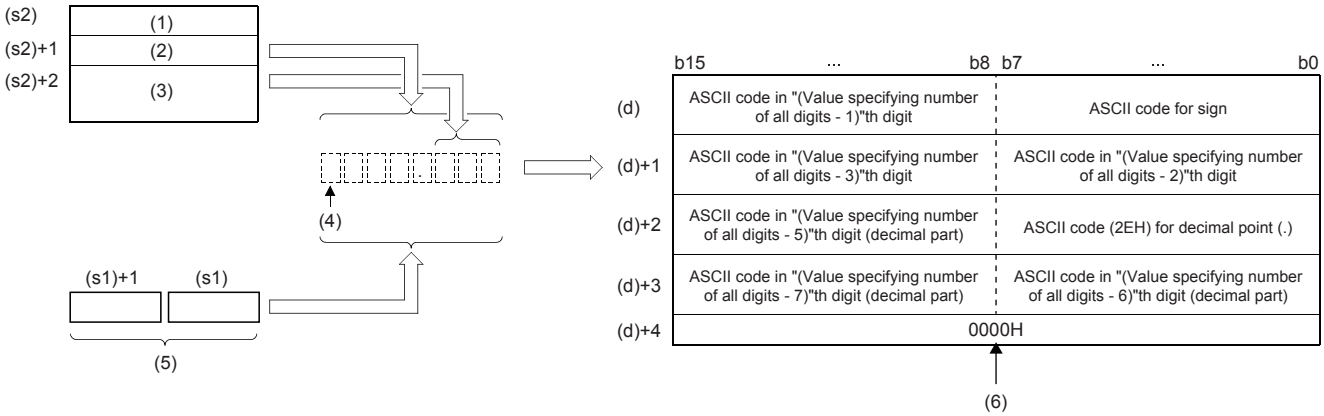
- These instructions convert the single-precision real number data stored in the device specified by (s1) into a character string according to the display specification stored in the device specified by (s2) and later, and store the string in the device specified by (d) and later. A real number can be directly specified as (s1).
- The data after conversion varies depending on the display specification stored in (s2).



(1): 0: Decimal part format, 1: Exponent format
(2): Total number of digits
(3): Number of digits of decimal part
(4): 2 to 24 can be set.

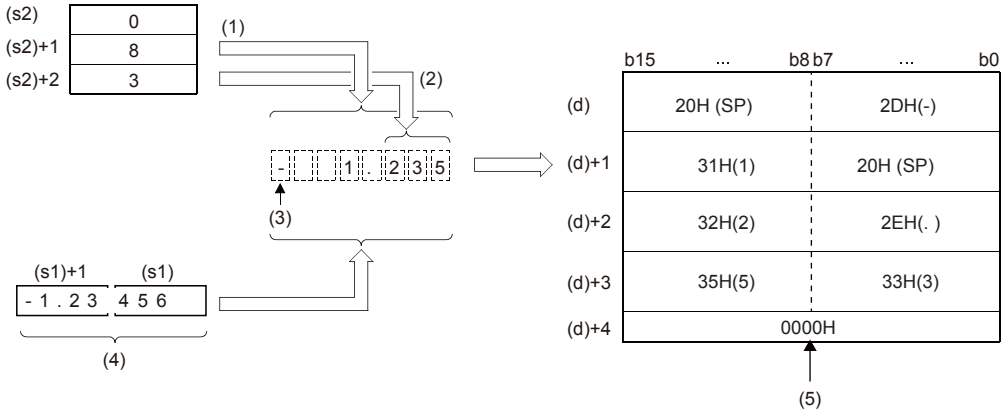
■Decimal point format

- When 0 is specified in (s2), the decimal point format is applied.



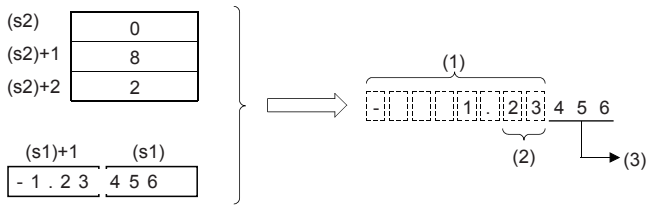
(1): Decimal point format
(2): Total number of digits
(3): Number of digits of decimal part
(4): Sign
(5): Single-precision real number
(6): "0000H" is automatically stored at the end of the character string.

- When the number of decimal part digits is 0, the number of digits that can be specified by (s2)+1 is "the number of digits (24 at a maximum) ≥ 2". For other than 0, the number of digits that can be specified by (s2)+1 is "the number of digits (24 at a maximum) ≥ (the number of decimal point digits + 3)".
- The number of digits in the decimal part that can be specified by (s2)+2 is 0 to 7. Note that the number of digits in the decimal part must be smaller than the total number of digits minus 3.
- For example, when the total number of digits is "8", the number of digits of the decimal part is "3", and "-1.23456" is specified, data is stored in (d) and later as shown below:



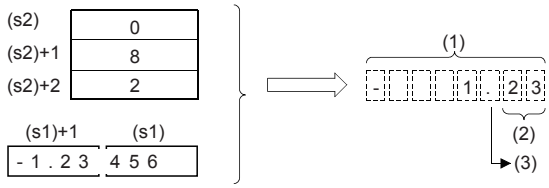
(1): Total number of digits
(2): Number of digits of decimal part
(3): Sign
(4): Single-precision real number
(5): "0000H" is automatically stored at the end of the character string.

- The converted character string data are stored in the device areas specified by (d) and later as shown below.
  - As sign data, "20H" (space) is stored if the single-precision real number is positive, and "2DH" (-) is stored if the data is negative.
  - If the decimal part of the single-precision real number data cannot be accommodated in the number of digits of the decimal part, lower digits of the decimal part are rounded off.



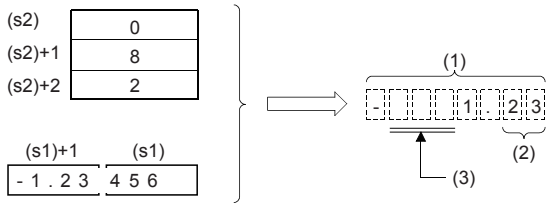
(1): Total number of digits (2): Number of digits of decimal part (3): These digits are rounded off.

  - If the number of digits in the decimal part is set to other than 0, "2EH" (.) is automatically stored at the position before the specified number of digits. If the number of digits in the decimal part is 0, "2EH" (.) is not stored.



(1): Total number of digits
(2): Number of digits of decimal part
(3): Added automatically

  - When the total number of digits subtracted by the digits for sign, decimal point, and decimal part is larger than the integer part of the single-precision real number data, "20H (space)" is stored between the sign and the integer part.

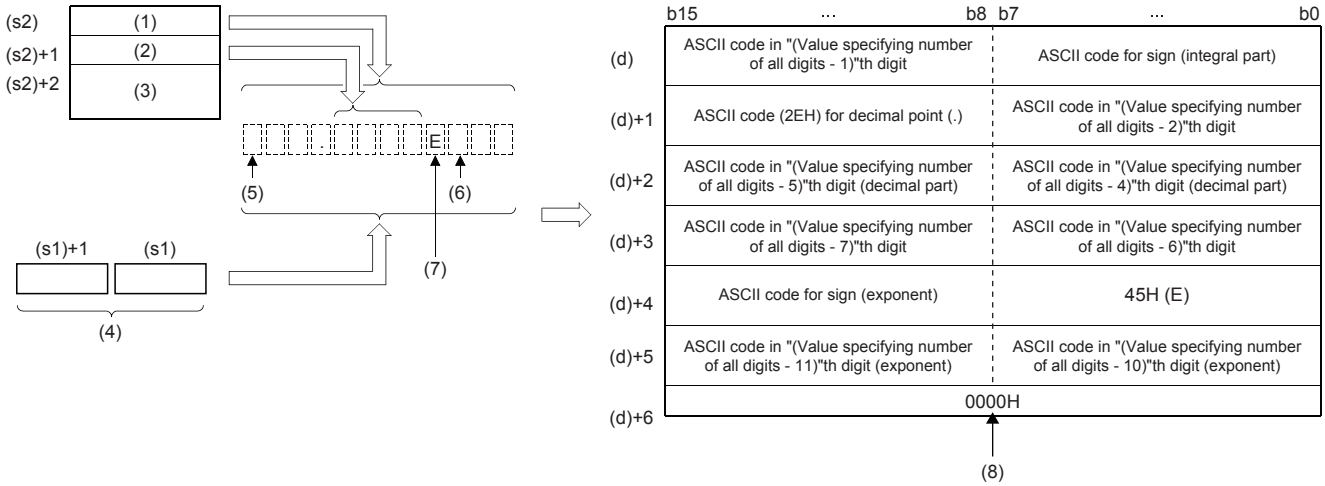


(1): Total number of digits
(2): Number of digits of decimal part
(3): "20H (SP)" is stored.

  - The value "00H" is automatically stored at the end of the converted character string.

■Exponent format

• When 1 is specified in (s2), the exponent format is applied.

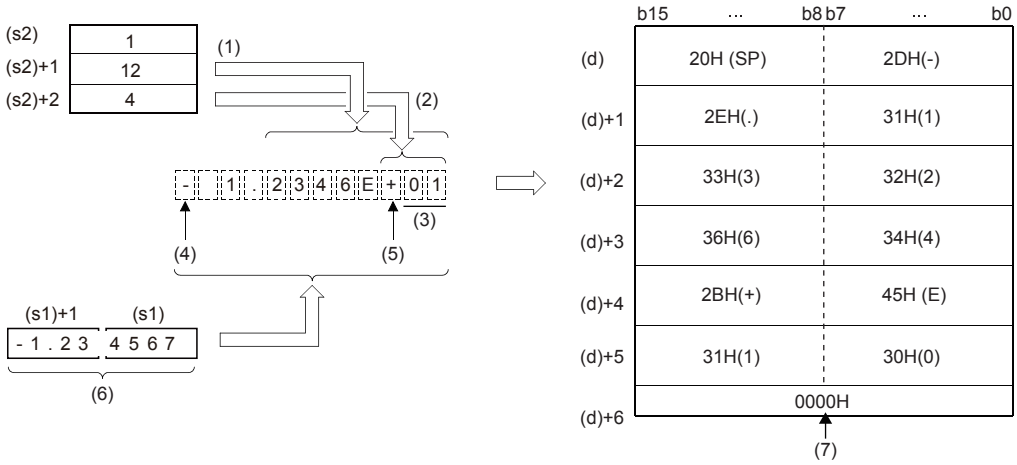


(1): Exponent format
(2): Total number of digits
(3): Number of digits of decimal part
(4): Single-precision real number
(5): Sign (integer part)
(6): Sign (exponent part)
(7): Added automatically
(8): "0000H" is automatically stored at the end of the character string.

• When the number of decimal part digits is 0, the number of digits that can be specified by (s2)+1 is "the number of digits (24 at a maximum) ≥ 6". For other than 0, the number of digits that can be specified by (s2)+1 is "the number of digits (24 at a maximum) > (the number of decimal point digits + 7)".

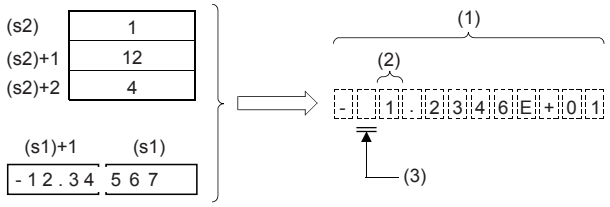
• The number of digits in the decimal part that can be specified by (s2)+2 is 0 to 7. Note that the number of digits in the decimal part must be equal to or smaller than the total number of digits minus 7.

• For example, when the total number of digits is "12", the number of digits of the decimal part is "4", and "-12.34567" is specified, data is stored in (d) and later as shown below:



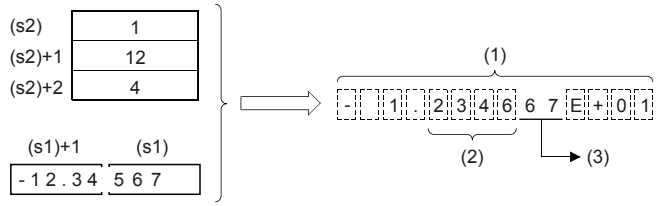
(1): Total number of digits
(2): Number of digits of decimal part
(3): Fixed to 2 digit
(4): Sign (integer part)
(5): Sign (exponent part)
(6): Single-precision real number
(7): "0000H" is automatically stored at the end of the character string.

- The converted character string data are stored in the device areas specified by (d) and later as shown below.
  - As sign data of the integral part, "20H" (space) is stored if the single-precision real number is positive, and "2DH" (-) is stored if the data is negative.
  - The integer part is fixed to 1 digit. "20H (space)" is stored between the integer part and the sign.



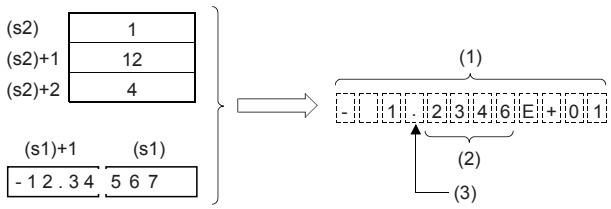
(1): Total number of digits
(2): Fixed to 1 digit
(3): "20H (SP)" is stored.

  - If the decimal part of the single-precision real number data cannot be accommodated in the number of digits of the decimal part, lower digits of the decimal part are rounded.



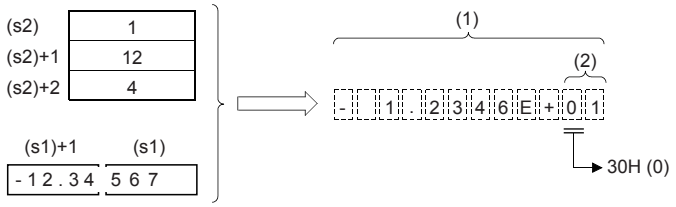
(1): Total number of digits
(2): Number of digits of decimal part
(3): These values are rounded down.

  - If the number of digits in the decimal part is set to other than 0, "2EH" (.) is automatically stored at the position before the specified number of digits. If the number of digits in the decimal part is 0, "2EH" (.) is not stored.



(1): Total number of digits
(2): Number of digits of decimal part
(3): Added automatically

  - For the sign of the exponent part, "2BH (+)" is stored when the exponent is positive, and "2DH (-)" is stored when the exponent is negative.
  - The exponent part is fixed to 2 digits. When the exponent part is 1 digit, "30H (0)" is stored after the sign of the exponent part.

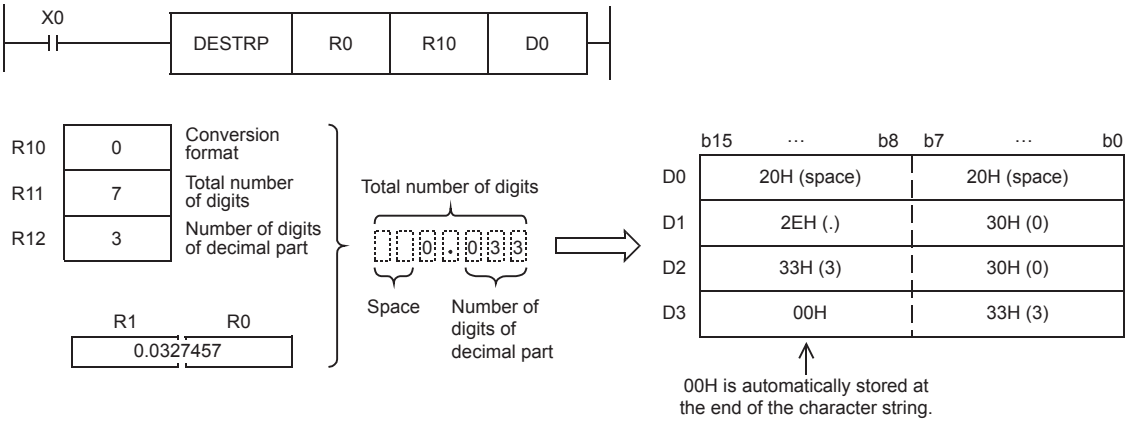


(1): Total number of digits
(2): Fixed to 2 digit

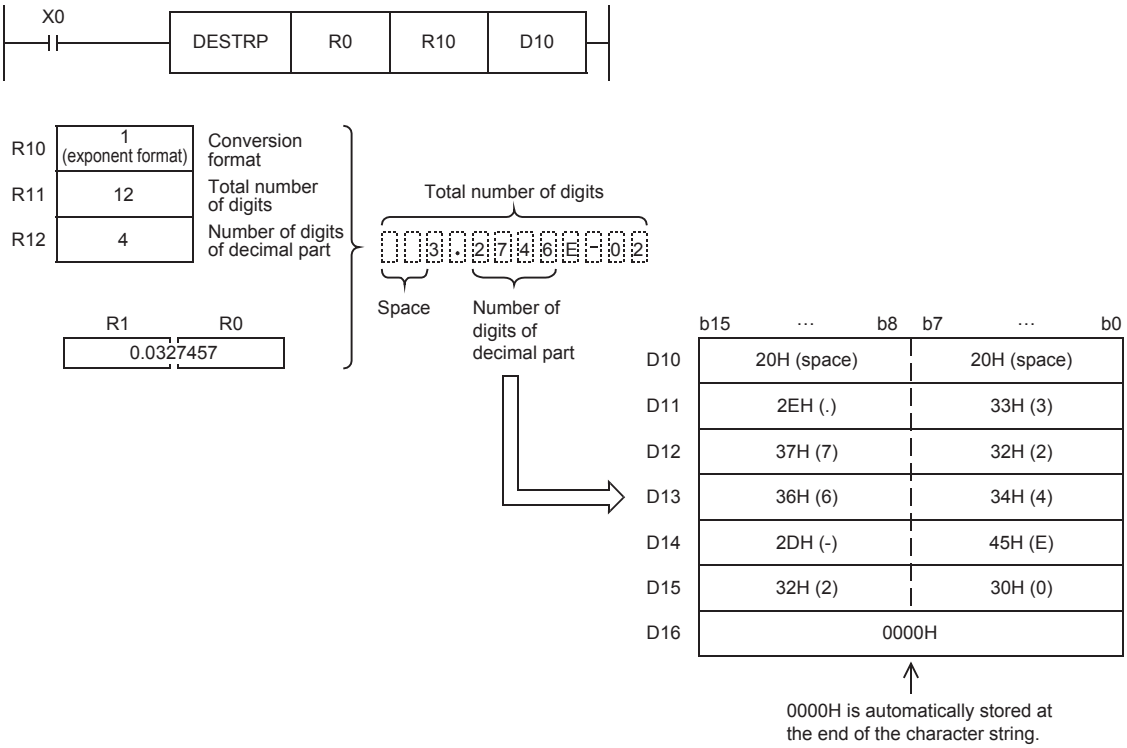
  - The value "00H" is automatically stored at the end of the converted character string.

Program example

In the program example shown below, the contents (single-precision real number data) of R0 and R1 are converted according to the contents specified by R10 to R12, and then stored to D0 and later when X0 turns ON.



In the program shown below, the contents (single-precision real number data) of R0 and R1 are converted according to the contents specified by R10 to R12, and then stored to D10 and later when X0 turns ON.



Operation error

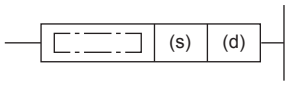
Error code (SD0/SD8067)	Description
2820H	The device specified by (s2) exceeds the corresponding device range.
3401H	The number of total digits specified by (s1)+1 exceeds 24.
	The format specified by (s2) is any value other than "0" or "1".
	The total number of digits specified by (s2)+1 is not within the following range in the decimal point format. When the number of digits of the decimal part is "0": Total number of digits ≥ 2 When the number of digits of the decimal part is any value other than "0": Total number of digits ≥ (Number of digits of decimal part + 3)
	The total number of digits specified by (s2)+1 is not within the following range in the exponent format. When the number of digits of the decimal part is "0": Total number of digits ≥ 6 When the number of digits of the decimal part is any value other than "0": Total number of digits ≥ (Number of digits of decimal part + +7)
	The number of digits of the decimal part specified by (s2)+2 is not within the following range. In the decimal part format ≤ (Total number of digits - 3) In the exponent format ≤ (Total number of digits - 7)
	When the conversion result exceeds the specified total number of digits
3402H	(s1) is not within the following range 0, $\pm 2^{-126} \leq (s1) < \pm 2^{128}$
	The specified device value is denormalized number, NaN (not a number), or $\pm\infty$.
3405H	The number of digits of the decimal part specified by (s2)+2 is not within the following range. 0 to 7
3406H	The device areas that store the character string specified by (d) exceed the corresponding device range.

Converting Unicode character string to Shift JIS character string

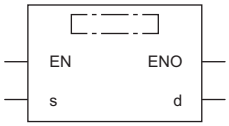
WS2SJIS(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert a Unicode character string to a Shift JIS character string.

Ladder	ST
	ENO:=WS2SJIS(EN,s,d); ENO:=WS2SJISP(EN,s,d);

FBD/LD



Setting data

■Description, range, data type

Operand	Description	Range	Data type	Data type (label)
(s)	Unicode string (up to 255 characters) to be converted or start device containing the Unicode string	—	Unicode string	ANYSTRING_DOUBLE
(d)	Start device for storing the converted Shift JIS character string	—	String	ANYSTRING_SINGLE
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

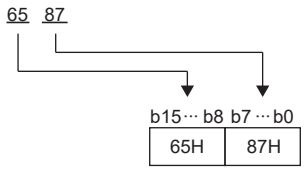
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	○	—
(d)	—	○	—	—	—	—	○	—	—	—	—

Processing details

- These instruction convert the Unicode character string in the device specified by (s) to the shift JIS character string, and stores the converted data in the device specified by (d).
- Specify the Unicode string in (s) in little endian.

Ex.

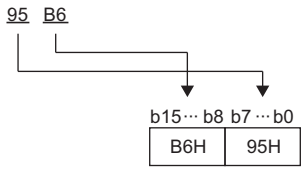
A Unicode string “6587H” is specified.



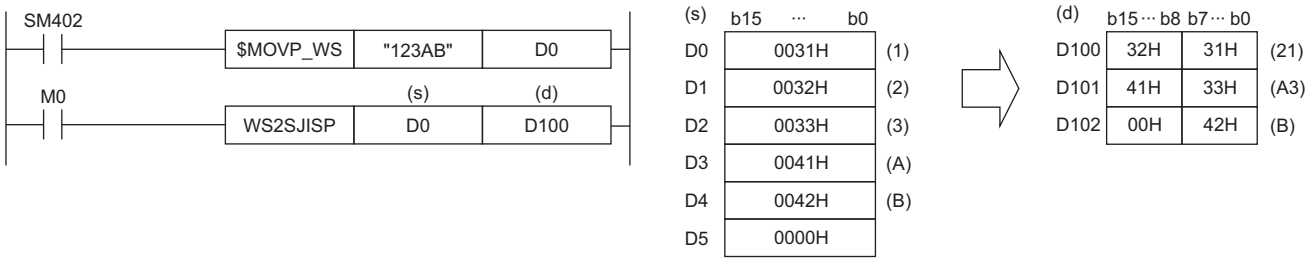
• The shift JIS string in (d) is stored in big endian.

Ex.

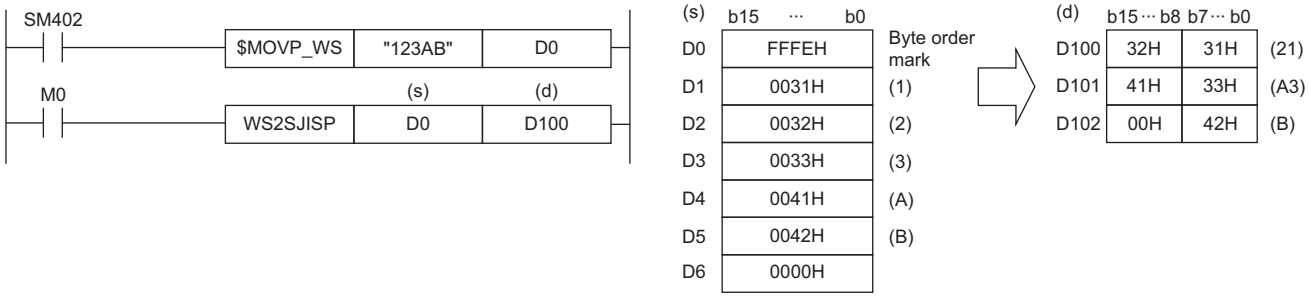
A shift JIS string “95B6H” is specified with “B695H”.



• When a byte order mark is not used, conversion from Unicode to shift JIS occurs as follows.



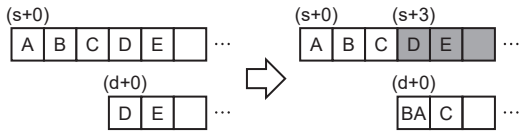
• When a byte order mark (FEFFH) is added, conversion from Unicode to shift JIS occurs as follows.



Precautions

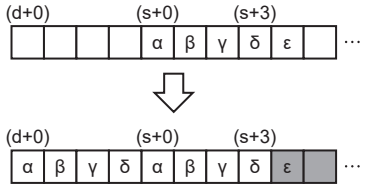
- If the Unicode string specified in (s) contains a character code that cannot be converted, the string to the character causing the error is written to the device No. specified in (d).
- 32-bit Unicode strings cannot be converted to shift JIS.
- If the device ranges specified in (s) and (d) overlap, the overlap may be detected in advance. The operation in each case of overlap of the device ranges is shown below.

[When the overlap is detected in advance]



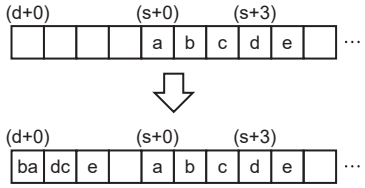
If the overlap of the device ranges is detected in advance, the string to the overlapped point is converted and written to (d).
Since the overlap of the device ranges in (s+3) is revealed in advance, the string in (s+0) to (s+2) is converted and written to (d+0) and (d+1).
An overlap error (2821H) occurs.

[When the overlap cannot be detected in advance (error completion)]



Since the device ranges overlap when the string in (s+3) is converted, the next destination to store is not found.
An overlap error (2821H) occurs.

[When the overlap cannot be detected in advance (normal completion)]



As the result of string conversion, the device ranges do not overlap, and the program terminates normally.

Operation error

Error code (SD0/SD8067)	Description
2820H	0000H does not exist between setting areas in the device/label memory after the device No. specified in (s).
2821H	The ranges of data in the devices specified by (s) and (d) are overlapping.
3401H	Byte order mark FEFFH (big endian) is added to the character string in the device specified by (s).
	The range of data in the device specified by (s) includes a character code that cannot be converted.
3405H	The character string in the device specified by (s) exceeds 512 characters.*1
3406H	All converted shift JIS (ASCII) strings cannot be stored to the number of points to the last No. in each setting area of the relevant device/label memory after the device No. specified in (d).

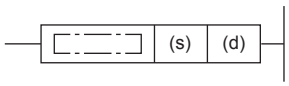
*1 A two-byte character such as a kanji character represented in shift JIS code should be counted 2.

Converting shift JIS character string to Unicode character string (without byte order mark)

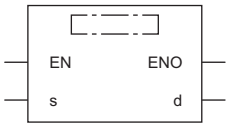
SJIS2WS(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert a Shift JIS character string to a Unicode character string.

Ladder	ST
	ENO:=SJIS2WS(EN,s,d); ENO:=SJIS2WSP(EN,s,d);

FBD/LD



Setting data

■Description, range, data type

Operand	Description	Range	Data type	Data type (label)
(s)	Shift JIS string (up to 255 characters) to be converted or start device containing the shift JIS string	—	String	ANYSTRING_SINGLE
(d)	Start device for storing the converted Unicode character string	—	Unicode string	ANYSTRING_DOUBLE
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

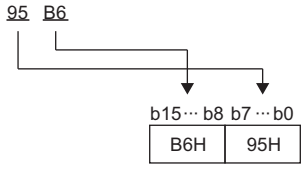
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	○	—
(d)	—	○	—	—	—	—	○	—	—	—	—

Processing details

- These instructions convert the shift JIS character string in the device specified by (s) to a Unicode character string, and store the converted data in the device specified by (d).
- Specify the shift JIS string in (s) in big endian.

Ex.

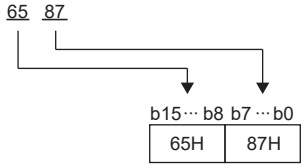
A shift JIS string “95B6H” is specified with “B695H”.



• The Unicode string in (d) is stored in little endian.

Ex.

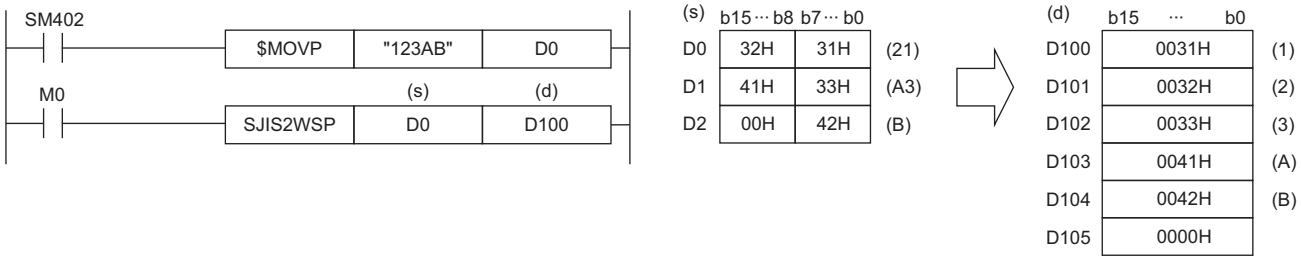
A Unicode string “6587H” is specified.



• The SJIS2WS(P) instruction does not add a byte order mark to the beginning of the data in the device specified by (d). To add a byte order mark, use the SJIS2WSB(P) instruction.

☞ Page 634 SJIS2WSB(P)

• The following figure shows the operation for converting shift JIS to Unicode.



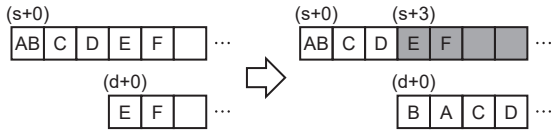
Precautions

• When the data length of the ASCII (shift JIS) string specified in (s) is 0, only the one character (one word) of termination NULL is written to (d)+0.

• If the ASCII (shift JIS) string specified in (s) contains a character code that cannot be converted, the string to the character causing the error is written to the device No. specified in (d).

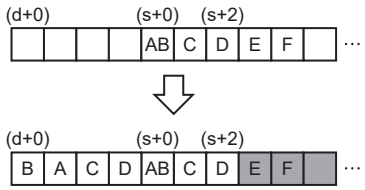
• If the device ranges specified in (s) and (d) overlap, the overlap may be detected in advance. The operation in each case of overlap of the device ranges is shown below.

[When the overlap is detected in advance]



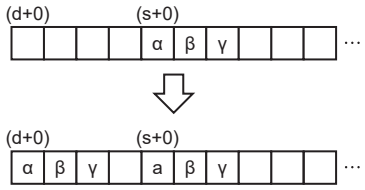
If the overlap of the device ranges is detected in advance, the string to the overlapped point is converted and written to (d).
Since the overlap of the device ranges in (s+3) is revealed in advance, the string in (s+0) to (s+2) is converted and written to (d+0) to (d+3).
An overlap error (2821H) occurs.

[When the overlap cannot be detected in advance (error completion)]



Since the device ranges overlap when the string in (s+2) is converted, the next destination to store is not found.
An overlap error (2821H) occurs.

[When the overlap cannot be detected in advance (normal completion)]



As the result of string conversion, the device ranges do not overlap, and the program terminates normally.

Operation error

Error code (SD0/SD8067)	Description
2820H	0000H does not exist between setting areas in the device/label memory after the device No. specified in (s).
2821H	The ranges of data in the devices specified by (s) and (d) are overlapping.
3401H	The range of data in the device specified by (s) includes a character code that cannot be converted.
3405H	The character string in the device specified by (s) exceeds 512 characters.*1
3406H	All converted shift Unicode strings cannot be stored to the number of points to the last No. in each setting area of the relevant device/label memory after the device No. specified in (d).

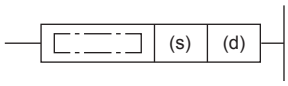
*1 A two-byte character such as a kanji character represented in shift JIS code should be counted 2.

Converting shift JIS character string to Unicode (with byte order mark)

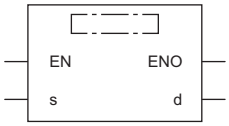
SJIS2WSB(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert a shift JIS character string to a Unicode character string, and adds a byte order mark to the start of the converted data.

Ladder	ST
	ENO:=SJIS2WSB(EN,s,d); ENO:=SJIS2WSBP(EN,s,d);

FBD/LD



Setting data

■Description, range, data type

Operand	Description	Range	Data type	Data type (label)
(s)	Shift JIS string (up to 255 characters) to be converted or start device containing the shift JIS string	—	String	ANYSTRING_SINGLE
(d)	Start device for storing the converted Unicode character string	—	Unicode string	ANYSTRING_DOUBLE
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

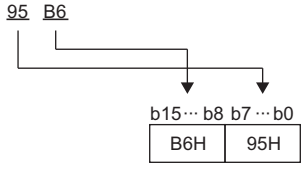
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	○	—
(d)	—	○	—	—	—	—	○	—	—	—	—

Processing details

- These instructions convert the shift JIS character string in the device specified by (s) to the Unicode character string, add a byte order mark to the start of the converted data, and store it in the device specified by (d).
- Specify the shift JIS string in (s) in big endian.

Ex.

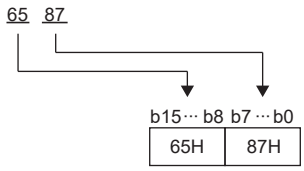
A shift JIS string “95B6H” is specified with “B695H”.



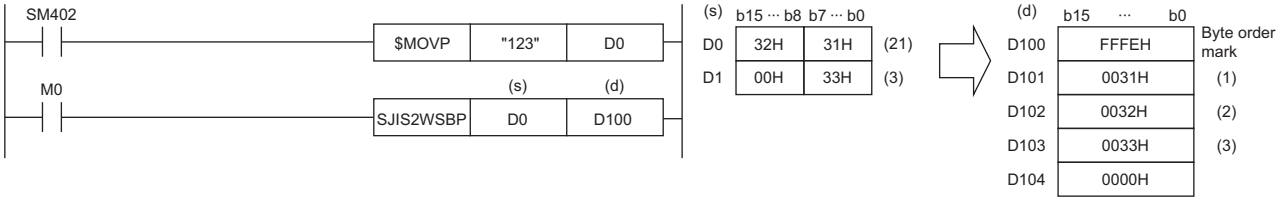
• The Unicode string in (d) is stored in little endian.

Ex.

A Unicode string “6587H” is specified.



• The following figure shows the operation for converting shift JIS to Unicode.



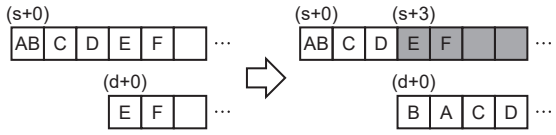
Precautions

• When the data length of the ASCII (Shift JIS) string specified in (s) is 0, only the byte order mark and one character (one word) of termination NULL are written to (d)+0.

• If the ASCII (Shift JIS) string specified in (s) contains a character code that cannot be converted, the string to the character causing the error including the byte order mark is written to the device No. specified in (d).

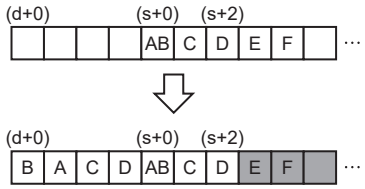
• If the device ranges specified in (s) and (d) overlap, the overlap may be detected in advance. The operation in each case of overlap of the device ranges is shown below.

[When the overlap is detected in advance]



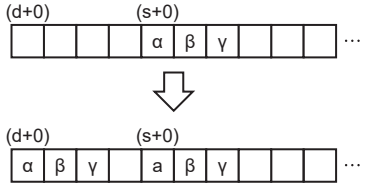
If the overlap of the device ranges is detected in advance, the string to the overlapped point is converted and written to (d).
Since the overlap of the device ranges in (s+3) is revealed in advance, the string in (s+0) to (s+2) is converted and written to (d+0) to (d+3).
An overlap error (2821H) occurs.

[When the overlap cannot be detected in advance (error completion)]



Since the device ranges overlap when the string in (s+2) is converted, the next destination to store is not found.
An overlap error (2821H) occurs.

[When the overlap cannot be detected in advance (normal completion)]



As the result of string conversion, the device ranges do not overlap, and the program terminates normally.

Operation error

Error code (SD0/SD8067)	Description
2820H	0000H does not exist between setting areas in the device/label memory after the device No. specified in (s).
2821H	The ranges of data in the devices specified by (s) and (d) are overlapping.
3401H	The range of data in the device specified by (s) includes a character code that cannot be converted.
3405H	The character string in the device specified by (s) exceeds 512 characters.*1
3406H	All converted shift Unicode strings cannot be stored to the number of points to the last No. in each setting area of the relevant device/label memory after the device No. specified in (d).

*1 A two-byte character such as a kanji character represented in shift JIS code should be counted 2.

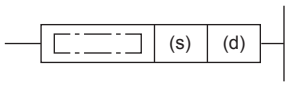
Detecting a character string length

LEN(P)

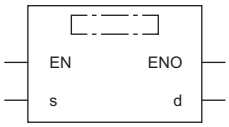
FX5S FX5UJ FX5U FX5UC

These instructions detect the length of the character string specified by (s), and store the length in the device specified by (d) and later.

These instructions handle data stored in the device specified by (s) to the device storing 00H as a character string.

Ladder diagram	Structured text*1
	ENO:=LENP(EN,s,d);

FBD/LD*1



*1 The LEN instruction is not supported by the ST language and the FBD/LD language. Use LEN of the standard function.
☞ Page 1337 LEN(_E)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Character string or head device number storing a character string	—	Character string	ANYSTRING_SINGLE
(d)	Device number storing the detected character string length	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

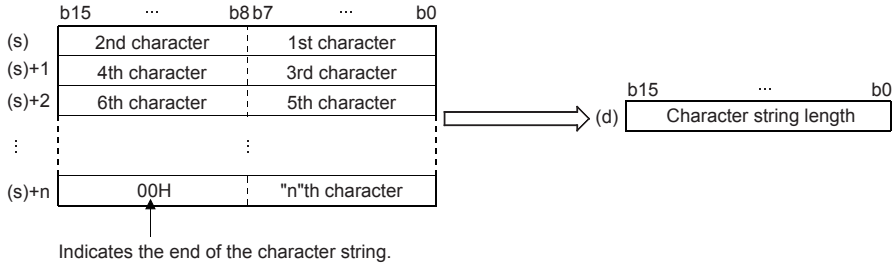
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

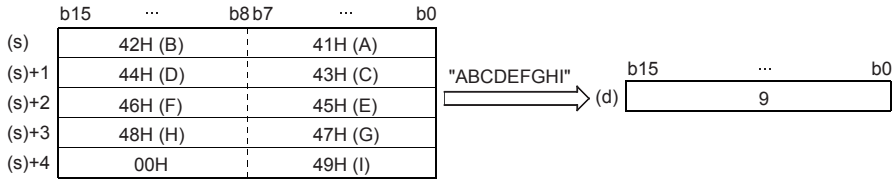
Processing details

- These instructions detect the length of the character string specified by (s), and store the length in the device specified by (d) and later.
- These instructions handle data stored in the device specified by (s) to the device storing 00H as a character string.



Ex.

When "ABCDEFGHI" is stored in (s) and later

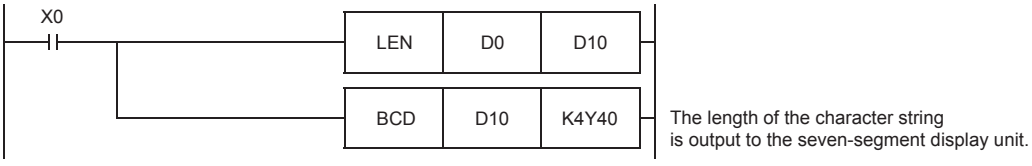


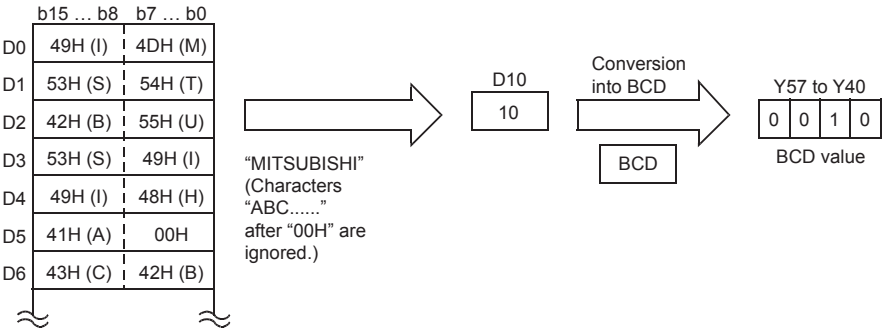
Precautions

The LEN(P) instructions can handle character codes other than ASCII codes, but the character string length is handled in byte units (8 bits). Accordingly, in the case of character codes in which 2 bytes express 1 character such as shift JIS codes, the length of 1 character is detected as "2".

Program example

In the program example shown below, the length of a character string stored in D0 and later is output in 4-digit BCD to Y40 to Y57 when X0 turns ON.





Operation error

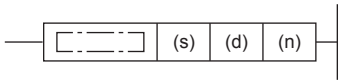
Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s) and later, "00H" does not exist.
3405H	The character string specified by (s) has more than 16383 characters.

Extracting character string data from the right

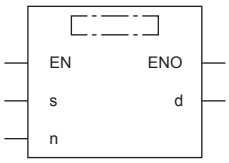
RIGHT(P)

FX5S FX5UJ FX5U FX5UC

These instructions extract "n" characters of the character string data stored in the device specified by (s) and later from the right end (from the end), and store the extracted characters in the device specified by (d) and later.

Ladder diagram	Structured text*1
	ENO:=RIGHTP(EN,s,n,d);

FBD/LD*1



*1 The RIGHT instruction is not supported by the ST language and the FBD/LD language. Use RIGHT of the standard function.
☞ Page 1339 LEFT(_E), RIGHT(_E)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Character string or head device number storing a character string	—	Character string	ANYSTRING_SINGLE
(d)	Head device number for storing "n" characters extracted from the right of the device specified by (s)	—	Character string	ANYSTRING_SINGLE
(n)	Number of characters to be extracted	1 to 16383	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

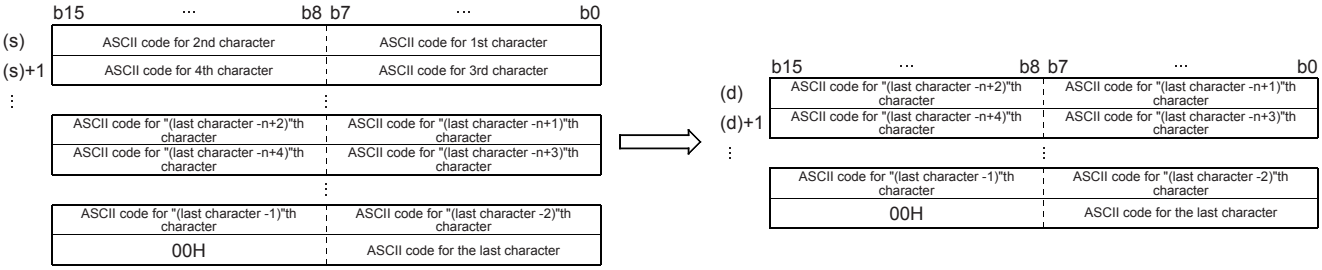
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

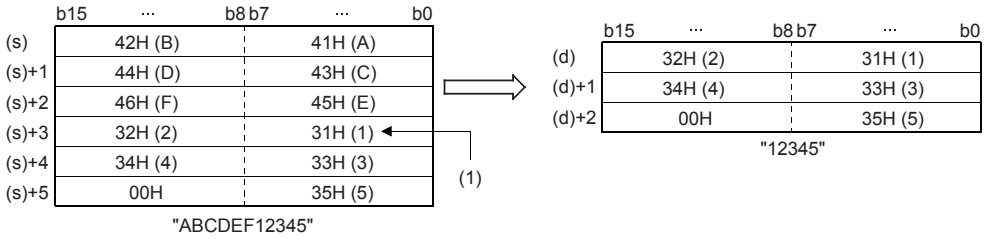
- These instructions extract "n" characters of the character string data stored in the device specified by (s) and later from the right end (from the end), and store the extracted characters in the device specified by (d) and later.



- A character string stored in (s) indicates data stored in devices from the specified device until "00H" is first detected in units of 1 byte.

Ex.

When 5 is specified in (n)



(1): ASCII code for 5th character

- A NULL code (00H), which indicates an end of a character string, is automatically added at the end of the character string data.
- When the number of extracted characters is odd, "00H" is stored in the upper byte of a device storing the last character. When the number of extracted characters is even, "0000H" is stored in the device after the last character.
- When the number of characters specified by (n) is 0, a NULL code (00H) is stored in (d).

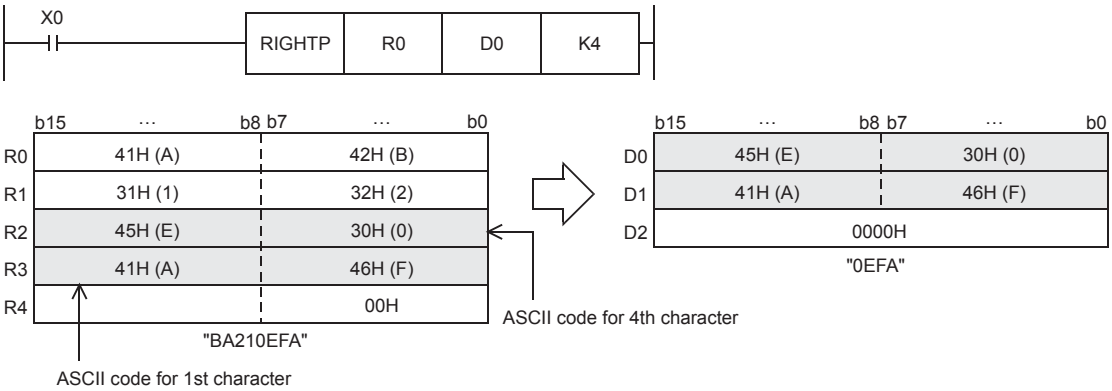
Precautions

When handling character codes other than ASCII codes, note the following points:

- The number of characters is handled in byte units (8 bits). Accordingly, in the case of character codes in which 2 bytes express 1 character such as shift JIS codes, 1 character is detected as "2".
- When extracting characters from a character string including character codes in which 2 bytes express 1 character such as shift JIS codes, consider the number of characters to be extracted in units of character codes for 1 character. Note that the expected character code is not retrieved if only 1 byte is extracted out of a 2-byte character code.

Program example

In the program example shown below, 4 characters are extracted from the right end of the character string data stored in R0 and later, and stored to D0 and later when X0 turns ON.



Operation error

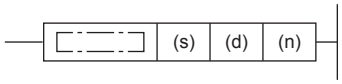
Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s) and later, "00H" does not exist.
3405H	(n) is not within the following range 0 to 16383
	The character string specified by (s) has more than 16383 characters.
	"n" exceeds the number of characters specified by (s)
3406H	The (n) points of data in the device starting from the one specified by (d) exceed the corresponding device range.

Extracting character string data from the left

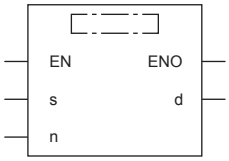
LEFT(P)

FX5S FX5UJ FX5U FX5UC

These instructions extract "n" characters of the character string data stored in the device specified by (s) and later from the left end (from the start), and store the extracted characters in the device specified by (d) and later.

Ladder diagram	Structured text*1
	ENO:=LEFTP(EN,s,n,d);

FBD/LD*1



*1 The LEFT instruction is not supported by the ST language and the FBD/LD language. Use LEFT of the standard function.
☞ Page 1339 LEFT(_E), RIGHT(_E)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Character string or head device number storing a character string	—	Character string	ANYSTRING_SINGLE
(d)	Head device number for storing "n" characters extracted from the left of the device specified by (s)	—	Character string	ANYSTRING_SINGLE
(n)	Number of characters to be extracted	1 to 16383	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

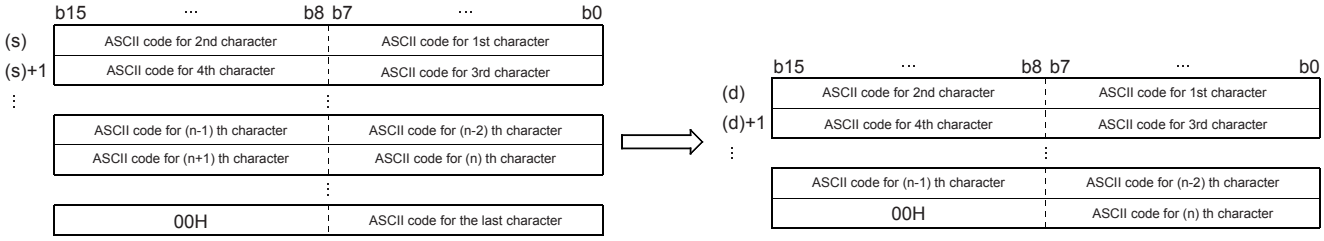
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

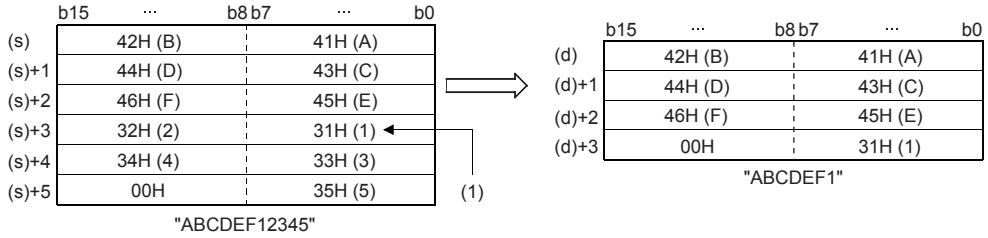
- These instructions extract "n" characters of the character string data stored in the device specified by (s) and later from the left end (from the start), and store the extracted characters in the device specified by (d) and later.



- A character string stored in (s) indicates data stored in devices from the specified device until "00H" is first detected in units of 1 byte.

Ex.

When 7 is specified in (n)



(1): ASCII code for 7th character

- A NULL code (00H), which indicates an end of a character string, is automatically added at the end of the character string data.
- When the number of extracted characters is odd, "00H" is stored in the upper byte of a device storing the last character. When the number of extracted characters is even, "0000H" is stored in the device after the last character.
- When the number of characters specified by (n) is 0, a NULL code (00H) is stored in (d).

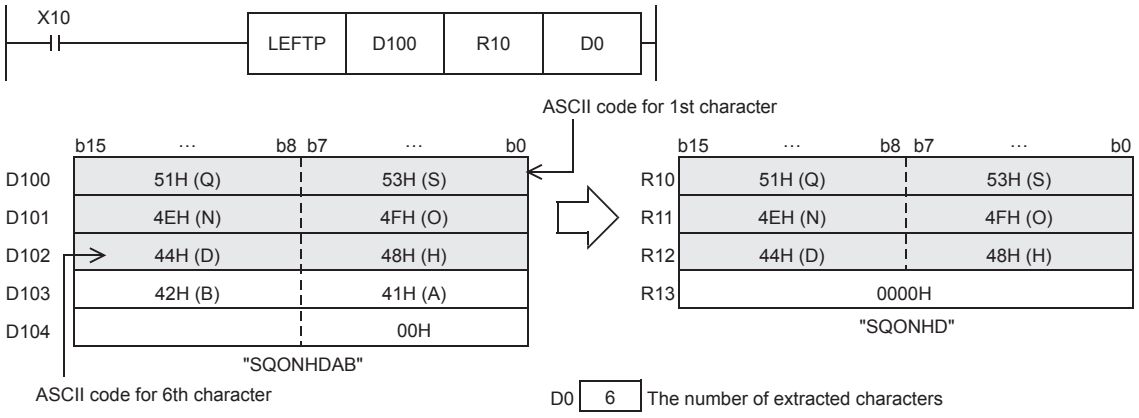
Precautions

When handling character codes other than ASCII codes, note the following points:

- The number of characters is handled in byte units (8 bits). Accordingly, in the case of character codes in which 2 bytes express 1 character such as shift JIS codes, 1 character is detected as "2".
- When extracting characters from a character string including character codes in which 2 bytes express 1 character such as shift JIS codes, consider the number of characters to be extracted in units of character codes for 1 character. Note that the expected character code is not retrieved if only 1 byte is extracted out of a 2-byte character code.

Program example

In the program example shown below, the number of characters which is equivalent to the number stored in D0 are extracted from the left end of the character string data stored in D100 and later, and stored to R10 and later when X10 turns ON.



Operation error

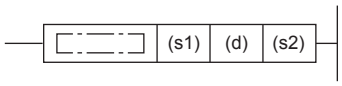
Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s) and later, "00H" does not exist.
3405H	(n) is not within the following range 0 to 16383
	The character string specified by (s) has more than 16383 characters.
	"n" exceeds the number of characters specified by (s)
3406H	The (n) points of data in the device starting from the one specified by (d) exceed the corresponding device range.

Storing the specified number of character strings

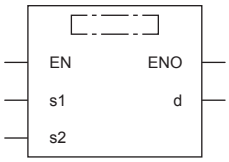
MIDR(P)

FX5S FX5UJ FX5U FX5UC

These instructions extract the number of characters specified by (s2)+1 of the character string data stored in the device specified by (s1) and later from the position specified by (s2), and store the extracted characters in the device specified by (d) and later.

Ladder diagram	Structured text
	ENO:=MIDR(EN,s1,s2,d); ENO:=MIDRP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Character string or head device number storing a character string	—	Character string	ANYSTRING_SINGLE
(d)	Head device number for storing the character string data of the operation result	—	Character string	ANYSTRING_SINGLE
(s2)	Head device number for storing the number of characters and position of the start character (s2)+0: Position of the start character, (s2)+1: Number of characters	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

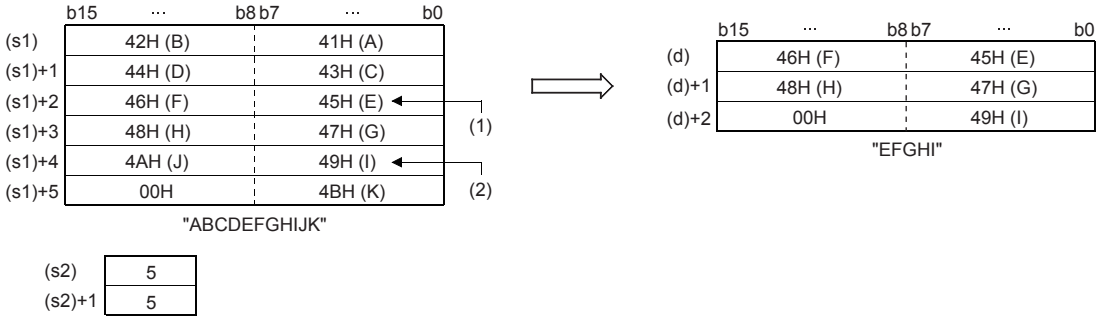
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

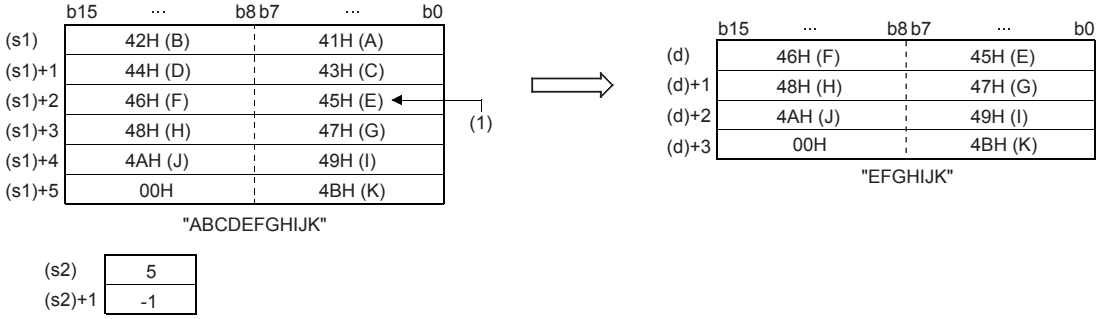
- These instructions extract the number of characters specified by (s2)+1 of the character string data stored in the device specified by (s1) and later from the position specified by (s2), and store the extracted characters in the device specified by (d) and later.



(s2)	5
(s2)+1	5

(1): 5th character position (s2)
(2): ASCII code for 5th character (s2)+1

- A character string stored in (s1) indicates data stored in devices from the specified device until "00H" is first detected in units of byte.
- A NULL code (00H), which indicates an end of a character string, is automatically added at the end of the character string data.
- When the number of extracted characters "(s2)+1" is odd, "00H" is stored in the upper byte of a device storing the last character. When the number of extracted characters "(s2)+1" is even, "0000H" is stored in the device after the last character.
- If the number of characters specified by (s2)+1 is 0, no processing is performed.
- When (s2)+1 (the number of characters to be extracted) is "-1", the entire character string stored in (s1) and later is stored to (d) and later.



(s2)	5
(s2)+1	-1

(1): 5th character position (s2)

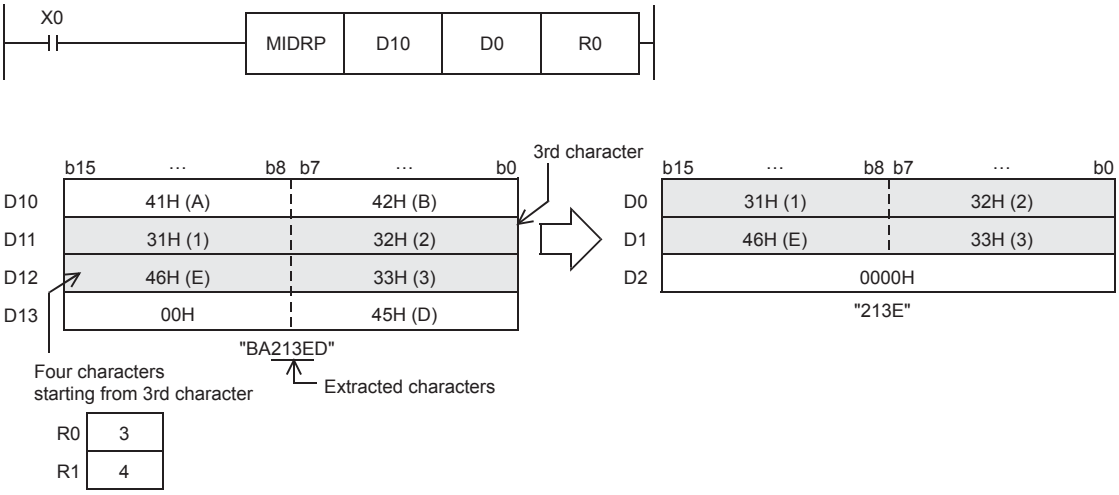
Precautions

When handling character codes other than ASCII codes, note the following points:

- The number of characters is handled in byte units (8 bits). Accordingly, in the case of character codes in which 2 bytes express 1 character such as shift JIS codes, 1 character is detected as "2".
- When extracting characters from a character string including character codes in which 2 bytes express 1 character such as shift JIS codes, consider the number of characters to be extracted in units of character codes for 1 character. Note that the expected character code is not retrieved if only 1 byte is extracted out of a 2-byte character code.

Program example

In the program example shown below, four characters are extracted from the 3rd character from the left end of the character string data stored in D10 and later, and then stored to D0 and later when X0 turns ON.



Operation error

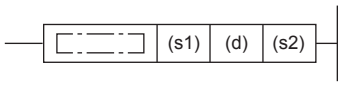
Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s1) and later, "00H" does not exist.
3405H	The value stored in a device specified in (s2)+1 is -2 or lower.
	The value stored in a device specified in (s2) exceeds the number of characters of (s1).
	A negative value is specified in (s2).
	The value stored in a device specified in (s2)+1 exceeds the number of characters of (s1).
	The character string specified by (s1) has more than 16383 characters.
	The total of the values stored in devices specified in (s2) and (s2)+1 exceeds the number of characters of (s1).
3406H	The number of characters from the position specified by (d) to (s2)+1 exceeds the corresponding device range.

Replacing the specified number of character strings

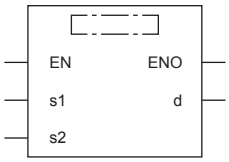
MIDW(P)

FX5S FX5UJ FX5U FX5UC

These instructions extract the number of characters specified by (s2)+1 from the character string data stored in the device specified by (s1) and later, and store the extracted data in the position specified by (s2) and later of the character string data stored in the device specified by (d) and later.

Ladder diagram	Structured text
	ENO:=MIDW(EN,s1,s2,d); ENO:=MIDWP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Character string or head device number storing a character string	—	Character string	ANYSTRING_SINGLE
(d)	Head device number for storing the character string data of the operation result	—	Character string	ANYSTRING_SINGLE
(s2)	Head device number for storing the number of characters and position of the start character (s2)+0: Position of the start character, (s2)+1: Number of characters	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

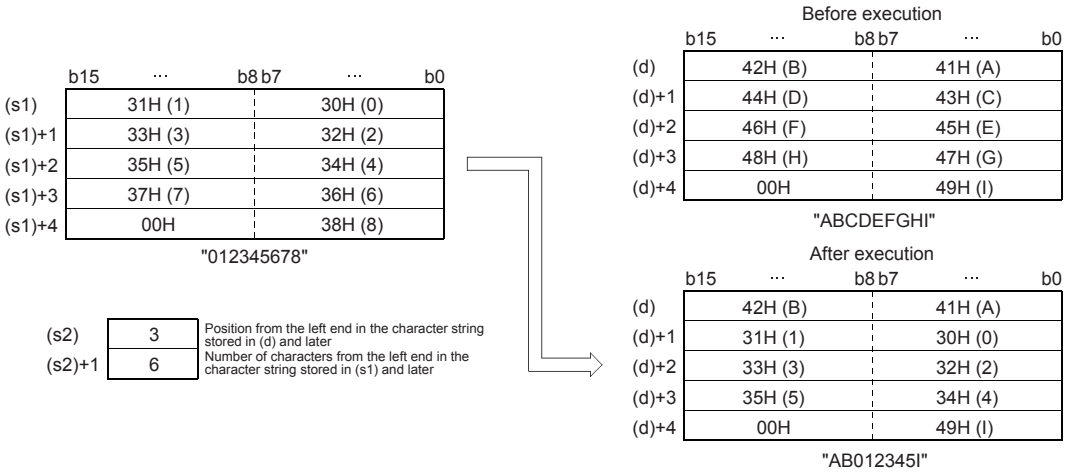
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	—	—	○	—	—	—	—

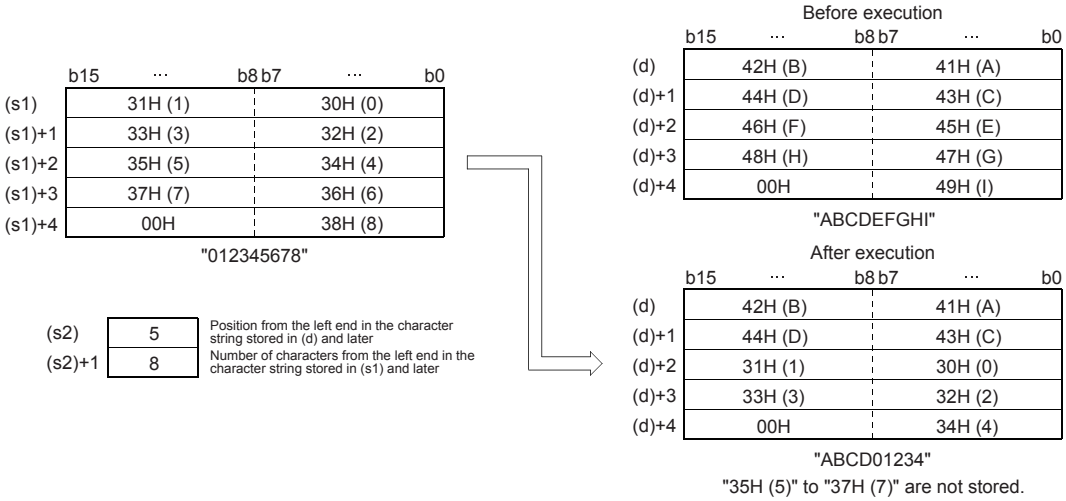
*1 T, ST, and C cannot be used.

Processing details

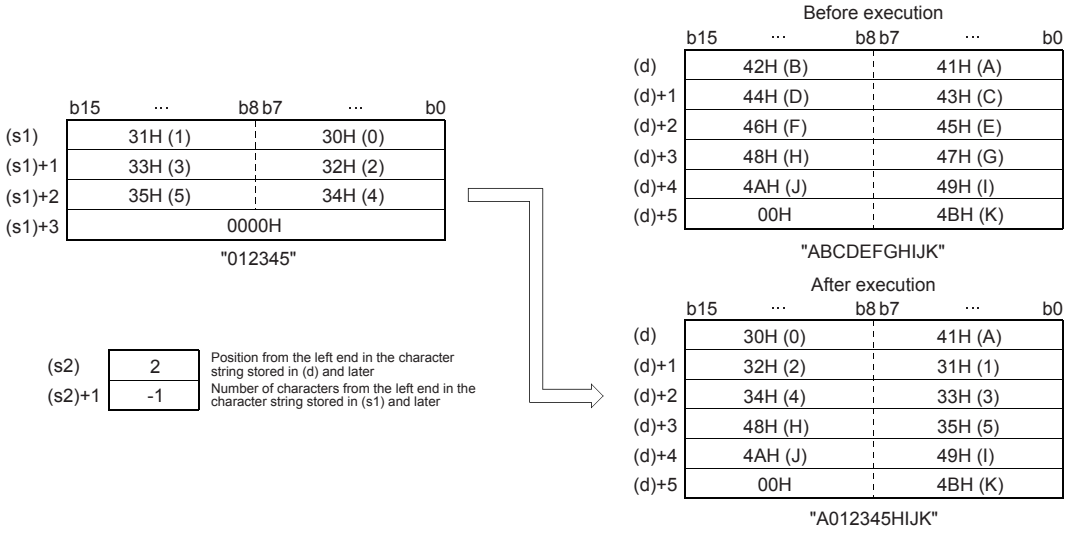
- These instructions extract the number of characters specified by (s2)+1 from the character string data stored in the device specified by (s1) and later, and store the extracted data in the position specified by (s2) and later of the character string data stored in the device specified by (d) and later.



- A character string stored in (s1) or (d) indicates data stored in devices from the specified device until "00H" is first detected in units of 1 byte.
- A NULL code (00H), which indicates an end of a character string, is automatically added at the end of the character string data.
- If the number of characters specified by (s2)+1 is 0, no processing is performed.
- When the number of characters specified by (s2)+1 exceeds the last character of the character string specified by (d), data is stored up to the last character of (d).



- When (s2)+1 (the number of characters to be extracted) is "-1", the entire character string stored in (s1) and later is stored to (d) and later.



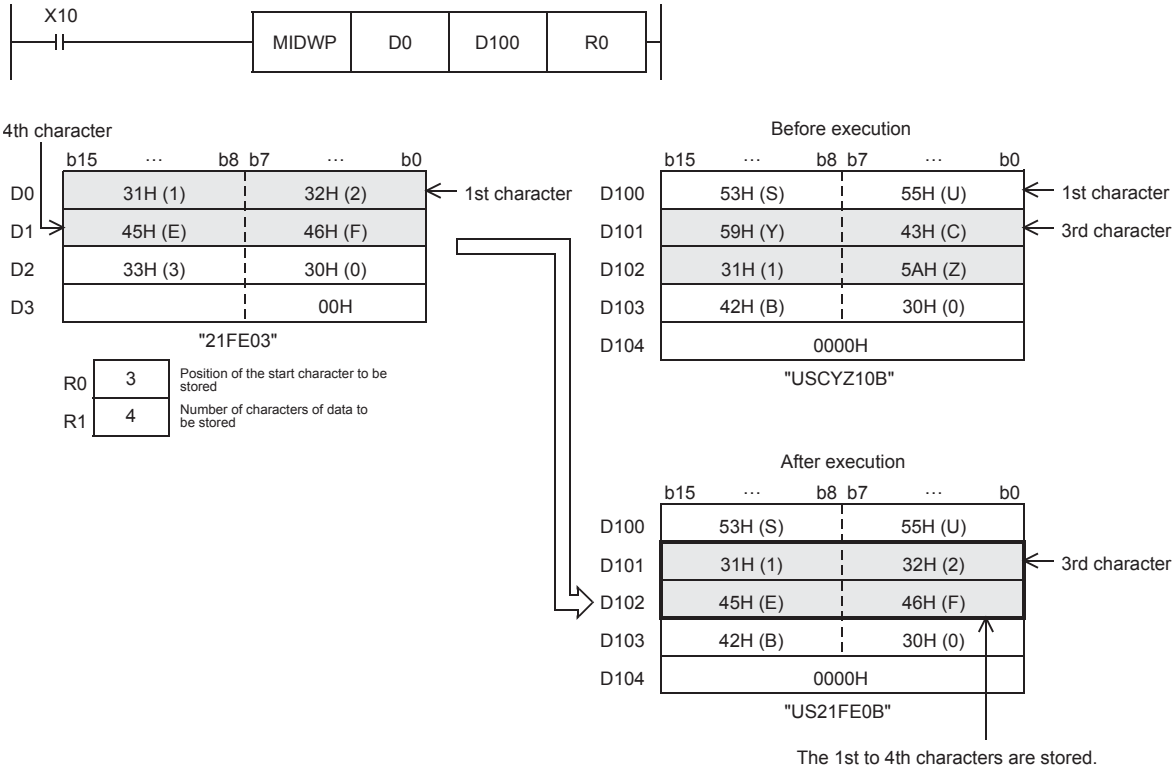
Precautions

When handling character codes other than ASCII codes, note the following points:

- The number of characters is handled in byte units (8 bits). Accordingly, in the case of character codes in which 2 bytes express 1 character such as shift JIS codes, 1 character is detected as "2".
- When extracting characters from a character string including character codes in which 2 bytes express 1 character such as shift JIS codes, consider the number of characters to be extracted in units of character codes for 1 character. Note that the expected character code is not retrieved if only 1 byte is extracted out of a 2-byte character code.

Program example

In the program example shown below, 4 characters are extracted from the character string data stored in D0 and later, and stored to the 3rd character (from the left end) and later for the character string data stored in D100 and later when X10 turns ON.



Operation error

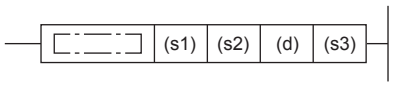
Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s1) and later, "00H" does not exist.
	The device specified by (d) exceeds the corresponding device range.
3405H	The value stored in a device specified in (s2)+1 is -2 or lower.
	The value stored in a device specified in (s2) exceeds the number of characters of (d).
	The value stored in a device specified in (s2)+1 exceeds the number of characters of (s1).
	The character string specified by (s1) has more than 16383 characters.
	The character string specified by (d) has more than 16383 characters.

Searching character string

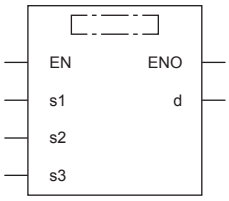
INSTR(P)

FX5S FX5UJ FX5U FX5UC

These instructions search the character string data stored in the device specified by (s2) and later starting from the (s3)th character from the left, for the character string data stored in the device specified by (s1) and later and store the search result in the device specified by (d).

Ladder diagram	Structured text
	ENO:=INSTR(EN,s1,s2,s3,d); ENO:=INSTRP(EN,s1,s2,s3,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Character string to be searched for or head device number storing a character string to be searched for	—	Character string	ANYSTRING_SINGLE
(s2)	Character string to be searched or head device number storing a character string to be searched	—	Character string	ANYSTRING_SINGLE
(d)	Head device number storing search result	—	16-bit signed binary	ANY16
(s3)	Search start position	1 to 16383	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

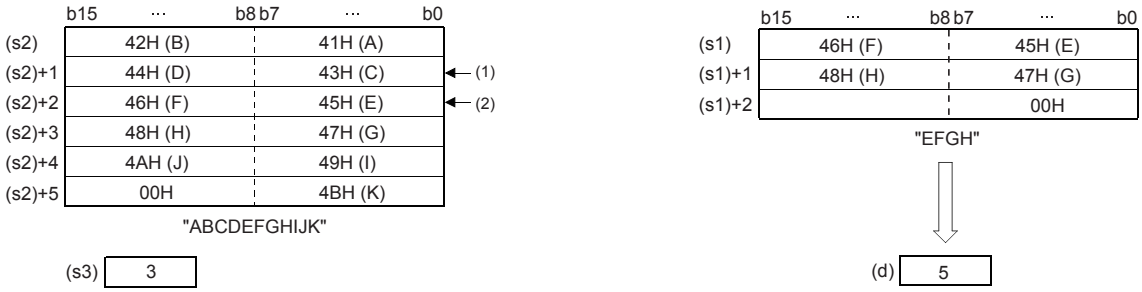
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	—	—	—	—	○	—	—	○	—
(s2)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(s3)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

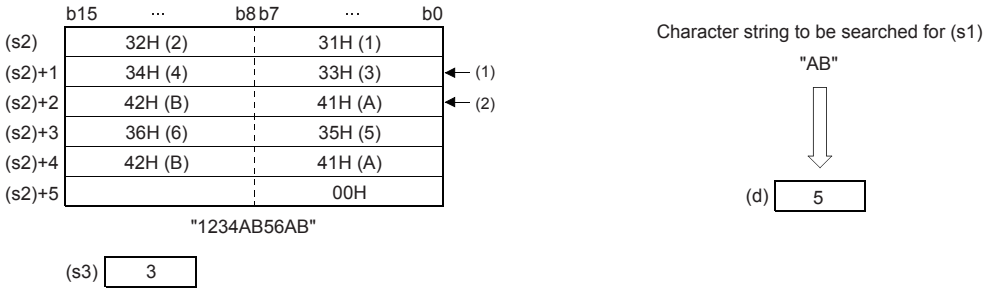
Processing details

- These instructions search the character string data stored in the device specified by (s2) and later starting from the (s3)th character from the left, for the character string data stored in the device specified by (s1) and later and store the search result in the device specified by (d). The search result stores the position where the first detected character is located from the start character in the character string data stored in (s2).



(1): Search start position (s3): 3rd character
(2): Fifth character from the head character

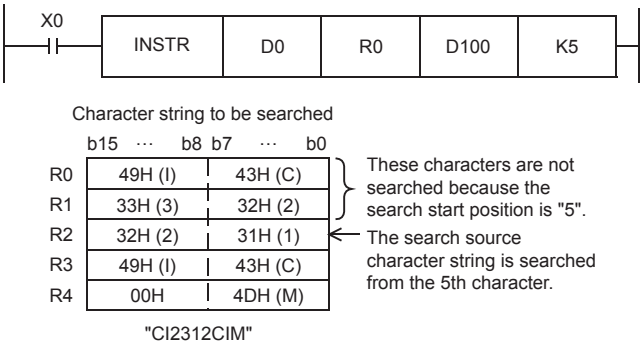
- If no matched character string data is found, 0 is stored in (d).
- When the search start position "s3" is a negative number or "0", search processing is not executed.
- A character string can be directly specified in the character string (s1).

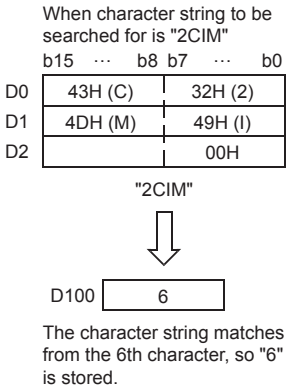


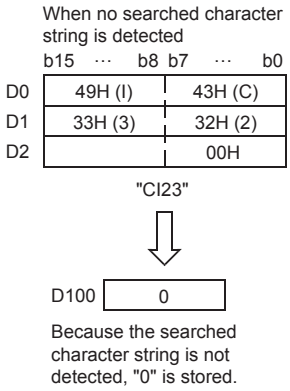
(1): Search start position (s3): 3rd character
(2): Fifth character from the head character

Program example

In the program shown below, the character string stored in D0 and later is searched from the 5th character of the character string "CI2312CIM" stored in R0 and later when X0 is set to ON. The search results are stored in D100.







Operation error

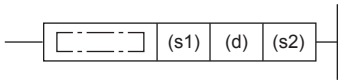
Error code (SD0/SD8067)	Description
2820H	No NULL code (00H) exists in the corresponding device range of the device specified by (s1) and later.
	No NULL code (00H) exists in the corresponding device range of the device specified by (s2) and later.
3405H	The value stored in a device specified in (s3) exceeds the number of characters of (s2).
	The character string specified by (s1) has more than 16383 characters.
	The character string specified by (s2) has more than 16383 characters.

Inserting character string

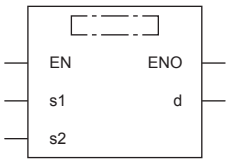
STRINS(P)

FX5S FX5UJ FX5U FX5UC

These instructions insert the character string specified by (s1) at the (s2)th character from the start of the character string specified by (d).

Ladder diagram	Structured text
	ENO:=STRINS(EN,s1,s2,d); ENO:=STRINSP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Character string to be inserted or head device number storing the character string to be inserted	—	Character string	ANYSTRING_SINGLE
(d)	Head device number storing a character string to which another character string is inserted	—	Character string	ANYSTRING_SINGLE
(s2)	Insertion position (in units of bytes)	1 to 16383	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

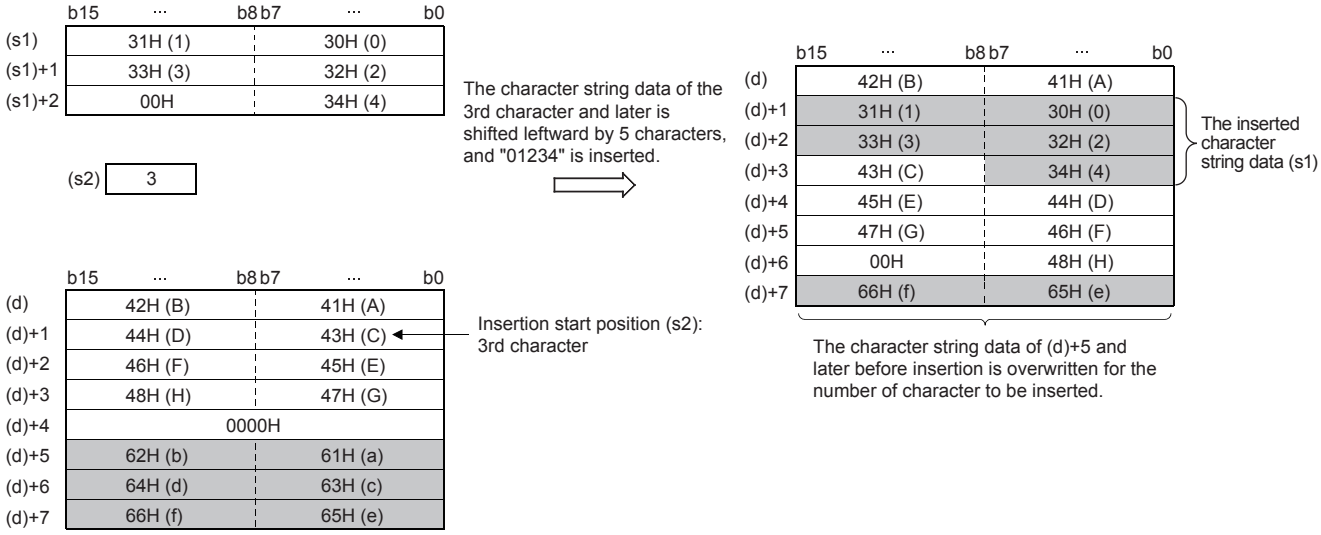
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions insert the character string specified by (s1) at the (s2)th character from the start of the character string specified by (d).



- When the number of characters after insertion, (s1)+(d), is even, a NULL code (00H) is stored in the device (1 word) after the last device storing the character string.
- When the number of characters after insertion, (s1)+(d), is odd, a NULL code (00H) is stored in the last device (upper 8 bits) of the character string.
- If the number of characters exceeding (d) by one character is specified in (s2), the character string in (s1) is added to the end of the character string in (d).

Operation error

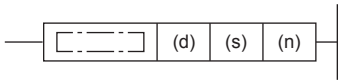
Error code (SD0/SD8067)	Description
2820H	No NULL code (00H) exists in the corresponding device range of the device specified by (s1) and later.
	No NULL code (00H) exists in the corresponding device range of the device specified by (d) and later.
2821H	A device of the character strings (s1) and (d) overlaps.
	The device storing the character string after insertion, (s1)+(d), overlaps with the character string-storing device of (s1).
3405H	The character string specified by (s1) has more than 16383 characters.
	The character string specified by (d) has more than 16383 characters.
	(s2) is not within the range (1≤(s2)≤16383)
	The value specified by (s2) exceeds "the number of characters of the character string (d) + 1".
3406H	The character string after insertion, (s1)+(d), has more than 16383 characters.
	The character string after insertion, (s1)+(d), exceeds the corresponding device range.

Deleting character string

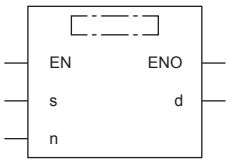
STRDEL(P)

FX5S FX5UJ FX5U FX5UC

These instructions delete (n) characters starting from the (s)th character (deletion start position) from the start of the character string data specified by (d).

Ladder diagram	Structured text
	ENO:=STRDEL(EN,s,n,d); ENO:=STRDELP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number storing a character string having characters to be deleted	—	Character string	ANYSTRING_SINGLE
(s)	Deletion start position	1 to 16383	16-bit signed binary	ANY16
(n)	Number of characters to be deleted	0 to 16384	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

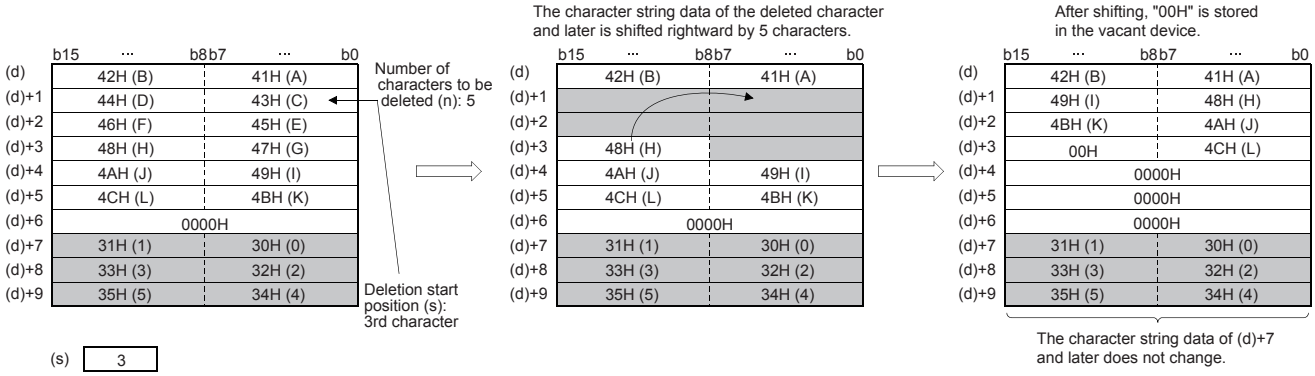
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○*1	—	—	—	—	○	—	—	—	—
(s)	—	○	○	○	—	—	○	○	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions delete (n) characters starting from the (s)th character (deletion start position) from the start of the character string data specified by (d).



- When the number of characters after deletion, (d), is even, a NULL code (00H) is stored in the device after the last device storing the character string.
- When the number of characters after deletion, (d), is odd, a NULL code (00H) is stored in the last device (upper 8 bits) of the character string.
- The character string after the deleted character string is shifted by (n) characters, a NULL code (00H) is stored in vacant devices.

Operation error

Error code (SD0/SD8067)	Description
2820H	No NULL code (00H) exists in the corresponding device range of the device specified by (d) and later.
3405H	The character string specified by (d) has more than 16383 characters.
	(s) is not within the range (1≤(s)≤16383)
	The value specified by (s) exceeds the number of characters of the character string (d).
	The value specified by (n) exceeds the number of characters from (s) to the last of the character string (d).

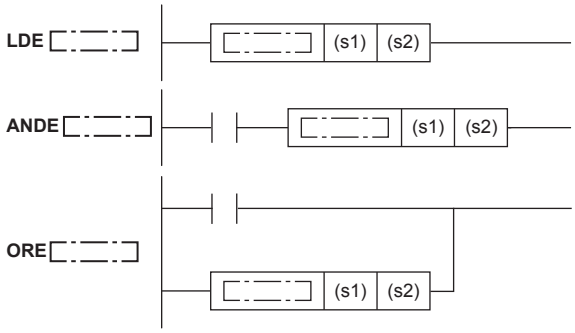
8.10 Real Number Instruction

Comparing single-precision real numbers

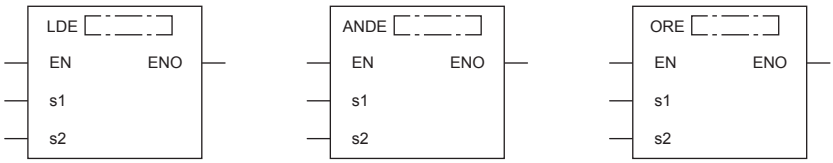
LDE□, ANDE□, ORE□

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation between the single-precision real number in the device specified by (s1) and the single-precision real number in the device specified by (s2). (Devices are used as a normally open contact.)

Ladder diagram	Structured text*1
 ("=", "<>", ">", "<=", "<", ">=" enters □.)	ENO:=LDE_□(EN,s1,s2); ENO:=ANDE_□(EN,s1,s2); ENO:=ORE_□(EN,s1,s2); ("EQ", "NE", "GT", "LE", "LT", "GE" enters □.)*2

FBD/LD



("_EQ", "_NE", "_GT", "_LE", "_LT", "_GE" enters □.)*2

*1 Supported by engineering tool version "1.035M" and later.

*2 EQ is =, NE is <>, GT is >, LE is <=, LT is <, and GE is >=.

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Comparison data or the head device number where the comparison data is stored	0, 2^{-126}<|(s1)|<2^{128}	Single-precision real number	ANYREAL_32
(s2)	Comparison data or the head device number where the comparison data is stored	0, 2^{-126}<|(s2)|<2^{128}	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	—	○	—	—
(s2)	—	○	○	—	○	—	○	—	○	—	—

Processing details

- These instructions perform a comparison operation between the single-precision real number in the device specified by (s1) and the single-precision real number in the device specified by (s2). (Devices are used as a normally open contact.)
- The following table lists the comparison operation results of each instruction.

Instruction symbol	Condition	Result	Instruction symbol	Condition	Result
E=	(s1)=(s2)	Conductive state	E=	(s1)≠(s2)	Non-conductive state
E<>	(s1)≠(s2)		E<>	(s1)=(s2)	
E>	(s1)>(s2)		E>	(s1)≤(s2)	
E<=	(s1)≤(s2)		E<=	(s1)>(s2)	
E<	(s1)<(s2)		E<	(s1)≥(s2)	
E>=	(s1)≥(s2)		E>=	(s1)<(s2)	

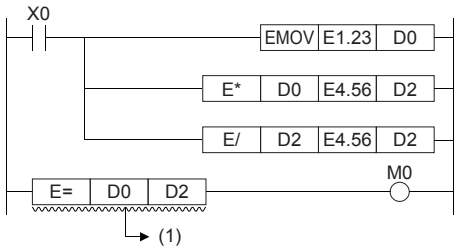
- When an input value is set from the engineering tool, a rounding error may occur.

Operation error

There is no operation error.

Point

When the E= instruction is used, note that values in the devices may not be equal.



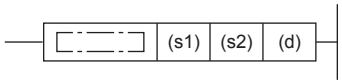
(1): Values in the devices may not be equal.

Single-precision real number comparison

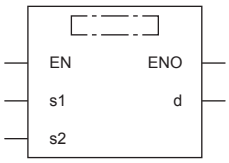
DECMP(P)

FX5S FX5UJ FX5U FX5UC

These instructions compare two data values (single-precision real number), and output the result (larger, same or smaller) to three consecutive bit devices.

Ladder diagram	Structured text
	ENO:=DECMP(EN,s1,s2,d); ENO:=DECMPP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Comparison data or the number of the device where the comparison data is stored	0, $2^{-126} \leq \lvert(s1)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Comparison data or the number of the device where the comparison data is stored	0, $2^{-126} \leq \lvert(s2)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Start bit device number to which comparison result is output (Three devices are occupied).	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

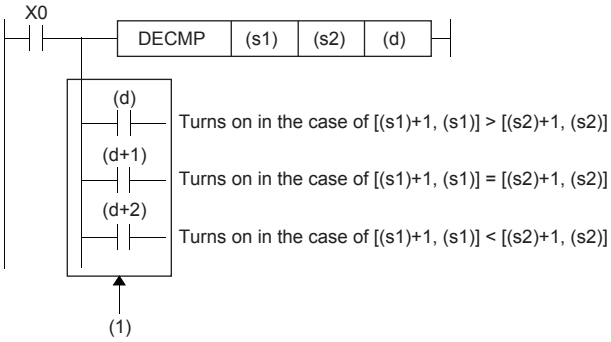
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	○	○	—	—
(s2)	—	○	○	—	○	—	○	○	○	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions compare the comparison value (s1) with the comparison source (s2) as floating point data, and one of the bits among (d), (d)+1, and (d)+2 turns on according to the result (smaller, same or larger).



(1): Even if the command input X0 turns off before the DECMP instruction is fully executed, (d) to (d)+2 hold the status.

- When the constant (K or H) is specified the device specified by (s1) and (s2), these instructions convert the binary value into single-precision real number automatically.

Precautions

- Three devices ((d), (d)+1, and (d)+2) specified by (d) are occupied. Note that these devices are not used for any other purpose.

Operation error

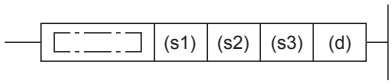
Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
3402H	The specified device value is denormalized number, NaN (not a number), or ±∞.

Single-precision real number data band comparison

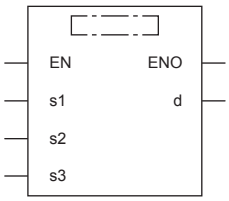
DEZCP(P)

FX5S FX5UJ FX5U FX5UC

These instructions compare the comparison range of two points, upper and lower, with the binary floating point, and output the result to three consecutive bit devices in accordance with the larger, smaller, and band.

Ladder diagram	Structured text
	ENO:=DEZCP (EN, s1, s2, s3, d); ENO:=DEZCPP(EN, s1, s2, s3, d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Comparison data or the number of the device where the comparison data is stored	0, $2^{-126} \leq \lvert(s1)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Comparison data or the number of the device where the comparison data is stored	0, $2^{-126} \leq \lvert(s2)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(s3)	Comparison data or the number of the device where the comparison data is stored	0, $2^{-126} \leq \lvert(s3)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Start bit device number to which comparison result is output (Three devices are occupied).	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

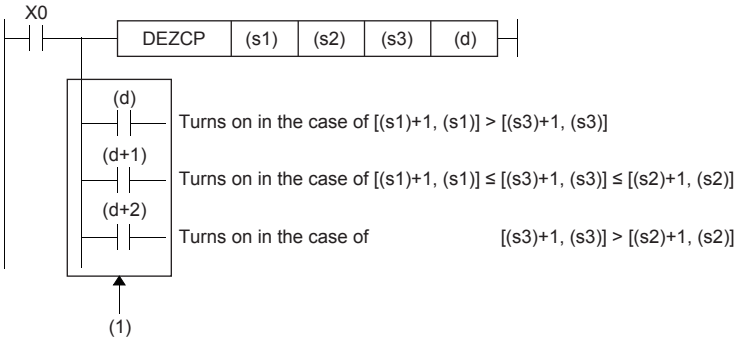
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	○	○	—	—
(s2)	—	○	○	—	○	—	○	○	○	—	—
(s3)	—	○	○	—	○	—	○	○	○	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions compare the comparison values (s1) and (s2) with the comparison source (s3) as floating point data, and one of the bits among (d), (d)+1, and (d)+2 turns on according to the result (smaller, within the range or larger).



(1): Even if the command input X0 turns off before the DEZCP instruction is fully executed, (d) to (d)+2 hold the status.

- When the constant (K or H) is specified the device specified by (s1), (s2) and (s3), these instructions convert the binary value into single-precision real number automatically.

Precautions

- Three devices ((d), (d)+1, and (d)+2) specified by (d) are occupied. Note that these devices are not used for any other purpose.
- The size relationship of the comparison data should be [(s1)+1, (s1)] ≤ [(s2)+1, (s2)]. If the relationship is [(s1)+1, (s1)] > [(s2)+1, (s2)], the value of [(s2)+1, (s2)] is regarded as the same as that of [(s1)+1, (s1)], and is compared.

Operation error

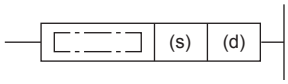
Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
3402H	The specified device value is denormalized number, NaN (not a number), or ±∞.

Adding single-precision real numbers

E+(P) [For 2 operands]

FX5S FX5UJ FX5U FX5UC

These instructions add the single-precision real number in the device specified by (s) to the single-precision real number in the device specified by (d), and store the result in the device specified by (d).

Ladder diagram	Structured text
	Not supported. (☞ Page 667 E+(P) [For 3 operands])

FBD/LD
Not supported. (☞ Page 667 E+(P) [For 3 operands])

Setting data

■Descriptions, ranges, and data types

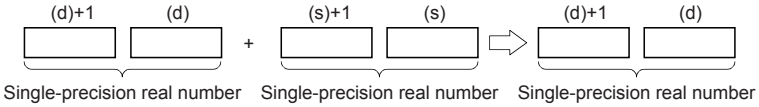
Operand	Description	Range	Data type	Data type (label)
(s)	Addend data or the head device number where the data that is added to another is stored	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Head device number where the data to which another is added is stored	—	Single-precision real number	ANYREAL_32

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions add the single-precision real number in the device specified by (s) to the single-precision real number in the device specified by (d), and store the result in the device specified by (d).



- Values in the devices specified (stored) by (s) and (d) should be 0 or $2^{-126} \leq |\text{specified value (stored value)}| < 2^{128}$.
- When an input value is set from the engineering tool, a rounding error may occur.
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result $\geq 2^{128}$	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result $< 2^{-126}$	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result $\geq 2^{128}$	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

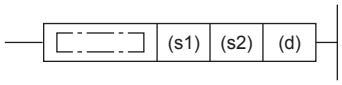
Operation error

Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or $\pm\infty$.
	The value stored in specified device is outside the following range 0, $2^{-126} \leq |\text{Specified device value}| < 2^{128}$

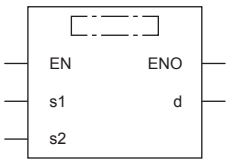
E+(P) [For 3 operands]

FX5S FX5UJ FX5U FX5UC

These instructions add the single-precision real number in the device specified by (s2) to the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).

Ladder diagram	Structured text
	ENO:=EPLUS(EN, s1, s2, d); ENO:=EPLUSP(EN, s1, s2, d);

FBD/LD



("EPLUS", "EPLUSP" enters □.)

Setting data

■Descriptions, ranges, and data types

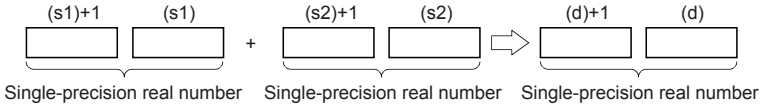
Operand	Description	Range	Data type	Data type (label)
(s1)	Augend data or the head device number where the data to which another is added is stored	0, $2^{-126} \leq \lvert(s1)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Addend data or the head device number where the data that is added to another is stored	0, $2^{-126} \leq \lvert(s2)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	—	○	—	—
(s2)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions add the single-precision real number in the device specified by (s2) to the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).



- Values in the devices specified (stored) by (s1), (s2), and (d) should be 0 or 2^{-126} |specified value (stored value)|$\leq 2^{128}$.
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result $\geq 2^{128}$	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result $\geq 2^{128}$	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Operation error

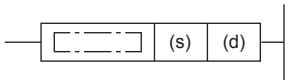
Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.
	The value stored in specified device is outside the following range 0, $2^{-126} \leq$ |specified device value| $< 2^{128}$

Subtracting single-precision real numbers

E-(P) [For 2 operands]

FX5S FX5UJ FX5U FX5UC

These instructions subtract the single-precision real number in the device specified by (s) from the single-precision real number in the device specified by (d), and store the result in the device specified by (d).

Ladder diagram	Structured text
	Not supported. (☞ Page 671 E-(P) [For 3 operands])

FBD/LD
Not supported. (☞ Page 671 E-(P) [For 3 operands])

Setting data

■Descriptions, ranges, and data types

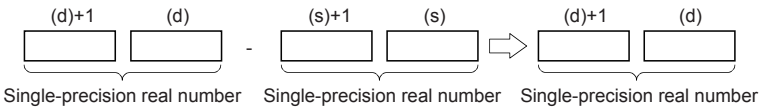
Operand	Description	Range	Data type	Data type (label)
(s)	Subtrahend data or the head device number where the data to be subtracted from another is stored	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Head device number where the data from which another is to be subtracted is stored	—	Single-precision real number	ANYREAL_32

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions subtract the single-precision real number in the device specified by (s) from the single-precision real number in the device specified by (d), and store the result in the device specified by (d).



- Values in the devices specified (stored) by (s) and (d) should be 0 or $2^{-126} \leq$ |specified value (stored value)| $< 2^{128}$.
- When an input value is set from the engineering tool, a rounding error may be occur.
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result $\geq 2^{128}$	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result $< 2^{-126}$	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result $\geq 2^{128}$	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

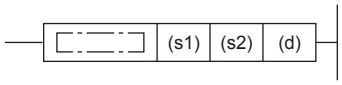
Operation error

Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.
	The value stored in specified device is outside the following range 0, $2^{-126} \leq$ |specified device value| $< 2^{128}$

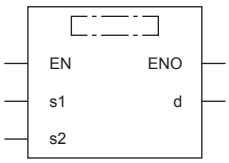
E-(P) [For 3 operands]

FX5S FX5UJ FX5U FX5UC

These instructions subtract the single-precision real number in the device specified by (s2) from the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).

Ladder diagram	Structured text
	ENO:=EMINUS(EN, s1, s2, d); ENO:=EMINUSP(EN, s1, s2, d);

FBD/LD



("EMINUS", "EMINUSP" enters □.)

Setting data

■Descriptions, ranges, and data types

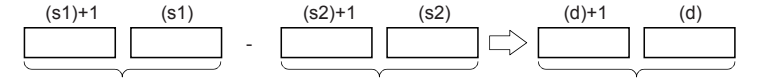
Operand	Description	Range	Data type	Data type (label)
(s1)	Minuend data or head device number where the data from which another is to be subtracted is stored	0, $2^{-126} \leq \lvert(s1)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Subtrahend data or head device number where the data to be subtracted from another is stored	0, $2^{-126} \leq \lvert(s2)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	—	○	—	—
(s2)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions subtract the single-precision real number in the device specified by (s2) from the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).



Single-precision real number Single-precision real number Single-precision real number

- Values in the devices specified (stored) by (s1), (s2), and (d) should be 0 or 2^{-126} |specified value (stored value)|$\leq 2^{128}$.
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result $\geq 2^{128}$	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result $\geq 2^{128}$	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Operation error

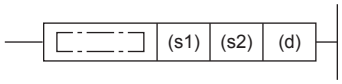
Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.
	The value stored in specified device is outside the following range 0, $2^{-126} \leq$ |specified device value| $< 2^{128}$

Adding single-precision real numbers

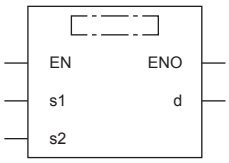
DEADD(P)

FX5S FX5UJ FX5U FX5UC

These instructions add the single-precision real number in the device specified by (s2) to the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DEADD(EN,s1,s2,d); ENO:=DEADDP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

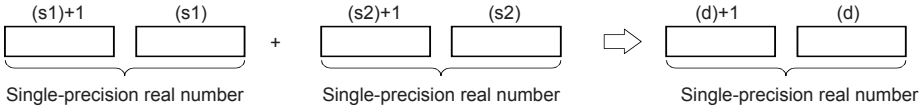
Operand	Description	Range	Data type	Data type (label)
(s1)	Augend data or head device number where the data to which another is added is stored	0, $2^{-126} \leq \lvert(s1)\rvert \leq 2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Addend data or head device number where the data that is added to another is stored	0, $2^{-126} \leq \lvert(s2)\rvert \leq 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

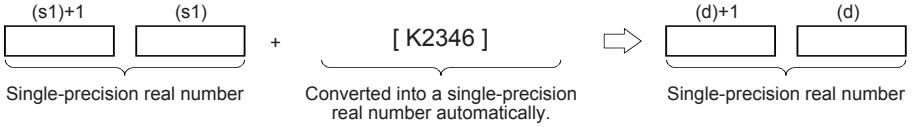
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	○	○	—	—
(s2)	—	○	○	—	○	—	○	○	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions add the single-precision real number in the device specified by (s2) to the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).



- When the constant (K or H) is specified in (s1) and (s2), these instructions convert values into single-precision real number automatically.



- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Precautions

The same device number can be specified for (s1), (s2), and (d). In this case, note that the addition result changes in every operation cycle when the continuous operation type instruction (DEADD) is used.

Operation error

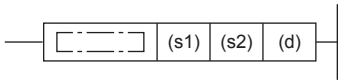
Error code (SD0/SD8067)	Description
3402H	The specified device value is denormalized number, NaN (not a number), or ±∞.

Subtracting single-precision real numbers

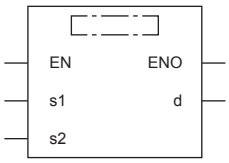
DESUB(P)

FX5S FX5UJ FX5U FX5UC

These instructions subtract the single-precision real number in the device specified by (s2) from the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DESUB(EN,s1,s2,d); ENO:=DESUBP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

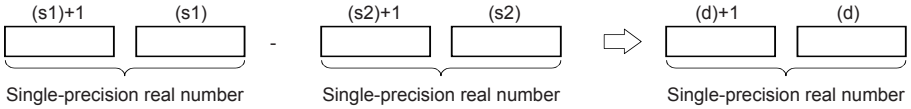
Operand	Description	Range	Data type	Data type (label)
(s1)	Minuend data or head device number where the data from which another is subtracted is stored	0, $2^{-126} \leq \lvert(s1)\rvert \leq 2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Minuend data or head device number where the data that is subtracted another is stored	0, $2^{-126} \leq \lvert(s2)\rvert \leq 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

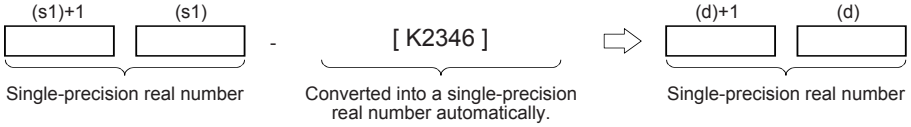
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	○	○	—	—
(s2)	—	○	○	—	○	—	○	○	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions subtract the single-precision real number in the device specified by (s2) from the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).



- When the constant (K or H) is specified in (s1) and (s2), these instructions convert values into single-precision real number automatically.



- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Precautions

The same device number can be specified for (s1), (s2), and (d). In this case, note that the subtraction result changes in every operation cycle when the continuous operation type instruction (DESUB) is used.

Operation error

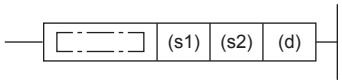
Error code (SD0/SD8067)	Description
3402H	The specified device value is denormalized number, NaN (not a number), or ±∞.

Multiplying single-precision real numbers

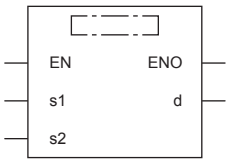
E*(P)

FX5S FX5UJ FX5U FX5UC

These instructions multiply the single-precision real number in the device specified by (s2) to the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=EMULTI(EN,s1,s2,d); ENO:=EMULTIP(EN,s1,s2,d);

FBD/LD



("EMULTI", "EMULTIP" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

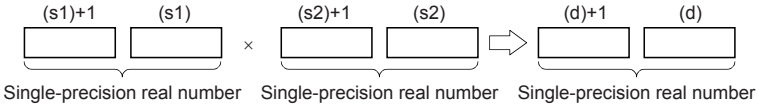
Operand	Description	Range	Data type	Data type (label)
(s1)	Multiplicand data or head device number where the data to be multiplied by another is stored	0, $2^{-126} \leq \lvert(s1)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Multiplier data or head device number where the data by which another is to be multiplied is stored	0, $2^{-126} \leq \lvert(s2)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	—	○	—	—
(s2)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions multiply the single-precision real number in the device specified by (s2) to the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).



- Values in the devices specified (stored) by (s1), (s2), and (d) should be 0 or 2^{-126} |specified value (stored value)|$\leq 2^{128}$.
- When an input value is set from the engineering tool, a rounding error may occur.
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result $\geq 2^{128}$	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result $\geq 2^{128}$	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Operation error

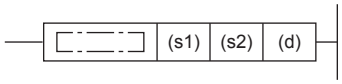
Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.
	The value stored in specified device is outside the following range 0, $2^{-126} \leq$ |specified device value| $< 2^{128}$

Dividing single-precision real numbers

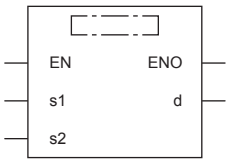
E/(P)

FX5S FX5UJ FX5U FX5UC

These instructions divide the single-precision real number in the device specified by (s1) by the single-precision real number in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=EDIVISION(EN,s1,s2,d); ENO:=EDIVISIONP(EN,s1,s2,d);

FBD/LD



("EDIVISION", "EDIVISIONP" enters □.)

*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

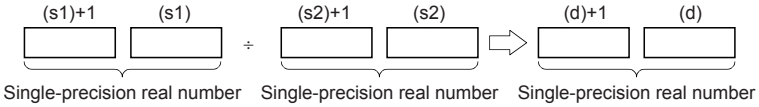
Operand	Description	Range	Data type	Data type (label)
(s1)	Dividend data or head device number where the data which is divided by another is stored.	0, $2^{-126} \leq \lvert(s1)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Divisor data or head device number where the data that divides another is stored.	0, $2^{-126} \leq \lvert(s2)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	—	○	—	—
(s2)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions divide the single-precision real number in the device specified by (s1) by the single-precision real number in the device specified by (s2), and store the result in the device specified by (d).



- Values in the devices specified (stored) by (s1), (s2), and (d) should be 0 or 2^{-126}<|specified value (stored value)|≤2^{128}.
- When an input value is set from the engineering tool, a rounding error may occur.
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Operation error

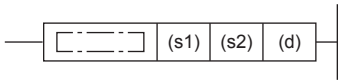
Error code (SD0/SD8067)	Description
3400H	The divisor is 0.
3402H	The specified device value is denormalized number, NaN (not a number), or ±∞.
	The value stored in specified device is outside the following range 0, 2^{-126}≤|specified device value|<2^{128}

Multiplying single-precision real numbers

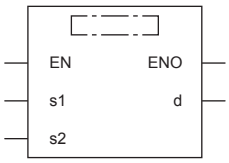
DEMUL(P)

FX5S FX5UJ FX5U FX5UC

These instructions multiply the single-precision real number in the device specified by (s2) to the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DEMUL(EN,s1,s2,d); ENO:=DEMULP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

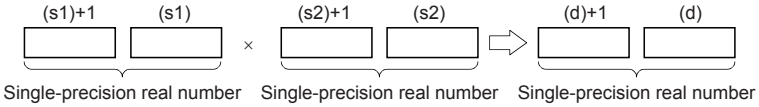
Operand	Description	Range	Data type	Data type (label)
(s1)	Multiplicand data or head device number where the data to be multiplied by another is stored	0, $2^{-126} \leq \lvert(s1)\rvert \leq 2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Multiplier data or head device number where the data by which another is to be multiplied is stored	0, $2^{-126} \leq \lvert(s2)\rvert \leq 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

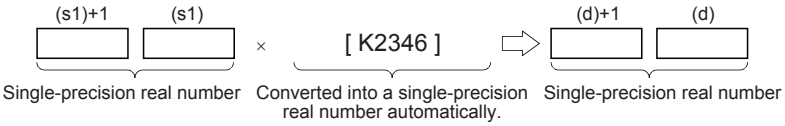
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	○	○	—	—
(s2)	—	○	○	—	○	—	○	○	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions multiply the single-precision real number in the device specified by (s2) to the single-precision real number in the device specified by (s1), and store the result in the device specified by (d).



- When the constant (K or H) is specified in (s1) and (s2), these instructions convert values into single-precision real number automatically.



- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Operation error

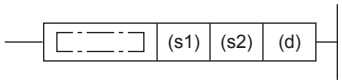
Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.
	The value stored in specified device is outside the following range 0, $2^{-126} \leq$ |specified device value| $< 2^{128}$

Dividing single-precision real numbers

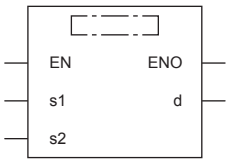
DEDIV(P)

FX5S FX5UJ FX5U FX5UC

These instructions divide the single-precision real number in the device specified by (s1) by the single-precision real number in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DEDIV(EN,s1,s2,d); ENO:=DEDIVP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

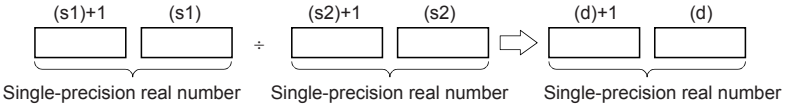
Operand	Description	Range	Data type	Data type (label)
(s1)	Dividend data or head device number where the data which is divided by another is stored.	—	Single-precision real number	ANYREAL_32
(s2)	Divisor data or head device number where the data that divides another is stored.	—	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

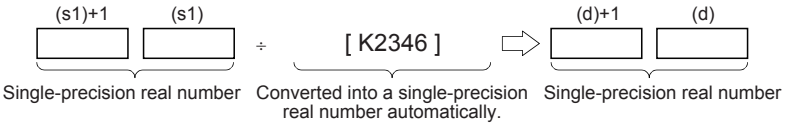
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	○	○	—	—
(s2)	—	○	○	—	○	—	○	○	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions divide the single-precision real number in the device specified by (s1) by the single-precision real number in the device specified by (s2), and store the result in the device specified by (d).



- When the constant (K or H) is specified in (s1) and (s2), these instructions convert values into single-precision real number automatically.



- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Operation error

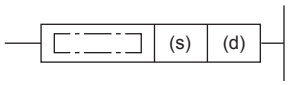
Error code (SD0/SD8067)	Description
3400H	The divisor is 0.
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.
	The value stored in specified device is outside the following range 0, $2^{-126} \leq$ |specified device value| $< 2^{128}$

Converting 16-bit signed binary data to single-precision real number

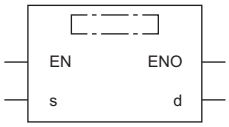
INT2FLT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit signed binary data in the device specified by (s) to single-precision real number, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=INT2FLT(EN,s,d); ENO:=INT2FLTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

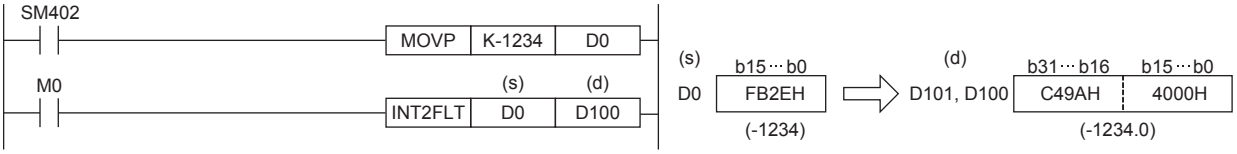
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	-32768 to +32767	16-bit signed binary	ANY16_S
(d)	Data after conversion	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions convert the 16-bit signed binary data in the device specified by (s) to single-precision real number, and store the converted data in the device specified by (d).



Operation error

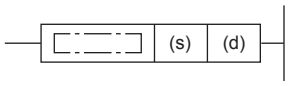
There is no operation error.

Converting 16-bit unsigned binary data to single-precision real number

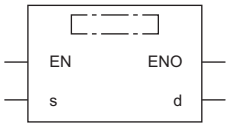
UINT2FLT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit unsigned binary data in the device specified by (s) to single-precision real number, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=UINT2FLT(EN,s,d); ENO:=UINT2FLTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

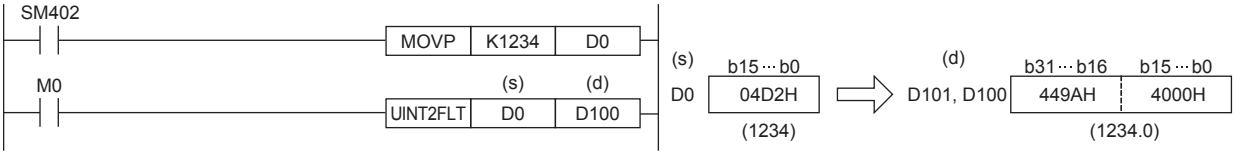
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	0 to 65535	16-bit unsigned binary	ANY32_U
(d)	Data after conversion	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions convert the 16-bit unsigned binary data in the device specified by (s) to single-precision real number, and store the converted data in the device specified by (d).



Operation error

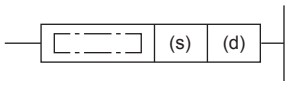
There is no operation error.

Converting 32-bit signed binary data to single-precision real number

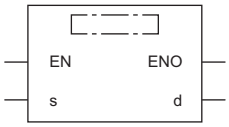
DINT2FLT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit signed binary data in the device specified by (s) to single-precision real number, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=DINT2FLT(EN,s,d); ENO:=DINT2FLTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

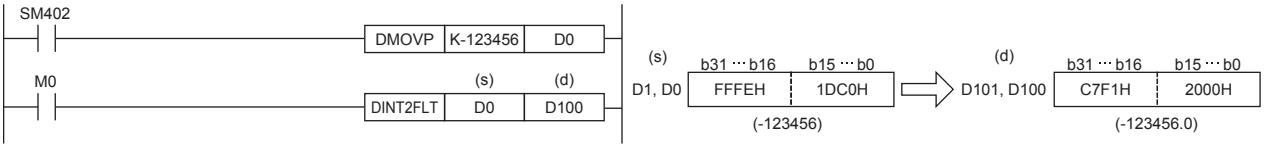
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
(d)	Data after conversion	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions convert the 32-bit signed binary data in the device specified by (s) to single-precision real number, and store the converted data in the device specified by (d).



Operation error

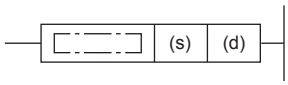
There is no operation error.

Converting 32-bit unsigned binary data to single-precision real number

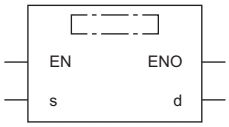
UDINT2FLT(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit unsigned binary data in the device specified by (s) to single-precision real number, and store the converted data in the device specified by (d).

Ladder diagram	Structured text*1
	ENO:=UDINT2FLT(EN,s,d); ENO:=UDINT2FLTP(EN,s,d);

FBD/LD



*1 Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

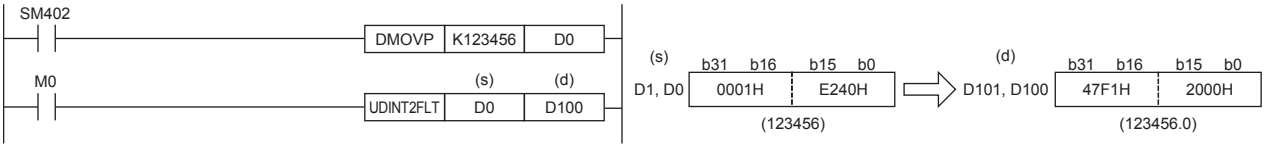
Operand	Description	Range	Data type	Data type (label)
(s)	Data before conversion	0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	Data after conversion	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions convert the 32-bit unsigned binary data in the device specified by (s) to single-precision real number, and store the converted data in the device specified by (d).



Operation error

There is no operation error.

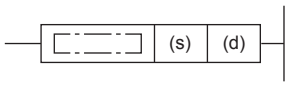
Converting character string to single-precision real number

EVAL(P)/DEVAL(P)

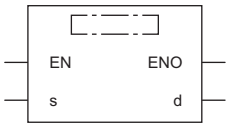
FX5S FX5UJ FX5U FX5UC

These instructions convert the character strings in the device areas specified by (s) and later to single-precision real number, and store the converted data in the device specified by (d).

The EVAL(P) instructions can also be used as DEVAL(P).

Ladder diagram	Structured text
	ENO:=EVAL(EN,s,d); ENO:=EVALP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Character string data to be converted to single-precision real number or head device number where the character string data is stored	—	Character string	ANYSTRING_SINGLE
(d)	Head device number storing converted single-precision real number	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

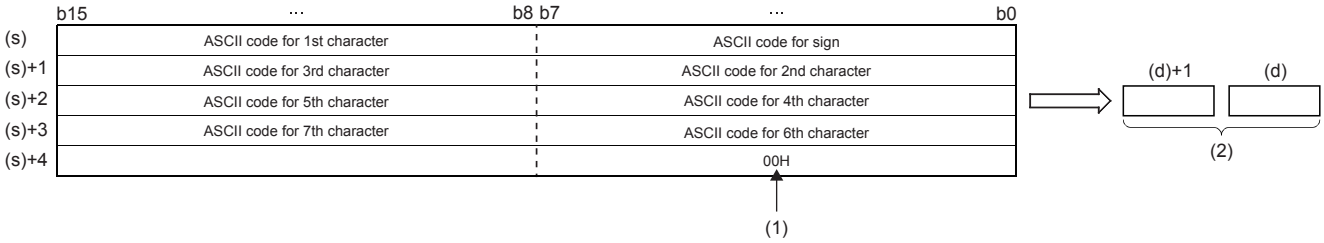
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	○	—
(d)	—	○	○	—	○	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions convert the character strings in the device areas specified by (s) and later to single-precision real number, and store the converted data in the device specified by (d).
- A specified character string may be in the decimal point format or exponent format. A character string in either format can be converted into single-precision real number.

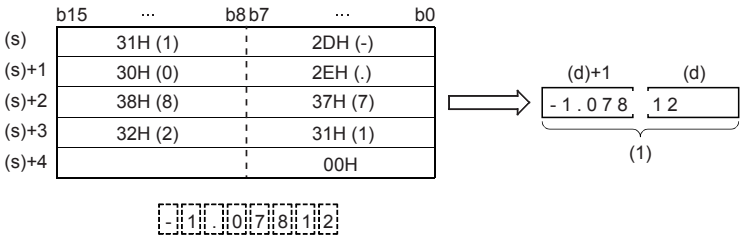


(1): Indicates the end of the character string.
(2): Single-precision real number

- A character string can consist of up to 24 characters. 20H (space) and 30H (0) in a character string are counted as one character each.

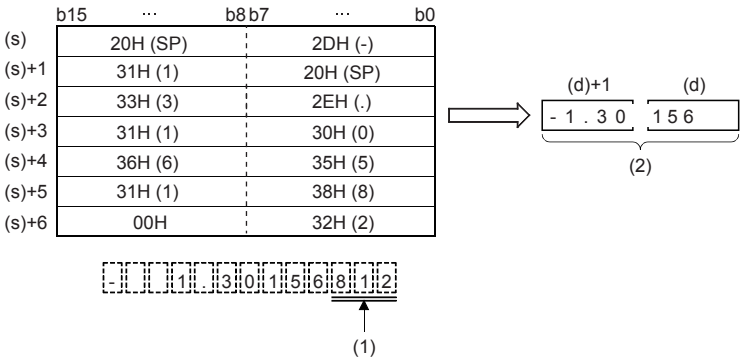
■Decimal point format

• When the character string specified by (s) is decimal point format, the operation is executed as follows.



(1): Single-precision real number

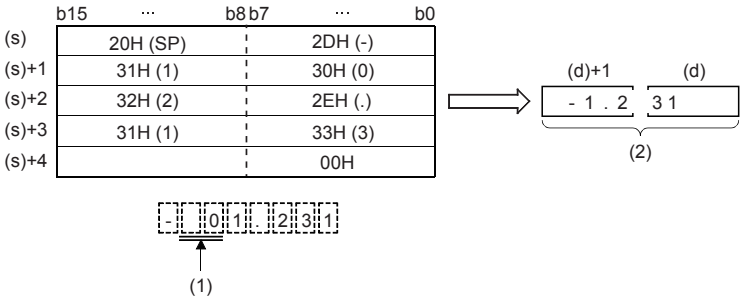
• With regard to character string, six digits excluding the sign, decimal point and exponent part are valid, and the 7th and later digits are discarded during conversion.



(1): These values are discarded.
(2): Single-precision real number

• When 2BH (+) is specified as the sign in the floating point format or when the sign is omitted, a character string is converted into a positive value. It is handled as negative value during conversion when the sign is set to 2DH (-).

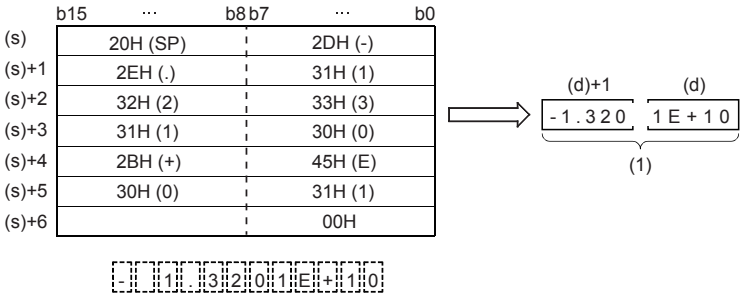
• When 20H (space) or 30H (0) exists between numbers except the first 0 in a character string specified by (s), 20H or 30H is ignored during conversion.



(1): Ignored
(2): Single-precision real number

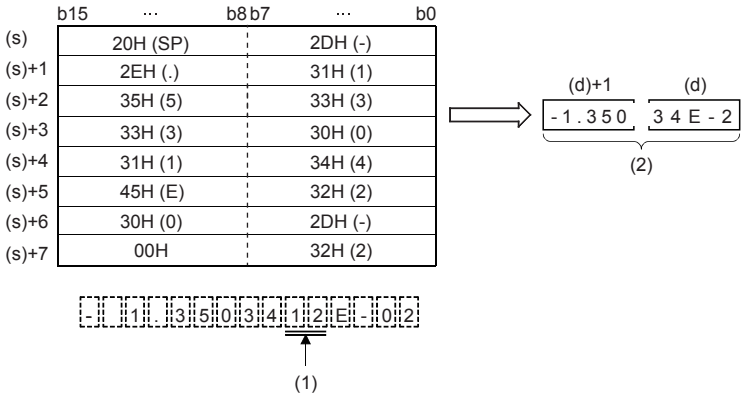
■Exponent format

• When the character string specified by (s) is in exponent format, the operation is executed as follows.



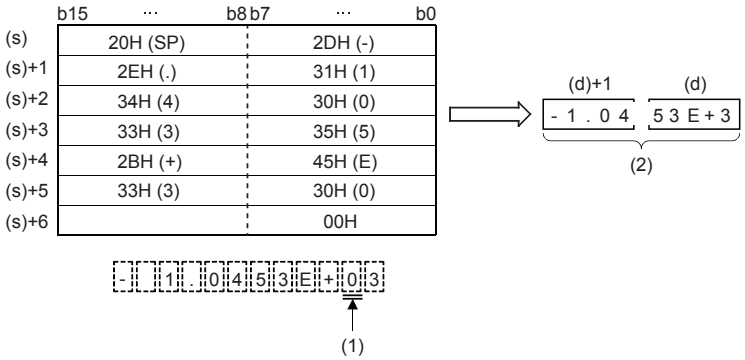
(1): Single-precision real number

- With regard to character string, six digits excluding the sign, decimal point and exponent part are valid, and the 7th and later digits are discarded during conversion.



(1): These values are discarded.
(2): Single-precision real number

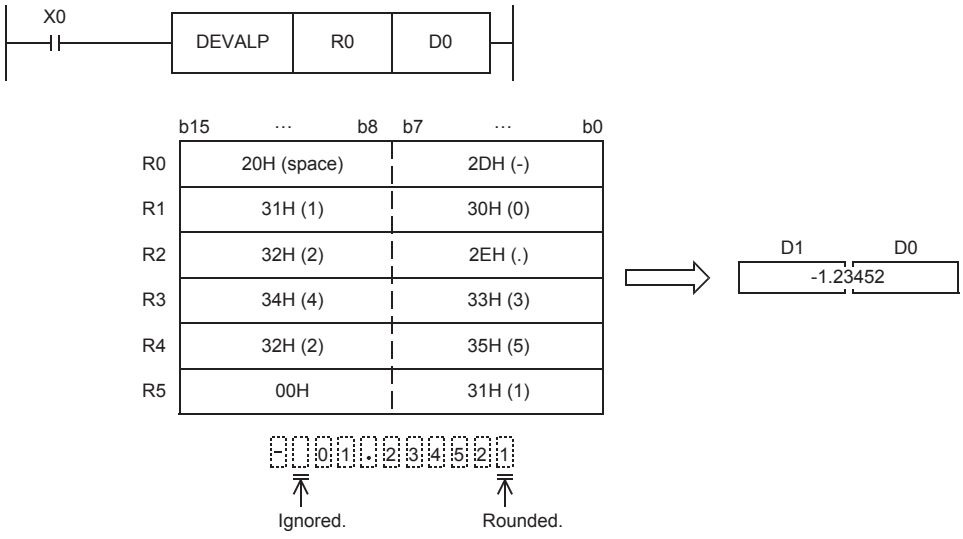
- String data in the exponent format is handled as positive value during conversion when the sign of the exponent part is set to 2BH (+) or when the sign is omitted. When 2DH (-) is specified as the sign, a character string is converted into a negative value.
- When 20H (space) or 30H (0) exists between numbers except the first 0 in a character string specified by (s), 20H or 30H is ignored during conversion.
- When 30H (0) exists between a number and "E" in a character string in the exponent format, 30H is ignored during conversion.



(1): Ignored
(2): Single-precision real number

Program example

In the program example shown below, a character string stored in R0 and later is converted into single-precision real number, and stored to D0 and D1 when X0 turns ON.



Operation error

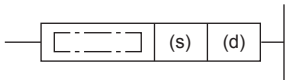
Error code (SD0/SD8067)	Description
2820H	00H does not exist in the corresponding device range starting from (s)
3401H	Characters other than 30 (0) to 39 (9) exist in a character string specified by (s)
	2EH (.) exists in two or more positions in a character string specified by (s)
	Any character other than 45H (E), 2BH (+), or 2DH (-) exists in the exponent part specified by (s), or two or more exponent parts exist
3405H	The number of characters after (s) is 0 or more than 24

Converting binary floating point to decimal floating point

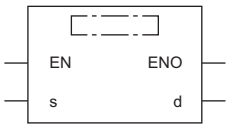
DEBCD(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the binary floating point specified by (s) to decimal floating point, and store the converted data in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DEBCD(EN,s,d); ENO:=DEBCDP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

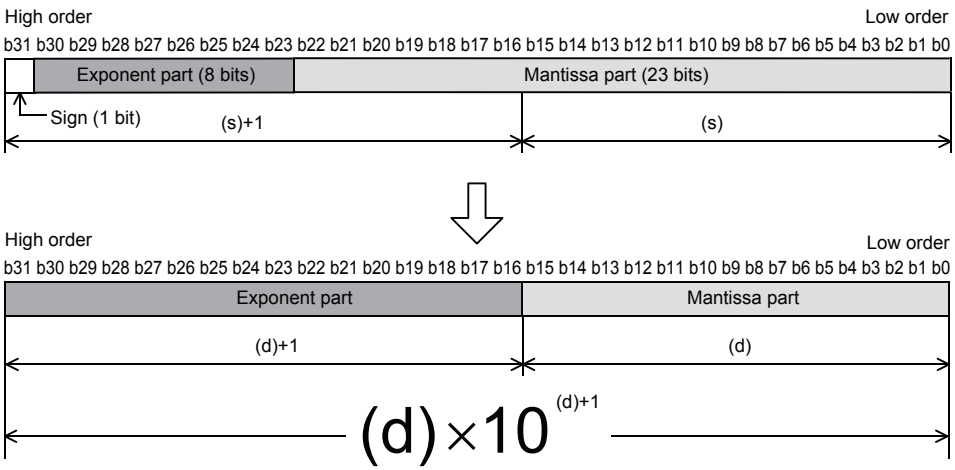
Operand	Description	Range	Data type	Data type (label)
(s)	Head device number storing binary floating point data	—	Single-precision real number	ANYREAL_32
(d)	Device number storing converted decimal floating point	—	Real number	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	—	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions convert the binary floating point specified by (s) to decimal floating point, and store the converted data in the device specified by (d).



Precautions

In floating point operations, all data is handled in binary floating point. Because binary floating point is difficult to understand (requiring a dedicated monitoring method), it is converted into scientific notation (decimal floating point) so that monitoring can be easily executed by peripheral equipment.

Operation error

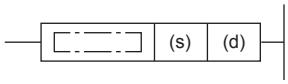
Error code (SD0/SD8067)	Description
3402H	The specified device value is denormalized number, NaN (not a number), or ±∞.

Converting decimal floating point to binary floating point

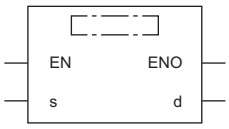
DEBIN(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the decimal floating point specified by (s) to the binary floating point, and store the converted data in the device specified by (d).

Ladder diagram	Structured text
	ENO:=DEBIN(EN,s,d); ENO:=DEBINP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

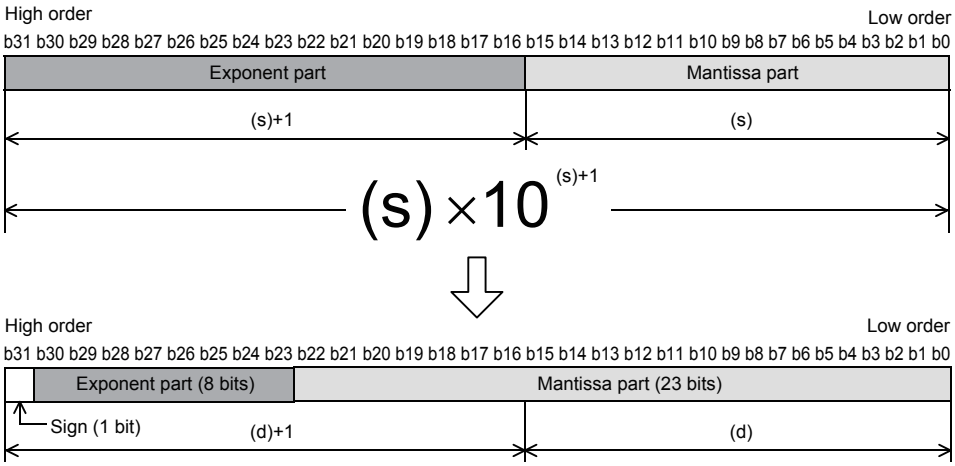
Operand	Description	Range	Data type	Data type (label)
(s)	Head device number storing decimal floating-point data	—	Real number	ANY32
(d)	Device number storing converted binary floating-point data	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	—	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions convert the decimal floating point specified by (s) to the binary floating point, and store the converted data in the device specified by (d).



• The table below shows the related devices.

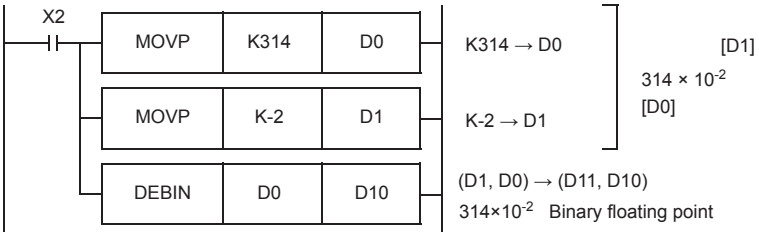
Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Program example

In the program shown below, a numeric value containing the decimal point is converted into binary floating point.

Converting "3.14" into binary floating point

3.14 = 314 × 10^{-2} (decimal floating point)



Operation error

There is no operation error.

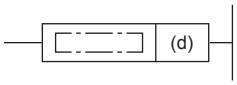
Inverting the sign of single-precision real number

ENEG(P)/DENEG(P)

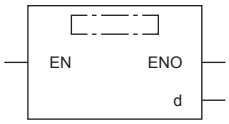
FX5S FX5UJ FX5U FX5UC

These instructions invert the sign of the single-precision real number specified by (d), and store the data of the device specified by (d).

The ENEG(P) instructions can also be used as DENEG(P).

Ladder diagram	Structured text
	ENO:=ENEG(EN,d); ENO:=ENEGP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

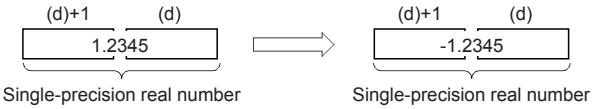
Operand	Description	Range	Data type	Data type (label)
(d)	Head device number storing single-precision real number whose sign is to be inverted	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

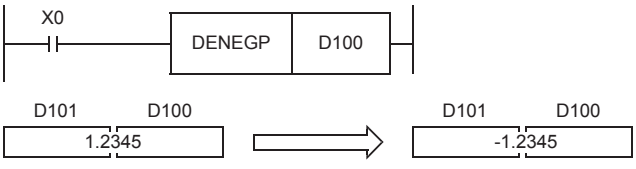
- These instructions invert the sign of the single-precision real number specified by (d), and store the data in the device specified by (d).



- Use these instructions for inverting the positive and negative sign.

Program example

In the program example shown below, the single-precision real number stored in D100 and D101 is inverted, and the negation result is stored to D100 and D101 when X0 turns ON.



Operation error

There is no operation error.

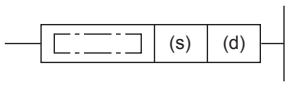
Transferring single-precision real number data

EMOV(P)/DEMOV(P)

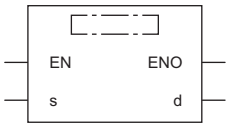
FX5S FX5UJ FX5U FX5UC

These instructions transfer the single-precision real number data stored in the device specified by (s) to the device specified by (d).

The EMOV(P) instructions can also be used as DEMOV(P).

Ladder diagram	Structured text
	ENO:=EMOV(EN,s,d); ENO:=EMOVP(EN,s,d)

FBD/LD



Setting data

■Descriptions, ranges, and data types

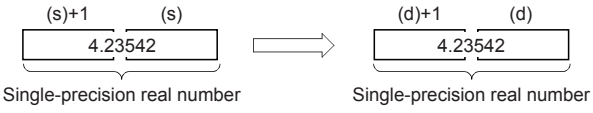
Operand	Description	Range	Data type	Data type (label)
(s)	Data to be transferred or head device number where the data to be transferred is stored	0, 2^{-126}<|(s)|<2^{128}	Single-precision real number	ANYREAL_32
(d)	Device number storing the data in transfer destination	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

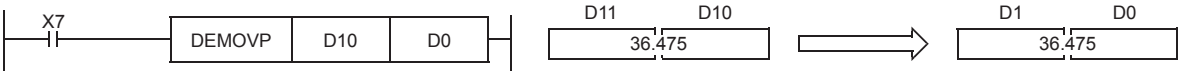
Processing details

- These instructions transfer the single-precision real number data stored in the device specified by (s) to the device specified by (d).



Program example

In the program example shown below, a single-precision real number stored in D11 and D10 is transferred to D1 and D0 when X7 turns ON.



In the program shown below, a single-precision real number "-1.23" is transferred to D11 and D10 when X7 turns ON.

Operation error

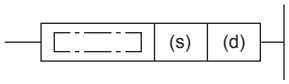
There is no operation error.

Calculating the sine of single-precision real number

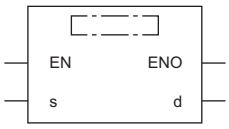
SIN(P)/DSIN(P)

FX5S FX5UJ FX5U FX5UC

These instructions calculate the sine of the angle specified by (s), and store the operation result in the device specified by (d).
The SIN(P) instructions can also be used as DSIN(P).

Ladder diagram	Structured text*1
	ENO:=SINP(EN,s,d);

FBD/LD*1



*1 The SIN instruction is not supported by the ST language and the FBD/LD language. Use SIN of the standard function.
☞ Page 1290 SIN(_E)

Setting data

■Descriptions, ranges, and data types

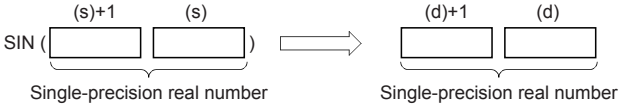
Operand	Description	Range	Data type	Data type (label)
(s)	Angle data or head device number where the angle data is stored	—	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions calculate the sine of the angle specified by (s), and store the operation result in the device specified by (d).



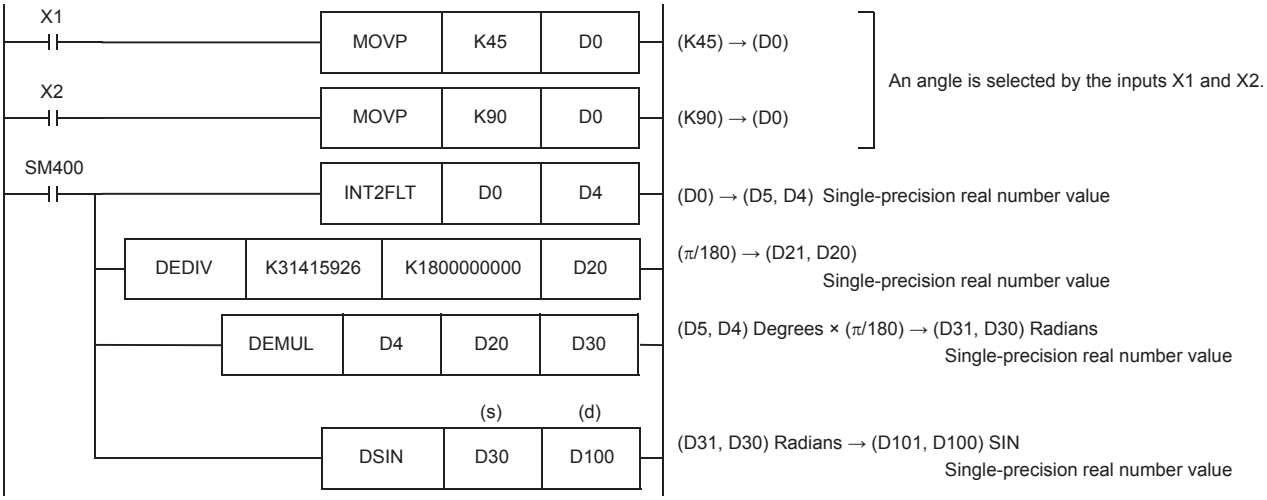
- Set the angle data in radians (angle×π÷180).
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.

Device	Name	Description	
		Condition	Operation
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Program example

This program designates the angle (45° or 90°) and calculates the SIN value when X1 or X2 is set to ON.



Operation error

Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.

Point

For the angle↔radian conversion, refer to the DRAD(P) and DDEG(P) instructions.
(☞ Page 716 Converting single-precision real number angle to radian, Page 718 Converting single-precision real number radian to angle)

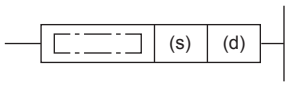
Calculating the cosine of single-precision real number

COS(P)/DCOS(P)

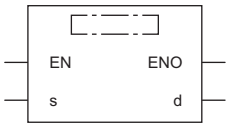
FX5S FX5UJ FX5U FX5UC

These instructions calculate the cosine of the angle specified by (s), and store the operation result in the device specified by (d).

The COS(P) instructions can also be used as DCOS(P).

Ladder diagram	Structured text*1
	ENO:=COSP(EN,s,d);

FBD/LD*1



*1 The COS instruction is not supported by the ST language and the FBD/LD language. Use COS of the standard function.
☞ Page 1291 COS(_E)

Setting data

■Descriptions, ranges, and data types

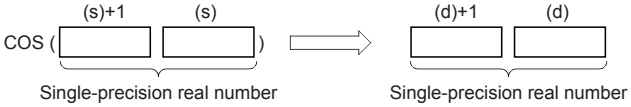
Operand	Description	Range	Data type	Data type (label)
(s)	Angle data or head device number where the angle data is stored	—	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions calculate the cosine of the angle specified by (s), and store the operation result in the device specified by (d).



- Set the angle data in radians (angle×π÷180).
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.

Device	Name	Description	
		Condition	Operation
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Operation error

Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.

Point

For the angle↔radian conversion, refer to the DRAD(P) and DDEG(P) instructions.
(☞ Page 716 Converting single-precision real number angle to radian, Page 718 Converting single-precision real number radian to angle)

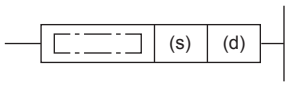
Calculating the tangent of single-precision real number

TAN(P)/DTAN(P)

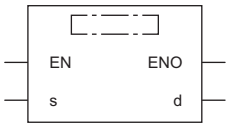
FX5S FX5UJ FX5U FX5UC

These instructions calculate the tangent of the angle specified by (s), and store the operation result in the device specified by (d).

The TAN(P) instructions can also be used as DTAN(P).

Ladder diagram	Structured text*1
	ENO:=TANP(EN,s,d);

FBD/LD*1



*1 The TAN instruction is not supported by the ST language and the FBD/LD language. Use TAN of the standard function.
☞ Page 1292 TAN(_E)

Setting data

■Descriptions, ranges, and data types

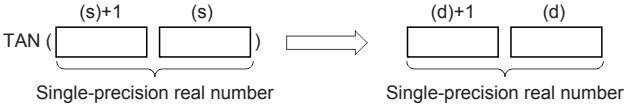
Operand	Description	Range	Data type	Data type (label)
(s)	Angle data or head device number where the angle data is stored	—	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions calculate the tangent of the angle specified by (s), and store the operation result in the device specified by (d).



- Set the angle data in radians (angle×π÷180).
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.

Device	Name	Description	
		Condition	Operation
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Precautions

When the angle specified by (s) is π/2 radian or (3/2)π radian, no error occurs because an operation error occurs in a radian value.

Operation error

Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.

Point

For the angle↔radian conversion, refer to the DRAD(P) and DDEG(P) instructions.
(☞ Page 716 Converting single-precision real number angle to radian, Page 718 Converting single-precision real number radian to angle)

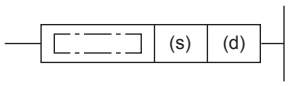
Calculating the arc sine of single-precision real number

ASIN(P)/DASIN(P)

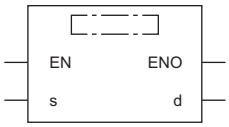
FX5S FX5UJ FX5U FX5UC

These instructions calculate the angle from the sine of the angle specified by (s), and store the operation result in the word device specified by (d).

The ASIN(P) instructions can also be used as DASIN(P).

Ladder diagram	Structured text*1
	ENO:=ASINP(EN,s,d);

FBD/LD*1



*1 The ASIN instruction is not supported by the ST language and the FBD/LD language. Use ASIN of the standard function.
☞ Page 1294 ASIN(_E)

Setting data

■Descriptions, ranges, and data types

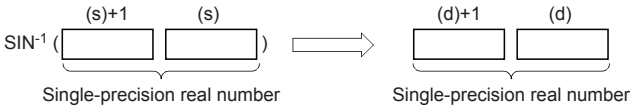
Operand	Description	Range	Data type	Data type (label)
(s)	A sine value used in SIN^{-1} (arc sine) operation or head device number storing the sine value	-1.0 to +1.0	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	$-\pi/2$ to $+\pi/2$	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions calculate the angle from the sine of the angle specified by (s), and store the operation result in the device specified by (d).

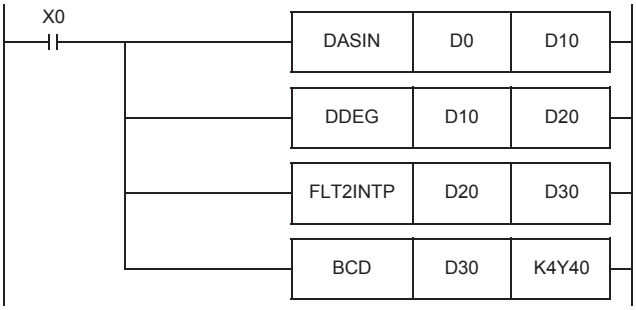


- The sine value specified by (s) can be set ranging from -1.0 to 1.0.
- The angle (operation result) stored in (d) is expressed in radians (from $(-\pi/2)$ to $(\pi/2)$).
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Program example

In the program example shown below, the SIN^{-1} value of data (single-precision real number) stored in D0 and D1 is calculated, and the angle is output in 4-digit BCD to Y40 to Y57 when X0 turns ON.



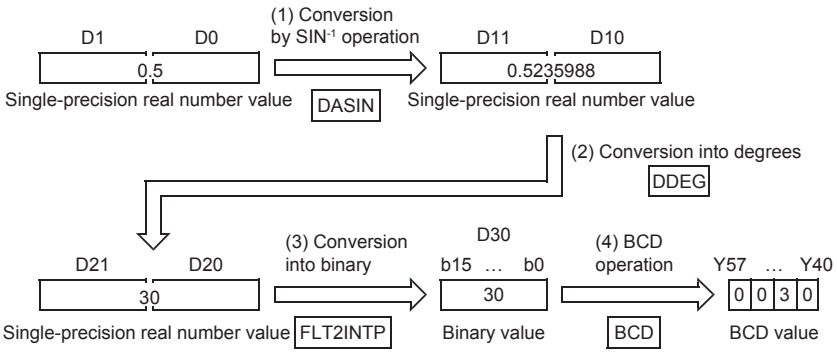
The angle (in radians) is calculated by the SIN^{-1} operation (1).

The value in radians is converted into the value in degrees (2).

The angle expressed in single-precision real number is converted into an integer (binary) (3).

The angle expressed in integer (binary) is output to the seven-segment display unit (4).

- Operation when "0.5" is stored in D0 and D1



Operation error

Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.
3405H	A value specified in (s) is outside the range from -1.0 to 1.0.

Point

For the radian↔angle conversion, refer to the DRAD(P) and DDEG(P) instructions.
(☞ Page 716 Converting single-precision real number angle to radian, Page 718 Converting single-precision real number radian to angle)

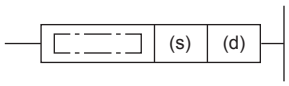
Calculating the arc cosine of single-precision real number

ACOS(P)/DACOS(P)

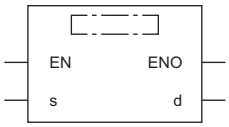
FX5S FX5UJ FX5U FX5UC

These instructions calculate the angle from the cosine of the angle specified by (s), and store the operation result in the word device specified by (d).

The ACOS(P) instructions can also be used as DACOS(P).

Ladder diagram	Structured text*1
	ENO:=ACOSP(EN,s,d)

FBD/LD*1



*1 The ACOS instruction is not supported by the ST language and the FBD/LD language. Use ACOS of the standard function.
☞ Page 1296 ACOS(_E)

Setting data

■Descriptions, ranges, and data types

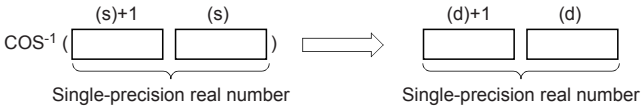
Operand	Description	Range	Data type	Data type (label)
(s)	A cosine value used in COS^{-1} (arc cosine) operation or head device number storing the cosine value	-1.0 to +1.0	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	0 to π	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions calculate the angle from the cosine of the angle specified by (s), and store the operation result in the device specified by (d).

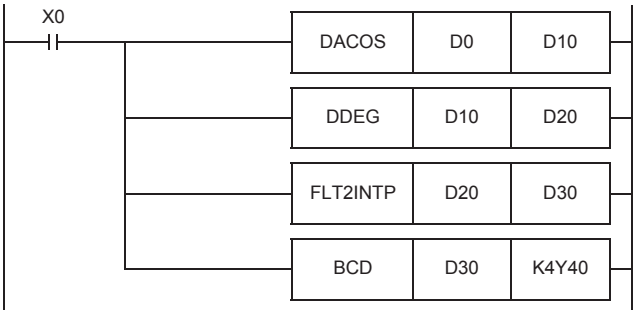


- The cosine value specified by (s) can be set ranging from -1.0 to 1.0
- The angle (operation result) stored in (d) is expressed in radians (0 to π).
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Program example

In the program example shown below, the COS^{-1} value of data (single-precision real number) stored in D0 and D1 is calculated, and the angle is output in 4-digit BCD to Y40 to Y57 when X0 turns ON.



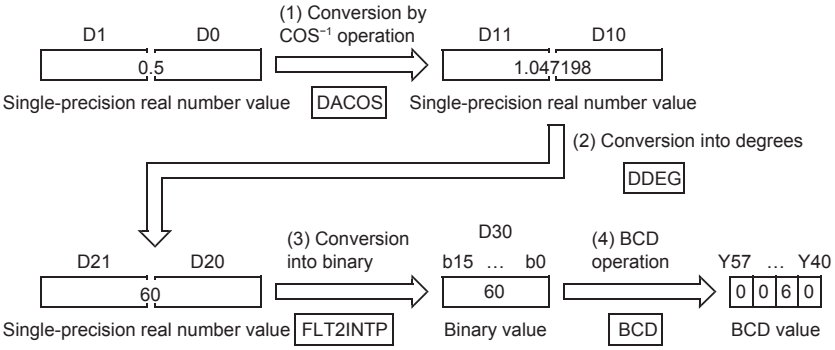
The angle (in radians) is calculated by the COS^{-1} operation (1).

The value in radians is converted into the value in degrees (2).

The angle in single-precision real number is converted into an integer (3).

The angle expressed in integer (binary) is output to the seven-segment display unit (4).

- Operation when "0.5" is stored in D0 and D1



Operation error

Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.
3405H	A value specified in (s) is outside the range from -1.0 to 1.0.

Point

For the radian↔angle conversion, refer to the DRAD(P) and DDEG(P) instructions.
(☞ Page 716 Converting single-precision real number angle to radian, Page 718 Converting single-precision real number radian to angle)

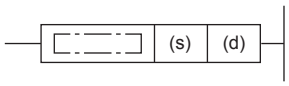
Calculating the arc tangent of single-precision real number

ATAN(P)/DATAN(P)

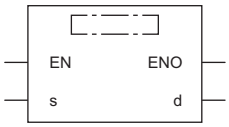
FX5S FX5UJ FX5U FX5UC

These instructions calculate the angle from the tangent of the angle specified by (s), and store the operation result in the word device specified by (d).

The ATAN(P) instructions can also be used as DATAN(P).

Ladder diagram	Structured text*1
	ENO:=ATANP(EN,s,d);

FBD/LD*1



*1 The ATAN instruction is not supported by the ST language and the FBD/LD language. Use ATAN of the standard function.
☞ Page 1298 ATAN(_E)

Setting data

■Descriptions, ranges, and data types

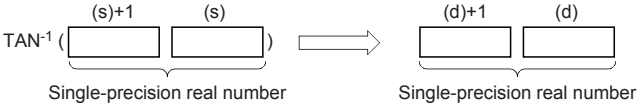
Operand	Description	Range	Data type	Data type (label)
(s)	A tangent value used in the TAN^{-1} (arc tangent) operation or head device number storing the tangent value	—	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	$-\pi/2$ to $+\pi/2$	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions calculate the angle from the tangent of the angle specified by (s), and store the operation result in the device specified by (d).



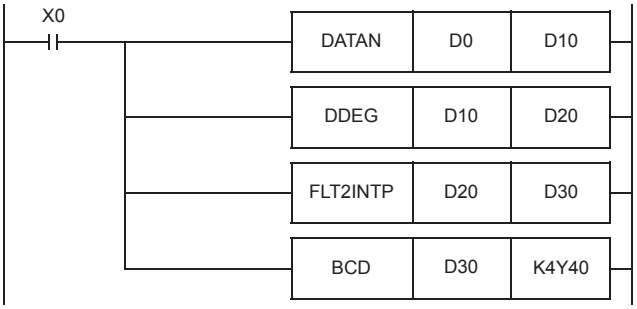
- The angle (operation result) stored in (d) is expressed in radians (from $-\pi/2$ to $\pi/2$).
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.

Device	Name	Description	
		Condition	Operation
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Program example

In the program example shown below, the TAN^{-1} value of data (single-precision real number) stored in D0 and D1 is calculated, and the angle is output in 4-digit BCD to Y40 to Y57 when X0 turns ON.



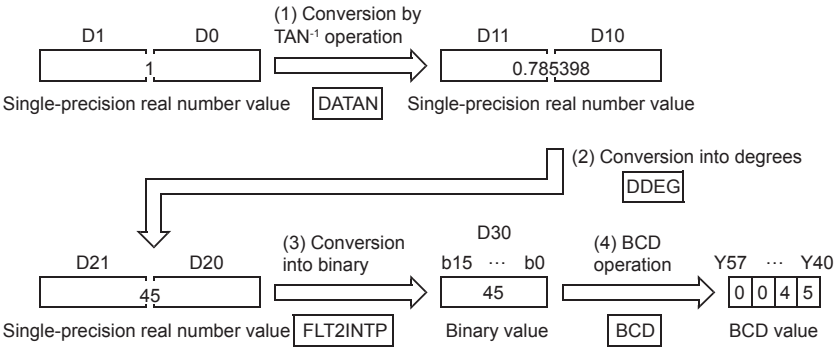
The angle (in radians) is calculated by the TAN^{-1} operation (1).

The value in radians is converted into the value in degrees (2).

The angle expressed in single-precision real number is converted into an integer (binary) (3).

The angle expressed in integer (binary) is output to the seven-segment display unit

- Operation when "1" is stored in D0 and D1



Operation error

Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.

Point

For the radian↔angle conversion, refer to the DRAD(P) and DDEG(P) instructions.
(☞ Page 716 Converting single-precision real number angle to radian, Page 718 Converting single-precision real number radian to angle)

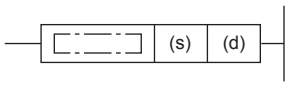
Converting single-precision real number angle to radian

RAD(P)/DRAD(P)

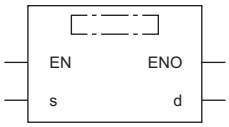
FX5S FX5UJ FX5U FX5UC

These instructions convert a unit of angle from degrees (DEG.) specified by (s) into radians, and store the converted angle in the device specified by (d).

The RAD(P) instructions can also be used as DRAD(P).

Ladder diagram	Structured text
	ENO:=RAD(EN,s,d); ENO:=RADP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

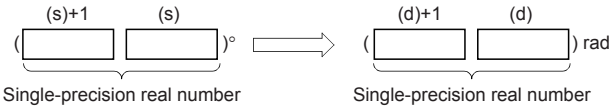
Operand	Description	Range	Data type	Data type (label)
(s)	A value in degrees to be converted into a value in radians or the start number storing the value in degrees	—	Single-precision real number	ANYREAL_32
(d)	Head device number storing a value in radians acquired by conversion	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions convert a unit of angle from degrees (DEG.) specified by (s) into radians, and store the converted angle in the device specified by (d).



- The conversion from degrees into radians is executed as follows:

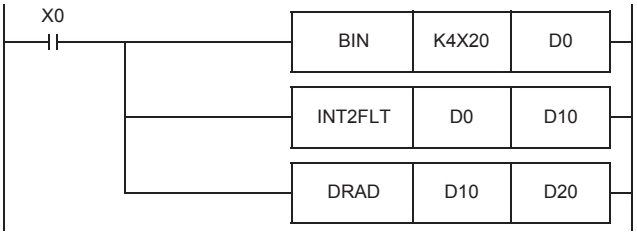
$$\text{Radians} = \text{Degrees} \times \frac{\pi}{180}$$

- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Program example

In the program example shown below, a 4-digit BCD value set in degrees in X20 to X37 is converted into a single-precision real number in radians, and stored to D20 and D21 when X0 turns ON.

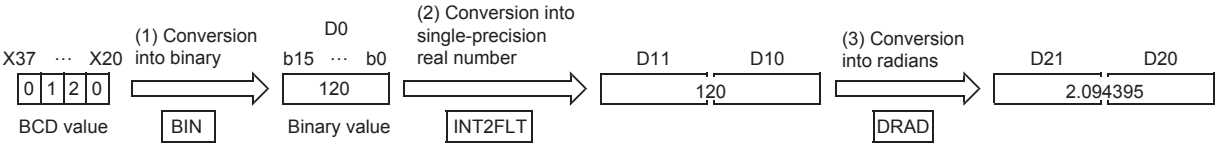


Angle to be converted into radians is input (1).

The input angle is converted into single-precision real number (2).

The angle is converted from degrees into radians (3).

- Operation when "120" is specified in X20 to X37



Operation error

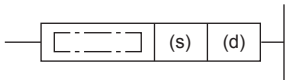
Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.

Converting single-precision real number radian to angle

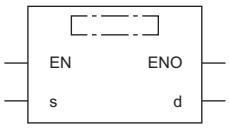
DEG(P)/DDEG(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert a unit of angle from radians specified by (s) into degrees (DEG.), and store the converted angle in the device specified by (d). The DEG(P) instructions can also be used as DDEG(P).

Ladder diagram	Structured text
	ENO:=DEG(EN,s,d); ENO:=DEGP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

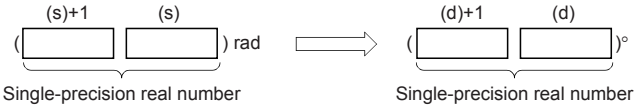
Operand	Description	Range	Data type	Data type (label)
(s)	A value in radians to be converted into a value in degrees or the head device number storing a value in radians	—	Single-precision real number	ANYREAL_32
(d)	Head device number storing a value in degrees acquired by conversion	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions convert a unit of angle from radians specified by (s) into degrees (DEG.), and store the converted angle in the device specified by (d).



- The conversion from radians into degrees is executed as follows:

$$\text{Degrees} = \text{Radians} \times \frac{180}{\pi}$$

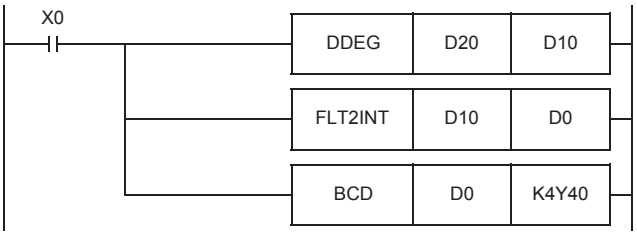
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.

Device	Name	Description	
		Condition	Operation
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Program example

In the program example shown below, a single-precision real number set in radians in D20 and D21 is converted into a BCD value in degrees, and stored to Y40 and Y57 when X0 turns ON.

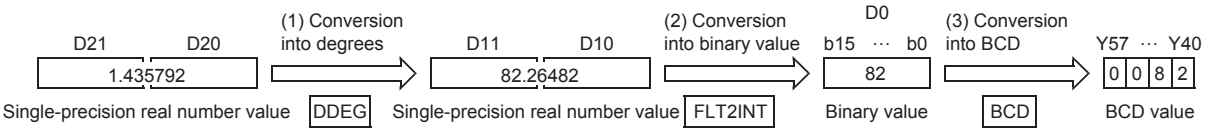


A value in radians is converted into a value in degrees (1).

The angle in single-precision real number is converted into an integer (2).

The converted integer is output to the seven-segment display unit (3).

- Operation when "1.435792" is specified in D20 and D21



Operation error

Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.

Calculating the square root of single-precision real number

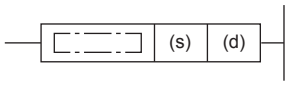
DESQR(P)/ESQRT(P)

FX5S FX5UJ FX5U FX5UC

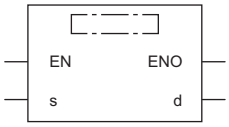
These instructions calculate the square root of a value specified by (s), and store the operation result in the device specified by (d).

The DESQR(P) instructions can also be used as ESQRT(P).

Refer to ☞ Page 1452 Added and Changed Functions for details on the ESQRT(P) instruction.

Ladder diagram	Structured text
	ENO:=DESQR(EN,s,d); ENO:=DESQRP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

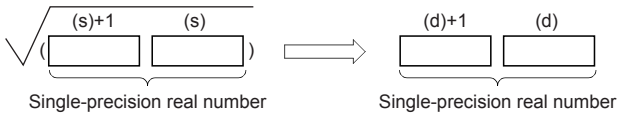
Operand	Description	Range	Data type	Data type (label)
(s)	Data whose square root is calculated or head device number where the data is stored	—	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	○	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions calculate the square root of a value specified by (s), and store the operation result in the device specified by (d).



- Only a positive value can be set in (s). (The square root operation cannot be executed for a negative value).
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.

Operation error

Error code (SD0/SD8067)	Description
2822H	Device that cannot be specified is specified.
3402H	The specified device value is denormalized number, NaN (not a number), or ±∞.
3405H	The value stored in a device specified in (s) is negative.

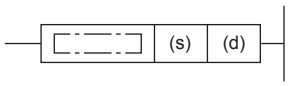
Calculating the exponent of single-precision real number

EXP(P)/DEXP(P)

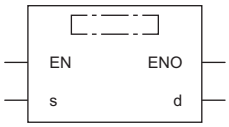
FX5S FX5UJ FX5U FX5UC

These instructions calculate the exponent of a value specified by (s), and store the operation result in the device specified by (d).

The EXP(P) instructions can also be used as DEXP(P).

Ladder diagram	Structured text*1
	ENO:=EXPP(EN,s,d);

FBD/LD*1



*1 The EXP instruction is not supported by the ST language and the FBD/LD language. Use EXP of the standard function.
☞ Page 1289 EXP(_E)

Setting data

■Descriptions, ranges, and data types

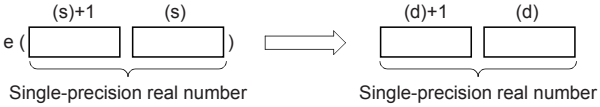
Operand	Description	Range	Data type	Data type (label)
(s)	Data whose exponent is calculated or head device number where the data is stored	—	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions calculate the exponent of a value specified by (s), and store the operation result in the device specified by (d).



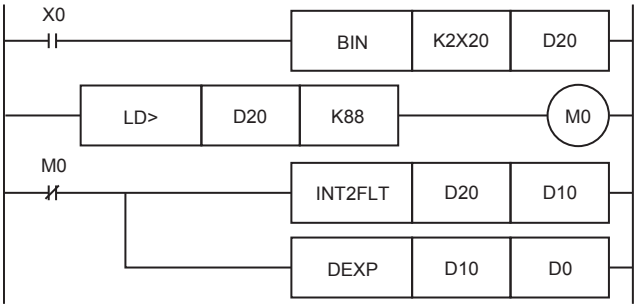
- In the exponential operation, the base (e) is set to "2.71828".
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.

Device	Name	Description	
		Condition	Operation
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Program example

In the program example shown below, the exponential operation is executed for a value set in the 2-digit BCD format in X20 to X27, and the operation result is stored in the single-precision real number format to D0 and D1 when X0 turns ON.



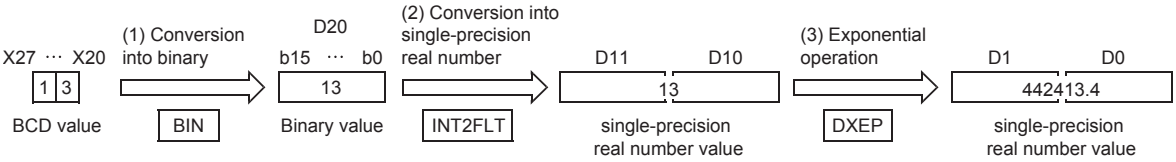
Data used in the exponential operation is input (1).

The range of the value to be operated is checked.*1

The input data is converted into single-precision real number (2).

The exponential operation is executed (3).

- Operation when "13" is specified in X20 to X27



*1 The operation result becomes less than "2^{128}" when the BCD value set in X20 to X27 is "88" or less because of "$\log_e 2^{128}$ = 88.7". If a value "89" or more is set, an operation error occurs. To prevent this operation error, when a value more than "89" is set, M0 is set to ON so that the exponential operation is not executed.

Operation error

Error code (SD0/SD8067)	Description
3402H	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.

Point

- The EXP(P) instructions execute operations in natural logarithm. For obtaining a value in common logarithm, specify a common logarithm value divided by 0.4342945 in (s).

$$10^X = e^{\frac{X}{0.4342945}}$$

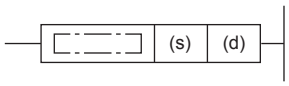
Calculating the natural logarithm of single-precision real number

LOG(P)/DLOGE(P)

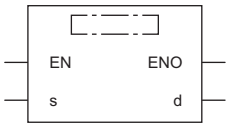
FX5S FX5UJ FX5U FX5UC

These instructions calculate the logarithm whose base is natural logarithm e of a value specified by (s), and store the operation result in the device specified by (d).

The LOG(P) instructions can also be used as DLOGE(P).

Ladder diagram	Structured text*1
	ENO:=LOGP(EN,s,d);

FBD/LD*1



*1 The LOG instruction is not supported by the ST language and the FBD/LD language. Use LOG of the standard function.
☞ Page 1287 LOG(_E)

Setting data

■Descriptions, ranges, and data types

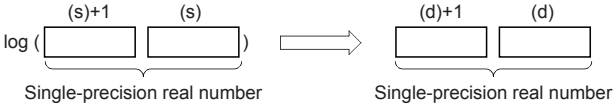
Operand	Description	Range	Data type	Data type (label)
(s)	Data whose natural logarithm is calculated or head device number where the data is stored	—	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions calculate the logarithm whose base is natural logarithm e of a value specified by (s), and store the operation result in the device specified by (d).



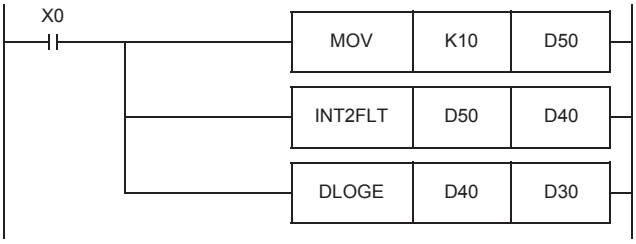
- Only a positive value can be set in (s). (The natural logarithm operation cannot be executed for a negative value).
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.

Device	Name	Description	
		Condition	Operation
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Program example

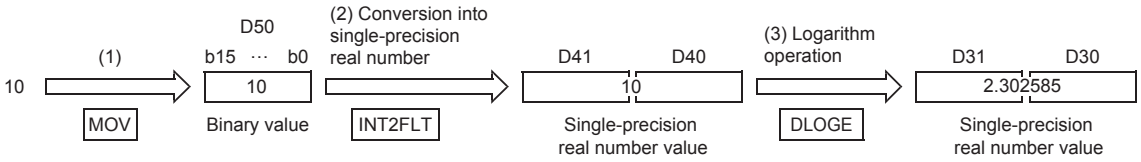
In the program example shown below, natural logarithm of "10" set in D50 is calculated, and stored to D30 and D31 when X0 turns ON.



Data used in the natural logarithm operation is set (1).

The data to be used is converted into single-precision real number (2).

The natural logarithm operation is executed (3).



Operation error

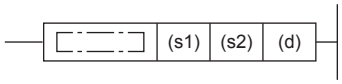
Error code (SD0/SD8067)	Description
3402H	The specified device value is denormalized number, NaN (not a number), or ±∞.
3405H	The value stored in a device specified in (s) is negative.
	The value stored in a device specified in (s) is 0.

Calculating the exponentiation of single-precision real number

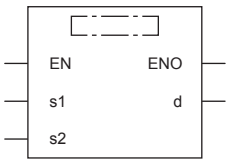
POW(P)

FX5S FX5UJ FX5U FX5UC

These instructions raise float (single precision) data stored in a device specified by (s1) by the single-precision real number specified by (s2), and store the operation result in a device specified by (d).

Ladder diagram	Structured text
	ENO:=POW(EN,s1,s2,d); ENO:=POWP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

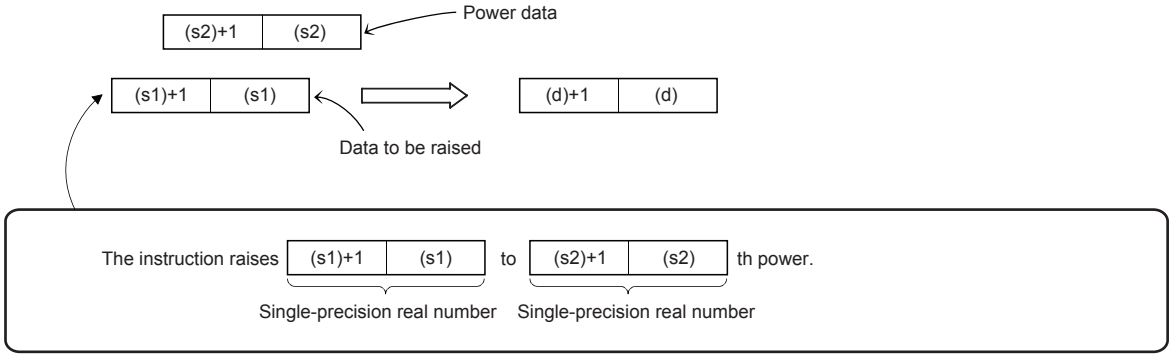
Operand	Description	Range	Data type	Data type (label)
(s1)	Data to be raised, or head device number which stores such data	0, $2^{-126} \leq \lvert(s1)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Power data, or head device number which stores such data	0, $2^{-126} \leq \lvert(s2)\rvert < 2^{128}$	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	—	○	—	—
(s2)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions raise float (single precision) data stored in a device specified by (s1) by the single-precision real number specified by (s2), and store the operation result in a device specified by (d).



- Values in the devices specified (stored) by (s1) and (s2) should be 0 or $2^{-126} \leq$|specified value (stored value)| $< 2^{128}$.
- When the operation result is -0 or underflow occurs, the operation result is regarded as 0.
- When an input value is set from the engineering tool, a rounding error may be occur.

Operation error

Error code (SD0/SD8067)	Description
3402H	The value specified by (s1) or (s2) is outside the following range. 0, $2^{-126} \leq$|specified value (stored value)|$< 2^{128}$
	The specified device value is -0, denormalized number, NaN (not a number), or ±∞.
3403H	The operation result is within the following range. (An overflow has occurred.) $2^{128} \leq$ |operation result|

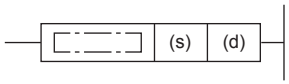
Calculating the common logarithm of single-precision real number

LOG10(P)/DLOG10(P)

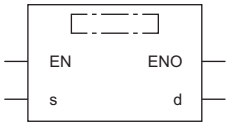
FX5S FX5UJ FX5U FX5UC

These instructions calculate the common logarithm (the logarithm whose base is 10) of a value specified by (s), and store the operation result in the device specified by (d).

The LOG10(P) instructions can also be used as DLOG10(P).

Ladder diagram	Structured text
	ENO:=LOG10(EN,s,d); ENO:=LOG10P(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Data whose common logarithm is calculated or head device number where the data is stored	—	Single-precision real number	ANYREAL_32
(d)	Head device number for storing the operation result	—	Single-precision real number	ANYREAL_32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	○	—	—
(d)	—	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions calculate the common logarithm (the logarithm whose base is 10) of a value specified by (s), and store the operation result in the device specified by (d).

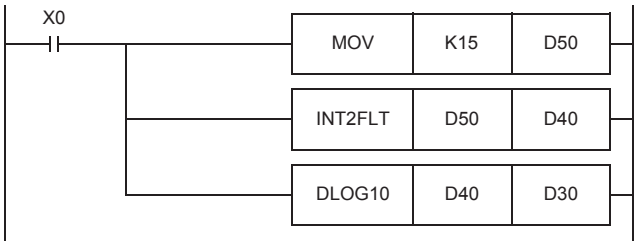
- Only a positive value can be set in (s). (The common logarithm operation cannot be executed for a negative value).
- The table below shows the related devices.

Device	Name	Description	
		Condition	Operation
SM700	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM700 turns on.

Device	Name	Description	
		Condition	Operation
SM8020	Zero	The operation result is true "0". (The mantissa part is "0").	The zero flag SM8020 turns on.
SM8021	Borrow	The absolute value of the operation result < 2^{-126}	The value of (d) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag SM8021 turns on.
SM8022	Carry	The absolute value of the operation result ≥ 2^{128}	The value of (d) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag SM8022 turns on.

Program example

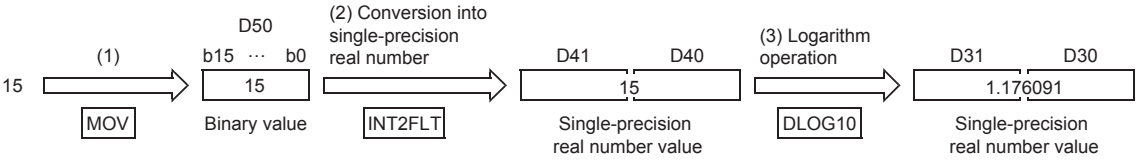
In the program example shown below, common logarithm of "15" set in D50 is calculated, and stored to D30 and D31 when X0 turns ON.



Data used in the common logarithm operation is set (1).

The data to be used is converted into single-precision real number (2).

The common logarithm operation is executed (3).



Operation error

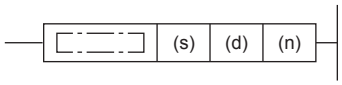
Error code (SD0/SD8067)	Description
3402H	The specified device value is denormalized number, NaN (not a number), or ±∞.
3405H	The value stored in a device specified in (s) is negative.
	The value stored in a device specified in (s) is 0.

Searching the maximum value of single-precision real number

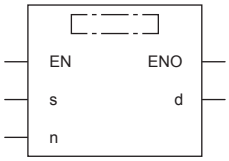
EMAX(P)

FX5S FX5UJ FX5U FX5UC

These instructions search for the maximum value in the (n) point(s) of single-precision real number block data in the device starting from the one specified by (s), and store the maximum value in the device areas specified by (d) and (d)+1.
These instructions also store the location of the first maximum value from (s) in the device specified by (d)+2 and the number of maximum values in the device specified by (d)+3.

Ladder diagram	Structured text*1
	ENO:=EMAXP(EN,s,n,d);

FBD/LD*1



*1 The EMAX instruction is not supported by the ST language and the FBD/LD language. Use MAX of the standard function.
☞ Page 1327 MAX(_E), MIN(_E)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Search target data	—	Single-precision real number	ANYREAL_32
(d)	Search result	—	Single-precision real number	—*1 (ANY_REAL_32_ARRAY)
(n)	Number of search target data points	—	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

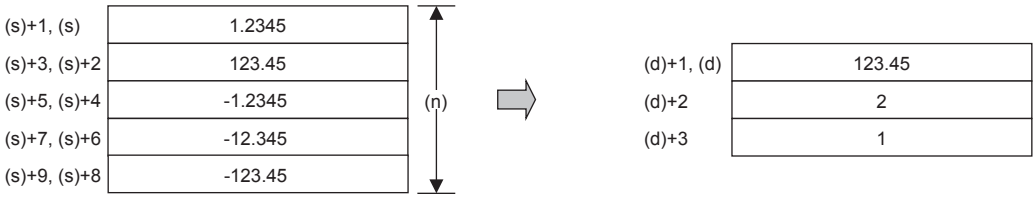
*1 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions search for the maximum value in the (n) point(s) of single-precision real number block data in the device starting from the one specified by (s), and store the maximum value in the device areas specified by (d). These instructions also store the location of the first maximum value from (s) in the device specified by (d)+2 and the number of maximum values in the device specified by (d)+3.
- The start of the block data in the device specified by (s) is counted as 1st point when the location is counted.



(d)+1, (d): Maximum value
(d)+2: Position
(d)+3: Number of data

- The following values are stored in (d).

	Data type	Description
(d)	Single-precision real number	Maximum value
(d)+1		
(d)+2	16-bit data	Maximum value position
(d)+3	16-bit data	Number of maximum values

- When (n) is 0, the processing is not performed.

Operation error

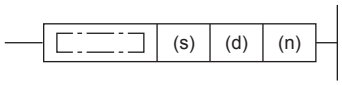
Error code (SD0/SD8067)	Description
2820H	The device areas specified by (s) exceed the corresponding device range.
	The device areas specified by (d) exceed the corresponding device range.
3402H	The block data in the device areas specified by (s) includes a value other than single-precision real number.

Searching the minimum value of single-precision real number

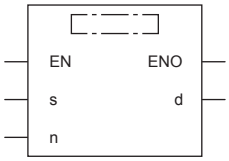
EMIN(P)

FX5S FX5UJ FX5U FX5UC

These instructions search for the minimum value in the (n) point(s) of single-precision real number block data in the device starting from the one specified by (s), and store the minimum value in the device areas specified by (d) and (d)+1.
These instructions also store the location of the first minimum value from (s) in the device specified by (d)+2 and the number of minimum values in the device specified by (d)+3.

Ladder diagram	Structured text*1
	ENO:=EMINP(EN,s,n,d);

FBD/LD*1



*1 The EMIN instruction is not supported by the ST language and the FBD/LD language. Use MIN of the standard function.
☞ Page 1327 MAX(_E), MIN(_E)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Search target data	—	Single-precision real number	ANYREAL_32
(d)	Search result	—	Single-precision real number	—*1 (ANY_REAL_32_ARRAY)
(n)	Number of search target data points	—	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

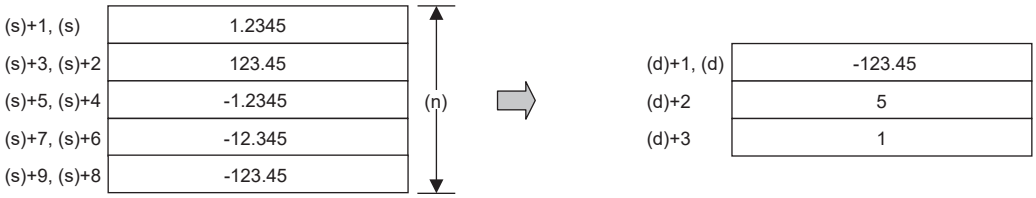
*1 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions search for the minimum value in the (n) point(s) of single-precision real number block data in the device starting from the one specified by (s), and store the minimum value in the device areas specified by (d) and (d)+1. These instructions also store the location of the first minimum value from (s) in the device specified by (d)+2 and the number of minimum values in the device specified by (d)+3.
- The start of the block data in the device specified by (s) is counted as 1st point when the location is counted.



(d)+1, (d): Minimum value
(d)+2: Position
(d)+3: Number of data

- The following values are stored in (d).

	Data type	Description
(d)	Single-precision real number	Minimum value
(d)+1		
(d)+2	16-bit data	Minimum value position
(d)+3	16-bit data	Number of minimum values

- When (n) is 0, the processing is not performed.

Operation error

Error code (SD0/SD8067)	Description
2820H	The device areas specified by (s) exceed the corresponding device range.
	The device areas specified by (d) exceed the corresponding device range.
3402H	The block data in the device areas specified by (s) includes a value other than single-precision real number.

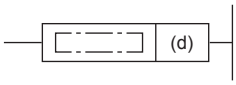
8.11 Random Number Instruction

Generating random number

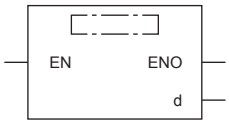
RND(P)

FX5S FX5UJ FX5U FX5UC

These instructions generate a pseudo-random number ranging from 0 to 32767, and store it as a random number to a device specified by (d).

Ladder diagram	Structured text
	ENO:=RND(EN,d); ENO:=RNDP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number storing a random number	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions generate a pseudo-random number ranging from 0 to 32767, and store it as a random number to a device specified by (d).
- In the pseudo-random number sequence, the source value of a random number is calculated every time, and this instruction calculates a pseudo-random number using the source value.

  Pseudo-random number calculation equation:

  (SD8311, SD8310)=(SD8311, SD8310)*1×1103515245+12345

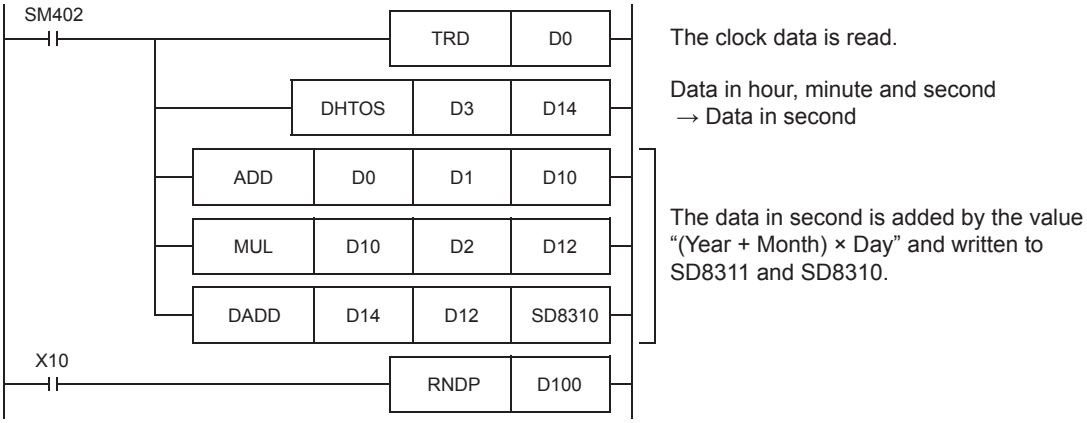
  (d) ="([SD8311, SD8310]>>16)&<logical product>00007FFFh"

*1 To (SD8311, SD8310), write a non-negative value (0 to 2147483647) only once when the CPU module mode switches from STOP to RUN. (K1 is written to (SD8311, SD8310) as the initial value when the power is restored.)

Program example

When the PLC mode switches from STOP to RUN, the time data converted into seconds and added by the value “(Year + Month) × Day” is written to SD8311 and SD8310.

In the program example shown below, a random number is stored to D100 every time X10 turns ON.



Operation error

There is no operation error.

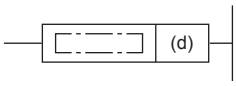
8.12 Index Register Operation Instruction

Saving all data of the index register

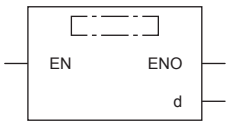
ZPUSH(P)

FX5S FX5UJ FX5U FX5UC

These instructions save the contents of index registers and long index registers in the devices specified by (d) and later.

Ladder diagram	Structured text
	ENO:=ZPUSH(EN,d); ENO:=ZPUSHP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number for saving the data of index registers and long index registers	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

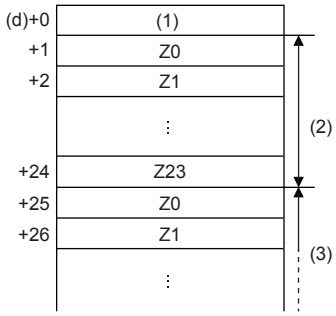
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○	—	—	—	—	○	—	—	—	—

Processing details

- These instructions save the contents of index registers and long index registers in the devices specified by (d) and later.
- When the contents of index registers are saved, "1" is added to (d).
- The contents of index registers and long index registers for 24 words are saved. The number of points differs for each CPU module as explained below.
  - FX5S/FX5U/FX5UC CPU module: The assigned range of index registers and long index register points is saved.
  - FX5UJ CPU module: The contents for 20 points of the index register and 2 points of the long index register are saved.
- The ZPOP(P) instructions are used to return the data. The ZPUSH(P) and ZPOP(P) instructions are used in pairs, and by using the same device in (d) a nesting structure can be adopted. (☞ Page 739 Returning all data of the index register)
- When a nesting structure is adopted, the areas to be used are added to (d) and later every time the ZPUSH(P) instructions are used. Check the number of index registers and long index registers by SD300 and SD302, and secure the areas for the number of instructions to be used in advance.

- The following shows the areas of (d) and later to be used.



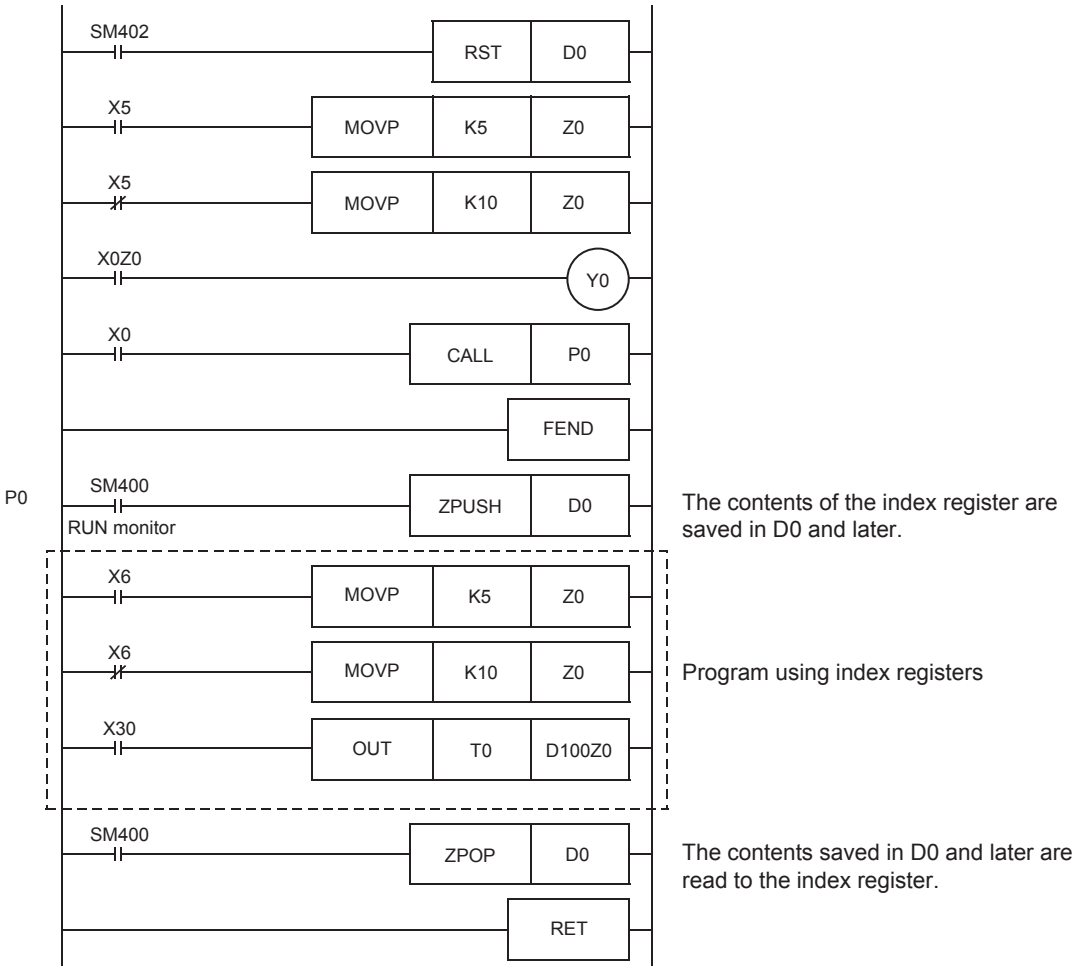
(1): Number of times of batch-storage
(2): 1st nesting (24 words for one nesting)
(3): 2nd nesting

Precautions

- When a nesting structure is not adopted, clear (d) before executing the ZPUSH(P) instructions.
- When a nesting structure is adopted, clear (d) before executing the first ZPUSH(P) instructions.
- When the ZPOP(P) instructions are used to return the data of index registers, use the ZPOP(P) instructions corresponding to the ZPUSH(P) instructions that were used for saving the data.
  ZPUSH(P) (One setting data) → ZPOP(P) (One setting data)
  ZPUSH(P) (Two setting data) → ZPOP(P) (Two setting data)
- Secure the areas so that the save destination specified by (d) do not exceed the device range.

Program example

In the program shown below, the contents of the index register (Z) before execution of the subroutine program are batch-stored in D0 and later when index registers are used in the subroutine program after the pointer P0.



Operation error

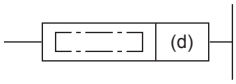
Error code (SD0/SD8067)	Description
2820H	The range of points used in (d) or later exceeds the range of the target device/label area.
3405H	(d) is negative.

Returning all data of the index register

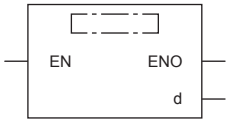
ZPOP(P)

FX5S FX5UJ FX5U FX5UC

These instructions read the data saved in the devices specified by (d) and later to index registers and long index registers.

Ladder diagram	Structured text
	ENO:=ZPOP(EN,d); ENO:=ZPOPP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number for returning the data of index registers	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○	—	—	—	—	○	—	—	—	—

Processing details

- These instructions read the data saved in the devices specified by (d) and later to index registers and long index registers.
- When the saved contents of the index registers and long index registers are read, "1" is subtracted from (d).
- The ZPUSH(P) instructions are used to temporarily save the data. The ZPUSH(P) and ZPOP(P) instructions are used in pairs.

Program example

For a program example, refer to ☞ Page 736 Saving all data of the index register.

Operation error

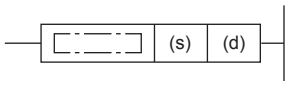
Error code (SD0/SD8067)	Description
2820H	The range of points used in (d) or later exceeds the range of the target device/label area.
3405H	(d) is 0 or negative.

Saving the selected data of the index register and long index register

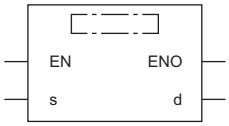
ZPUSH(P)

FX5S FX5UJ FX5U FX5UC

These instructions save the contents of index registers and long index registers within the range specified by (s) in the devices specified by (d) and later.

Ladder diagram	Structured text
	ENO:=ZPUSH_2(EN,s,d); ENO:=ZPUSHP_2(EN,s,d);

FBD/LD



("ZPUSH_2", "ZPUSHP_2" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Type of the index register or long index register to be saved	0 to 2	16-bit unsigned binary	ANY16
(d)	Head device number for saving the data of index registers	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	—	○	—	—	○	○	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—

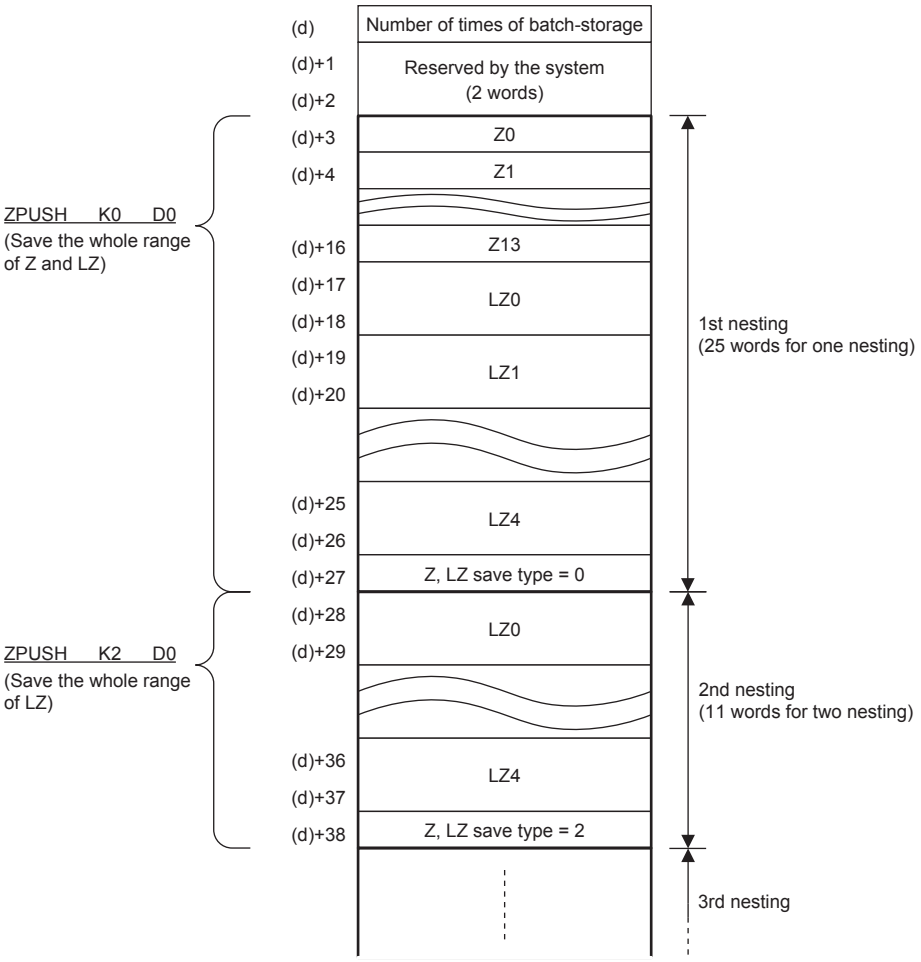
Processing details

- These instructions save the contents of index registers and long index registers within the range specified by (s) in the devices specified by (d) and later. The type of the index register or long index register saved is stored in the end of the saved data.
- When the contents of the index registers and long index registers are saved, "1" is added to (d).
- The following shows values specified by (s) and the index register or long index register to be saved.

(s) value	Z or LZ to be saved
0	Z, LZ (whole range)
1	Z (whole range)
2	LZ (whole range)

- The selected data of index register/long index register return instructions (ZPOP(P) instructions) are used to return the data. The selected data of index register/long index register save instructions (ZPUSH(P) instructions) and the selected data of index register/long index register return instructions (ZPOP(P) instructions) can be used in pairs and to adopt a nesting structure. (☞ Page 742 Returning the selected data of the index register and long index register)

- When a nesting structure is adopted, the areas to be used are added to (d) and later every time the selected data of index register/long index register save instructions (ZPUSH(P) instructions) are executed. Check the number of index registers and long index registers by SD300 and SD302, and secure the areas for the number of instructions to be used in advance.
- The following shows the areas of (d) and later used for the instructions (when Z0 to 13 and LZ0 to 4 are used).



Precautions

- When a nesting structure is not adopted, clear (d) before executing the ZPUSH(P) instructions.
- When a nesting structure is adopted, clear (d) before executing the first ZPUSH(P) instructions.
- When the ZPOP(P) instructions are used to return the data of index registers, use the ZPOP(P) instructions corresponding to the ZPUSH(P) instructions that were used for saving the data.
  ZPUSH(P) (One setting data) → ZPOP(P) (One setting data)
  ZPUSH(P) (Two setting data) → ZPOP(P) (Two setting data)
- Do not change the values of (d)+1 and (d)+2 because they are used by the system. Do not change the values of the Z and LZ save types stored in the devices specified by (d) and later because they are used by the system.
- Secure the areas so that the save destination specified by (d) does not exceed the device range.

Operation error

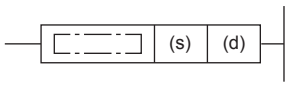
Error code (SD0/SD8067)	Description
2820H	The range of points used in (d) or later exceeds the range of the target device/label area.
3405H	A value other than 0 to 2 is specified in (s).
	When the number of index registers is 0, "1" is specified in (s).
	When the number of long index registers is 0, "2" is specified in (s).

Returning the selected data of the index register and long index register

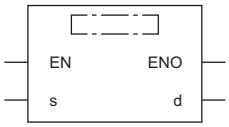
ZPOP(P)

FX5S FX5UJ FX5U FX5UC

These instructions read the data saved in the devices specified by (d) and later to index registers and long index registers.

Ladder diagram	Structured text
	ENO:=ZPOP_2(EN,s,d); ENO:=ZPOPP_2(EN,s,d);

FBD/LD



("ZPOP_2, ZPOPP_2" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Dummy	—	16-bit unsigned binary	ANY16
(d)	Head device number for returning the data of index registers	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	—	○	—	—	○	○	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—

Processing details

- These instructions read the data saved in the devices specified by (d) and later to index registers and long index registers.
- When the saved contents of the index registers and long index registers are read, "1" is subtracted from (d).
- The data specified by (s) is regarded as dummy data and ignored.

Operation error

Error code (SD0/SD8067)	Description
2820H	The range of points used in (d) or later exceeds the range of the target device/label area.
3405H	(d) is 0 or negative.

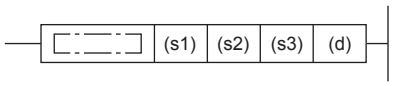
8.13 Data Control Instruction

Upper and lower limit control of 16-bit binary data

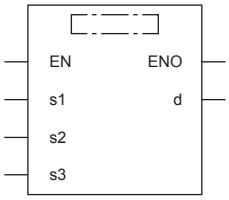
LIMIT(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions control the output value to be stored in the device specified by (d) by checking the input value (16-bit binary data) in the device specified by (s3) with the upper and lower limit values specified by (s1) and (s2).

Ladder diagram	Structured text*1	
	ENO:=LIMITP(EN,s1,s2,s3,d);	ENO:=LIMITP_U(EN,s1,s2,s3,d);

FBD/LD*1



*1 The LIMIT and LIMIT_U instructions are not supported by the ST language and the FBD/LD language. Use LIMIT of the standard function.
☞ Page 1329 LIMIT(_E)

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	LIMIT(P)	Lower limit value (minimum output value)	-32768 to 32767	16-bit signed binary	ANY16_S
	LIMIT(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	LIMIT(P)	Upper limit value (maximum output value)	-32768 to 32767	16-bit signed binary	ANY16_S
	LIMIT(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s3)	LIMIT(P)	Input value controlled by the upper and lower limit values	-32768 to 32767	16-bit signed binary	ANY16_S
	LIMIT(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	LIMIT(P)	Head device number storing the output value controlled by the upper and lower limit values	—	16-bit signed binary	ANY16_S
	LIMIT(P)_U			16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

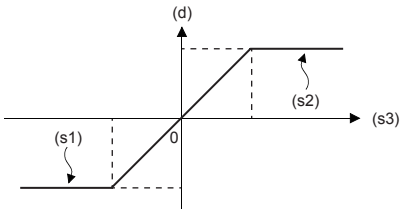
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions control the output value to be stored in the device specified by (d) by checking the input value (16-bit binary data) in the device specified by (s3) with the upper and lower limit values specified by (s1) and (s2). The output value is controlled as follows.

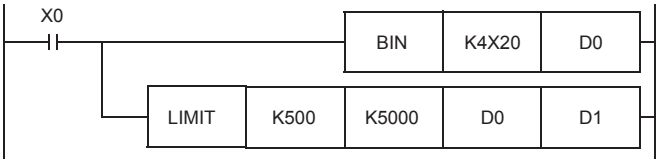
Condition	Output value
Lower limit value (s1) > Input value (s3)	Lower limit value (s1)
Upper limit value (s2) < Input value (s3)	Upper limit value (s2)
Lower limit value (s1) ≤ Input value (s3) ≤ Upper limit value (s2)	Input value (s3)



- To control the input value only with the upper limit, set the minimum value within the setting range in (s1).
- To control the input value only with the lower limit, set the maximum value within the setting range in (s2).

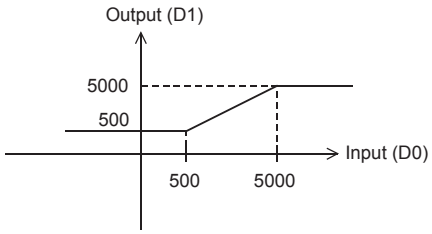
Program example

In the program example shown below, the BCD data set in X20 to X37 is controlled by the limit values “500” to “5000”, and the controlled value is output to D1 when X0 turns ON.



Operation

- In the case of “D0 < 500”, “500” is output to D1.
- In the case of “500 ≤ D0 ≤ 5000”, the value of D0 is output to D1.
- In the case of “D0 > 5000”, “5000” is output to D1.



Operation error

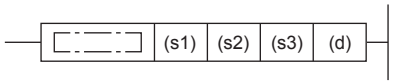
Error code (SD0/SD8067)	Description
3405H	The lower limit value specified by (s1) is greater than the upper limit value specified by (s2).

Upper and lower limit control of 32-bit binary data

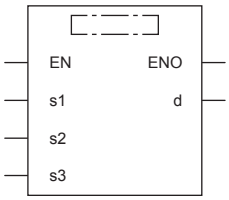
DLIMIT(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions control the output value to be stored in the device specified by (d) by checking the input value (32-bit binary data) in the device specified by (s3) with the upper and lower limit values specified by (s1) and (s2).

Ladder diagram	Structured text*1	
	ENO:=DLIMITP(EN,s1,s2,s3,d);	ENO:=DLIMITP_U(EN,s1,s2,s3,d);

FBD/LD*1



*1 The DLIMIT and DLIMIT_U instructions are not supported by the ST language and the FBD/LD language. Use LIMIT of the standard function.
☞ Page 1329 LIMIT(_E)

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DLIMIT(P)	Lower limit value (minimum output value)	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DLIMIT(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DLIMIT(P)	Upper limit value (maximum output value)	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DLIMIT(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s3)	DLIMIT(P)	Input value controlled by the upper and lower limit values	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DLIMIT(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DLIMIT(P)	Head device number storing the output value controlled by the upper and lower limit values	—	32-bit signed binary	ANY32_S
	DLIMIT(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

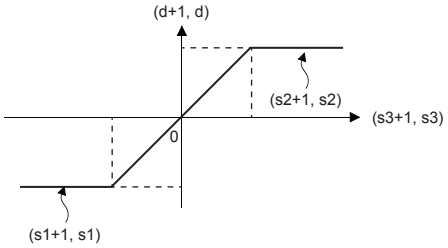
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions control the output value to be stored in the device specified by (d) by checking the input value (32-bit binary data) in the device specified by (s3) with the upper and lower limit values specified by (s1) and (s2). The output value is controlled as follows.

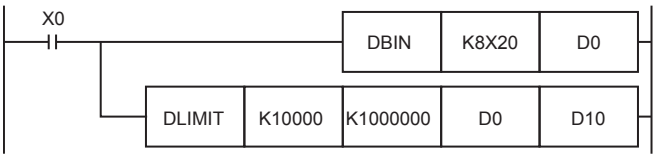
Condition	Output value
Lower limit value ((s1), (s1)+1) > Input value ((s3), (s3)+1)	Lower limit value ((s1), (s1)+1)
Upper limit value ((s2), (s2)+1) < Input value ((s3), (s3)+1)	Upper limit value ((s2), (s2)+1)
Lower limit value ((s1), (s1)+1) ≤ Input value ((s3), (s3)+1) ≤ Upper limit value ((s2), (s2)+1)	Input value ((s3), (s3)+1)



- To control the input value only with the upper limit, set the minimum value within the setting range in (s1).
- To control the input value only with the lower limit, set the maximum value within the setting range in (s2).

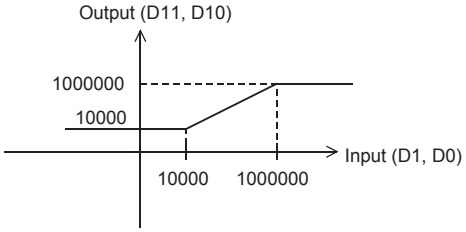
Program example

In the program example shown below, the BCD data set in X20 to X57 is controlled by the limit values “10000” and “1000000”, and the controlled value is output to D11 and D10 when X0 turns ON.



Operation

- In the case of “(D1, D0) < 10000”, “10000” is output to (D11, D10).
- In the case of “10000 ≤ (D1, D0) ≤ 1000000”, the value of (D1, D0) is output to (D11, D10).
- In the case of “(D1, D0) > 1000000”, “1000000” is output to (D11, D10).



Operation error

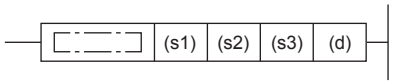
Error code (SD0/SD8067)	Description
3405H	The lower limit value specified by (s1) is greater than the upper limit value specified by (s2).

Dead band control of 16-bit binary data

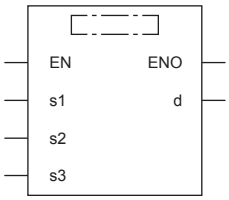
BAND(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions control the output value to be stored in the device specified by (d) by checking the input value (16-bit binary data) in the device specified by (s3) with the upper and lower limit values of the dead band specified by (s1) and (s2).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	BAND(P)	Lower limit value of the dead band (no-output band)	-32768 to +32767	16-bit signed binary	ANY16_S
	BAND(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	BAND(P)	Upper limit value of the dead band (no-output band)	-32768 to +32767	16-bit signed binary	ANY16_S
	BAND(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s3)	BAND(P)	Input value controlled by the dead band	-32768 to +32767	16-bit signed binary	ANY16_S
	BAND(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	BAND(P)	Head device number for storing the output value controlled by the dead band	—	16-bit signed binary	ANY16_S
	BAND(P)_U			16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

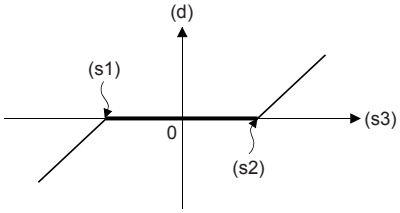
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions control the output value to be stored in the device specified by (d) by checking the input value (16-bit binary data) in the device specified by (s3) with the upper and lower limit values of the dead band specified by (s1) and (s2). The output value is controlled as follows.

Condition	Output value
Lower limit value of the dead band (s1) > Input value (s3)	Input value (s3) - Lower limit value of the dead band (s1)
Upper limit value of the dead band (s2) < Input value (s3)	Input value (s3) - Upper limit value of the dead band (s2)
Lower limit value of the dead band (s1) ≤ Input value (s3) ≤ Upper limit value of the dead band (s2)	0



- When the output value to be stored in the device specified by (d) is a 16-bit signed binary value and the operation result exceeds the range of -32768 to 32767, the output value is calculated as follows.

Ex.

When (s1) is 10 and (s3) is -32768: Output value = -32768-10 = 8000H-000AH = 7FFFH = 32758

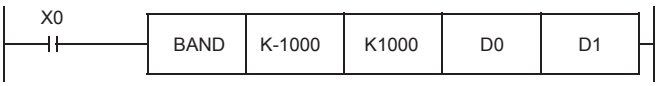
- When the output value to be stored in the device specified by (d) is a 16-bit unsigned binary value and the operation result exceeds the range of 0 to 65535, the output value is calculated as follows.

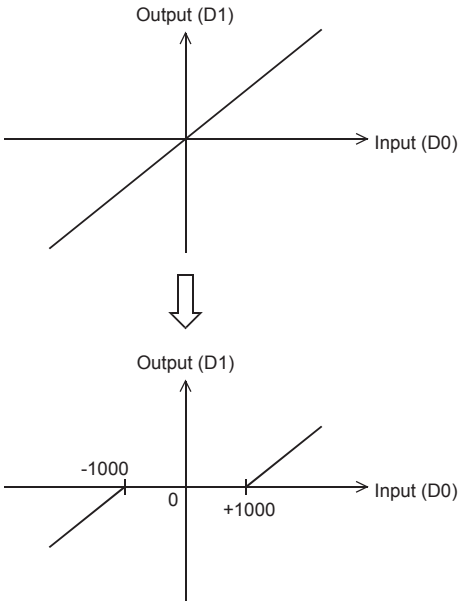
Ex.

When (s1) is 100 and (s3) is 50: Output value = 50-100 = 0032H-0064H = FFCEH = 65486

Program example

In the program example shown below, the data of D0 is controlled by the dead band of the limit values “-1000” to “+1000”, and the controlled value is output to D1 when X0 is set to ON.





Operation

- In the case of “D0 < -1000”, “D0 - (-1000)” is output to D1.
- In the case of “-1000 ≤ D0 ≤ +1000”, “0” is output to D1.
- In the case of “+1000 < D0”, “D0 - 1000” is output to D1.

Operation error

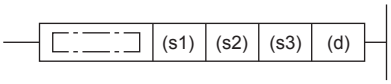
Error code (SD0/SD8067)	Description
3405H	The lower limit value specified by (s1) is greater than the upper limit value specified by (s2).

Dead band control of 32-bit binary data

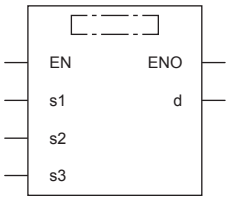
DBAND(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions control the output value to be stored in the device specified by (d) by checking the input value (32-bit binary data) in the device specified by (s3) with the upper and lower limit values of the dead band specified by (s1) and (s2).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DBAND(P)	Lower limit value of the dead band (no-output band)	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DBAND(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DBAND(P)	Upper limit value of the dead band (no-output band)	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DBAND(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s3)	DBAND(P)	Input value controlled by the dead band	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DBAND(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DBAND(P)	Head device number for storing the output value controlled by the dead band	—	32-bit signed binary	ANY32_S
	DBAND(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

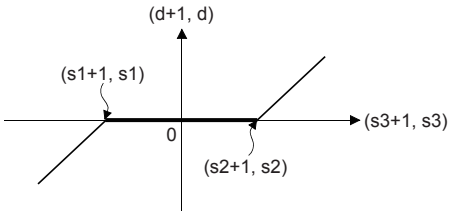
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions control the output value to be stored in the device specified by (d) by checking the input value (32-bit binary data) in the device specified by (s3) with the upper and lower limit values of the dead band specified by (s1) and (s2). The output value is controlled as follows.

Condition	Output value
Lower limit value of the dead band ((s1), (s1)+1) > Input value ((s3), (s3)+1)	Input value ((s3), (s3)+1) - Lower limit value of the dead band ((s1), (s1)+1)
Upper limit value of the dead band ((s2), (s2)+1) < Input value ((s3), (s3)+1)	Input value ((s3), (s3)+1) - Upper limit value of the dead band ((s2), (s2)+1)
Lower limit value of the dead band ((s1), (s1)+1) ≤ Input value ((s3), (s3)+1) ≤ Upper limit value of the dead band ((s2), (s2)+1)	0



- When the output value to be stored in the device specified by (d) is a 32-bit signed binary value and the operation result exceeds the range of -2147483648 to 2147483647, the output value is calculated as follows.

Ex.

When (s1) and (s1)+1 are 1000, and (s3) and (s3)+1 are -2147483648: Output value = -2147483648-1000 = 80000000H-000003E8H = 7FFFFC18H = 2147482648

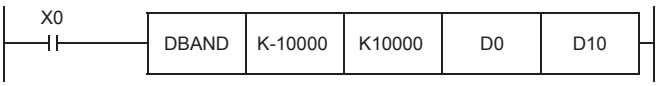
- When the output values to be stored in the devices specified by (d) and (d)+1 are 32-bit unsigned binary values and the operation result exceeds the range of 0 to 4294967295, the output value is calculated as follows.

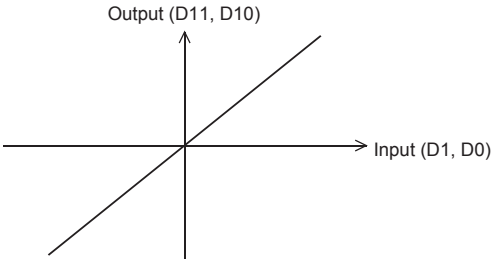
Ex.

When (s1) and (s1)+1 are 100, and (s3) and (s3)+1 are 50: Output value = 50-100 = 00000032H-00000064H = FFFFFFCEH = 4294967246

Program example

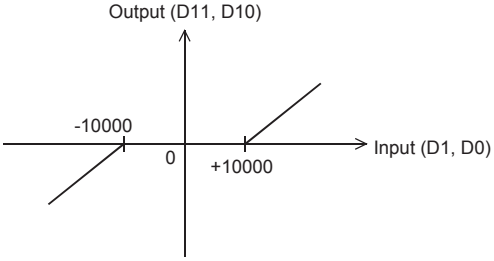
In the program example shown below, the data of D0 and D1 is controlled by the dead band of the limit values “-10000” to “+10000”, and the controlled values are output to D10 and D11 when X0 is set to ON.





Operation

- In the case of “(D1, D0) < -10000”, “(D1, D0) - (-10000)” is output to (D11, D10).
- In the case of “-10000 ≤ (D1, D0) ≤ +10000”, “0” is output to (D11, D10).
- In the case of “(D1, D0) > +10000”, “(D1, D0) - 10000” is output to (D11, D10).



Operation error

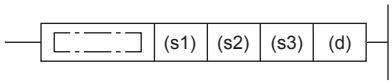
Error code (SD0/SD8067)	Description
3405H	The lower limit value specified by (s1) is greater than the upper limit value specified by (s2).

Zone control of 16-bit binary data

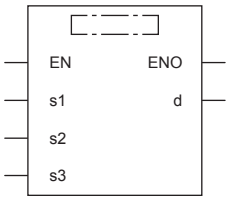
ZONE(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add the bias value specified by (s1) or (s2) to the input value specified by (s3), and store the operation result in the device specified by (d).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	ZONE(P)	Negative bias value to be added to the input value	-32768 to +32767	16-bit signed binary	ANY16_S
	ZONE(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	ZONE(P)	Positive bias value to be added to the input value	-32768 to +32767	16-bit signed binary	ANY16_S
	ZONE(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s3)	ZONE(P)	Input value for performing the zone control	-32768 to +32767	16-bit signed binary	ANY16_S
	ZONE(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(d)	ZONE(P)	Head device number storing the output value controlled by the zone	—	16-bit signed binary	ANY16_S
	ZONE(P)_U			16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

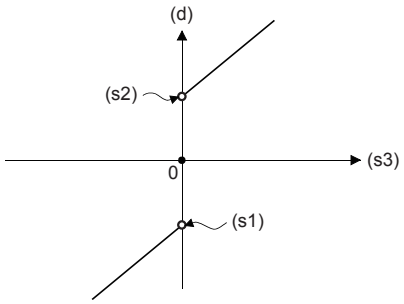
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions add the bias value specified by (s1) or (s2) to the input value (16-bit binary data) specified by (s3), and store the operation result in the device specified by (d). The bias value is controlled as follows.

Condition	Output value
Input value (s3) < 0	Input value (s3) + Negative bias value (s1)
Input value (s3) = 0	0
Input value (s3) > 0	Input value (s3) + Positive bias value (s2)



- When the output value to be stored in the device specified by (d) is a 16-bit signed binary value and the operation result exceeds the range of -32768 to 32767, the output value is calculated as follows.

Ex.

When (s1) is -100 and (s3) is -32768: Output value = -32768+(-100) = 8000H-FF9CH = 7F9CH = 32668

- When the output value to be stored in the device specified by (d) is a 16-bit unsigned binary value and the operation result exceeds the range of 0 to 65535, the output value is calculated as follows.

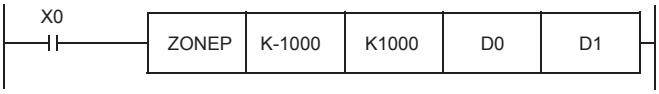
Ex.

When (s2) is 100 and (s3) is 65535: Output value =65535+100 = FFFFH-0064H = 0063H = 99

- When the ZONE(P)_U instructions are used, (s1) is regarded as dummy data and ignored.

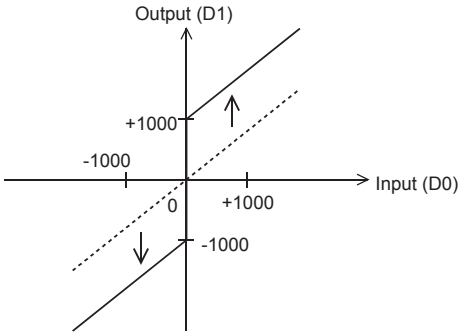
Program example

In the program example shown below, the data of D0 is controlled by the zone of the limit values “-1000” to “+1000”, and the controlled value is output to D1 when X0 is set to ON.



Operation

- In the case of “D0 < 0”, “D0 + (-1000)” is output to D1.
- In the case of “D0 = 0”, “0” is output to D1.
- In the case of “0 < D0”, “D0 + 1000” is output to D1.



Operation error

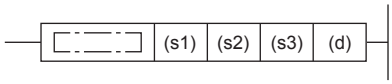
There is no operation error.

Zone control of 32-bit binary data

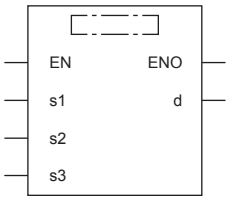
DZONE(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add the bias value specified by (s1) or (s2) to the input value specified by (s3), and store the operation result in the device specified by (d).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DZONE(P)	Negative bias value to be added to the input value	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DZONE(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DZONE(P)	Positive bias value to be added to the input value	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DZONE(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s3)	DZONE(P)	Input value for performing the zone control	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DZONE(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DZONE(P)	Head device number storing the output value controlled by the zone	—	32-bit signed binary	ANY32_S
	DZONE(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

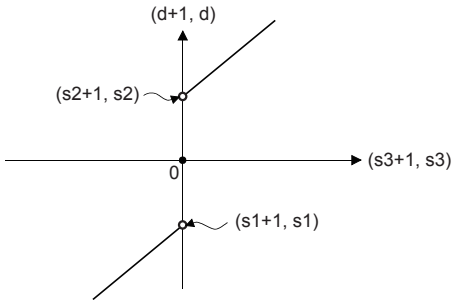
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions add the bias value specified by (s1) or (s2) to the input value (32-bit binary data) specified by (s3), and store the operation result in the device specified by (d). The bias value is controlled as follows.

Condition	Output value
Input value ((s3), (s3)+1) < 0	Input value ((s3), (s3)+1) + Negative bias value (s1), (s1)+1
Input value ((s3), (s3)+1) = 0	0
Input value ((s3), (s3)+1) > 0	Input value ((s3), (s3)+1) + Positive bias value (s2), (s2)+1



- When the output values to be stored in the devices specified by (d) and (d)+1 are 32-bit signed binary values and the operation result exceeds the range of -2147483648 to 2147483647, the output value is calculated as follows.

Ex.

When (s1) and (s1)+1 are -1000, and (s3) and (s3)+1 are -2147483648: Output value = -2147483648+(-1000) = 80000000H-FFFFFC18H = 7FFFFC18H = 2147482648

- When the output values to be stored in the devices specified by (d) and (d)+1 are 32-bit unsigned binary values and the operation result exceeds the range of 0 to 4294967295, the output value is calculated as follows.

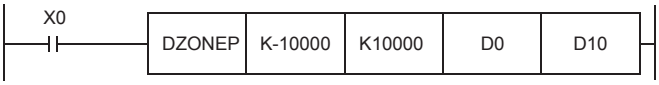
Ex.

When (s2) and (s2)+1 are 1000, and (s3) and (s3)+1 are 4294967295: Output value =4294967295+1000 = FFFFFFFFH-00003E8H = 000003E7H = 999

- When the DZONE(P)_U instructions are used, (s1) and (s1)+1 are regarded as dummy data and ignored.

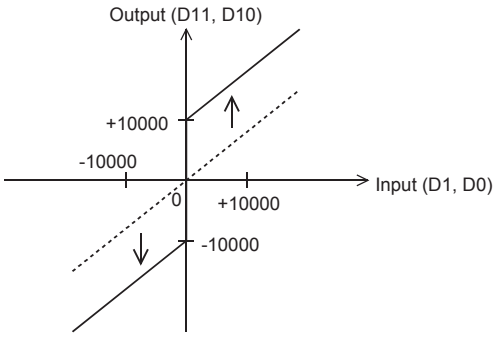
Program example

In the program example shown below, the data of D0 and D1 is controlled by the zone of the limit values “-10000” to “+10000”, and the controlled values are output to D10 and D11 when X0 is set to ON.



Operation

- In the case of “(D1, D0) < 0”, “(D1, D0) + (-10000)” is output to (D11, D10).
- In the case of “(D1, D0) = 0”, the “0” is output to (D11, D10).
- In the case of “0 < (D1, D0)”, “(D1, D0) + 10000” is output to (D11, D10).



Operation error

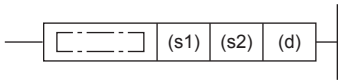
There is no operation error.

Scaling 16-bit binary data (point coordinates)

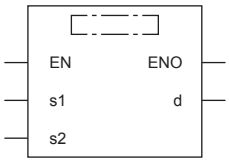
SCL(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions process the scaling conversion data (in 16-bit data units) specified by (s2) by scaling it based on the input value specified by (s1), and store the operation result in the device specified by (d).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	SCL(P)	Input value used in scaling or head device number storing the input value	-32768 to +32767	16-bit signed binary	ANY16_S
	SCL(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	SCL(P)	Head device number where the scaling conversion data is stored	—	16-bit signed binary*1	ANY16_S
	SCL(P)_U			16-bit unsigned binary*1	ANY16_U
(d)	SCL(P)	Head device number storing the output value controlled by scaling	—	16-bit signed binary	ANY16_S
	SCL(P)_U			16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 The number of coordinate points of (s2) is 16-bit unsigned binary data.

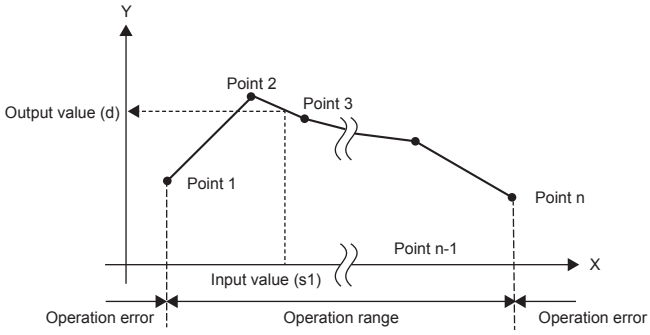
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

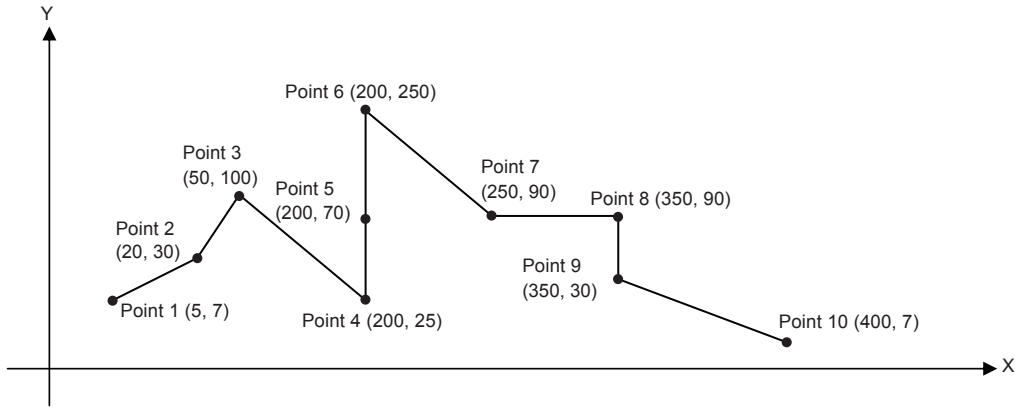
- These instructions process the scaling conversion data (in 16-bit data units) specified by (s2) by scaling it based on the input value specified by (s1), and store the operation result in the device number specified by (d). The scaling conversion is performed based on the scaling conversion data stored in the device specified by (s2) and later.

Setting item ("n" indicates the number of coordinate points specified by (s2).)		Device assignment
Number of coordinate points		(s2)
Point 1	X coordinate	(s2)+1
	Y coordinate	(s2)+2
Point 2	X coordinate	(s2)+3
	Y coordinate	(s2)+4
⋮		
Point n	X coordinate	(s2)+2n-1
	Y coordinate	(s2)+2n



- If the operation result is not an integer, the number in the first decimal place is rounded off.
- Set the X coordinate data of the scaling conversion data in the ascending order.
- Set (s1) within the scaling conversion data range (device value of (s2)).
- If the same X coordinate is specified by multiple points, the Y coordinate value of the point whose number is the largest is output.
- Set the number of coordinate points for the scaling conversion data within the range of 1 to 65535.
- Setting example of the conversion table for scaling

  In the case of the conversion characteristics for scaling shown in the figure below, set each value as shown in the following data table.



Setting item		Setting device and setting contents		
		When R0 is specified in (s2)		Setting details
Number of coordinate points		(s2)	R0	K10
Point 1	X coordinate	(s2)+1	R1	K5
	Y coordinate	(s2)+2	R2	K7
Point 2	X coordinate	(s2)+3	R3	K20
	Y coordinate	(s2)+4	R4	K30
Point 3*1	X coordinate	(s2)+5	R5	K50
	Y coordinate	(s2)+6	R6	K100

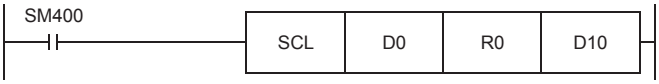
Setting item		Setting device and setting contents		
		When R0 is specified in (s2)		Setting details
Point 4*1	X coordinate	(s2)+7	R7	K200
	Y coordinate	(s2)+8	R8	K25
Point 5*1	X coordinate	(s2)+9	R9	K200
	Y coordinate	(s2)+10	R10	K70
Point 6	X coordinate	(s2)+11	R11	K200
	Y coordinate	(s2)+12	R12	K250
Point 7	X coordinate	(s2)+13	R13	K250
	Y coordinate	(s2)+14	R14	K90
Point 8	X coordinate	(s2)+15	R15	K350
	Y coordinate	(s2)+16	R16	K90
Point 9*2	X coordinate	(s2)+17	R17	K350
	Y coordinate	(s2)+18	R18	K30
Point 10*2	X coordinate	(s2)+19	R19	K400
	Y coordinate	(s2)+20	R20	K7

*1 When coordinates are specified using three points, the output value can be set to an intermediate value.
In this example, the output value (intermediate value) is specified by the Y coordinate of the point 5.
Even if the X coordinate is the same at three points or more, the value at the second point is output.

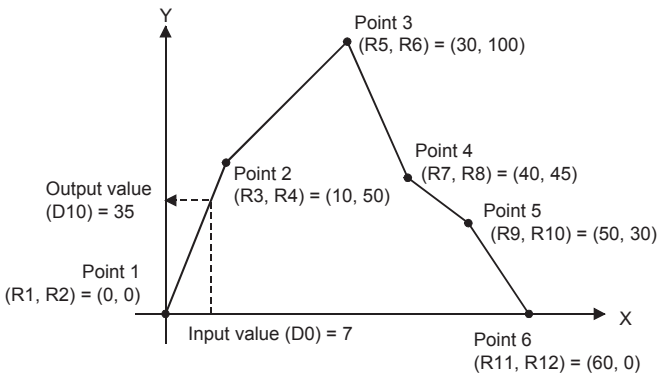
*2 When coordinates are specified using two points, the output value is the Y coordinate at the next point.
In this example, the output value is specified by the Y coordinate of the point 9.

Program example

In the program example shown below, the value input to D0 is processed by scaling based on the conversion table for scaling set in R0 and later, and output to D10. It is assumed that the values for the conversion table for scaling are set in R0 and later beforehand.



• Operation



• Conversion table for scaling

Set item		Device	Setting contents
Number of coordinate points		R0	K6
Point 1	X coordinate	R1	K0
	Y coordinate	R2	K0
Point 2	X coordinate	R3	K10
	Y coordinate	R4	K50
Point 3	X coordinate	R5	K30
	Y coordinate	R6	K100
Point 4	X coordinate	R7	K40
	Y coordinate	R8	K45
Point 5	X coordinate	R9	K50
	Y coordinate	R10	K30
Point 6	X coordinate	R11	K60
	Y coordinate	R12	K0

Operation error

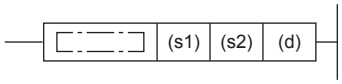
Error code (SD0/SD8067)	Description
3405H	The Xn data is not set in the ascending order in the data table. However, the instructions before the occurrence of an error are executed.
	The input value specified by (s1) is out of the range for the set scaling conversion data.
	The value in the middle of operation exceeds the 32-bit data range. In this case, verify that the distance between points is not "65535" or more. If the distance is "65535" or more, reduce the distance between points.
	The number of coordinate points from the device specified by (s2) is 0 or less.

Scaling 32-bit binary data (point coordinates)

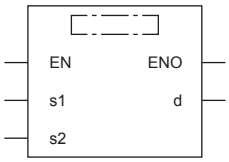
DSCL(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions process the scaling conversion data (in 32-bit data units) specified by (s2) by scaling it based on the input value specified by (s1), and store the operation result in the device specified by (d).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DSCL(P)	Input value used in scaling or head device number storing the input value	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DSCL(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DSCL(P)	Head device number where the scaling conversion data is stored	—	32-bit signed binary*1	ANY32_S
	DSCL(P)_U			32-bit unsigned binary*1	ANY32_U
(d)	DSCL(P)	Head device number storing the output value controlled by scaling	—	32-bit signed binary	ANY32_S
	DSCL(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 The numbers of coordinate points of (s2)+1 and (s2) are 32-bit unsigned binary data.

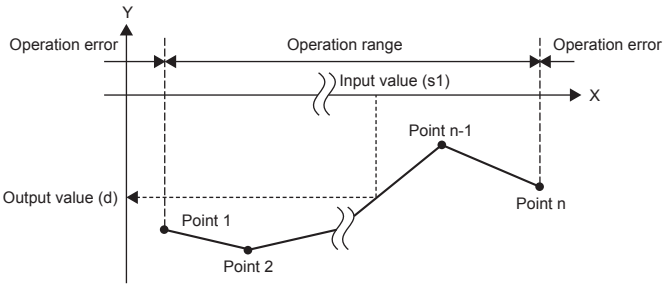
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

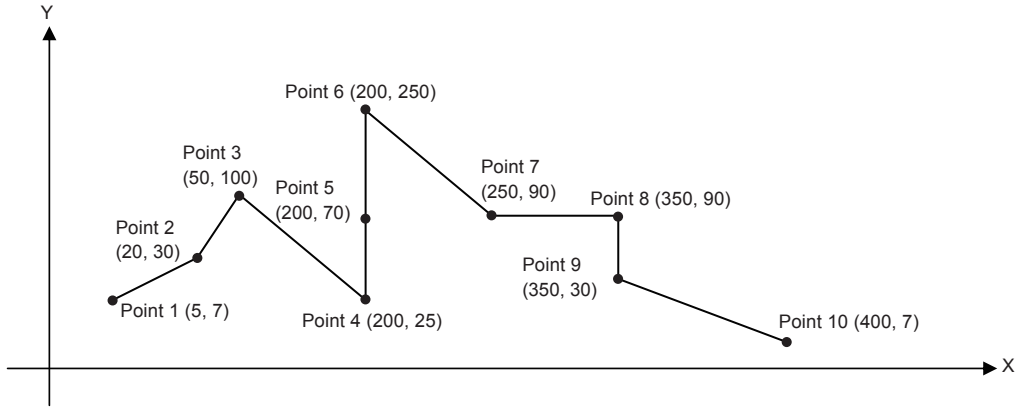
- These instructions process the scaling conversion data (in 32-bit data units) specified by (s2) by scaling it based on the input value specified by (s1), and store the operation result in the device number specified by (d). The scaling conversion is performed based on the scaling conversion data stored in the device specified by (s2) and later.

Setting item ("n" indicates the number of coordinate points specified by (s2).)		Device assignment
Number of coordinate points		(s2)+1, (s2)
Point 1	X coordinate	(s2)+3, (s2)+2
	Y coordinate	(s2)+5, (s2)+4
Point 2	X coordinate	(s2)+7, (s2)+6
	Y coordinate	(s2)+9, (s2)+8
⋮		
Point n	X coordinate	(s2)+4n-1, (s2)+4n-2
	Y coordinate	(s2)+4n+1, (s2)+4n



- If the operation result is not an integer, the number in the first decimal place is rounded off.
- Set the X coordinate data of the scaling conversion data in the ascending order.
- Set (s1) within the scaling conversion data range (device values of (s2) and (s2)+1).
- If the same X coordinate is specified by multiple points, the Y coordinate value of the point whose number is the largest is output.
- Set the number of coordinate points for the scaling conversion data within the range of 1 to 4294967295.
- Setting example of the conversion table for scaling

  In the case of the conversion characteristics for scaling shown in the figure below, set each value as shown in the following data table.



Setting item		Setting device and setting contents		
		When R0 is specified in (s2)		Setting details
Number of coordinate points		(s2)+1, (s2)	R1, R0	K10
Point 1	X coordinate	(s2)+3, (s2)+2	R3, R2	K5
	Y coordinate	(s2)+5, (s2)+4	R5, R4	K7
Point 2	X coordinate	(s2)+7, (s2)+6	R7, R6	K20
	Y coordinate	(s2)+9, (s2)+8	R9, R8	K30
Point 3	X coordinate	(s2)+11, (s2)+10	R11, R10	K50
	Y coordinate	(s2)+13, (s2)+12	R13, R12	K100

Setting item		Setting device and setting contents		
		When R0 is specified in (s2)		Setting details
Point 4*1	X coordinate	(s2)+15, (s2)+14	R15, R14	K200
	Y coordinate	(s2)+17, (s2)+16	R17, R16	K25
Point 5*1	X coordinate	(s2)+19, (s2)+18	R19, R18	K200
	Y coordinate	(s2)+21, (s2)+20	R21, R20	K70
Point 6*1	X coordinate	(s2)+23, (s2)+22	R23, R22	K200
	Y coordinate	(s2)+25, (s2)+24	R25, R24	K250
Point 7	X coordinate	(s2)+27, (s2)+26	R27, R26	K250
	Y coordinate	(s2)+29, (s2)+28	R29, R28	K90
Point 8*2	X coordinate	(s2)+31, (s2)+30	R31, R30	K350
	Y coordinate	(s2)+33, (s2)+32	R33, R32	K90
Point 9*2	X coordinate	(s2)+35, (s2)+34	R35, R34	K350
	Y coordinate	(s2)+37, (s2)+36	R37, R36	K30
Point 10	X coordinate	(s2)+39, (s2)+38	R39, R38	K400
	Y coordinate	(s2)+41, (s2)+40	R41, R40	K7

*1 When coordinates are specified using three points, the output value can be set to an intermediate value.
In this example, the output value (intermediate value) is specified by the Y coordinate of the point 5.
Even if the X coordinate is the same at three points or more, the value at the second point is output.

*2 When coordinates are specified using two points, the output value is the Y coordinate at the next point.
In this example, the output value is specified by the Y coordinate of the point 9.

Operation error

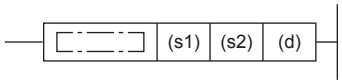
Error code (SD0/SD8067)	Description
3405H	The Xn data is not set in the ascending order in the data table. However, the instructions before the occurrence of an error are executed.
	The input value specified by (s1) is out of the range for the set scaling conversion data.
	The value in the middle of operation exceeds the 32-bit data range. In this case, verify that the distance between points is not "65535" or more. If the distance is "65535" or more, reduce the distance between points.
	The number of coordinate points from the device specified by (s2) is 0 or less.

Scaling 16-bit binary data (XY coordinates)

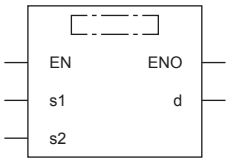
SCL2(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions process the scaling conversion data (in 16-bit data units) specified by (s2) by scaling it based on the input value specified by (s1), and store the operation result in the device specified by (d).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	SCL2(P)	Input value used in scaling or head device number storing the input value	-32768 to +32767	16-bit signed binary	ANY16_S
	SCL2(P)_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	SCL2(P)	Head device number where the scaling conversion data is stored	—	16-bit signed binary*1	ANY16_S
	SCL2(P)_U			16-bit unsigned binary*1	ANY16_U
(d)	SCL2(P)	Head device number storing the output value controlled by scaling	—	16-bit signed binary	ANY16_S
	SCL2(P)_U			16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 The number of coordinate points of (s2) is 16-bit unsigned binary data.

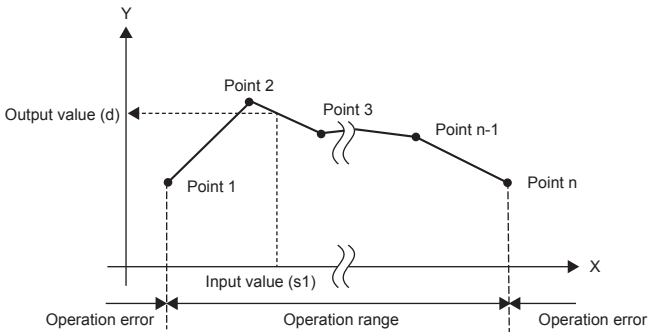
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions process the scaling conversion data (in 16-bit data units) specified by (s2) by scaling it based on the input value specified by (s1), and store the operation result in the device number specified by (d). The scaling conversion is performed based on the scaling conversion data stored in the device specified by (s2) and later.

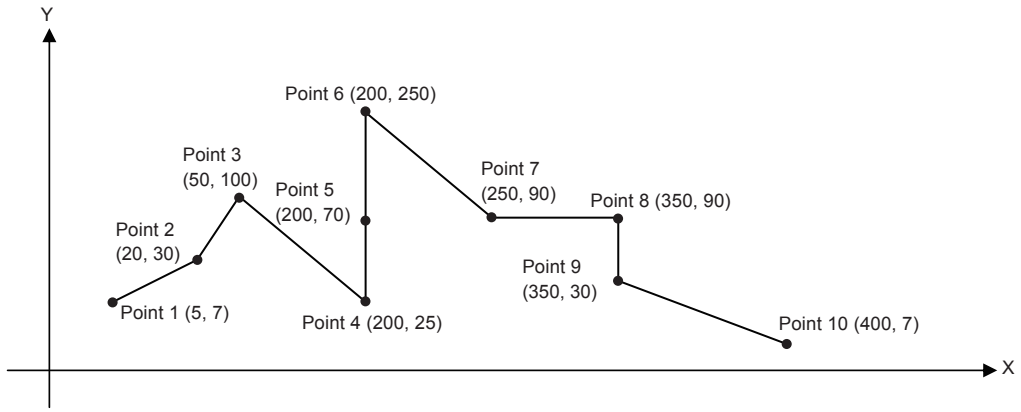
Setting item ("n" indicates the number of coordinate points specified by (s2).)		Device assignment
Number of coordinate points		(s2)
X coordinate	Point 1	(s2)+1
	Point 2	(s2)+2
	⋮	⋮
	Point n	(s2)+n
Y coordinate	Point 1	(s2)+n+1
	Point 2	(s2)+n+2
	⋮	⋮
	Point n	(s2)+2n



- If the operation result is not an integer, the number in the first decimal place is rounded off.
- Set the X coordinate data of the scaling conversion data in the ascending order.
- Set (s1) within the scaling conversion data range (device value of (s2)).
- If the same X coordinate is specified by multiple points, the Y coordinate value of the point whose number is the largest is output.
- Set the number of coordinate points for the scaling conversion data within the range of 1 to 65535.

• Setting example of the conversion table for scaling

In the case of the conversion characteristics for scaling shown in the figure below, set each value as shown in the following data table.



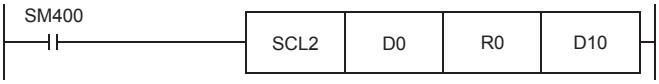
Setting item		Setting device and setting contents		
		When R0 is specified in (s2)		Setting details
Number of coordinate points		(s2)	R0	K10
X coordinate	Point 1	(s2)+1	R1	K5
	Point 2	(s2)+2	R2	K20
	Point 3	(s2)+3	R3	K50
	Point 4*1	(s2)+4	R4	K200
	Point 5*1	(s2)+5	R5	K200
	Point 6*1	(s2)+6	R6	K200
	Point 7	(s2)+7	R7	K250
	Point 8*2	(s2)+8	R8	K350
	Point 9*2	(s2)+9	R9	K350
	Point 10	(s2)+10	R10	K400
Y coordinate	Point 1	(s2)+11	R11	K7
	Point 2	(s2)+12	R12	K30
	Point 3	(s2)+13	R13	K100
	Point 4*1	(s2)+14	R14	K25
	Point 5*1	(s2)+15	R15	K70
	Point 6*1	(s2)+16	R16	K250
	Point 7	(s2)+17	R17	K90
	Point 8*2	(s2)+18	R18	K90
	Point 9*2	(s2)+19	R19	K30
	Point 10	(s2)+20	R20	K7

*1 When coordinates are specified using three points as shown in the points 4, 5 and 6, the output value can be set to an intermediate value.
In this example, the output value (intermediate value) is specified by the Y coordinate of the point 5.
Even if the X coordinate is the same at three points or more, the value at the second point is output.

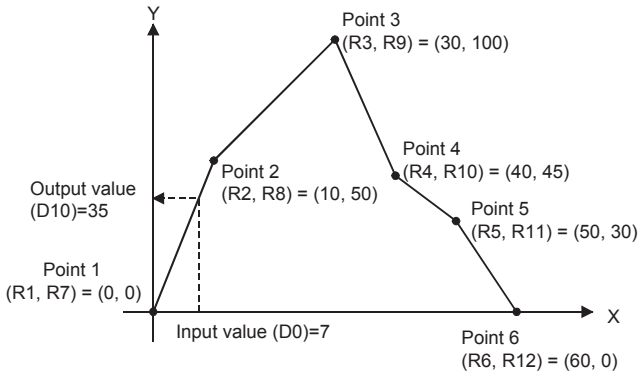
*2 When coordinates are specified using two points as shown in the points 8 and 9, the output value is the Y coordinate at the next point.
In this example, the output value is specified by the Y coordinate of the point 9.

Program example

In the program example shown below, the value input to D0 is processed by scaling based on the conversion table for scaling set in R0 and later, and output to D10. It is assumed that the values for the conversion table for scaling are set in R0 and later beforehand.



• Operation



• Conversion table for scaling

Set item		Device	Setting contents
Number of coordinate points		R0	K6
X coordinate	Point 1	R1	K0
	Point 2	R2	K10
	Point 3	R3	K30
	Point 4	R4	K40
	Point 5	R5	K50
	Point 6	R6	K60
Y coordinate	Point 1	R7	K0
	Point 2	R8	K50
	Point 3	R9	K100
	Point 4	R10	K45
	Point 5	R11	K30
	Point 6	R12	K0

Operation error

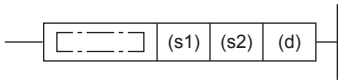
Error code (SD0/SD8067)	Description
3405H	The Xn data is not set in the ascending order in the data table. However, the instructions before the occurrence of an error are executed.
	The input value specified by (s1) is out of the range for the set scaling conversion data.
	The value in the middle of operation exceeds the 32-bit data range. In this case, verify that the distance between points is not "65535" or more. If the distance is "65535" or more, reduce the distance between points.
	The number of coordinate points from the device specified by (s2) is 0 or less.

Scaling 32-bit binary data (XY coordinates)

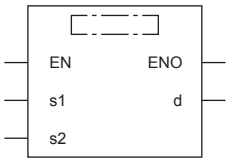
DSCL2(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions process the scaling conversion data (in 32-bit data units) specified by (s2) by scaling it based on the input value specified by (s1), and store the operation result in the device specified by (d).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DSCL2(P)	Input value used in scaling or head device number storing the input value	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DSCL2(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DSCL2(P)	Head device number where the scaling conversion data is stored	—	32-bit signed binary*1	ANY32_S
	DSCL2(P)_U			32-bit unsigned binary*1	ANY32_U
(d)	DSCL2(P)	Head device number storing the output value controlled by scaling	—	32-bit signed binary	ANY32_S
	DSCL2(P)_U			32-bit unsigned binary	ANY32_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 The numbers of coordinate points of (s2)+1 and (s2) are 32-bit unsigned binary data.

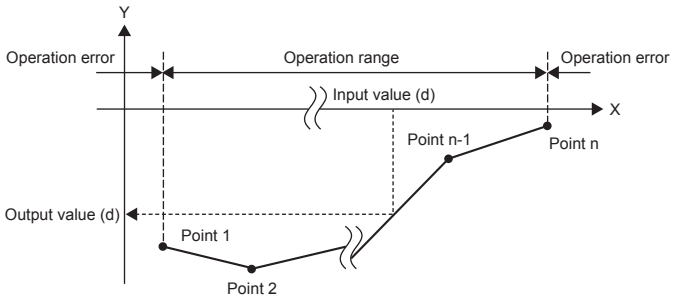
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions process the scaling conversion data (in 32-bit data units) specified by (s2) by scaling it based on the input value specified by (s1), and store the operation result in the device number specified by (d). The scaling conversion is performed based on the scaling conversion data stored in the device specified by (s2) and later.

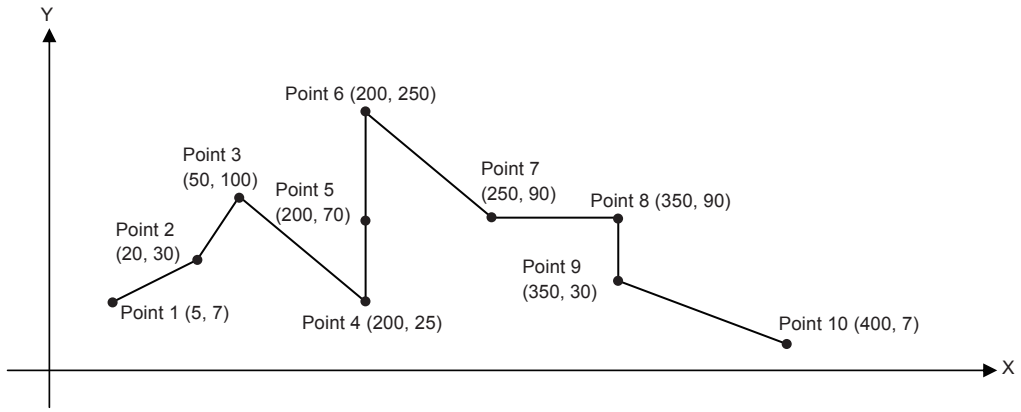
Setting item ("n" indicates the number of coordinate points specified by (s2).)		Device assignment
Number of coordinate points		(s2)+1, (s2)
X coordinate	Point 1	(s2)+3, (s2)+2
	Point 2	(s2)+5, (s2)+4
	⋮	⋮
	Point n	(s2)+2n+1, (s2)+2n
Y coordinate	Point 1	(s2)+2n+3, (s2)+2n+2
	Point 2	(s2)+2n+5, (s2)+2n+4
	⋮	⋮
	Point n	(s2)+4n+1, (s2)+4n



- If the operation result is not an integer, the number in the first decimal place is rounded off.
- Set the X coordinate data of the scaling conversion data in the ascending order.
- Set (s1) within the scaling conversion data range (device values of (s2) and (s2)+1).
- If the same X coordinate is specified by multiple points, the Y coordinate value of the point whose number is the largest is output.
- Set the number of coordinate points for the scaling conversion data within the range of 1 to 4294967295.

• Setting example of the conversion table for scaling

In the case of the conversion characteristics for scaling shown in the figure below, set each value as shown in the following data table.



Setting item		Setting device and setting contents		
		When R0 is specified in (s2)		Setting details
Number of coordinate points		(s2)+1, (s2)	R1, R0	K10
X coordinate	Point 1	(s2)+3, (s2)+2	R3, R2	K5
	Point 2	(s2)+5, (s2)+4	R5, R4	K20
	Point 3	(s2)+7, (s2)+6	R7, R6	K50
	Point 4*1	(s2)+9, (s2)+8	R9, R8	K200
	Point 5*1	(s2)+11, (s2)+10	R11, R10	K200
	Point 6*1	(s2)+13, (s2)+12	R13, R12	K200
	Point 7	(s2)+15, (s2)+14	R15, R14	K250
	Point 8*2	(s2)+17, (s2)+16	R17, R16	K350
	Point 9*2	(s2)+19, (s2)+18	R19, R18	K350
	Point 10	(s2)+21, (s2)+20	R21, R20	K400
Y coordinate	Point 1	(s2)+23, (s2)+22	R23, R22	K7
	Point 2	(s2)+25, (s2)+24	R25, R24	K30
	Point 3	(s2)+27, (s2)+26	R27, R26	K100
	Point 4*1	(s2)+29, (s2)+28	R29, R28	K25
	Point 5*1	(s2)+31, (s2)+30	R31, R30	K70
	Point 6*1	(s2)+33, (s2)+32	R33, R32	K250
	Point 7	(s2)+35, (s2)+34	R35, R34	K90
	Point 8*2	(s2)+37, (s2)+36	R37, R36	K90
	Point 9*2	(s2)+39, (s2)+38	R39, R38	K30
	Point 10	(s2)+41, (s2)+40	R41, R40	K7

*1 When coordinates are specified using three points as shown in the points 4, 5 and 6, the output value can be set to an intermediate value.
In this example, the output value (intermediate value) is specified by the Y coordinate of the point 5.
Even if the X coordinate is the same at three points or more, the value at the second point is output.

*2 When coordinates are specified using two points as shown in the points 8 and 9, the output value is the Y coordinate at the next point.
In this example, the output value is specified by the Y coordinate of the point 9.

Operation error

Error code (SD0/SD8067)	Description
3405H	The Xn data is not set in the ascending order in the data table. However, the instructions before the occurrence of an error are executed.
	The input value specified by (s1) is out of the range for the set scaling conversion data.
	The value in the middle of operation exceeds the 32-bit data range. In this case, verify that the distance between points is not "65535" or more. If the distance is "65535" or more, reduce the distance between points.
	The number of coordinate points from the device specified by (s2) is 0 or less.

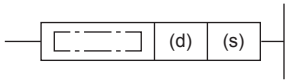
8.14 Special Timer Instruction

Teaching timer

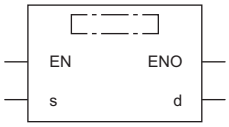
TTMR

FX5S FX5UJ FX5U FX5UC

This instruction measures the period of time in which TTMR instruction is ON.
Use this instruction to adjust the set value of a timer by a pushbutton switch.

Ladder diagram	Structured text
	ENO:=TTMR(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Device storing the teaching data	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 2)
(s)	Magnification applied to the teaching data	0 to 2	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○	—	—	—	—	○	—	—	—	—
(s)	○	○	○	○	—	—	○	○	—	—	—

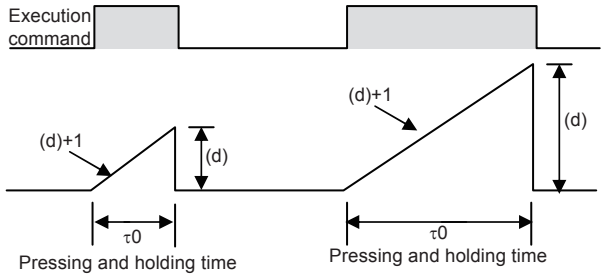
■Control data

Operand: (d)			
Device	Description	Setting range	Set by
+0	Teaching time	—	System
+1	Current value of the pressing and holding time	—	System

8

Processing details

- This instruction measures the period of time to press and hold the command input (pushbutton switch) in 1-second units, multiplies the measured value by the magnification (10^{s}) which is specified by (s), and stores it in the device specified by (d).



- The table below shows the actual value indicated by (d) depending on the magnification specified by (s) and the pressing and holding time τ0.

(s)	Magnification	(d)
K0	τ0	(d)×1
K1	10τ0	(d)×10
K2	100τ0	(d)×100

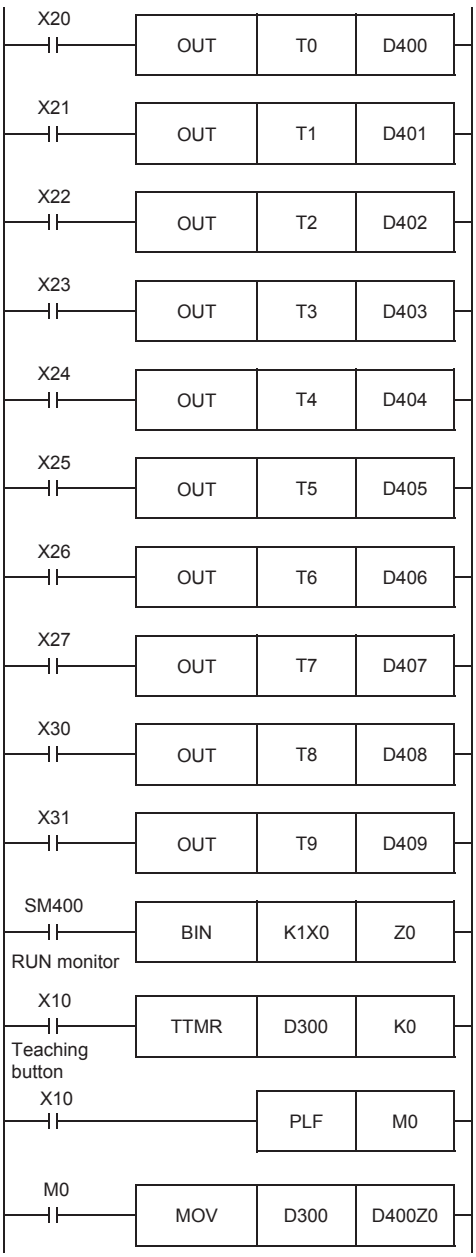
Precautions

- When the command contact turns from on to off, the current value (d)+1 of the pressing and holding time is cleared, and the teaching time (d) will not change any more.
- Two devices are occupied from a device specified as the teaching time (d). Make sure that such devices are not used in other controls for the machine.

Program example

• Writing the teaching time to 10 types of data registers

Suppose that the set value is written to D400 to D409 in advance.



10 timers to be set
Because the timers T0 to T9 are set to 100 ms, the actual operating time (sec) is 1/10 of the teaching data.

Selecting a timer by a digital switch
An input to the 1-digit digital switch connected to X0 to X3 is converted into binary format, and transferred to Z0.

Measuring the teaching time
The time (sec) in which X10 is pressed and held is stored to D300.

Recovery of the teaching time is detected.

Writing the set value of a timer
The teaching time (D300) is transferred to the device (D400Z0) for setting a timer selected by the digital switch.

Operation error

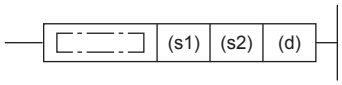
Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
3405H	The value specified by (s) is outside the following range. 0 to 2

Special function timer

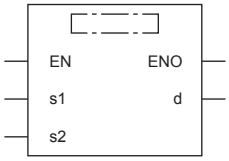
STMR

FX5S FX5UJ FX5U FX5UC

This instruction uses the four devices from the device specified by (d) to perform four types of timer output.

Ladder diagram	Structured text
	ENO:=STMR(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Used timer number (operates as a 100 ms timer)	—	Device name	ANY16
(s2)	Timer set value	1 to 32767	16-bit signed binary	ANY16
(d)	Start bit number to be output	—	Bit	ANYBIT_ARRAY (Number of elements: 4)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○*2	—	—	—	—	—	—	—	—	—

*1 Only T can be used.

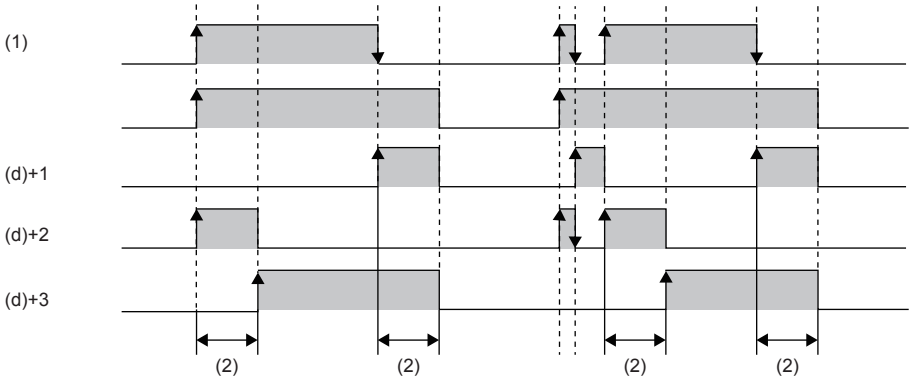
*2 T, ST, and C cannot be used.

■Control data

Operand: (d)			
Device	**Description**	**Setting range**	**Set by**
+0	Off delay timer output: Turns on at the rising edge of the command of the STMR instruction and turns off when the time specified by (s2) elapses after the falling edge.	—	System
+1	One-shot timer output after turning off: Turns on at the falling edge of the command of the STMR instruction and turns off when the time specified by (s2) elapses.	—	System
+2	One-shot timer output after turning on Turns on at the rising edge of the command of the STMR instruction and turns off when the command of the STMR instruction is turned off or when the time specified by (s2) elapses.	—	System
+3	On delay timer + Off delay timer output: Turns on at the falling edge of the timer coil and turns off when the time specified by (s2) elapsed after the falling edge of the command of the STMR instruction.	—	System

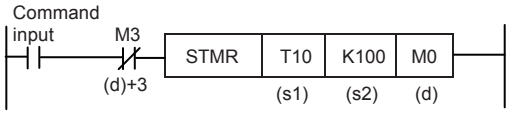
Processing details

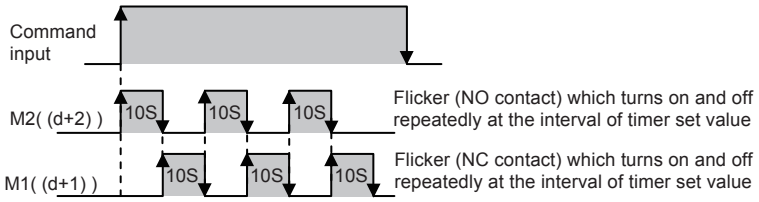
- This instruction uses the four devices from the device specified by (d) to perform four types of timer output.



(1): Command of the STMR instruction
(2): Setting value specified by (s2)

- The flickering effect is produced using (d)+1 and (d)+2 with the following program (T10 is assigned to (s1), K100 is assigned to (s2), and M0 is assigned to (d)), which turns on/off the STMR instruction at the normally closed contact of (d)+3.





- A value in the range of 0 to 32767 (0 to 3276.7 seconds) can be specified in (s2).

Precautions

- The timer number specified in this instruction cannot be used in other general circuits (such as OUT instruction). If the timer number is used in other general circuits, the timer malfunctions.
- The timer specified by (s1) starts counting as a 100 ms timer on the rising edge of the command contact.
- Four devices are occupied from a device specified in (d). Make sure that such devices are not used in other controls for the machine.
- If the command contact is turned off, (d), (d)+1, and (d)+3 turn off when the set time elapses. (d)+2 and the timer (s1) are immediately reset.

Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
3405H	The value specified by (s2) is outside the following range. 1 to 32767

8.15 Special Counter Instruction

Signed 32-bit bi-directional counters

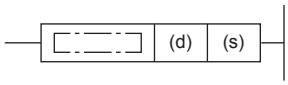
UDCNTF

FX5S FX5UJ FX5U FX5UC

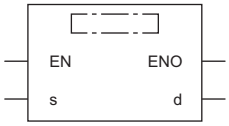
This instruction increments the current value of the counter specified by (d) by 1 when the operation result up to UDCNTF instruction changes from OFF to ON, and when the counter reaches the end of its count, NO contact becomes turns ON and NC contact becomes turns OFF. When the long counter specified by (d) is a high-speed counter, the instruction operates to start and stop the high-speed counter.

The high-speed pulse input/output module is not supported.

For details on the high-speed counter, refer to MELSEC iQ-F FX5 User's Manual (Application).

Ladder diagram	Structured text
	ENO:= UDCNTF(EN, s, d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Long counter number	—	Device name	ANY32
(s)	Long counter set value	-2147483648 to 2147483647	32-bit signed binary	ANY32_S*1
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

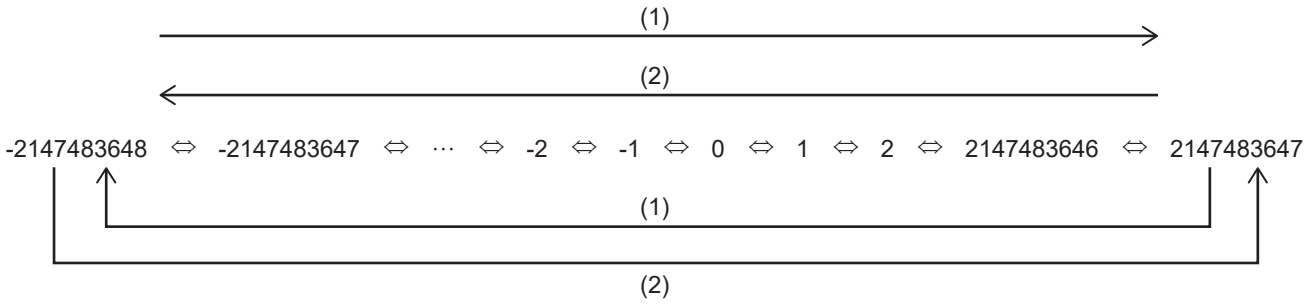
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	—	—	—	○	—	—	—	—	—	—
(s)	—	○*1	○	—	—	—	—	○*2	—	—	—

*1 T, ST, and C cannot be used.

*2 Only decimal constant (K) can be used.

Processing details

- This instruction increments the current value of the counter specified by (d) by 1 when the operation result up to UDCNTF instruction changes from OFF to ON, and when the counter reaches the end of its count, NO contact becomes turns ON and NC contact becomes turns OFF.
- When the long counter specified by (d) is a high-speed counter, the instruction operates to start and stop the high-speed counter, and the current counter value is updated by turning ON/OFF an external input instead of the drive contact. Counting up or down can be switched by whether the supported special device (SM4580 to SM4595) is on or off.
- When the long counter specified by (d) is a normal counter, the counting direction of LC0 to LC34 can be switched by turning on or off the corresponding special device (SM8200 to SM8234). For LC35 or later, only up counting is supported.
- During up-counting, the output contact turns ON when the current value changes from a value less than the set value to a value not less than the set value.
- During down-counting, the output contact turns OFF when the current value changes from a value not less than the set value to a value equivalent to the “set value -1” or less.
- Counting is continued also when the drive contact changes from OFF to ON after the output contact changes. (Only when specify other than a high-speed counter for (d))
- When unsigned (0 to 4294967295) is assigned to the LC, the OUT LC instruction is used. For the OUT LC instruction, refer to ☞ Page 137 OUT LC.
- The current value operates as a ring counter.



(1): Up counting
(2): Down counting

Precautions

- The last number of a word device cannot be input as the long counter set value.
- Indirect specification cannot be input as the long counter set value.
- When the set value or more is written to the current value of the up counter by the DMOV instruction, the counter is counted but the output contact does not turn on.
- When the value less than the set value is written to the current value of the down counter by the DMOV instruction, the counter is counted but the output contact does not turn off.
- When the long counter is cleared by the RST instruction, the current value will be cleared, and the contact will be turned off. After the counter is cleared, the counter will not count under the UDCNTF instruction until the RST instruction is executed in the contact OFF status.

Operation error

There is no operation error.

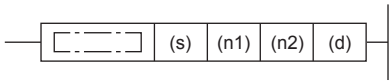
8.16 Shortcut Control Instruction

Rotary table shortest direction control

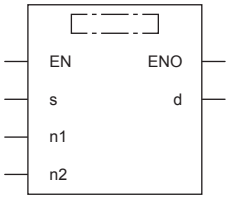
ROTC

FX5S FX5UJ FX5U FX5UC

This instruction is suitable for efficient control of the rotary table for putting/taking a product on/off the rotary table.

Ladder diagram	Structured text
	ENO:=ROTC(EN,s,n1,n2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

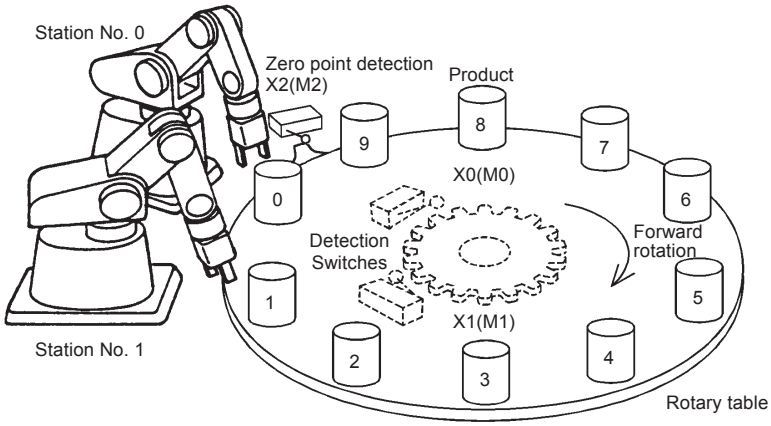
Operand	Description		Range	Data type	Data type (label)
(s)	Registers specifying the calling condition (Set them in advance using the transfer instruction.)	(s)+0: Works as a register for counting.	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
		(s)+1: Sets the station No. to be called.			
		(s)+2: Sets the product No. to be called.			
(n1)	Number of divisions		2 to 32767	16-bit unsigned binary	ANY16
(n2)	Number of low-speed sections		0 to 32767	16-bit unsigned binary	ANY16
(d)	Registers (bit devices) specifying the calling condition (Construct an internal contact circuit in advance which is driven by the input signal (X).)	(d): A phase signal	—	Bit	ANYBIT_ARRAY (Number of elements: 8)
		(d)+1: B phase signal			
		(d)+2: Zero point detection signal			
		(d)+3: Forward rotation at high-speed			
		(d)+4: Forward rotation at low-speed			
		(d)+5: Stop			
		(d)+6: Backward rotation at low-speed			
		(d)+7: Backward rotation at high-speed			
EN	Execution condition		—	Bit	BOOL
ENO	Execution result		—	Bit	BOOL

■Applicable devices

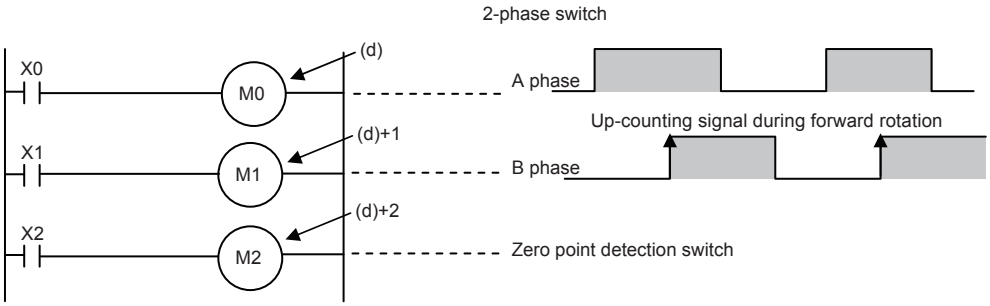
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	—	—	—	—	—	—	—	—	—	—

Processing details

- The table rotation is controlled by conditions of "n2", (s), and (d) so that a product can be efficiently put on or taken off the rotary table divided into "n1" (=10) sections as shown in the figure below. When the following conditions are specified, forward/backward rotation and high-speed/low-speed/stop are output to (d)+3 to (d)+7.



- Provide a 2-phase switch (X0 and X1) for detecting the rotation direction (forward or backward) of the table and the switch X2 which turns ON when the product No. 0 reaches the station No. 0. X0 to X2 are replaced with internal contacts of (d) to (d)+2. Any head device number can be specified by X or (d).



- The counter (s) detects which product number is located at the station No. 0.
- Set the station No. to be called in (s)+1.
- Set the product No. to be called in (s)+2.
- Specify the number of divisions (n1) of the table, and number of low-speed sections (n2).

Precautions

- When the command input is set to ON and this instruction is executed, the result will be automatically output to (d)+3 to (d)+7. When the command input is set to OFF, (d)+3 to (d)+7 are set to OFF accordingly.
- For example, when the rotation detection signal ((d) to (d)+2) is activated 10 times in one division, set a value multiplied by "10" to each division, station No. to be called and product No. to be called. As a result, an intermediate value of the division number can be set to a low-speed section.
- When the zero point detection signal (M2) turns ON while the command input is ON, the contents of the register for counting (s) are cleared to "0". This clear operation should be executed before starting the operation.
- Up to four ROTC instructions can be used simultaneously.

Operation error

Error code (SD0/SD8067)	Description
1811H	The number of the ROTC instructions which are used simultaneously exceeds four.
2820H	The device range specified by (s) exceeds the corresponding device range.
	The device range specified by (d) exceeds the corresponding device range.
3405H	The value specified by (n1) is outside the following range. 2 to 32767
	The value specified by (n2) is outside the following range. 0 to 32767
	The value specified by (n1) or (n2) is in the following condition. (n1) < (n2)
	Either (s), (s)+1, or (s)+2 is negative.
	Either (s), (s)+1, or (s)+2 is equal to (n1) or larger.

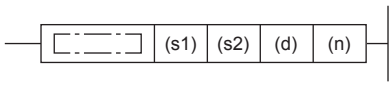
8.17 Ramp Signal Instruction

Ramp signal

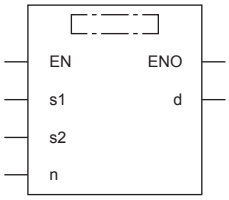
RAMPF

FX5S FX5UJ FX5U FX5UC

This instruction obtains the data which changes between the start value (initial value) and the end value (target value) over the specified "n" times.

Ladder diagram	Structured text
	ENO:=RAMPF(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

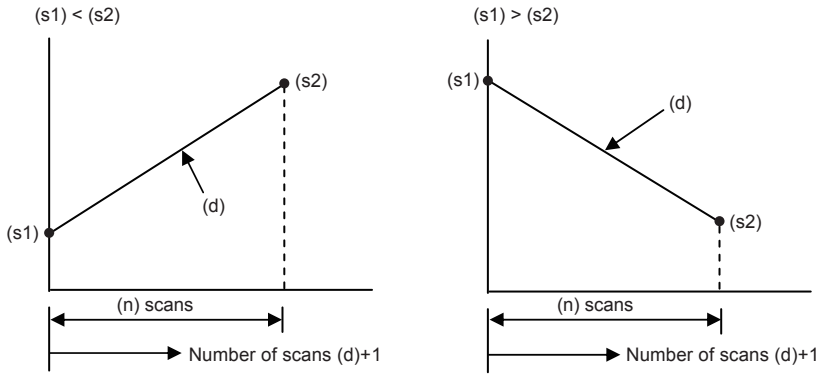
Operand	Description	Range	Data type	Data type (label)
(s1)	Initial value of ramp	—	16-bit signed binary	ANY16
(s2)	Target value of ramp	—	16-bit signed binary	ANY16
(d)	(d)+0: Current value	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 2)
	(d)+1: Number of scans			
(n)	Ramp transfer time (scan)	1 to 32767	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

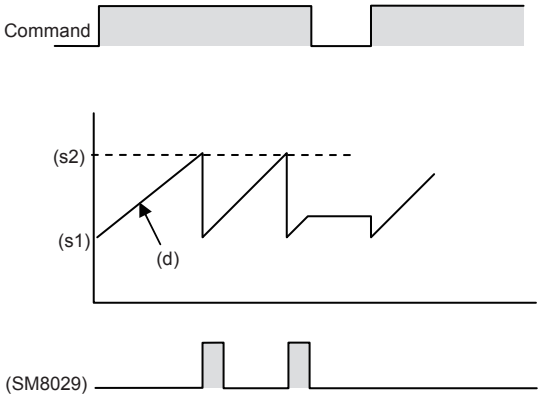
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- When the start value (s1) and the end value (s2) have been specified and the command input is set to ON, the value obtained by adding a value divided equally by "n" times to (s1) in the next operation cycle is stored to (d). By combining this instruction and an analog output, the cushion start/stop command can be output.



- The number of scans ("0" to "n") is stored in (d)+1.
- The time from start to the end value is the operation cycle multiplied by "n" times.
- If the command input is set to OFF in the middle of operation, execution is paused. (The current value stored in (d) is held, and the number of scans stored in (d)+1 is cleared.) When the command input is set to ON again, (d) is cleared, and the operation is started from (s1).
- After transfer is completed, the instruction execution complete flag SM8029 turns ON, and the (d) value is returned to the (s1) value.



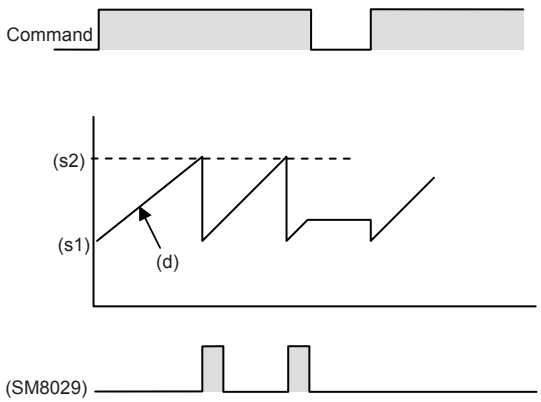
- When the operation result is acquired at a constant time interval (constant scan mode), write a prescribed scan time (which is longer than the actual scan time) to SD8039 and set SM8039 to ON. For example, when "20 ms" is written to SD8039 and "n" is set to 100, the (d) value will change from (s1) to (s2) in 2 seconds.
- The value used in the constant scan mode can be set in the parameter setting of an engineering tool (constant scan execution interval setting of CPU parameter).

For details on the constant scan, refer to MELSEC iQ-F FX5 User's Manual (Application).

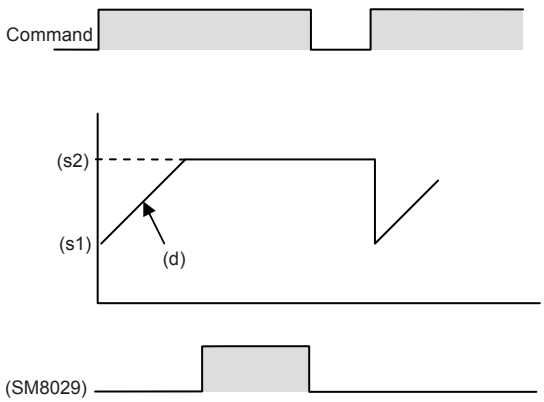
For details on the engineering tool, refer to GX Works3 Operating Manual.

• The contents of (d) are changed as follows depending on the ON/OFF status of the mode flag SM8026.

When SM8026 is off



When SM8026 is on



Precautions

To specify a latched (battery backed) type device as (d) when setting the CPU module to the RUN mode while the command input is ON, clear (d) in advance.

Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
3405H	The value specified by (n) is outside the following range. 1 to 32767

8.18 Pulse Related Instruction

Instruction correspondence table for CPU module built-in positioning and high-speed pulse input/output module

The table for instructions corresponding to the CPU module and the high-speed pulse input/output module are shown below.

○: Supported, —: Not supported

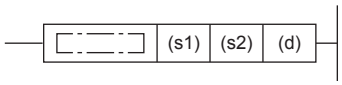
Instruction	CPU module	High-speed pulse input/output module
SPD	○	—
DSPD	○	—
PLSY [For the FX3 compatible operand specification]	○	—
PLSY [For the FX5 operand specification]	○	—
DPLSY [For the FX3 compatible operand specification]	○	—
DPLSY [For the FX5 operand specification]	○	—
PWM	○	○
DPWM	○	○

Measuring the density of 16 bit binary pulses

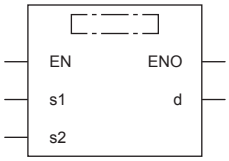
SPD

FX5S FX5UJ FX5U FX5UC

This instruction counts the number of times the device input specified by (s1) turns off → on only for the time (in 16-bit data units) specified by (s2) × 1ms and stores the operation result in the device specified by (d). The high-speed pulse input/output module is not supported.

Ladder diagram	Structured text
	ENO:=SPD(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Pulse input	—	Bit/Word	ANY_ELEMENTARY*1
(s2)	Measurement time (Unit: ms)	-32768 to +32767	16-bit signed binary	ANY16
(d)	Head device number for storing the measurement result	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

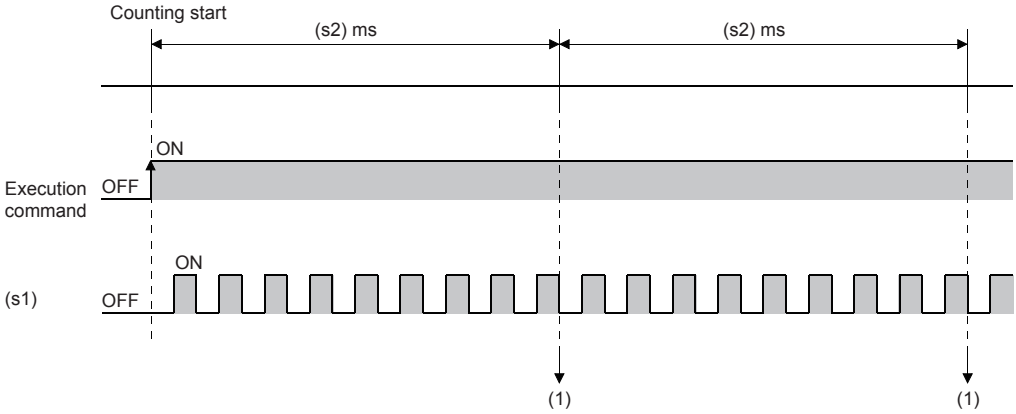
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○*1	○*2	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○	—	○	—	—	○	—	—	—	—

*1 Only X can be used for a bit device. Designate with the following range.
- FX5S/FX5UJ CPU module: X0 to X7
- FX5U/FX5UC CPU module: X0 to X17
Note that bit device digit designation cannot be used.

*2 When a word device is specified, specify one of the channel numbers (CH1 to CH8).
When FX3 compatible function of a high-speed counter is valid, a channel number cannot be specified. Only X can be used for a bit device.
If the channel numbers is specified in (s1), an error occurs.

Processing details

- This instruction counts the number of times the device input specified by (s1) turns off → on only for the time (in 16-bit data units) specified by (s2) × 1ms and stores the operation result in the device specified by (d).



(1): The elapsed time is judged using the 1ms interrupt and the counted result is stored in (d).

- The channel number of the high-speed counter specified by (s1) interlocks with the channel number in which parameters are set.
- When a word device is specified by (s1), this instruction counts the number of pulses by the high-speed counter setting of the channel number corresponding to each word device.
- When a bit device is specified by (s1), the following input assignment devices (shaded area) are valid.
- When the high-speed counter FX3 compatibility function is enabled, only the 1-phase 1-input pulse input mode (S/W) (general-purpose input assignment of the 1-phase 1-input counter (switching S/W up or down)) is enabled.

[FX5S/FX5UJ CPU module]

General-purpose input assignment of the 1-phase 1-input counter (switching S/W up or down)

U/D: UP/DOWN pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	U/D(A)	P					E									
CH2		U/D(A)	P					E								
CH3			U/D(A)	P					E							
CH4				U/D(A)	P					E						
CH5					U/D(A)	P					E					
CH6						U/D(A)	P					E				
CH7							U/D(A)	P					E			
CH8								U/D(A)	P					E		

General-purpose input assignment of the 1-phase 1-input counter (switching H/W up or down)

C: Pulse input, D: Direction input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	C(A)	D(B)	P				E									
CH2		C(A)	D(B)	P				E								
CH3			C(A)	D(B)	P				E							
CH4				C(A)	D(B)	P				E						
CH5					C(A)	D(B)	P				E					
CH6						C(A)	D(B)	P				E				
CH7							C(A)	D(B)	P				E			
CH8								C(A)	D(B)	P				E		

General-purpose input assignment of the 1-phase 2-input counter

U: UP pulse input, D: DOWN pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	U(A)	D(B)	P				E									
CH2		U(A)	D(B)	P				E								
CH3			U(A)	D(B)	P				E							
CH4				U(A)	D(B)	P				E						
CH5					U(A)	D(B)	P				E					
CH6						U(A)	D(B)	P				E				
CH7							U(A)	D(B)	P				E			

General-purpose input assignment of the 2-phase 2-input counter

A: A phase pulse input, B: B phase pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	A	B	P				E									
CH4				A	B	P				E						
CH6						A	B	P				E				
CH7							A	B	P				E			

If one of X1, X2, X4, X5, X7 is specified as a device, an error occurs.

[FX5U/FX5UC CPU module]

General-purpose input assignment of the 1-phase 1-input counter (switching S/W up or down)

U/D: UP/DOWN pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	U/D(A)								P	E						
CH2		U/D(A)									P	E				
CH3			U/D(A)										P	E		
CH4				U/D(A)											P	E
CH5					U/D(A)				P	E						
CH6						U/D(A)					P	E				
CH7							U/D(A)						P	E		
CH8								U/D(A)							P	E

If one of X10 to X17 is specified as a device, an error occurs.

General-purpose input assignment of the 1-phase 1-input counter (switching H/W up or down)

C: Pulse input, D: Direction input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	C(A)	D(B)							P	E						
CH2			C(A)	D(B)							P	E				
CH3					C(A)	D(B)							P	E		
CH4							C(A)	D(B)							P	E
CH5									C(A)	D(B)	P	E				
CH6											C(A)	D(B)	P	E		
CH7													C(A)	D(B)	P	E
CH8															C(A)	D(B)

If one of X1, X3, X5, X7, X11, X13, X15, X17 is specified as a device, an error occurs.

General-purpose input assignment of the 1-phase 2-input counter

U: UP pulse input, D: DOWN pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	U(A)	D(B)							P	E						
CH2			U(A)	D(B)							P	E				
CH3					U(A)	D(B)							P	E		
CH4							U(A)	D(B)							P	E
CH5									U(A)	D(B)	P	E				
CH6											U(A)	D(B)	P	E		
CH7													U(A)	D(B)	P	E
CH8															U(A)	D(B)

If one of X1, X3, X5, X7, X11, X13, X15, X17 is specified as a device, an error occurs.

General-purpose input assignment of the 2-phase 2-input counter

A: A phase pulse input, B: B phase pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	A	B							P	E						
CH2			A	B							P	E				
CH3					A	B							P	E		
CH4							A	B							P	E
CH5									A	B	P	E				
CH6											A	B	P	E		
CH7													A	B	P	E
CH8															A	B

If one of X1, X3, X5, X7, X11, X13, X15, X17 is specified as a device, an error occurs.

• The table below shows the related devices.

Function	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
Monitor in operation	SM4500	SM4501	SM4502	SM4503	SM4504	SM4505	SM4506	SM4507
High-speed counter pulse density	SD4507, SD4506	SD4537, SD4536	SD4567, SD4566	SD4597, SD4596	SD4627, SD4626	SD4657, SD4656	SD4687, SD4686	SD4717, SD4716
Measurement unit time	SD4517, SD4516	SD4547, SD4546	SD4577, SD4576	SD4607, SD4606	SD4637, SD4636	SD4667, SD4666	SD4697, SD4696	SD4727, SD4726

• The table below shows the related device update timing.

Function	R/W	Update timing	Clear
Monitor in operation	R	• When the SPD instruction is executed • When the HIOEN instruction is executed	• Power-on • Reset
High-speed counter pulse density	R	• Each time the measurement unit time elapses	• Power-on • Reset
Measurement unit time	R/W	• When the SPD instruction is executed	• Power-on • Reset

Precautions

• The maximum input frequency of turning the inputs ON and OFF is shown below:

■FX5S/FX5UJ CPU module

Used input number	Maximum input frequency
X0, X1, X3, X4	100 kHz
X2, X5, X6, X7	10 kHz

■FX5U-32M□/ FX5UC-32M□ CPU module

Used input number	Maximum input frequency
X0 to X5	200 kHz
X6, X7	10 kHz

■FX5U-64M□/FX5U-80M□/FX5UC-64M□/FX5UC-96M□ CPU module

Used input number	Maximum input frequency
X0 to X7	200 kHz
X10 to X17	10 kHz

- When the SPD instruction is used, the UP/DOWN pulse input, preset input and enable input operate in accordance with the contents set by the parameters of the high-speed counter.
- When the measurement time is changed while the SPD instruction is executed, the changed time is applied every time the measurement time ends.
- When the current value of the high-speed counter is overwritten, a preset input is executed, or the high-speed counter is reset by the DHCMOV instruction while the SPD instruction is executed, the operation continues, but the pulse density cannot be measured normally.
- When the SPD instruction is used, pulses per unit time which exceeds the ring length of the high-speed counter cannot be input. When pulses are input, the pulse density cannot be measured normally.
- The measurement time specified by (s2) overwrites the value stored in the SD device specified for the measurement unit time.
- When the measurement time specified by (s2) is outside the range from 1 to 32767, the specified measurement time is rounded into “1” with the sign.

Operation error

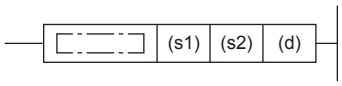
Error code (SD0/SD8067)	Description
1810H	The input specified in (s1) is already used by another instruction.
3405H	An unavailable bit device is set in (s1).
	A channel number other than 1 to 8 is specified in (s1).
	When FX3 compatible function of a high-speed counter is valid and a channel number is specified to (s1).
3600H	The channel number or device number in which parameters are not set in (s1) is specified.

Measuring the density of 32 bit binary pulses

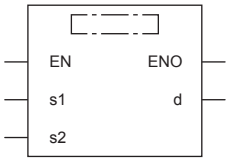
DSPD

FX5S FX5UJ FX5U FX5UC

This instruction counts the number of times the device input specified by (s1) turns off → on only for the time (in 32-bit data units) specified by (s2) × 1ms and stores the operation result in the device specified by (d). The high-speed pulse input/output module is not supported.

Ladder diagram	Structured text
	ENO:=DSPD(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Pulse input	—	Bit/32-bit unsigned binary	ANY_ELEMENTARY*1
(s2)	Measurement time (Unit: ms)	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d)	Head device number for storing the measurement result	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○*1	○*2	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	—	○	—	○	○	○	○	—	—	—	—

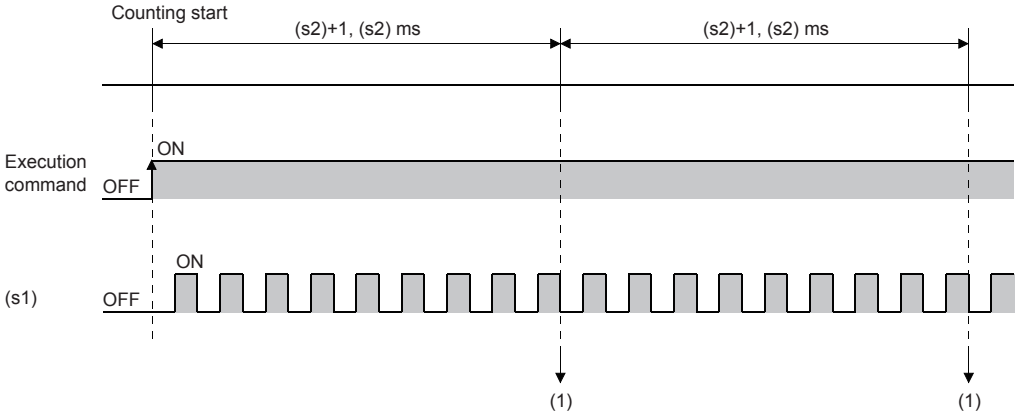
*1 Only X can be used for a bit device. Designate with the following range.
- FX5S/FX5UJ CPU module: X0 to X7
- FX5U/FX5UC CPU module: X0 to X17

Note that bit device digit designation cannot be used.

*2 When a word device is specified, specify one of the channel numbers (CH1 to CH8).
When FX3 compatible function of a high-speed counter is valid, a channel number cannot be specified. Only X can be used for a bit device.
If the channel numbers is specified in (s1), an error occurs.

Processing details

- This instruction counts the number of times the device input specified by (s1) turns off → on only for the time (in 32-bit data units) specified by (s2) × 1ms and stores the operation result in the device specified by (d).



(1): The elapsed time is judged using the 1ms interrupt and the counted result is stored in (d)+1 and (d).

- The channel number of the high-speed counter specified by (s1) interlocks with the channel number in which parameters are set.
- When a word device is specified by (s1), this instruction counts the number of pulses by the high-speed counter setting of the channel number corresponding to each word device.
- When a bit device is specified by (s1), the following input assignment devices (shaded area) are valid.
- When the high-speed counter FX3 compatibility function is enabled, only the 1-phase 1-input pulse input mode (S/W) (general-purpose input assignment of the 1-phase 1-input counter (switching S/W up or down)) is enabled.

[FX5S/FX5UJ CPU module]

General-purpose input assignment of the 1-phase 1-input counter (switching S/W up or down)

U/D: UP/DOWN pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	U/D(A)	P					E									
CH2		U/D(A)	P					E								
CH3			U/D(A)	P					E							
CH4				U/D(A)	P					E						
CH5					U/D(A)	P					E					
CH6						U/D(A)	P					E				
CH7							U/D(A)	P					E			
CH8								U/D(A)	P					E		

General-purpose input assignment of the 1-phase 1-input counter (switching H/W up or down)

C: Pulse input, D: Direction input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	C(A)	D(B)	P				E									
CH2		C(A)	D(B)	P				E								
CH3			C(A)	D(B)	P				E							
CH4				C(A)	D(B)	P				E						
CH5					C(A)	D(B)	P				E					
CH6						C(A)	D(B)	P				E				
CH7							C(A)	D(B)	P				E			
CH8								C(A)	D(B)	P				E		

General-purpose input assignment of the 1-phase 2-input counter

U: UP pulse input, D: DOWN pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	U(A)	D(B)	P				E									
CH2		U(A)	D(B)	P				E								
CH3			U(A)	D(B)	P				E							
CH4				U(A)	D(B)	P				E						
CH5					U(A)	D(B)	P				E					
CH6						U(A)	D(B)	P				E				
CH7							U(A)	D(B)	P				E			

General-purpose input assignment of the 2-phase 2-input counter

A: A phase pulse input, B: B phase pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	A	B	P				E									
CH4				A	B	P				E						
CH6						A	B	P				E				
CH7							A	B	P				E			

If one of X1, X2, X4, X5, X7 is specified as a device, an error occurs.

[FX5U/FX5UC CPU module]

General-purpose input assignment of the 1-phase 1-input counter (switching S/W up or down)

U/D: UP/DOWN pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	U/D(A)								P	E						
CH2		U/D(A)									P	E				
CH3			U/D(A)										P	E		
CH4				U/D(A)											P	E
CH5					U/D(A)				P	E						
CH6						U/D(A)					P	E				
CH7							U/D(A)						P	E		
CH8								U/D(A)							P	E

If one of X10 to X17 is specified as a device, an error occurs.

General-purpose input assignment of the 1-phase 1-input counter (switching H/W up or down)

C: Pulse input, D: Direction input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	C(A)	D(B)							P	E						
CH2			C(A)	D(B)							P	E				
CH3					C(A)	D(B)							P	E		
CH4							C(A)	D(B)							P	E
CH5									C(A)	D(B)	P	E				
CH6											C(A)	D(B)	P	E		
CH7													C(A)	D(B)	P	E
CH8															C(A)	D(B)

If one of X1, X3, X5, X7, X11, X13, X15, X17 is specified as a device, an error occurs.

General-purpose input assignment of the 1-phase 2-input counter

U: UP pulse input, D: DOWN pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	U(A)	D(B)							P	E						
CH2			U(A)	D(B)							P	E				
CH3					U(A)	D(B)							P	E		
CH4							U(A)	D(B)							P	E
CH5									U(A)	D(B)	P	E				
CH6											U(A)	D(B)	P	E		
CH7													U(A)	D(B)	P	E
CH8															U(A)	D(B)

If one of X1, X3, X5, X7, X11, X13, X15, X17 is specified as a device, an error occurs.

General-purpose input assignment of the 2-phase 2-input counter

A: A phase pulse input, B: B phase pulse input, P: Preset input (reset), E: Enable input (start)

	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
CH1	A	B							P	E						
CH2			A	B							P	E				
CH3					A	B							P	E		
CH4							A	B							P	E
CH5									A	B	P	E				
CH6											A	B	P	E		
CH7													A	B	P	E
CH8															A	B

If one of X1, X3, X5, X7, X11, X13, X15, X17 is specified as a device, an error occurs.

• The table below shows the related devices.

Function	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
Monitor in operation	SM4500	SM4501	SM4502	SM4503	SM4504	SM4505	SM4506	SM4507
High-speed counter pulse density	SD4507, SD4506	SD4537, SD4536	SD4567, SD4566	SD4597, SD4596	SD4627, SD4626	SD4657, SD4656	SD4687, SD4686	SD4717, SD4716
Measurement unit time	SD4517, SD4516	SD4547, SD4546	SD4577, SD4576	SD4607, SD4606	SD4637, SD4636	SD4667, SD4666	SD4697, SD4696	SD4727, SD4726

• The table below shows the related device update timing.

Function	R/W	Update timing	Clear
Monitor in operation	R	• When the DSPD instruction is executed • When the DHIOEN instruction is executed	• Power-on • Reset
High-speed counter pulse density	R	• Each time the measurement unit time elapses	• Power-on • Reset
Measurement unit time	R/W	• When the DSPD instruction is executed	• Power-on • Reset

Precautions

• The maximum input frequency of turning the inputs ON and OFF is shown below:

■FX5S/FX5UJ CPU module

Used input number	Maximum input frequency
X0, X1, X3, X4	100 kHz
X2, X5, X6, X7	10 kHz

■FX5U-32M□/ FX5UC-32M□ CPU module

Used input number	Maximum input frequency
X0 to X5	200 kHz
X6, X7	10 kHz

■FX5U-64M□/FX5U-80M□/FX5UC-64M□/FX5UC-96M□ CPU module

Used input number	Maximum input frequency
X0 to X7	200 kHz
X10 to X17	10 kHz

- When the DSPD instruction is used, the UP/DOWN pulse input, preset input and enable input operate in accordance with the contents set by the parameters of the high-speed counter.
- When the measurement time is changed while the DSPD instruction is executed, the changed time is applied every time the measurement time ends.
- When the current value of the high-speed counter is overwritten, a preset input is executed, or the high-speed counter is reset by the DHCMOV instruction while the DSPD instruction is executed, the operation continues, but the pulse density cannot be measured normally.
- When the DSPD instruction is used, pulses per unit time which exceeds the ring length of the high-speed counter cannot be input. When pulses are input, the pulse density cannot be measured normally.
- The measurement time specified by (s2) overwrites the value stored in the SD device specified for the measurement unit time.
- When the measurement time specified by (s2) is outside the range from 1 to 2147483647, the specified measurement time is rounded into “1” with the sign.

Operation error

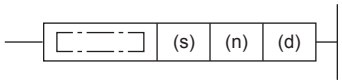
Error code (SD0/SD8067)	Description
1810H	The input specified in (s1) is already used by another instruction.
3405H	An unavailable bit device is set in (s1).
	A channel number other than 1 to 8 is specified in (s1).
	When FX3 compatible function of a high-speed counter is valid and a channel number is specified to (s1).
3600H	The channel number or device number in which parameters are not set in (s1) is specified.

16 bit binary pulse output

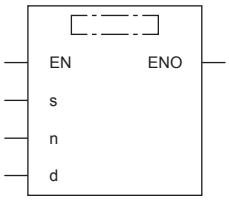
PLSY [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction outputs 16-bit pulse trains specified by the command speed (s) from the device specified by the output (d) for the amount of 16-bit pulses specified by the positioning address (n). Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=PLSY(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Command speed or word device number storing data	0 to 65535	16-bit unsigned binary	ANY16
(n)	Positioning address or word device number storing data	0 to 65535	16-bit unsigned binary	ANY16
(d)	Bit device number (Y) from which pulses are output	■FX5S/FX5U/FX5UC CPU module 0 to 3 ■FX5UJ CPU module 0 to 2	Bit	ANY_ELEMENTARY (BOOL)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

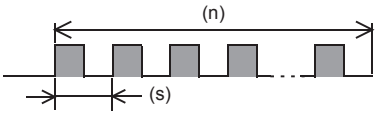
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—
(d)	○*1	—	—	—	—	—	—	—	—	—	—

*1 FX5S/FX5U/FX5UC CPU module: Only Y0 to Y3 devices can be used.
FX5UJ CPU module: Only Y0 to Y2 devices can be used.

Processing details

- This instruction outputs 16-bit pulse trains specified by the command speed (s) from the device specified by the output (d) for the amount of 16-bit pulses specified by the positioning address (n).



- Set the value from 0 to 65535 (in user unit) to the command speed (s), so that the command speed is 200 kpps or less when the command speed is converted to frequency.
- Set the value from 0 to 65535 (in user unit) to the positioning address (n), so that the positioning address is within the range from 0 to 2147483647 when the positioning address is converted to number of pulses.
- Specify the following Y device number in (d).
  - FX5S/FX5U/FX5UC CPU module: Y0 to Y3
  - FX5UJ CPU module: Y0 to Y2
- The following tables show the special relays and special registers related to the PLSY instruction.

[Special relays]

Axis number				Name	Descriptions
1	2	3	4		
SM5500	SM5501	SM5502	SM5503	Positioning instruction activation	ON: During activation, OFF: Not activated
SM5516	SM5517	SM5518	SM5519	Pulse output monitor	ON: During output, OFF: During stop
SM5532	SM5533	SM5534	SM5535	Positioning error occurrence	On: Error occurred, OFF: Error not occurred
SM5628	SM5629	SM5630	SM5631	Pulse output stop command	ON: Stop command is on, OFF: Stop command is off
SM5644	SM5645	SM5646	SM5647	Pulse deceleration stop command*1	ON: Deceleration stop command is on, OFF: Deceleration stop command is off
SM5660	SM5661	SM5662	SM5663	Forward limit	ON: Forward limit is on, OFF: Forward limit is off
SM5676	SM5677	SM5678	SM5679	Reverse limit	ON: Reverse limit is on, OFF: Reverse limit is off

*1 Because the PLSY instruction does not have the acceleration/deceleration function, the operation is stopped immediately even though the pulse deceleration stop command is turned on.

[Special registers]

Axis number				Name
1	2	3	4	
SD5500, SD5501	SD5540, SD5541	SD5580, SD5581	SD5620, SD5621	Current address (in user unit)
SD5502, SD5503	SD5542, SD5543	SD5582, SD5583	SD5622, SD5623	Current address (in pulse unit)
SD5504, SD5505	SD5544, SD5545	SD5584, SD5585	SD5624, SD5625	Current speed (in user unit)
SD5510	SD5550	SD5590	SD5630	Positioning error error code

[Special relays (FX3 compatible area)]

Axis number				Name
1	2	3	4	
SM8029				Instruction execution complete flag
SM8329				Instruction execution abnormal end flag
SM8340	SM8350	SM8360	SM8370	Pulse output monitoring
SM8348	SM8358	SM8368	SM8378	Positioning instruction activation

[Special registers (FX3 compatible area)]

Axis number				Name
1	2	3	4	
SD8136, SD8137		—	—	Total number of outputs for axis 1 and 2 of PLSY instruction
SD8140, SD8141	SD8142, SD8143	—	—	Total number of output pulses of PLSY instruction
SD8340, SD8341	SD8350, SD8351	SD8360, SD8361	SD8370, SD8371	Current address (in pulse unit)

Precautions

- The operation cannot be performed normally in an environment such as user program where the instruction cannot be executed at each scan or if the instruction is jumped by the CJ(P) instruction. However, the pulse output is continued.
- The same devices as the ones of position instruction, PWM output or general-purpose output cannot be used for the output in the PLSY instruction.
- When the pulse output is stopped, the operation is stopped immediately by the PLSY instruction. Note that the motor is stopped without deceleration and this may damage the system. For stopping method of the pulse output, refer to 📖MELSEC iQ-F FX5 User's Manual (Application).
- If the positioning address is 0 when the PLSY instruction is activated, pulses are output without limitation.
- Overwrite the positioning address during the pulse output to change the positioning address in operation. The written value is reflected at the first time that the instruction is executed after the device is overwritten. The positioning address becomes invalid if it is changed from 0 to a value other than 0 or from a value other than 0 to 0 during positioning operation.
- When the positioning address is changed during the pulse output, the operation is stopped immediately if the changed value is the number of pulses which have already been output or less.
- Overwrite the command speed during the pulse output to change the command speed in operation. The written value is reflected at the first time that the instruction is executed after the device is overwritten.
- When the numbers of pulses (by the pulses conversion) of the command speed and positioning address exceed the 32-bit range, an error occurs and the operation cannot be performed.
- The PLSY instruction always increases the current address because the setting of rotation direction is disabled due to the absence of direction.
- When the output mode is CW/CCW mode, output is always performed from the device set to CW.
- If reverse limit is used, it operates as forward limit.
- Do not set the value of 200 kpps or more by the frequency conversion when changing the command speed during the pulse output.
- If the command speed is set to 0 when the PLSY instruction is activated, the operation ends with an error and stops pulse output.
- If the command speed is changed to 0 during pulse output, the operation immediately stops without abnormal end flag. However, if the drive contact is not turned off, pulse output will restart when the command speed is changed.
- The command speed is changed to negative value during pulse output, it is the operation ends with an error.
- The following table shows the operation timing of the complete flag and abnormal end flag of the PLSY instruction.

	Complete flag (SM8029)	Abnormal end flag (SM8329)
ON condition	• From when the output of the specified positioning address is completed until the drive contact is turned off • Pulse deceleration stop command (when unlimited pulses are being output)	From the following stops until the drive contact is turned off • The specified axis is already used*1 • Pulse output stop command • Pulse deceleration stop command (when unlimited pulses are not being output) • Drive direction limit • All outputs disabled (SM8034) • Write during RUN • Positioning address error
ON→OFF condition	• When the drive contact is turned off	• When the drive contact is turned off

*1 The flag turns on only during one scan time when the activation contact of the instruction turns off and on.

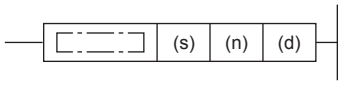
Operation error

Error code (SD0/SD8067)				Description
Axis 1	Axis 2	Axis 3	Axis 4	
SD5510	SD5550	SD5590	SD5630	
1810H				The axis number specified by (d) is used by another instruction.
3405H				The value specified by (s) is outside the following range. 0 to 65535
				The value specified by (n) is outside the following range. 0 to 65535
				The value specified by (d) is outside the following range. 0 to 3
3600H				The axis number specified by (d) is not set by parameters. A function which is set to be not used by parameters (such as interrupt input signal 1 and zero return relations) is used.
3631H	3632H	3633H	3634H	The numbers of pulses (by the pulses conversion) of the positioning address specified by (n) exceed the 32-bit range.
3641H	3642H	3643H	3644H	The numbers of pulses (by the pulses conversion) of the command speed specified by (s) exceed the 32-bit range.
3651H	3652H	3653H	3654H	The operation decelerates and stops by the forward limit or reverse limit during the pulse output or at the activating of the positioning.
3661H	3662H	3663H	3664H	The operation decelerates and stops by the pulse output stop command or special relay whose all outputs are disabled during the pulse output or at the activating of the positioning.

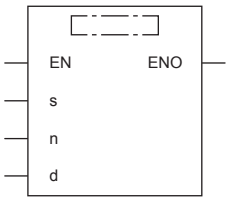
PLSY [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction outputs 16-bit pulse trains specified by the command speed (s) from the device specified by the output (d) for the amount of 16-bit pulses specified by the positioning address (n). Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=PLSY(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Command speed or word device number storing data	0 to 65535	16-bit unsigned binary	ANY16
(n)	Positioning address or word device number storing data	0 to 65535	16-bit unsigned binary	ANY16
(d)	Axis number from which pulses are to be output	■FX5S/FX5U/FX5UC CPU module K1 to K4 ■FX5UJ CPU module K1 to K3	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

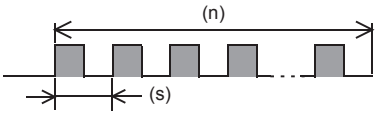
*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○	○	○	—	—	○	○	—	—	—

Processing details

- This instruction outputs 16-bit pulse trains specified by the command speed (s) from the device specified by the output (d) for the amount of 16-bit pulses specified by the positioning address (n).



- Set the value from 0 to 65535 (in user unit) to the command speed (s), so that the command speed is 200 kpps or less when the command speed is converted to frequency.
- Set the value from 0 to 65535 (in user unit) in the positioning address (n), so that the positioning address is within the range from 0 to 2147483647 when the positioning address is converted to number of pulses.
- Specify the axis number (K1 to K4) in which positioning parameters exist in (d).
- The following tables show the special relays and special registers related to the PLSY instruction.

[Special relays]

Axis number				Name	Descriptions
1	2	3	4		
SM5500	SM5501	SM5502	SM5503	Positioning instruction activation	ON: During activation, OFF: Not activated
SM5516	SM5517	SM5518	SM5519	Pulse output monitor	ON: During output, OFF: During stop
SM5532	SM5533	SM5534	SM5535	Positioning error occurrence	On: Error occurred, OFF: Error not occurred
SM5628	SM5629	SM5630	SM5631	Pulse output stop command	ON: Stop command is on, OFF: Stop command is off
SM5644	SM5645	SM5646	SM5647	Pulse deceleration stop command*1	ON: Deceleration stop command is on, OFF: Deceleration stop command is off
SM5660	SM5661	SM5662	SM5663	Forward limit	ON: Forward limit is on, OFF: Forward limit is off
SM5676	SM5677	SM5678	SM5679	Reverse limit	ON: Reverse limit is on, OFF: Reverse limit is off

*1 Because the PLSY instruction does not have the acceleration/deceleration function, the operation is stopped immediately even though the pulse deceleration stop command is turned on.

[Special registers]

Axis number				Name
1	2	3	4	
SD5500, SD5501	SD5540, SD5541	SD5580, SD5581	SD5620, SD5621	Current address (in user unit)
SD5502, SD5503	SD5542, SD5543	SD5582, SD5583	SD5622, SD5623	Current address (in pulse unit)
SD5504, SD5505	SD5544, SD5545	SD5584, SD5585	SD5624, SD5625	Current speed (in user unit)
SD5510	SD5550	SD5590	SD5630	Positioning error error code

[Special relays (FX3 compatible area)]

Axis number				Name
1	2	3	4	
SM8029				Instruction execution complete flag
SM8329				Instruction execution abnormal end flag
SM8340	SM8350	SM8360	SM8370	Pulse output monitoring
SM8348	SM8358	SM8368	SM8378	Positioning instruction activation

[Special registers (FX3 compatible area)]

Axis number				Name
1	2	3	4	
SD8136, SD8137		—	—	Total number of outputs for axis 1 and 2 of PLSY instruction
SD8140, SD8141	SD8142, SD8143	—	—	Total number of output pulses of PLSY instruction
SD8340, SD8341	SD8350, SD8351	SD8360, SD8361	SD8370, SD8371	Current address (in pulse unit)

Precautions

- The operation cannot be performed normally in an environment such as user program where the instruction cannot be executed at each scan or if the instruction is jumped by the CJ(P) instruction. However, the pulse output is continued.
- The same devices as the ones of position instruction, PWM output or general-purpose output cannot be used for the output in the PLSY instruction.
- When the pulse output is stopped, the operation is stopped immediately by the PLSY instruction. Note that the motor is stopped without deceleration and this may damage the system. For stopping method of the pulse output, refer to MELSEC iQ-F FX5 User's Manual (Application).
- If the positioning address is 0 when the PLSY instruction is activated, pulses are output without limitation.
- Overwrite the positioning address during the pulse output to change the positioning address in operation. The written value is reflected at the first time that the instruction is executed after the device is overwritten. The positioning address becomes invalid if it is changed from 0 to a value other than 0 or from a value other than 0 to 0 during positioning operation.
- When the positioning address is changed during the pulse output, the operation is stopped immediately if the changed value is the number of pulses which have already been output or less.
- Overwrite the command speed during the pulse output to change the command speed in operation. The written value is reflected at the first time that the instruction is executed after the device is overwritten.
- When the numbers of pulses (by the pulses conversion) of the command speed and positioning address exceed the 32-bit range, an error occurs and the operation cannot be performed.
- The PLSY instruction always increases the current address because the setting of rotation direction is disabled due to the absence of direction.
- When the output mode is CW/CCW mode, output is always performed from the device set to CW.
- If reverse limit is used, it operates as forward limit.
- Do not set the value of 200 kpps or more by the frequency conversion when changing the command speed during the pulse output.
- If the command speed is set to 0 when the PLSY instruction is activated, the operation ends with an error and stops pulse output.
- If the command speed is changed to 0 during pulse output, the operation immediately stops without abnormal end flag. However, if the drive contact is not turned off, pulse output will restart when the command speed is changed.
- The command speed is changed to negative value during pulse output, it is the operation ends with an error.
- The following table shows the operation timing of the complete flag and abnormal end flag of the PLSY instruction.

	Complete flag (SM8029)	Abnormal end flag (SM8329)
ON condition	• From when the output of the specified positioning address is completed until the drive contact is turned off • Pulse decelerate and stop command (when unlimited pulses are being output)	From the following stops until the drive contact is turned off • The specified axis is already used*1 • Pulse output stop command • Pulse deceleration stop command (when unlimited pulses are not being output) • Drive direction limit • All outputs disabled (SM8034) • Write during RUN • Positioning address error
ON→OFF condition	• When the drive contact is turned off	• When the drive contact is turned off

*1 The flag turns on only during one scan time when the activation contact of the instruction turns off and on.

Operation error

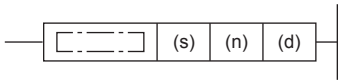
Error code (SD0/SD8067)				Description
Axis 1	Axis 2	Axis 3	Axis 4	
SD5510	SD5550	SD5590	SD5630	
1810H				The axis number specified by (d) is used by another instruction.
3405H				The value specified by (s) is outside the following range. 0 to 65535
				The value specified by (n) is outside the following range. 0 to 65535
				The value specified by (d) is outside the following range. 0 to 3
3600H				The axis number specified by (d) is not set by parameters. A function which is set to be not used by parameters (such as interrupt input signal 1 and zero return relations) is used.
3631H	3632H	3633H	3634H	The numbers of pulses (by the pulses conversion) of the positioning address specified by (n) exceed the 32-bit range.
3641H	3642H	3643H	3644H	The numbers of pulses (by the pulses conversion) of the command speed specified by (s) exceed the 32-bit range.
3651H	3652H	3653H	3654H	The operation decelerates and stops by the forward limit or reverse limit during the pulse output or at the activating of the positioning.
3661H	3662H	3663H	3664H	The operation decelerates and stops by the pulse output stop command or special relay whose all outputs are disabled during the pulse output or at the activating of the positioning.

32 bit binary pulse output

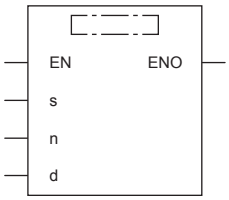
DPLSY [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction outputs 32-bit pulse trains specified by the command speed (s) from the device specified by the output (d) for the amount of 32-bit pulses specified by the positioning address (n). Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=DPLSY(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Command speed or word device number storing data	0 to 2147483647	32-bit unsigned binary	ANY32
(n)	Positioning address or word device number storing data	0 to 2147483647	32-bit unsigned binary	ANY32
(d)	Bit device number (Y) from which pulses are output	■FX5S/FX5U/FX5UC CPU module 0 to 3 ■FX5UJ CPU module 0 to 2	Bit	ANY_ELEMENTARY (BOOL)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

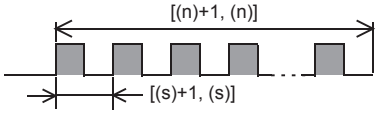
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(n)	○	○	○	○	○	○	○	○	—	—	—
(d)	○*1	—	—	—	—	—	—	—	—	—	—

*1 FX5S/FX5U/FX5UC CPU module: Only Y0 to Y3 devices can be used.
FX5UJ CPU module: Only Y0 to Y2 devices can be used.

Processing details

- This instruction outputs 32-bit pulse trains specified by the command speed (s) from the device specified by the output (d) for the amount of 32-bit pulses specified by the positioning address (n).



- Set the value from 0 to 2147483647 (in user unit) to the command speed (s), so that the command speed is 200 kpps or less when the command speed is converted to frequency.

- Set the value from 0 to 2147483647 (in user unit) to the positioning address (n), so that the positioning address is within the range from 0 to 2147483647 when the positioning address is converted to number of pulses.
- Specify the following Y device number in (d).
  - FX5S/FX5U/FX5UC CPU module: Y0 to Y3
  - FX5UJ CPU module: Y0 to Y2
- The following tables show the special relays and special registers related to the DPLSY instruction.

[Special relays]

Axis number				Name	Descriptions
1	2	3	4		
SM5500	SM5501	SM5502	SM5503	Positioning instruction activation	ON: During activation, OFF: Not activated
SM5516	SM5517	SM5518	SM5519	Pulse output monitor	ON: During output, OFF: During stop
SM5532	SM5533	SM5534	SM5535	Positioning error occurrence	On: Error occurred, OFF: Error not occurred
SM5628	SM5629	SM5630	SM5631	Pulse output stop command	ON: Stop command is on, OFF: Stop command is off
SM5644	SM5645	SM5646	SM5647	Pulse deceleration stop command*1	ON: Deceleration stop command is on, OFF: Deceleration stop command is off
SM5660	SM5661	SM5662	SM5663	Forward limit	ON: Forward limit is on, OFF: Forward limit is off
SM5676	SM5677	SM5678	SM5679	Reverse limit	ON: Reverse limit is on, OFF: Reverse limit is off

*1 Because the DPLSY instruction does not have the acceleration/deceleration function, the operation is stopped immediately even though the pulse deceleration stop command is turned on.

[Special registers]

Axis number				Name
1	2	3	4	
SD5500, SD5501	SD5540, SD5541	SD5580, SD5581	SD5620, SD5621	Current address (in user unit)
SD5502, SD5503	SD5542, SD5543	SD5582, SD5583	SD5622, SD5623	Current address (in pulse unit)
SD5504, SD5505	SD5544, SD5545	SD5584, SD5585	SD5624, SD5625	Current speed (in user unit)
SD5510	SD5550	SD5590	SD5630	Positioning error error code

[Special relays (FX3 compatible area)]

Axis number				Name
1	2	3	4	
SM8029				Instruction execution complete flag
SM8329				Instruction execution abnormal end flag
SM8340	SM8350	SM8360	SM8370	Pulse output monitoring
SM8348	SM8358	SM8368	SM8378	Positioning instruction activation

[Special registers (FX3 compatible area)]

Axis number				Name
1	2	3	4	
SD8136, SD8137		—	—	Total number of outputs for axis 1 and 2 of DPLSY instruction
SD8140, SD8141	SD8142, SD8143	—	—	Total number of output pulses of DPLSY instruction
SD8340, SD8341	SD8350, SD8351	SD8360, SD8361	SD8370, SD8371	Current address (in pulse unit)

Precautions

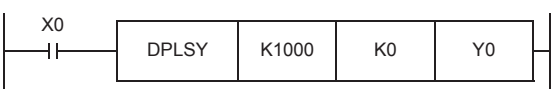
- The operation cannot be performed normally in an environment such as user program where the instruction cannot be executed at each scan or if the instruction is jumped by the CJ(P) instruction. However, the pulse output is continued.
- The same devices as the ones of position instruction, PWM output or general-purpose output cannot be used for the output in the DPLSY instruction.
- When the pulse output is stopped, the operation is stopped immediately by the DPLSY instruction. Note that the motor is stopped without deceleration and this may damage the system. For stopping method of the pulse output, refer to 📖MELSEC iQ-F FX5 User's Manual (Application).
- If the positioning address is 0 when the DPLSY instruction is activated, pulses are output without limitation.
- Overwrite the positioning address during the pulse output to change the positioning address in operation. The written value is reflected at the first time that the instruction is executed after the device is overwritten. The positioning address becomes invalid if it is changed from 0 to a value other than 0 or from a value other than 0 to 0 during positioning operation.
- When the positioning address is changed during the pulse output, the operation is stopped immediately if the changed value is the number of pulses which have already been output or less.
- Overwrite the command speed during the pulse output to change the command speed in operation. The written value is reflected at the first time that the instruction is executed after the device is overwritten.
- When the numbers of pulses (by the pulses conversion) of the command speed and positioning address exceed the 32-bit range, an error occurs and the operation cannot be performed.
- The DPLSY instruction always increases the current address because the setting of rotation direction is disabled due to the absence of direction.
- When the output mode is CW/CCW mode, output is always performed from the device set to CW.
- If reverse limit is used, it operates as forward limit.
- Do not set the value of 200 kpps or more by the frequency conversion when changing the command speed during the pulse output.
- If the command speed is set to 0 when the DPLSY instruction is activated, the operation ends with an error and stops pulse output.
- If the command speed is changed to 0 during pulse output, the operation immediately stops without abnormal end flag. However, if the drive contact is not turned off, pulse output will restart when the command speed is changed.
- The command speed is changed to negative value during pulse output, it is the operation ends with an error.
- The following table shows the operation timing of the complete flag and abnormal end flag of the DPLSY instruction.

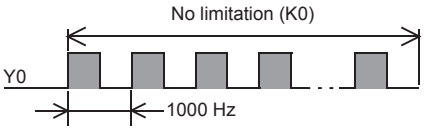
	Complete flag (SM8029)	Abnormal end flag (SM8329)
ON condition	• From when the output of the specified positioning address is completed until the drive contact is turned off • Pulse decelerate and stop command (when unlimited pulses are being output)	From the following stops until the drive contact is turned off • The specified axis is already used*1 • Pulse output stop command • Pulse deceleration stop command (when unlimited pulses are not being output) • Drive direction limit • All outputs disabled (SM8034) • Write during RUN • Positioning address error
ON→OFF condition	• When the drive contact is turned off	• When the drive contact is turned off

*1 The flag turns on only during one scan time when the activation contact of the instruction turns off and on.

Program example

When (n) is set to K0, pulses are output without any limitation.





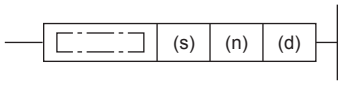
Operation error

Error code (SD0/SD8067)				Description
Axis 1	Axis 2	Axis 3	Axis 4	
SD5510	SD5550	SD5590	SD5630	
1810H				The axis number specified by (d) is used by another instruction.
3405H				The value specified by (s) is outside the following range. 0 to 65535
				The value specified by (n) is outside the following range. 0 to 65535
				The value specified by (d) is outside the following range. 0 to 3
3600H				The axis number specified by (d) is not set by parameters. A function which is set to be not used by parameters (such as interrupt input signal 1 and zero return relations) is used.
3631H	3632H	3633H	3634H	The numbers of pulses (by the pulses conversion) of the positioning address specified by (n) exceed the 32-bit range.
3641H	3642H	3643H	3644H	The numbers of pulses (by the pulses conversion) of the command speed specified by (s) exceed the 32-bit range.
3651H	3652H	3653H	3654H	The operation decelerates and stops by the forward limit or reverse limit during the pulse output or at the activating of the positioning.
3661H	3662H	3663H	3664H	The operation decelerates and stops by the pulse output stop command or special relay whose all outputs are disabled during the pulse output or at the activating of the positioning.

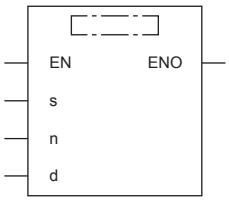
DPLSY [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction outputs 32-bit pulse trains specified by the command speed (s) from the device specified by the output (d) for the amount of 32-bit pulses specified by the positioning address (n). Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=DPLSY(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Command speed or word device number storing data	0 to 2147483647	32-bit signed binary	ANY32
(n)	Positioning address or word device number storing data	0 to 2147483647	32-bit signed binary	ANY32
(d)	Axis number from which pulses are to be output	■FX5S/FX5U/FX5UC CPU module K1 to K4 ■FX5UJ CPU module K1 to K3	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

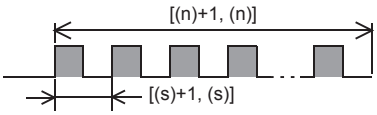
*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(n)	○	○	○	○	○	○	○	○	—	—	—
(d)	—	○	○	○	—	—	○	○	—	—	—

Processing details

- This instruction outputs 32-bit pulse trains specified by the command speed (s) from the device specified by the output (d) for the amount of 32-bit pulses specified by the positioning address (n).



- Set the value from 0 to 2147483647 (in user unit) to the command speed (s), so that the command speed is 200 kpps or less when the command speed is converted to frequency.
- Set the value from 0 to 2147483647 (in user unit) to the positioning address (n), so that the positioning address is within the range from 0 to 2147483647 when the positioning address is converted to number of pulses.
- Specify the axis number (K1 to K4) in which positioning parameters exist in (d).
- The following tables show the special relays and special registers related to the DPLSY instruction.

[Special relays]

Axis number				Name	Descriptions
1	2	3	4		
SM5500	SM5501	SM5502	SM5503	Positioning instruction activation	ON: During activation, OFF: Not activated
SM5516	SM5517	SM5518	SM5519	Pulse output monitor	ON: During output, OFF: During stop
SM5532	SM5533	SM5534	SM5535	Positioning error occurrence	On: Error occurred, OFF: Error not occurred
SM5628	SM5629	SM5630	SM5631	Pulse output stop command	ON: Stop command is on, OFF: Stop command is off
SM5644	SM5645	SM5646	SM5647	Pulse deceleration stop command*1	ON: Deceleration stop command is on, OFF: Deceleration stop command is off
SM5660	SM5661	SM5662	SM5663	Forward limit	ON: Forward limit is on, OFF: Forward limit is off
SM5676	SM5677	SM5678	SM5679	Reverse limit	ON: Reverse limit is on, OFF: Reverse limit is off

*1 Because the DPLSY instruction does not have the acceleration/deceleration function, the operation is stopped immediately even though the pulse deceleration stop command is turned on.

[Special registers]

Axis number				Name
1	2	3	4	
SD5500, SD5501	SD5540, SD5541	SD5580, SD5581	SD5620, SD5621	Current address (in user unit)
SD5502, SD5503	SD5542, SD5543	SD5582, SD5583	SD5622, SD5623	Current address (in pulse unit)
SD5504, SD5505	SD5544, SD5545	SD5584, SD5585	SD5624, SD5625	Current speed (in user unit)
SD5510	SD5550	SD5590	SD5630	Positioning error error code

[Special relays (FX3 compatible area)]

Axis number				Name
1	2	3	4	
SM8029				Instruction execution complete flag
SM8329				Instruction execution abnormal end flag
SM8340	SM8350	SM8360	SM8370	Pulse output monitoring
SM8348	SM8358	SM8368	SM8378	Positioning instruction activation

[Special registers (FX3 compatible area)]

Axis number				Name
1	2	3	4	
SD8136, SD8137		—	—	Total number of outputs for axis 1 and 2 of DPLSY instruction
SD8140, SD8141	SD8142, SD8143	—	—	Total number of output pulses of DPLSY instruction
SD8340, SD8341	SD8350, SD8351	SD8360, SD8361	SD8370, SD8371	Current address (in pulse unit)

Precautions

- The operation cannot be performed normally in an environment such as user program where the instruction cannot be executed at each scan or if the instruction is jumped by the CJ(P) instruction. However, the pulse output is continued.
- The same devices as the ones of position instruction, PWM output or general-purpose output cannot be used for the output in the DPLSY instruction.
- When the pulse output is stopped, the operation is stopped immediately by the DPLSY instruction. Note that the motor is stopped without deceleration and this may damage the system. For stopping method of the pulse output, refer to MELSEC iQ-F FX5 User's Manual (Application).
- If the positioning address is 0 when the DPLSY instruction is activated, pulses are output without limitation.
- Overwrite the positioning address during the pulse output to change the positioning address in operation. The written value is reflected at the first time that the instruction is executed after the device is overwritten. The positioning address becomes invalid if it is changed from 0 to a value other than 0 or from a value other than 0 to 0 during positioning operation.
- When the positioning address is changed during the pulse output, the operation is stopped immediately if the changed value is the number of pulses which have already been output or less.
- Overwrite the command speed during the pulse output to change the command speed in operation. The written value is reflected at the first time that the instruction is executed after the device is overwritten.
- When the numbers of pulses (by the pulses conversion) of the command speed and positioning address exceed the 32-bit range, an error occurs and the operation cannot be performed.
- The DPLSY instruction always increases the current address because the setting of rotation direction is disabled due to the absence of direction.
- When the output mode is CW/CCW mode, output is always performed from the device set to CW.
- If reverse limit is used, it operates as forward limit.
- Do not set the value of 200 kpps or more by the frequency conversion when changing the command speed during the pulse output.
- If the command speed is set to 0 when the DPLSY instruction is activated, the operation ends with an error and stops pulse output.
- If the command speed is changed to 0 during pulse output, the operation immediately stops without abnormal end flag. However, if the drive contact is not turned off, pulse output will restart when the command speed is changed.
- The command speed is changed to negative value during pulse output, it is the operation ends with an error.
- The following table shows the operation timing of the complete flag and abnormal end flag of the DPLSY instruction.

	Complete flag (SM8029)	**Abnormal end flag (SM8329)**
ON condition	• From when the output of the specified positioning address is completed until the drive contact is turned off • Pulse decelerate and stop command (when unlimited pulses are being output)	From the following stops until the drive contact is turned off • The specified axis is already used*1 • Pulse output stop command • Pulse deceleration stop command (when unlimited pulses are not being output) • Drive direction limit • All outputs disabled (SM8034) • Write during RUN • Positioning address error
ON→OFF condition	• When the drive contact is turned off	• When the drive contact is turned off

*1 The flag turns on only during one scan time when the activation contact of the instruction turns off and on.

Operation error

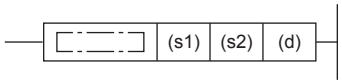
Error code (SD0/SD8067)				Description
Axis 1	Axis 2	Axis 3	Axis 4	
SD5510	SD5550	SD5590	SD5630	
1810H				The axis number specified by (d) is used by another instruction.
3405H				The value specified by (s) is outside the following range. 0 to 65535
				The value specified by (n) is outside the following range. 0 to 65535
				The value specified by (d) is outside the following range. 0 to 3
3600H				The axis number specified by (d) is not set by parameters. A function which is set to be not used by parameters (such as interrupt input signal 1 and zero return relations) is used.
3631H	3632H	3633H	3634H	The numbers of pulses (by the pulses conversion) of the positioning address specified by (n) exceed the 32-bit range.
3641H	3642H	3643H	3644H	The numbers of pulses (by the pulses conversion) of the command speed specified by (s) exceed the 32-bit range.
3651H	3652H	3653H	3654H	The operation decelerates and stops by the forward limit or reverse limit during the pulse output or at the activating of the positioning.
3661H	3662H	3663H	3664H	The operation decelerates and stops by the pulse output stop command or special relay whose all outputs are disabled during the pulse output or at the activating of the positioning.

16 bit binary pulse width modulation

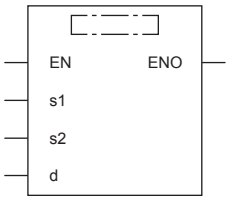
PWM

FX5S FX5UJ FX5U FX5UC

This instruction outputs the pulse (in 16-bit data units) of the ON time (in 16-bit data units) specified by (s1) and the period specified by (s2) to the output destination specified by (d).

Ladder diagram	Structured text
	ENO:=PWM(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	ON time or the device number storing the ON time	1 to 65535	16-bit unsigned binary	ANY16
(s2)	Period or the device number storing the period	1 to 65535	16-bit unsigned binary	ANY16
(d)	Channel number or device number from which pulses are to be output	—	Bit/16-bit unsigned binary	ANY_ELEMENTARY*1
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

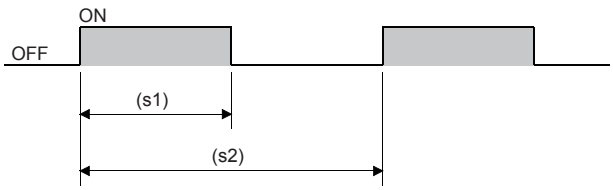
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○*1	○*2	○*2	○*2	—	—	○	○*2	—	—	—

*1 Only Y can be used for a bit device.
When a bit device is specified, specify one of Y0 to Y7. The high-speed pulse input/output module bit device (Y) cannot be specified.
If Y is specified, outputs are enabled when there is an unused channel number in the parameter setting and the specified Y number is not used.
The nibble of a bit device cannot be specified.

*2 When a word device or constant is specified, specify one of the channel numbers.
Specify K1 to K4 (axis 1 to 4) for the CPU module, and K5 to K12 (axis 5 to 12) for the high-speed pulse input/output module.

Processing details

- This instruction outputs the pulse of the ON time specified by (s1) and the period specified by (s2) to the output destination specified by (d).



- Time with a unit selected on the parameter setting screen (μs or ms) can be specified by (s1) and (s2).
- The pulse output destination channel number selected on the parameter setting screen can be specified by (d).
- This instruction store the number of pulses, pulse width, and period output from each channel to an SD device. The pulse width and period are stored in the units set by the parameters. When 0 is specified in the pulse output, pulses are output without any limitation.

Pulse output destination channel	Number of output pulses	R/W	Initial value	Timing of reflection on operation	Timing of clearing to initial value
CH1	SD5301, SD5300	R/W	0	• When the DHCMOV instruction is executed[*1] • When the PWM instruction is executed • END processing	STOP/PAUSE→RUN
CH2	SD5317, SD5316				
CH3	SD5333, SD5332				
CH4	SD5349, SD5348				
CH5	SD5365, SD5364				
CH6	SD5381, SD5380				
CH7	SD5397, SD5396				
CH8	SD5413, SD5412				
CH9	SD5429, SD5428				
CH10	SD5445, SD5444				
CH11	SD5461, SD5460				
CH12	SD5477, SD5476				

Pulse output destination channel	ON time	R/W	Initial value	Timing of reflection on operation	Timing of clearing to initial value
CH1	SD5303, SD5302	R/W	0[*2]	• When the DHCMOV instruction is executed[*1] • When this instruction is executed[*3] • END processing	STOP/PAUSE→RUN
CH2	SD5319, SD5318				
CH3	SD5335, SD5334				
CH4	SD5351, SD5350				
CH5	SD5367, SD5366				
CH6	SD5383, SD5382				
CH7	SD5399, SD5398				
CH8	SD5415, SD5414				
CH9	SD5431, SD5430				
CH10	SD5447, SD5446				
CH11	SD5463, SD5462				
CH12	SD5479, SD5478				

Pulse output destination channel	Period	R/W	Initial value	Timing of reflection on operation	Timing of clearing to initial value
CH1	SD5305, SD5304	R/W	0*2	• When the DHCMOV instruction is executed*1 • When this instruction is executed*3 • END processing	STOP/PAUSE→RUN
CH2	SD5321, SD5320				
CH3	SD5337, SD5336				
CH4	SD5353, SD5352				
CH5	SD5369, SD5368				
CH6	SD5385, SD5384				
CH7	SD5401, SD5400				
CH8	SD5417, SD5416				
CH9	SD5433, SD5432				
CH10	SD5449, SD5448				
CH11	SD5465, SD5464				
CH12	SD5481, SD5480				

*1 When the DHCMOV instruction is used, the latest value can be read. A writable device can be updated immediately.
*2 Parameter setting values are set to an SD device at STOP to RUN.
*3 When this instruction is executed, the pulse width and period specified (s1) and (s2) are set to an SD device.

• After the pulse output is started from each channel, the pulse output monitor turns on.

Pulse output destination channel	Pulse output monitor	R/W	Initial value	ON timing	OFF timing
CH1	SM5300	R	OFF	• When the HIOEN instruction is executed • When this instruction is executed	• Power on • Reset • RUN→STOP/PAUSE • When the specified number of pulses are output. • The drive contact is turned off
CH2	SM5301				
CH3	SM5302				
CH4	SM5303				
CH5	SM5304				
CH6	SM5305				
CH7	SM5306				
CH8	SM5307				
CH9	SM5308				
CH10	SM5309				
CH11	SM5310				
CH12	SM5311				

• This instruction stores the number of pulses output from each channel.

Pulse output destination channel	Monitoring the current number of output pulses	R/W	Initial value	Timing of reflection on operation	Timing of clearing to initial value
CH1	SD5307, SD5306	R/W	0	• When the DHCMOV instruction is executed → An SD device is updated • When the PWM instruction is executed • END processing	• Power-on • Reset • STOP/PAUSE→RUN
CH2	SD5323, SD5322				
CH3	SD5339, SD5338				
CH4	SD5355, SD5354				

- The number of output pulses set to an SD device is valid for this instruction as well. The setting values are always read and updated.
- When the specified number of output pulses is equal to or less than the number of pulses which have already been output, pulse output stops after outputting pulses which are being output.
- When the specified number of output pulses is larger than the number of pulses which have already been output, pulse output stops after outputting set number of pulses.
- When the number of output pulses is set from the no limitation output setting (number of output pulses is 0), the number of output pulses is not updated (because outputting pulses continues or stops in the no limitation output).
- The maximum number of output pulses which can be output when the PWM instruction is executed once (= maximum value which can be set to an SD device) is “2147483647”.
- The ON time and period can be set during the pulse output. Setting values are always read and updated.
- When the number of output pulses is 0 (no limitation output setting), the monitor of the current number of output pulses is set to 0.

- When the number of output pulses is specified, the output pulses are monitored. When the PWM output is executed several times, the monitor of the number of output pulses is an integrated value.
- The monitor of the current number of output pulses can be changed during the pulse output.
- The monitor of the current number of output pulses is updated when the number of pulses is counted at the falling edge of pulses in the positive logic and at the rising edge of pulses in the negative logic.
- When the output always remains ON or OFF, the monitor of the current number of output pulses does not change.
- The maximum value of the monitor of the current number of output pulses is “FFFFFFFFH”. After the current number of output pulses reaches the maximum value, the monitor of the current number of output pulses starts to count again from “0”.

Precautions

- Specify the ON time by (s1) and the period by (s2) so that [(s2)-(s1)] is equal to or larger than 3 μs.
- Specify the ON time by (s1) and the period by (s2) so that they are equal or larger than the following values.

[FX5S/FX5UJ CPU module]

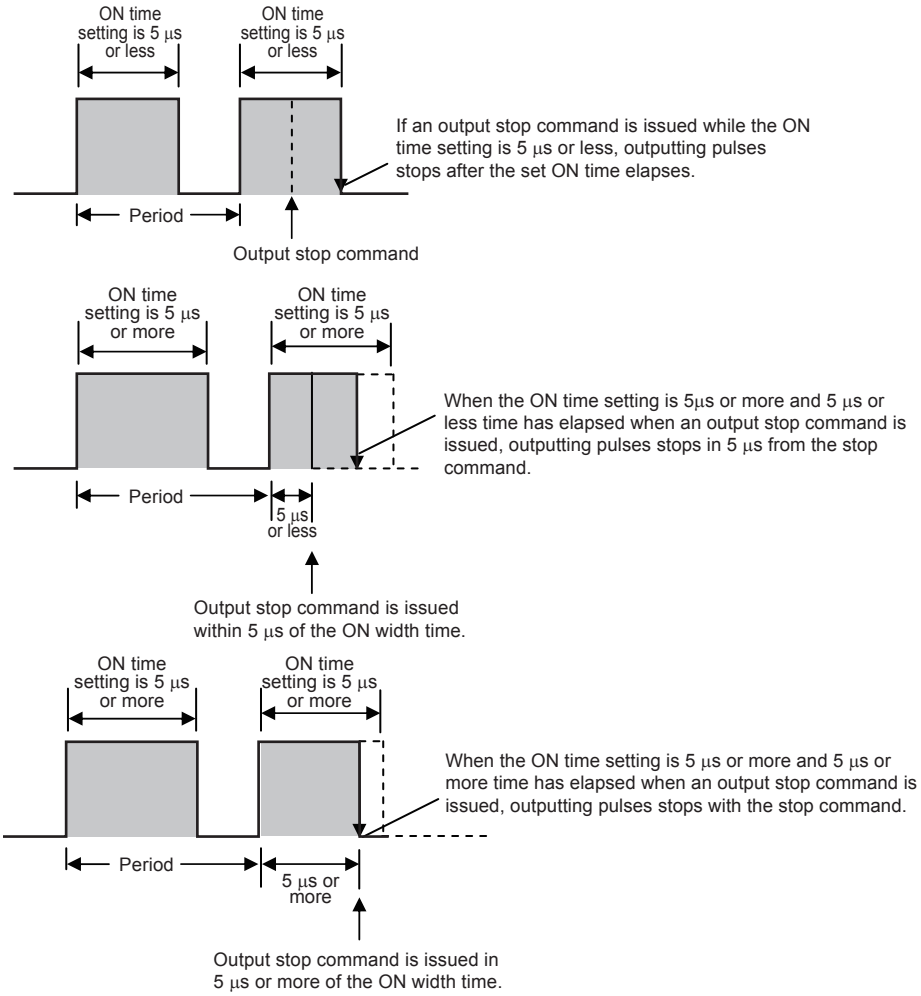
Output number	(s1) ON time	(s2) Period
CPU module		
Y0 to Y2	2 μs more	5 μs more
Y3 to Y7	200 μs more	400 μs more

[FX5U/FX5UC CPU module and high-speed pulse input/output module]

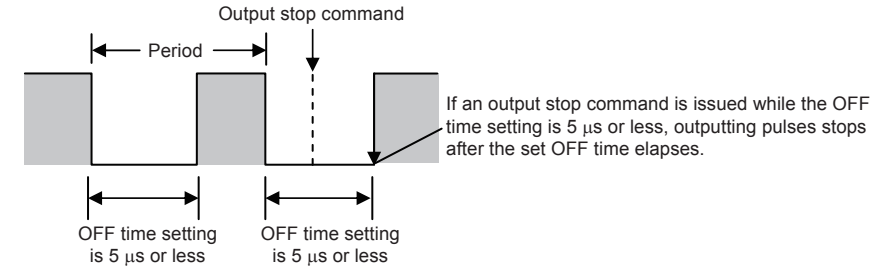
Output number		(s1) ON time	(s2) Period
CPU module	High-speed pulse input/ output module		
Y0 to Y3	Y0 to Y2	2 μs more	5 μs more
Y4 to Y7	—	200 μs more	400 μs more

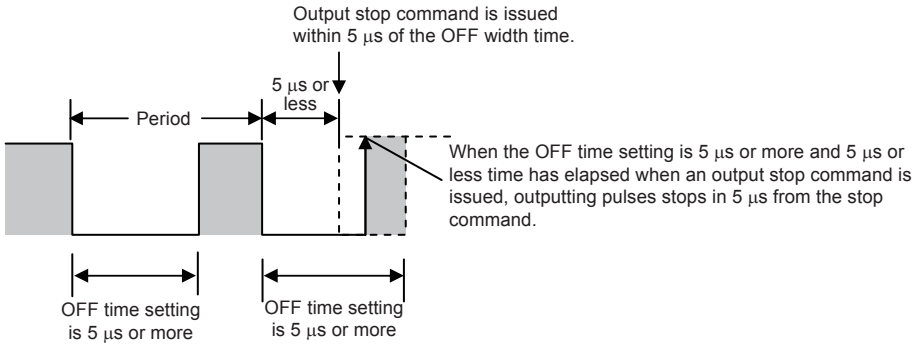
- When a channel number that is not selected for the PWM output in the parameter setting is specified for (d), this instruction is not executed. An operation error occurs.
- Do not specify the channel numbers of the high-speed pulse input/output module (5 to 12) in a program with interrupt priority set to "1".

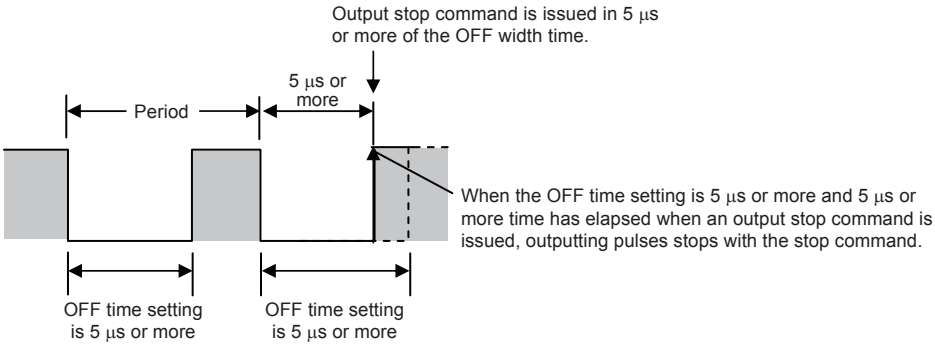
• Operations when the PWM output is stopped (while the output pulse is on)



- Operations when the PWM output is stopped (while the output pulse is off)



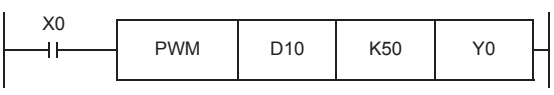


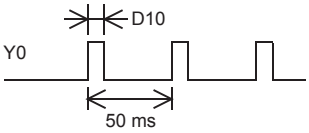


- The PWM output stops when SM8034 is on, and starts when SM8034 is off.
- PWM output does not stop, even if a pulse stop command for positioning is driven.
- When specifying the number of output pulses, executing the PWM instruction, and then outputting pulses again after the pulse output stops due to the completion of output of the specified number of pulses, turn OFF the contact which drove the PWM instruction. When driving PWM output by the HIOEN instruction, use the HIOEN instruction to turn PWM output off.
- When the period setting is equivalent to the ON time setting, the output always remains ON. The output ON state continues even after “Period x Number of output pulses” is finished in this condition.

Program example

When the contents of D10 are changed ranging from "0" to "50" in the program example shown below, the average output from Y0 will be ranging from 0 to 100%.
An error will occur if the contents of D10 exceed K50.





Operation error

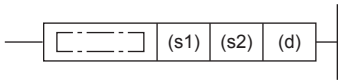
Error code (SD0/SD8067)	Description
1810H	The output destination specified by (d) is already used by another instruction (positioning instruction). (The PWM output is not executed.)
	A Y device is specified as the output destination specified by (d), and there is no unused channel number in the parameter setting.
2221H	The parameter set value is out of range.
3405H	Y10 or later is specified as the output destination specified by (d). (The PWM output stops.)
3580H	An instruction that cannot be used in an interrupt program is used
3600H	A channel number that is not selected in the parameter setting are specified for the output destination specified by (d). (The PWM output is not executed.)
3611H(CH1) 3612H(CH2) 3613H(CH3) 3614H(CH4) 3615H(CH5) 3616H(CH6) 3617H(CH7) 3618H(CH8) 3619H(CH9) 361AH(CH10) 361BH(CH11) 361CH(CH12)	The ON time specified by (s1) is larger than the period specified by (s2). (The PWM output stops.)
	Values of an SD device for setting pulse width and period of this instruction are incorrect. (The PWM output stops.)
	The ON time or period is less than “1”.
	The SD device specified for the number of output pulses stores a value outside the available range (0 to 2147483647).

32 bit binary pulse width modulation

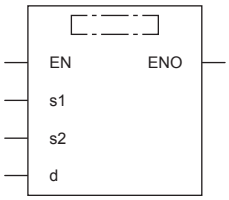
DPWM

FX5S FX5UJ FX5U FX5UC

This instruction outputs the pulse (in 32-bit data units) of the ON time (in 32-bit data units) specified by (s1) and the period specified by (s2) to the output destination specified by (d).

Ladder diagram	Structured text
	ENO:=DPWM(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	ON time or the device number storing the ON time	1 to 2147483647	32-bit unsigned binary	ANY32
(s2)	Period or the device number storing the period	1 to 2147483647	32-bit unsigned binary	ANY32
(d)	Channel number or device number from which pulses are to be output	—	Bit/16-bit unsigned binary	ANY_ELEMENTARY*1
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

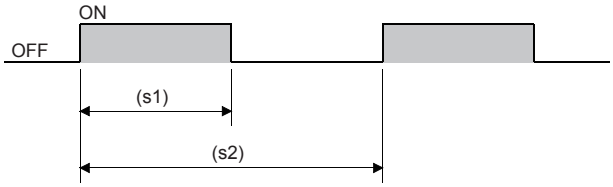
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○*1	○*2	○*2	○*2	—	—	○	○*2	—	—	—

*1 Only Y can be used for a bit device.
When a bit device is specified, specify one of Y0 to Y7. The high-speed pulse input/output module bit device (Y) cannot be specified.
If Y is specified, outputs are enabled when there is an unused channel number in the parameter setting and the specified Y number is not used.
The nibble of a bit device cannot be specified.

*2 When a word device or constant is specified, specify one of the CH numbers.
Specify K1 to K4 (axis 1 to 4) for the CPU module, and K5 to K12 (axis 5 to 12) for the high-speed pulse input/output module.

Processing details

- This instruction outputs the pulse of the ON time specified by (s1) and the period specified by (s2) to the output destination specified by (d).



- Time with a unit selected on the parameter setting screen (μs or ms) can be specified by (s1) and (s2).
- The pulse output destination channel number selected on the parameter setting screen can be specified by (d).
- This instruction stores the number of pulses, pulse width, and period output from each channel to an SD device. The pulse width and period are stored in the units set by the parameters. When 0 is specified in the pulse output, pulses are output without any limitation.

Pulse output destination channel	Number of output pulses	R/W	Initial value	Timing of reflection on operation	Timing of clearing to initial value
CH1	SD5301, SD5300	R/W	0	• When the DHCMOV instruction is executed[*1] • When the DPWM instruction is executed • END processing	STOP/PAUSE→RUN
CH2	SD5317, SD5316				
CH3	SD5333, SD5332				
CH4	SD5349, SD5348				
CH5	SD5365, SD5364				
CH6	SD5381, SD5380				
CH7	SD5397, SD5396				
CH8	SD5413, SD5412				
CH9	SD5429, SD5428				
CH10	SD5445, SD5444				
CH11	SD5461, SD5460				
CH12	SD5477, SD5476				

Pulse output destination channel	ON time	R/W	Initial value	Timing of reflection on operation	Timing of clearing to initial value
CH1	SD5303, SD5302	R/W	0[*2]	• When the DHCMOV instruction is executed[*1] • When the DPWM instruction is executed[*3] • END processing	STOP/PAUSE→RUN
CH2	SD5319, SD5318				
CH3	SD5335, SD5334				
CH4	SD5351, SD5350				
CH5	SD5367, SD5366				
CH6	SD5383, SD5382				
CH7	SD5399, SD5398				
CH8	SD5415, SD5414				
CH9	SD5431, SD5430				
CH10	SD5447, SD5446				
CH11	SD5463, SD5462				
CH12	SD5479, SD5478				

Pulse output destination channel	Period	R/W	Initial value	Timing of reflection on operation	Timing of clearing to initial value
CH1	SD5305, SD5304	R/W	0*2	• When the DHCMOV instruction is executed*1 • When the DPWM instruction is executed*3 • END processing	STOP/PAUSE→RUN
CH2	SD5321, SD5320				
CH3	SD5337, SD5336				
CH4	SD5353, SD5352				
CH5	SD5369, SD5368				
CH6	SD5385, SD5384				
CH7	SD5401, SD5400				
CH8	SD5417, SD5416				
CH9	SD5433, SD5432				
CH10	SD5449, SD5448				
CH11	SD5465, SD5464				
CH12	SD5481, SD5480				

*1 When the DHCMOV instruction is used, the latest value can be read. A writable device can be updated immediately.
*2 Parameter setting values are set to an SD device at STOP to RUN.
*3 When this instruction is executed, the pulse width and period specified (s1) and (s2) are set to an SD device.

• After the pulse output is started from each channel, the pulse output monitor turns on.

Pulse output destination channel	Pulse output monitor	R/W	Initial value	ON timing	OFF timing
CH1	SM5300	R	OFF	• When the DHIOEN instruction is executed • When the DPWM instruction is executed	• Power on • Reset • RUN→STOP/PAUSE • When the specified number of pulses are output. • The drive contact is turned off
CH2	SM5301				
CH3	SM5302				
CH4	SM5303				
CH5	SM5304				
CH6	SM5305				
CH7	SM5306				
CH8	SM5307				
CH9	SM5308				
CH10	SM5309				
CH11	SM5310				
CH12	SM5311				

• This instruction stores the number of pulses output from each channel.

Pulse output destination channel	Monitoring the current number of output pulses	R/W	Initial value	Timing of reflection on operation	Timing of clearing to initial value
CH1	SD5307, SD5306	R/W	0	• When the DHCMOV instruction is executed → An SD device is updated • END processing	• Power-on • Reset • STOP/PAUSE→RUN
CH2	SD5323, SD5322				
CH3	SD5339, SD5338				
CH4	SD5355, SD5354				

• The number of output pulses set to an SD device is valid for this instruction as well. The setting values are always read and updated.
• When the specified number of output pulses is equal to or less than the number of pulses which have already been output, pulse output stops after outputting pulses which are being output.
• When the specified number of output pulses is larger than the number of pulses which have already been output, pulse output stops after outputting set number of pulses.
• When the number of output pulses is set from the no limitation output setting (number of output pulses is 0), the number of output pulses is not updated (because outputting pulses continues or stops in the no limitation output).
• The maximum number of output pulses which can be output when the DPWM instruction is executed once (= maximum value which can be set to an SD device) is “2147483647”
• The ON time and period can be set during the pulse output. Setting values are always read and updated.
• When the number of output pulses is 0 (no limitation output setting), the monitor of the current number of output pulses is set to 0.

- When the number of output pulses is specified, the output pulses are monitored. When the DPWM output is executed several times, the monitor of the number of output pulses is an integrated value.
- The monitor of the current number of output pulses can be changed during the pulse output.
- The monitor of the current number of output pulses is updated when the number of pulses is counted at the falling edge of pulses in the positive logic and at the rising edge of pulses in the negative logic.
- When the output always remains ON or OFF, the monitor of the current number of output pulses does not change.
- The maximum value of the monitor of the current number of output pulses is “FFFFFFFFH”. After the current number of output pulses reaches the maximum value, the monitor of the current number of output pulses starts to count again from “0”.

Precautions

- Specify the ON time by (s1) and the period by (s2) so that [(s2)-(s1)] is equal to or larger than 3 μs.
- When a negative value is specified for the ON time by (s1) and the period by (s2), an operation error occurs. (In 16-bit instruction PWM, no error occurs.)
- Specify the ON time by (s1) and the period by (s2) so that they are equal or larger than the following values.

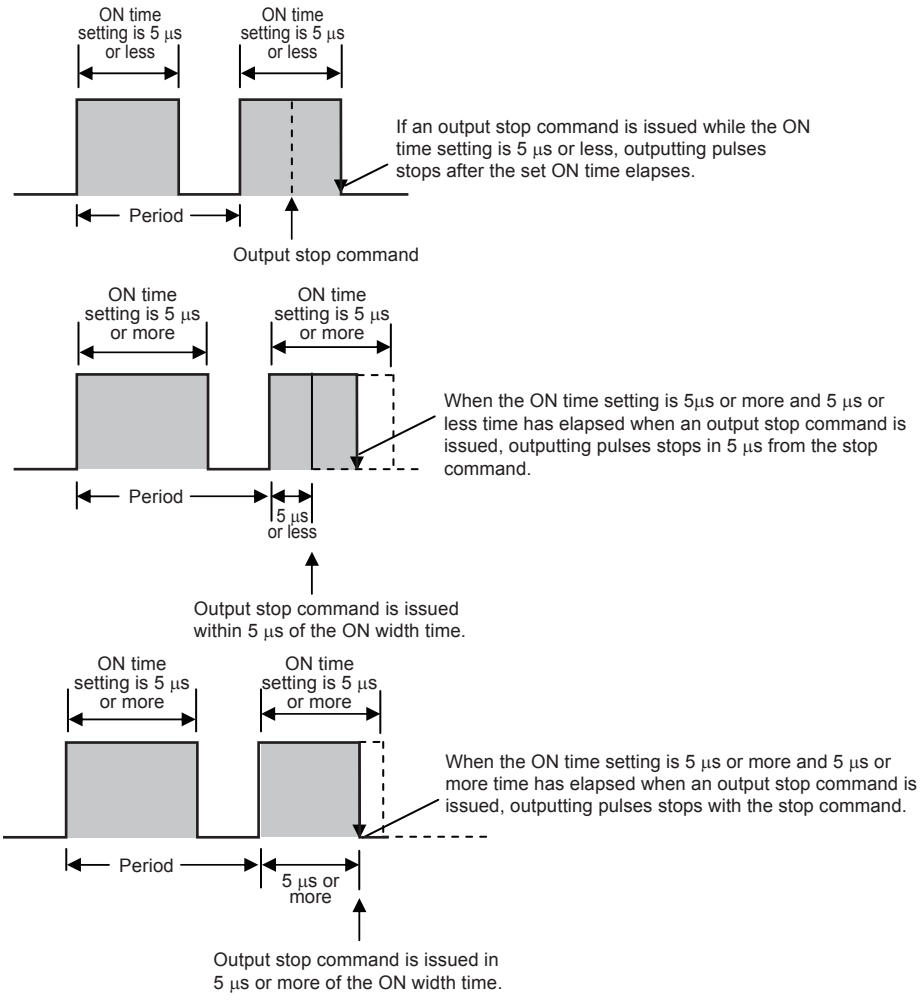
[FX5S/FX5UJ CPU module]

Output number	(s1) ON time	(s2) Period
CPU module		
Y0 to Y2	2 μs more	5 μs more
Y3 to Y7	200 μs more	400 μs more

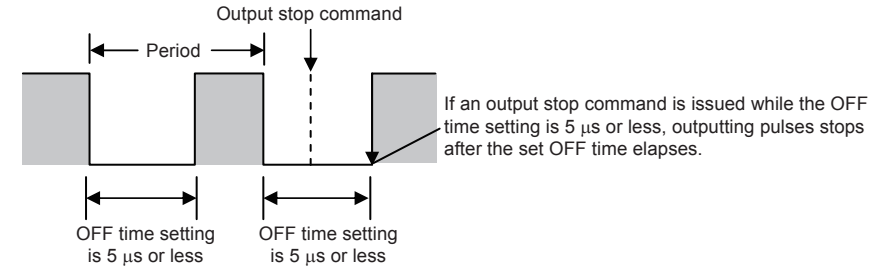
[FX5U/FX5UC CPU module and high-speed pulse input/output module]

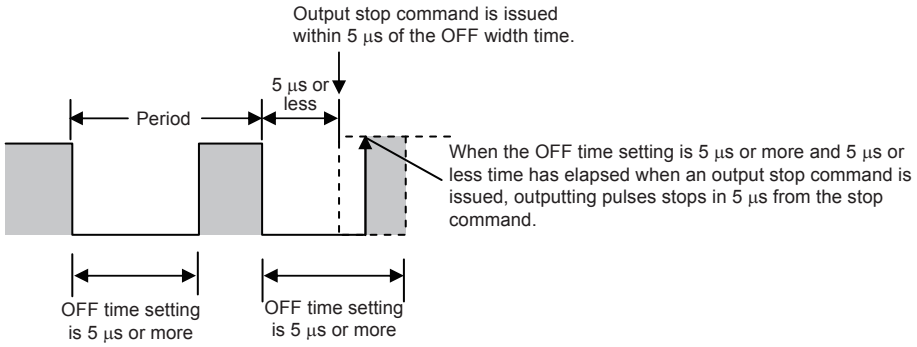
Output number		(s1) ON time	(s2) Period
CPU module	High-speed pulse input/ output module		
Y0 to Y3	Y0 to Y2	2 μs more	5 μs more
Y4 to Y7	—	200 μs more	400 μs more

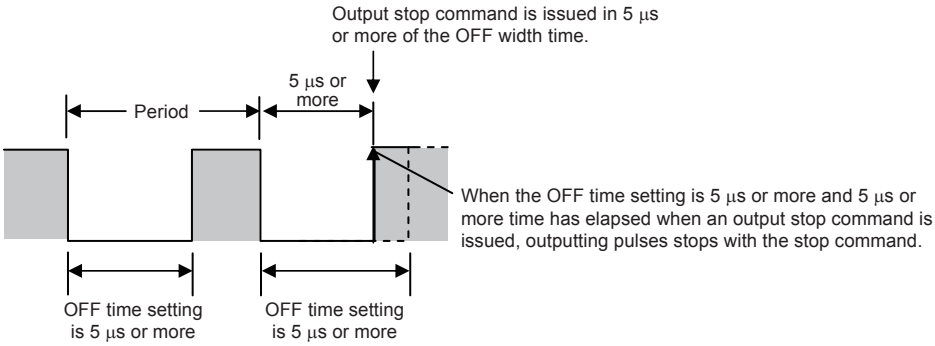
- When a channel number that is not selected for the DPWM output in the parameter setting is specified for (d), this instruction is not executed. An operation error occurs.
- Do not specify the channel numbers of the high-speed pulse input/output module (5 to 12) in a program with interrupt priority set to "1".
- Operations when the DPWM output is stopped (while the output pulse is on)



- Operations when the DPWM output is stopped (while the output pulse is off)







- The PWM output stops when SM8034 is on, and starts when SM8034 is off.
- PWM output does not stop, even if a pulse stop command for positioning is driven.
- When specifying the number of output pulses, executing the DPWM instruction, and then outputting pulses again after the pulse output stops due to the completion of output of the specified number of pulses, turn OFF the contact which drove the DPWM instruction. When driving PWM output by the DHIOEN instruction, use the DHIOEN instruction to turn PWM output off.
- When the period setting is equivalent to the ON time setting, the output always remains ON. The output ON state continues even after “Period × Number of output pulses” is finished in this condition.

Operation error

Error code (SD0/SD8067)	Description
1810H	The output destination specified by (d) is already used by another instruction (positioning instruction). (The PWM output is not executed.)
	A Y device is specified as the output destination specified by (d), and there is no unused channel number in the parameter setting
2221H	The parameter set value is out of range.
3405H	Y10 or later is specified as the output destination specified by (d). (The PWM output stops.)
3580H	An instruction that cannot be used in an interrupt program is used.
3600H	A channel number that is not selected in the parameter setting are specified for the output destination specified by (d). (The PWM output is not executed.)
3611H(CH1) 3612H(CH2) 3613H(CH3) 3614H(CH4) 3615H(CH5) 3616H(CH6) 3617H(CH7) 3618H(CH8) 3619H(CH9) 361AH(CH10) 361BH(CH11) 361CH(CH12)	The ON time specified by (s1) is larger than the period specified by (s2). (The PWM output stops.)
	In (s1) and (s2), a negative value is specified. (The PWM output stops.)
	Values of an SD device for setting pulse width and period of this instruction are incorrect. (The PWM output stops.)
	The ON time or period is less than “1”.
	The SD device specified for the number of output pulses stores a value outside the available range (0 to 2147483647).

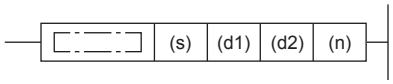
8.19 Input Matrix Instruction

Input matrix

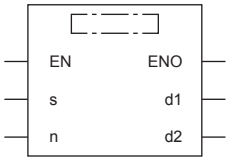
MTR

FX5S FX5UJ FX5U FX5UC

Reads matrix input as 8-point input × "n"-point output (transistor) in the time division method.

Ladder diagram	Structured text
	ENO:=MTR(EN, s, n, d1, d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Input device (X) number of matrix signal input X0, X10, X20 … final input device number (Only "0" is allowed in the least significant digit of device numbers.)	—	Bit	ANYBIT_ARRAY (Number of elements: 8)
(d1)	Head device (Y) number of matrix signal output Y0, Y10, Y20 … final output device number (Only "0" is allowed in the least significant digit of device numbers.))	—	Bit	ANY_BOOL
(d2)	Head bit device (Y, M or S) number of ON output destination Y0, Y10, Y20 … final Y number, M0, M10, M20 … final M number or S0,S10, S20 … final S number (Only "0" is allowed in the least significant digit of device numbers.)	—	Bit	ANY_BOOL
(n)	Number of columns in matrix input	2 to 8	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

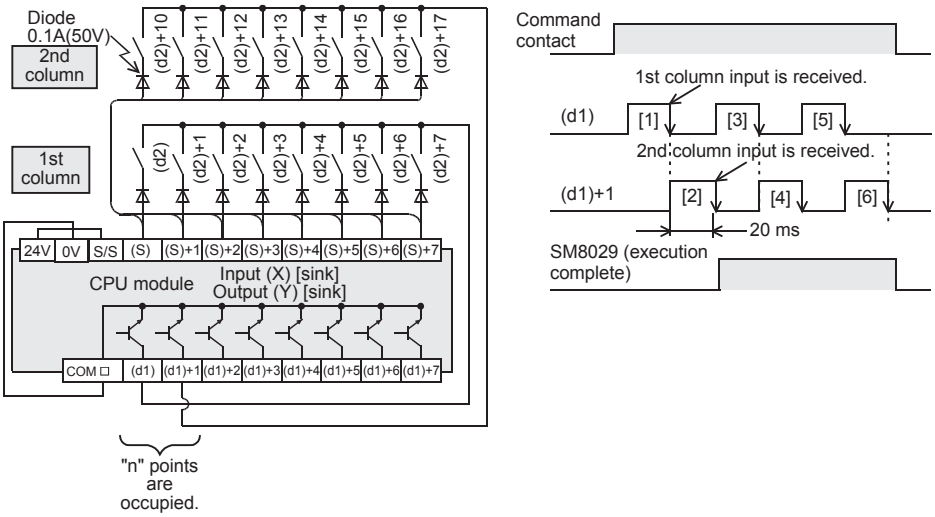
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○*1	—	—	—	—	—	—	—	—	—	—
(d1)	○*2	—	—	—	—	—	—	—	—	—	—
(d2)	○*3	—	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 Only X can be used.
*2 Only Y can be used.
*3 X cannot be used.

Processing details

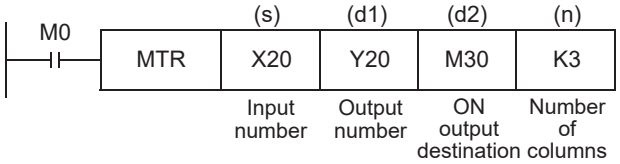
- An input signal of 8 points x "n" columns is controlled in the time division method using 8 inputs specified in (s) and transistor outputs specified in (d1). Each column is read in turn, and then output to devices specified in (d2).
- For each output, the I/O processing is executed immediately in turn in interrupt at every 10 ms or 20 ms.
- For the connection example of matrix input, refer to the following manual.

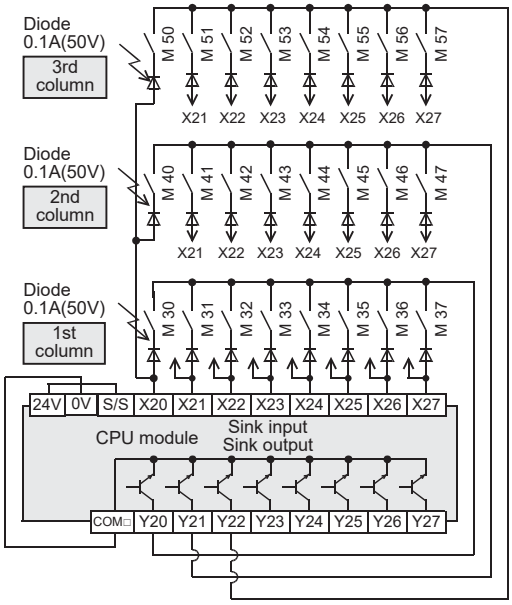
MELSEC iQ-F FX5S/FX5UJ/FX5U/FX5UC User's Manual (Hardware)

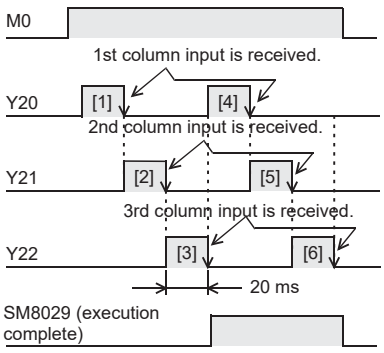


- In this program example, n=Three outputs (Y20, Y21 and Y22) are set to ON in turn repeatedly. Every time an output is set to ON, eight inputs in the 1st, 2nd and 3rd columns are received in turn repeatedly, and stored to M30 to M37, M40 to M47 and M50 to M57 respectively. For wiring details, refer to the following manual.

MELSEC iQ-F FX5S/FX5UJ/FX5U/FX5UC User's Manual (Hardware)





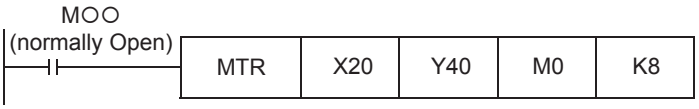


• The table below shows the related devices.

Device	Name	Description
SM8029	Instruction execution complete	ON: Turns ON after the matrix in the nth (last) column is input. OFF: Remains OFF while the matrix in the 1st to nth (last) columns is being input.

Precautions

- Eight devices are occupied from the device specified in (s).
- When specifying the output in (d2) , make sure that "n" output numbers specified in (d1) does not overlap the output specified in (d2).
- The MTR instruction can only be used once in a program.
- One diode of 0.1 A/50 V is required for each switch.
- Use the transistor output format.
- If write during RUN is executed while the MTR instruction is being executed, the control right is released by the END processing. The MTR instruction executed first in the next scan will acquire the control right next.
- For the MTR instruction, set the command input to normally Open.

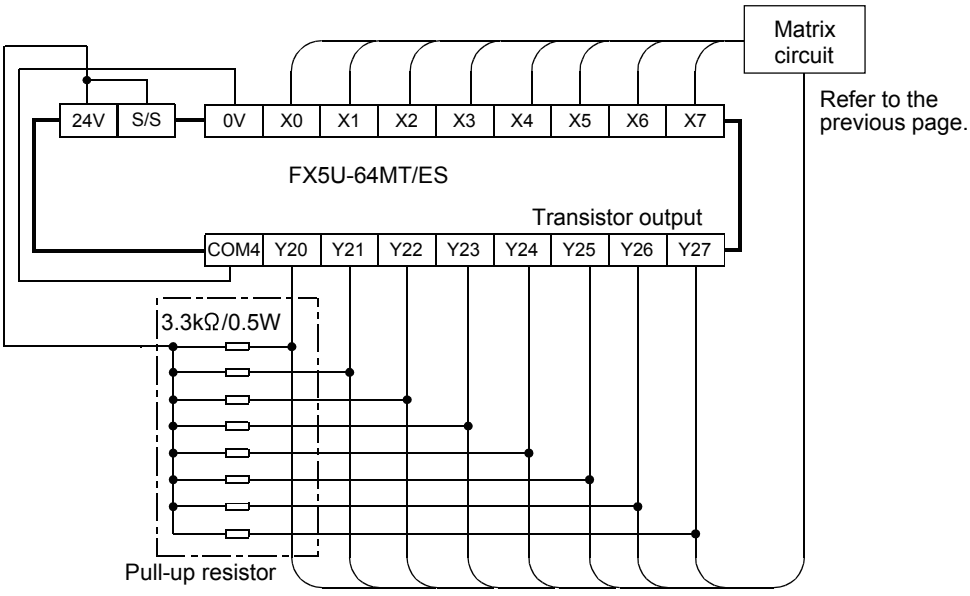


- Normally, the following inputs should be used with the MTR instruction.
  - FX5S/FX5UJ CPU module: X10 and higher
  - FX5U/FX5UC CPU module: X20 and higher
- When using inputs other than those above, the receiving speed will be faster. However, the output transistor recovery time may be longer and the input sensitivity will be high, so there may be erroneous inputs. Change the input response time of unit parameter setting to “5 ms” to input at intervals of 10 ms. To prevent erroneous input pulses, connect pull-up resistors (3.3 kΩ/0.5 W) to transistor outputs used in MTR instruction. For pull-up resistors, use the power supply shown in the table below.

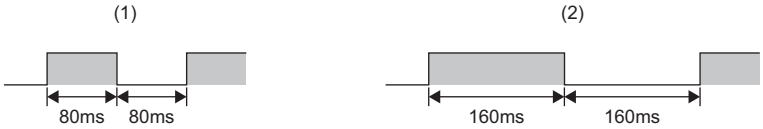
Programmable controller	Power supply used for pull-up resistors transistor output
AC power type CPU module.	Service power supply

Ex.

For FX5U-64MT/ES (sink input/sink output)



• Because 64 input points (8 rows × 8 columns) are received in a cycle of 80 or 160 ms, the ON/OFF duration of each input signal should be greater than or equal to the value shown below:



(1) FX5S/FX5UJ CPU module: When using X0 to X7 FX5U/FX5UC CPU module: When using X0 to X17

(2) FX5S/FX5UJ CPU module: When using X10 and higher FX5U/FX5UC CPU module: When using X20 and higher

Operation error

Error code (SD0/SD8067)	Description
3405H	The value specified by (n) is outside the following range. 2 to 8
2820H	• The device range specified by (s) exceeds the corresponding device range. • The device range specified by (d1) exceeds the corresponding device range. • The device range specified by (d2) exceeds the corresponding device range.
1811H	The number of times the MTR instruction is simultaneously driven exceeds 1.
3582H	The MTR instruction is used in the interrupt program.

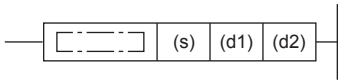
8.20 Initial State

Initial State

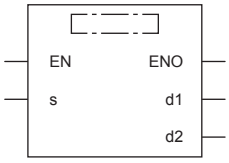
IST

FX5S FX5UJ FX5U FX5UC

Automatically controls the initial state and special relays in a step ladder program.

Ladder diagram	Structured text
	ENO:=IST(EN, s, d1, d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head bit device number of the selector switch in the operation mode	—	Bit	ANYBIT_ARRAY (Number of elements: 8)
(d1)	Smallest state relay number of practical state relays in the automatic mode ((d1) < (d2))	—	Bit	ANY_BOOL
(d2)	Largest state relay number of practical state relays in the automatic mode ((d1) < (d2))	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○*1	○*3	—	—	—	—	—	—	—	—	—
(d1)	○*2	—	—	—	—	—	—	—	—	—	—
(d2)	○*2	—	—	—	—	—	—	—	—	—	—

*1 S cannot be used.
*2 Only S can be used.
*3 T, ST, and C cannot be used.

Processing details

- Specify the head input in the operation mode in (s).
- Selector switches in the operation mode occupy eight devices from the head device.

- The switch functions shown in the table below are assigned to each of the devices specified for selector switches in the operation mode. When X20 is assigned, it is necessary to set X20 to X24 as rotary switches so that they do not turn ON at the same time. It is not necessary to wire unused switches, but they cannot be used for any other purpose because they are occupied by IST instruction.

Source	Device number (example)	Switch function	Descriptions
(s)	X20	Individual operation	Each load is turned ON and OFF by an individual pushbutton switch.
(s)+1	X21	Return to zero point	When the pushbutton switch for zero return is pressed, the machine automatically returns to the zero point.
(s)+2	X22	Stepping	Every time the start button is pressed, the machine performs one process.
(s)+3	X23	Cycle operation	When the start button is pressed while the machine is located at the zero point, the machine performs one cycle of automatic operation and stops at the zero point. If the stop button is pressed in the middle of one cycle, the machine stops immediately. When the start button is pressed after that, the machine performs the continuous operation from the last position, and automatically stops at the zero point.
(s)+4	X24	Continuous operation	When the start button is pressed while the machine is located at the zero point, the machine starts continuous operation. When the stop button is pressed, the machine finishes the current cycle until the zero point, and then stops at the zero point.
(s)+5	X25	Zero return start	Starts return to the zero point.
(s)+6	X26	Automatic start	Starts the stepping operation, cycle operation or continuous operation.
(s)+7	X27	Stop	Stops each operation.

- Specify the smallest device number of practical state relays in (d1) (for the automatic mode).
- Specify the largest device number of practical state relays in (d2) (for the automatic mode).

• While the command input is ON, the following devices are automatically switched and controlled. While the command input is OFF, the devices are not switched.

Device number	Descriptions	ON/OFF condition	
SM8040	STL transfer disable	ON condition	Always remains ON in the individual operation. Always remains ON in the stepping operation except when the [START] button is pressed. Turns ON when the [STOP] button is pressed during return to the zero point and in the cycle operation.
		OFF condition	Turns OFF when the [START] button is pressed in the stepping operation. Turns OFF after the [STOP] button is pressed during return to the zero point and in the cycle operation.
SM8041	Transfer start	ON condition	Turns ON when the [START] button is pressed in the stepping operation and cycle operation. Turns ON after the [START] button is pressed in the cycle operation.
		OFF condition	Turns OFF when the operation mode is changed from RUN to STOP. Always remains OFF in the individual operation and during return to the zero point. Turns OFF after the [STOP] button is pressed in the continuous operation.
SM8042	Start pulse	ON condition	Instantaneously turns ON when the [START] button is pressed.
		OFF condition	Remains OFF except in the ON condition.
SM8043	Zero return complete	ON condition	Turns ON when return to the zero point is completed (in the user program).
		OFF condition	Turns OFF when the operation mode is changed from RUN to STOP. Remains OFF while return to the zero point is not completed.
SM8044	Zero return condition	ON condition	Turns ON when the zero point condition is established (in the user program).
		OFF condition	Turns OFF when the operation mode is changed from RUN to STOP. Remains OFF while return to the zero point is not completed.
SM8045	All output reset disable	ON condition	Turns ON when all-output reset is not executed (in the user program).
		OFF condition	Turns OFF when all-output reset is executed (in the user program).
SM8046	STL state ON	ON condition	Turns ON when SM8047 (Enable STL monitoring) is ON and either step relay (device S) is ON.
		OFF condition	Turns OFF when SM8047 (Enable STL monitoring) is OFF or when SM8047 (STL monitor enable) is ON and all step relays (device S) are OFF.
SM8047	Enable STL monitoring	ON condition	Turns ON when the IST instruction command is given.
		OFF condition	Turns OFF when the step ladder is finished (in the user program).

Device number	Descriptions	ON/OFF condition	
S0	Individual operation initial state	ON condition	when a individual operation mode is selected.
		OFF condition	Except a individual operation mode.
S1	Zero return initial state	ON condition	when a zero return operation mode is selected.
		OFF condition	Except a zero return operation mode.
S2	Automatic operation initial state	ON condition	when a automatic operation mode is selected.
		OFF condition	Except a automatic operation mode.

• Do not program the following state relays as general state relays;

Device number	Descriptions	ON/OFF condition	
S0 to S9	Occupied for the initial state • S0 to S2 are used for individual operation, zero return and automatic operation as shown above. • S3 to S9 can be used arbitrarily.	ON condition	Turns ON when a step relay (device S) is selected for the initial state.
		OFF condition	Turns OFF when no step relay (device S) is selected.
S10 to S19	Occupied for zero return	ON condition	Turns ON when a step relay (device S) is selected for return to the zero point.
		OFF condition	Turns OFF when no step relay (device S) is selected.

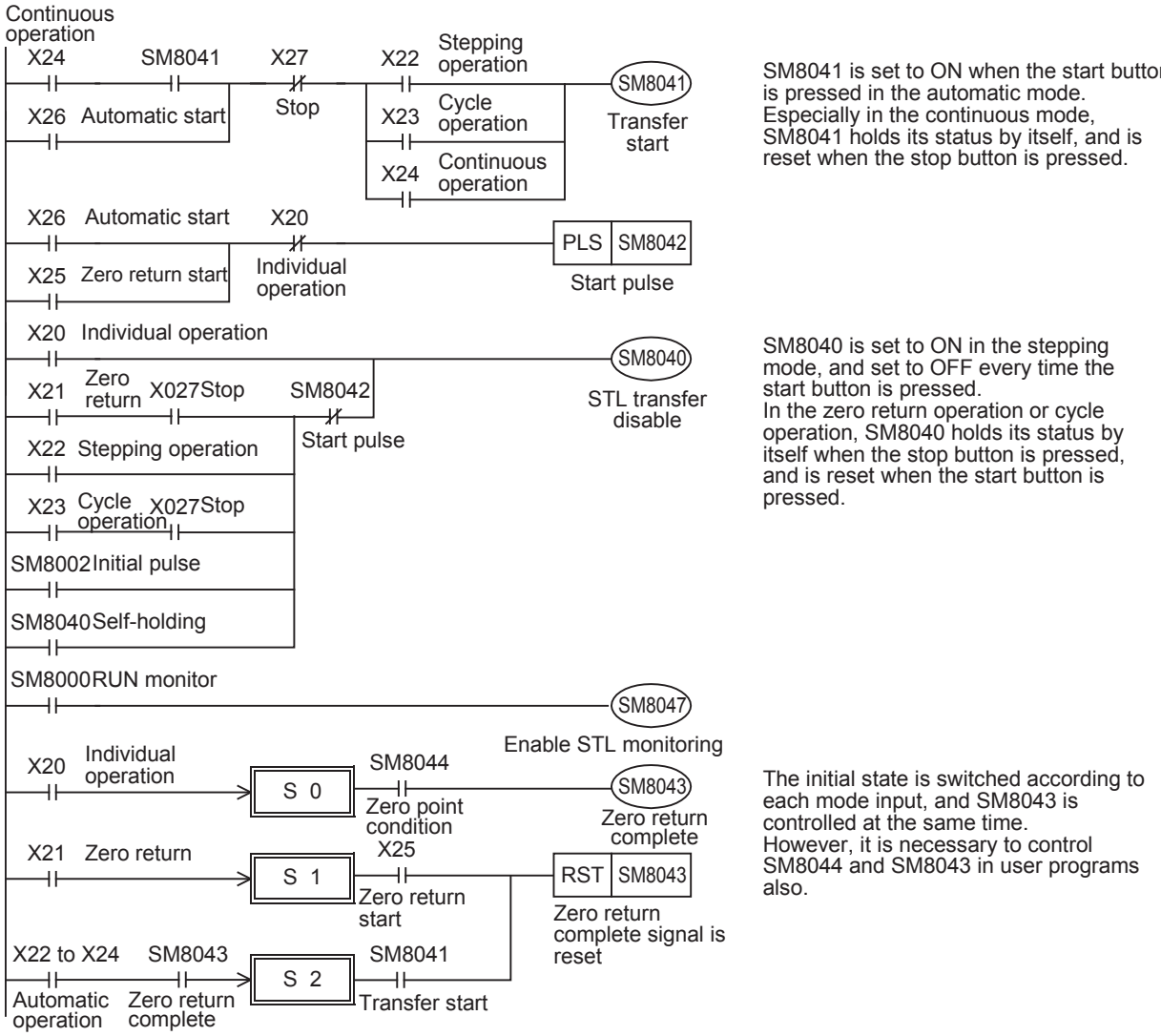
• If the devices are switched among individual operation (X20), zero return (X21) and automatic operation (X22, 23 and X24) while the zero return complete device (SM8043) is OFF, all outputs are set to OFF. Automatic operation can be started again after zero return is completed.

Precautions

- It is not necessary to use all switches for mode selection. When some switches are not used, leave the corresponding numbers in the unused status. Such numbers cannot be used for any other purpose.
- The IST instruction should be programmed earlier than a series of STL circuit such as state relays S0 to S2.
- Use the state relays S10 to S19 for the zero return operation. In the final state in the zero return operation, set SM8043 to ON, and then let it be reset to OFF by itself.
- The IST instruction can only be used once in a program.

■IST instruction equivalent circuit

- The details on special relays (SM) and initial state relays (S0 to S9) which are automatically controlled by the IST instruction is as shown in the equivalent circuit below. (Refer to the equivalent circuit below for reference.) This equivalent circuit cannot be programmed.

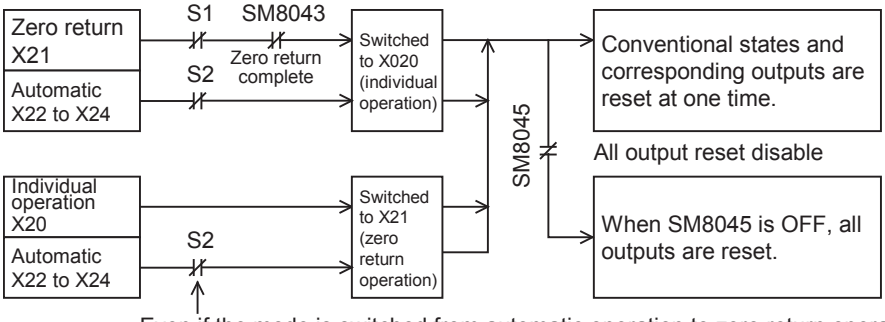


SM8041 is set to ON when the start button is pressed in the automatic mode. Especially in the continuous mode, SM8041 holds its status by itself, and is reset when the stop button is pressed.

SM8040 is set to ON in the stepping mode, and set to OFF every time the start button is pressed. In the zero return operation or cycle operation, SM8040 holds its status by itself when the stop button is pressed, and is reset when the start button is pressed.

The initial state is switched according to each mode input, and SM8043 is controlled at the same time. However, it is necessary to control SM8044 and SM8043 in user programs also.

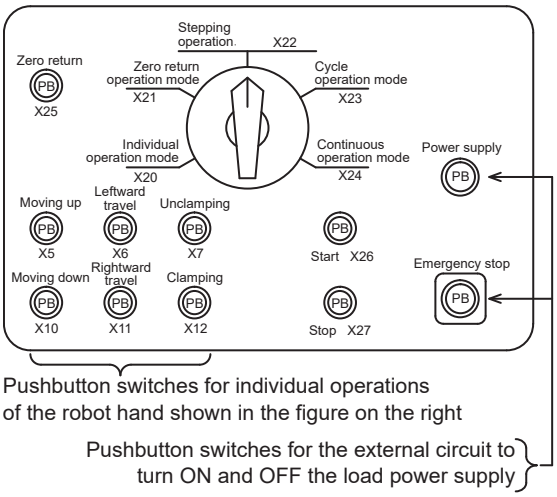
• When the operation mode is switched among the individual operation, zero return operation and automatic operation,all outputs and conventional states are reset at one time unless the machine is located in the zero point. (Reset of all outputs is not executed when SM8045 is driven.)



Even if the mode is switched from automatic operation to zero return operation while S2 is ON, state relays (except initial state relays) and outputs are not reset.

■Example of IST instruction introduction (example of workpiece transfer mechanism)

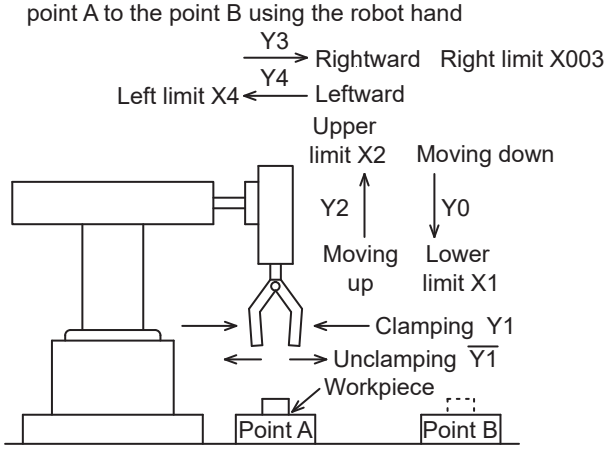
• Operation mode



Pushbutton switches for individual operations of the robot hand shown in the figure on the right

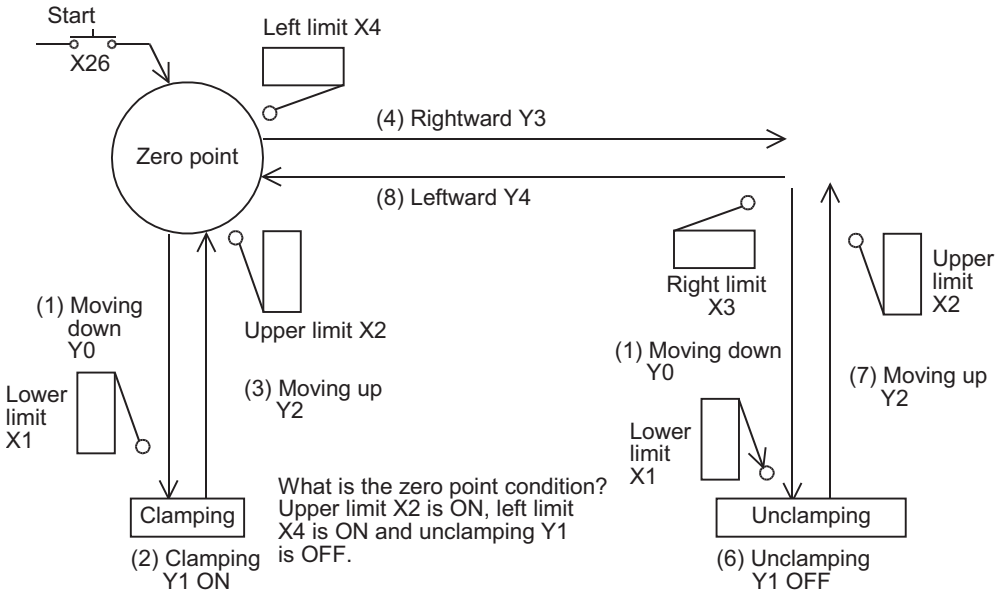
Pushbutton switches for the external circuit to turn ON and OFF the load power supply

Mechanism for transferring a workpiece from the point A to the point B using the robot hand



Operation mode		Contents of operation
Manual mode	Individual operation mode:	Each load is turned ON and OFF by an individual pushbutton switch.
	Zero return operation mode:	When the pushbutton switch for zero return is pressed, the machine automatically returns to the zero point.
Automatic mode	Stepping operation mode:	Every time the start button is pressed, the machine performs one process.
	Cycle operation mode	When the start button is pressed while the machine is located at the zero point, the machine performs one cycle of automatic operation and stops at the zero point. If the stop button is pressed in the middle of one cycle, the machine stops immediately. When the start button is pressed after that, the machine performs the continuous operation from the last position, and automatically stops at the zero point.
	Continuous operation mode	When the start button is pressed while the machine is located at the zero point, the machine starts continuous operation. When the stop button is pressed, the machine finishes the current cycle until the zero point, and then stops at the zero point.

• Transfer mechanism

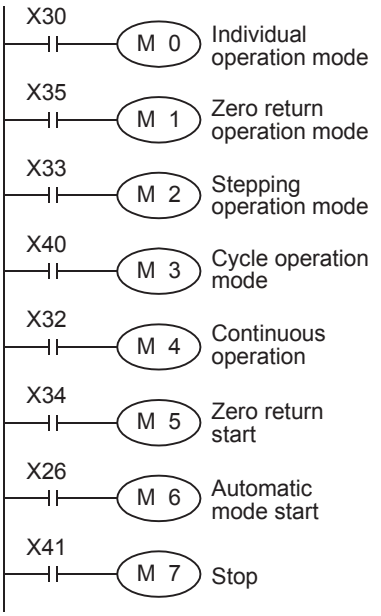


- For using IST instruction, it is necessary to assign inputs having consecutive device numbers as shown below for mode inputs. When using non-consecutive inputs or omitting some modes, change the layout by using an auxiliary relay as the head input for mode specification as shown in the figure below.

Input device	Assignment
X20	Individual operation mode
X21	Zero return operation mode
X22	Stepping operation mode
X23	Cycle operation mode
X24	Continuous operation mode
X25	Zero return start
X26	Automatic mode start
X27	Stop

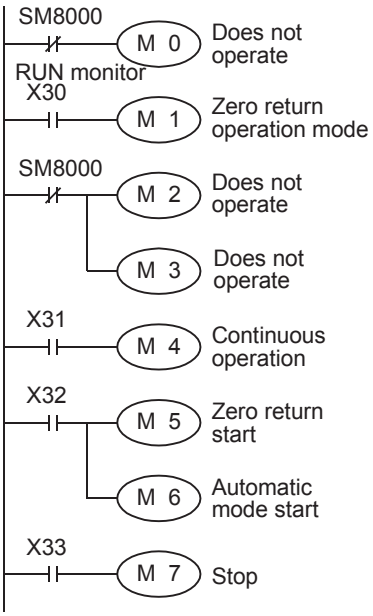
When inputs do not have consecutive device numbers

Example:
X30: Individual operation mode
X35: Zero return operation mode
X33: Stepping operation mode
X40: Cycle operation mode
X32: Continuous operation mode
X34: Zero return start
X26: Automatic mode start
X41: Stop



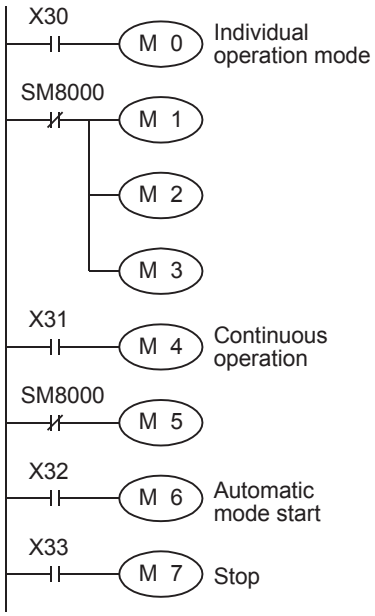
When only the continuous operation mode and zero return operation mode are used

Example:
X30: Zero return operation mode
X31: Continuous operation mode
X32: Automatic mode start zero return start
X33: Stop

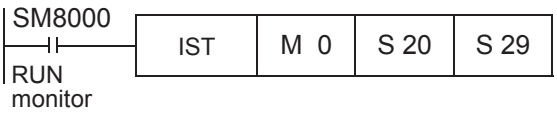


When only the continuous operation mode and individual operation mode are used

Example:
X30: Individual operation mode
X31: Continuous operation mode
X32: Automatic mode start
X33: Stop



In this example, M0 is used as the head input for mode specification.

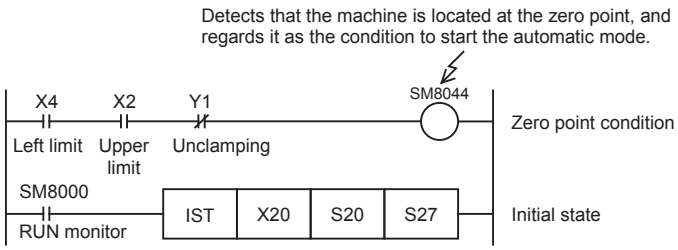


- Special relay (SM) used in the IST instruction are classified into two types. Some special relays are automatically controlled by the IST instruction itself according to the situation. Other special relays should be controlled by a program for preparation of operation or for purpose of control.

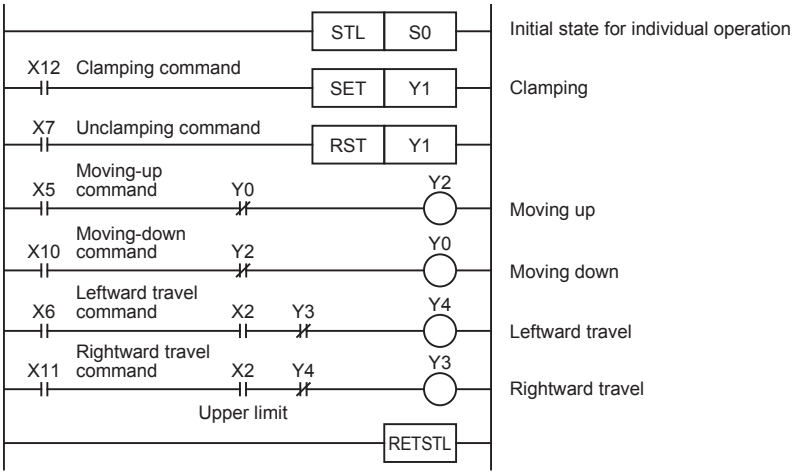
Special relay	Description	Remarks
SM8040 (STL transfer disable)	When this special relay turns ON, transfer of every state is disabled. Individual operation mode: SM8040 is always effective. Zero return operation mode and cycle operation mode: When the stop button is pressed, the operation is held until the start button is pressed. Stepping operation mode: SM8040 is always effective except when the start button is pressed. When the start button is pressed, SM8040 is not effective and transfer of states is allowed. Others: The operation is latched when the PLC mode switches from STOP to RUN, and reset when the start button is pressed. Even in the transfer disabled status, the operation is held for outputs in the states.	Special relays automatically controlled by the IST instruction
SM8041 (Transfer start)	This special relay allows transfer from the initial state S2 to the next state. Individual operation mode and zero return operation mode: SM8041 is not effective. Stepping operation mode and cycle operation mode: SM8041 is effective only while the start button is pressed and held. Continuous operation mode: The operation is latched when the start button is pressed, and cleared when the stop button is pressed.	
SM8042 (Start pulse)	SM8042 is activated instantaneously only when the start button is pressed.	
SM8047 (Enable STL monitoring)	When IST instruction is executed, SM8047 is set to ON. When the SM8047 turns ON, STL monitoring becomes valid, and state relay numbers (S0 to S899) in the ON status are stored in turn in the ascending order of device number to the special register SD8040 to SD8047. Up to eight state relay numbers in the ON status can be monitored. If either state relay is ON, the special auxiliary relay SM8046 is set to ON.	
SM8043 (Zero return complete)	Set this special relay (SM) to ON by a user program when the machine returns to the zero point in the zero return operation mode.	Special relays controlled by a sequence program
SM8044 (Zero point condition)	Detect the zero point condition of the machine, and drive this special relay. This signal is effective in every mode.	
SM8045 (All output reset disable)	When the mode is switched among individual operation mode, zero return operation mode and automatic mode, all outputs and operation state relays are reset if the machine is not located at the zero point. If SM8045 has been set to ON in advance, however, only operation state relays are reset.	

Program example

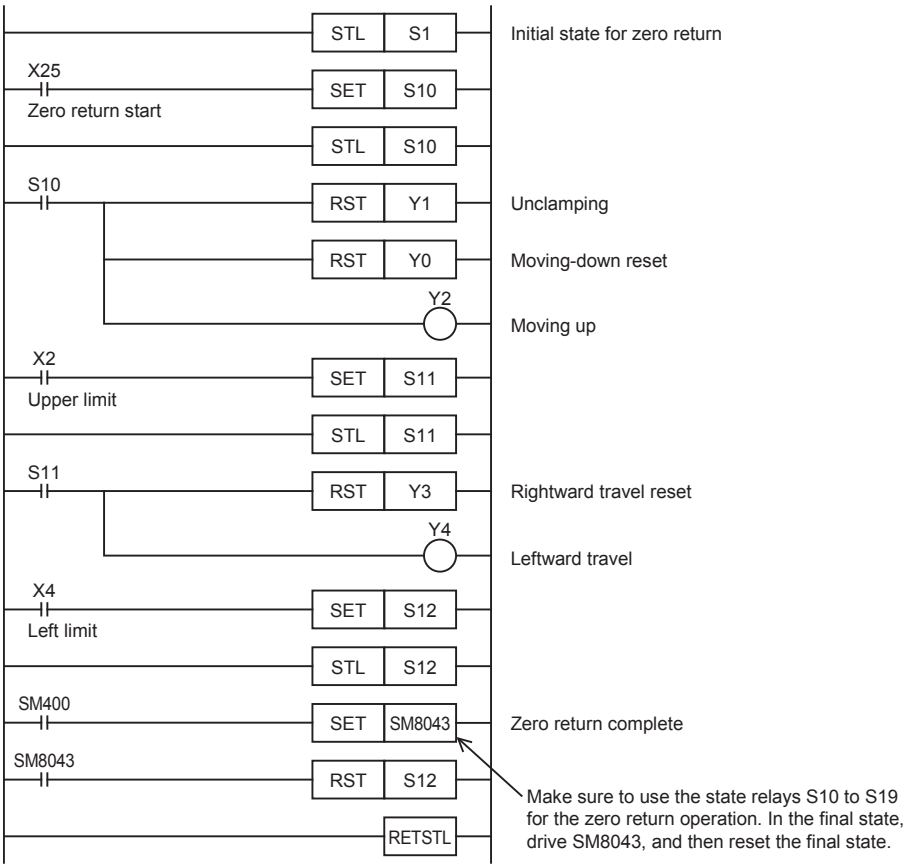
- While the machine is operating, the operation mode can be switched arbitrarily (among stepping operation, cycle operation and continuous operation) in the automatic mode. When the operation mode is switched between the individual operation mode, zero return operation mode and automatic mode while the machine is operating, all outputs are reset once to assure safety, after which the following mode becomes valid. (While SM8045 (All output reset disable) is ON, outputs are not reset at all.)



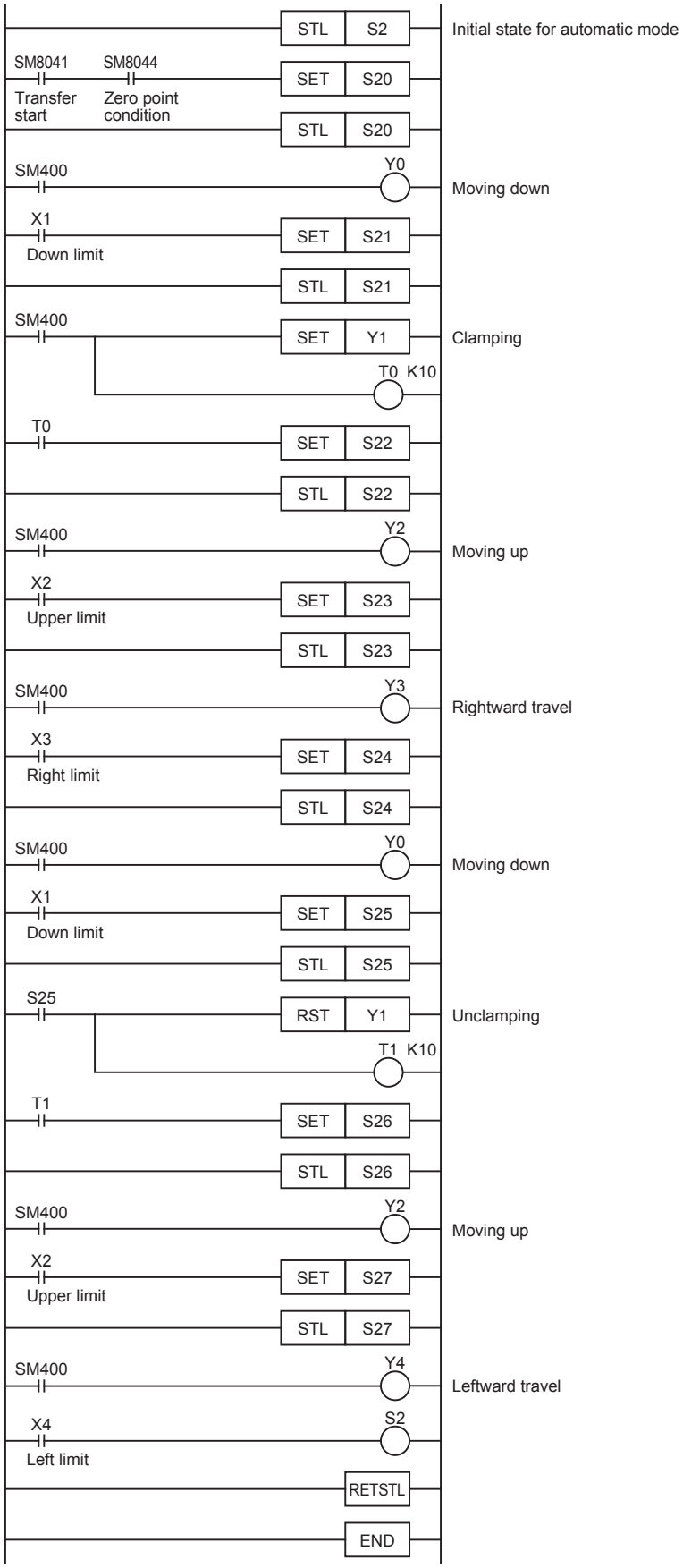
- Programming is not required when the individual operation mode is not provided.



- Programming is not required when the zero return operation mode is not provided. It is necessary to set SM8043 (zero return complete) to ON before starting the automatic mode.



• Automatic mode (stepping operation mode, cycle operation mode or continuous operation mode)



Operation error

Error code (SD0/SD8067)	Description
1811H	The number of times the IST instruction is simultaneously driven exceeds 1.
2820H	The device numbers specified in (d1) and (d2) show the following relationship: (d1) ≥ (d2)
	Eight points are not secured from the device specified in (s).
	An unavailable device is set in (s).
	An unavailable device is set in (d1).
	An unavailable device is set in (d2).

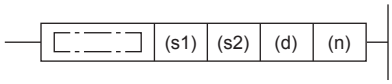
8.21 Drum Sequence

16-bit binary data absolute method

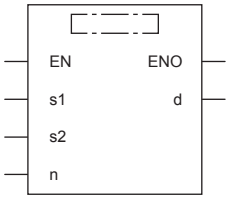
ABSD

FX5S FX5UJ FX5U FX5UC

This instruction creates many output patterns corresponding to the current value (16-bit binary data) of a counter.

Ladder diagram	Structured text
	ENO:=ABSD(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device number storing the data table (with rising and falling point data)	—	16-bit signed binary	ANY16
(s2)	Counter number for monitoring the current value compared with the data table	—	16-bit signed binary	ANY16
(d)	Head bit device number to be output	—	Bit	ANY_BOOL
(n)	Number of lines in the table and the number of output bit devices	1 to 64	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

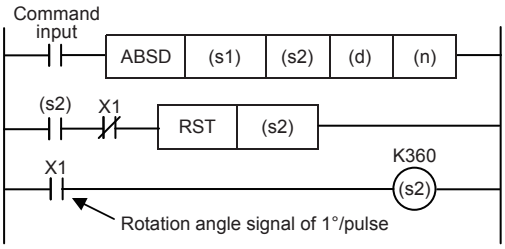
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	—	—	—	—
(s2)	—	○*1	—	—	—	—	○	—	—	—	—
(d)	○	○*2	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 Only C can be used.
*2 T, ST, and C cannot be used.

Processing details

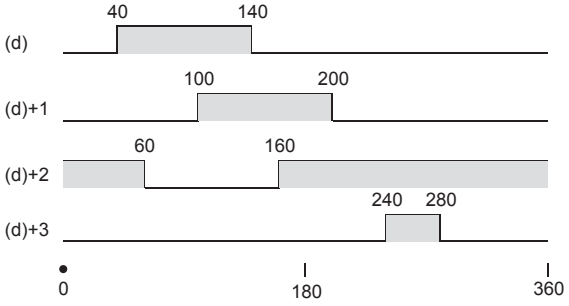
- In this example, outputs are controlled to on or off by one table rotation (0 to 360° using the rotation angle signal of 1°/pulse).
- The current value (s2) of the counter is compared with the data table with "n" lines starting from (s1) (which occupies "n" lines × 2 devices), and consecutive "n" outputs starting from (d) are controlled to on or off during one rotation.



- Write the following data to (s1) to (s1)+2(n)-1 in advance by a transfer instruction: For example, store 16-bit rising point data in even-numbered devices and 16-bit falling point data in odd-numbered devices.

Rising point		Falling point		Target output
—	Data value (example)	—	Data value (example)	
(s1)	40	(s1)+1	140	(d)
(s1)+2	100	(s1)+3	200	(d)+1
(s1)+4	160	(s1)+5	60	(d)+2
(s1)+6	240	(s1)+7	280	(d)+3
⋮	—	⋮	—	⋮
(s1)+2(n)-2		(s1)+2(n)-1		(d)+n-1

- The following figure shows the output patterns for device points (n) starting from (d) when the command input is set to on. Each rising point/falling point can be changed by overwriting the data in (s1) to (s1)+2(n)-1.



Precautions

- When specifying the nibble of a bit device to (s1), specify a multiple of 16 (0, 16, 32, 64 ...) as a device number and always specify K4 for the number of digits.
- The value of (n) determines the number of target outputs ($1 \leq (n) \leq 64$).
- Even if the command input is set to OFF, the ON/OFF status of outputs does not change.

Operation error

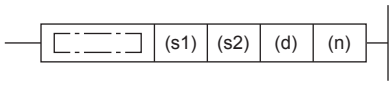
Error code (SD0/SD8067)	Description
2820H	The number of device points specified by (s1) or (d) is insufficient.
3405H	The value specified by (n) is outside the following range. 1 to 64

32-bit binary data absolute method

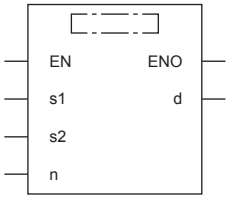
DABSD

FX5S FX5UJ FX5U FX5UC

This instruction creates many output patterns corresponding to the current value (32-bit binary data) of a counter.

Ladder diagram	Structured text
	ENO:=DABSD(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device number storing the data table (with rising and falling point data)	—	32-bit signed binary	ANY32
(s2)	Counter number for monitoring the current value compared with the data table	—	32-bit signed binary	ANY32
(d)	Head bit device number to be output	—	Bit	ANY_BOOL
(n)	Number of lines in the table and the number of output bit devices	1 to 64	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

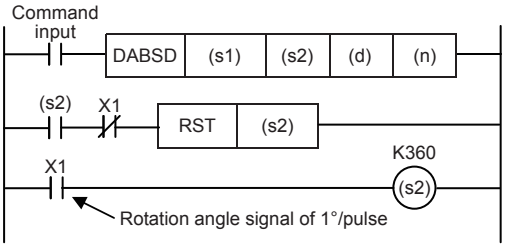
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	—	—	—	—
(s2)	—	○*1	—	—	○	—	○	—	—	—	—
(d)	○	○*2	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 Only C (32 bits) can be used.
*2 T, ST, and C cannot be used.

Processing details

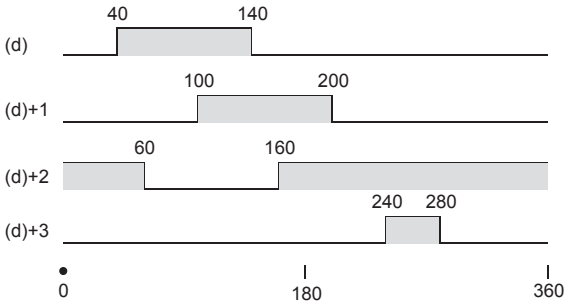
- In this example, outputs are controlled to on or off by one table rotation (0 to 360° using the rotation angle signal of 1°/pulse).
- The current value (s2) of the counter is compared with the data table with "n" lines starting from (s1) (which occupies "n" lines × 4 devices), and consecutive "n" outputs starting from (d) are controlled to on or off during one rotation.



- Write the following data to (s1), (s1)+1 to (s1)+4(n)-2, and (s1)+4(n)-1 in advance by a transfer instruction: For example, store 32-bit rising point data in even-numbered devices and 32-bit falling point data in odd-numbered devices.

Rising point		Falling point		Target output
—	Data value (example)	—	Data value (example)	
(s1)+1, (s1)	40	(s1)+3, (s1)+2	140	(d)
(s1)+5, (s1)+4	100	(s1)+7, (s1)+6	200	(d)+1
(s1)+9, (s1)+8	160	(s1)+11, (s1)+10	60	(d)+2
(s1)+13, (s1)+12	240	(s1)+15, (s1)+14	280	(d)+3
⋮	—	⋮	—	⋮
(s1)+4(n)-3, (s1)+4(n)-4		(s1)+4(n)-1, (s1)+4(n)-2		(d)+n-1

- The following figure shows the output patterns for device points (n) starting from (d) when the command input is set to on. Each rising point/falling point can be changed by overwriting the data in (s1) to (s1)+2(n)-1.



Precautions

- The DABSD instruction can specify a high-speed counter. When the high-speed counter is specified, the output pattern contains response delay caused by the scan cycle with regard to the current value of a counter.
- When specifying the nibble of a bit device to (s1), specify a multiple of 16 (0, 16, 32, 64 ...) as a device number and always specify K8 for the number of digits.
- The value of (n) determines the number of target outputs ($1 \leq (n) \leq 64$).
- Even if the command input is set to OFF, the ON/OFF status of outputs does not change.

Operation error

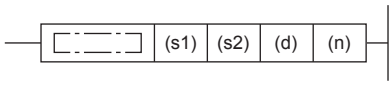
Error code (SD0/SD8067)	Description
2820H	The number of device points specified by (s1) or (d) is insufficient.
3405H	The value specified by (n) is outside the following range. 1 to 64

Relative method

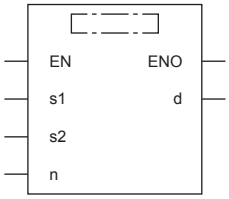
INCD

FX5S FX5UJ FX5U FX5UC

This instruction creates many output patterns using a pair of counters.

Ladder diagram	Structured text
	ENO:=INCD(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head word device number storing the set value	—	16-bit signed binary	ANY16
(s2)	Head counter number for monitoring current value is monitored	—	16-bit signed binary	ANY16
(d)	Head bit device number to be output	—	Bit	ANY_BOOL
(n)	Number of output bit devices	1 to 64	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	—	—	—	—
(s2)	—	○*1	—	—	—	—	○	—	—	—	—
(d)	○	○*2	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

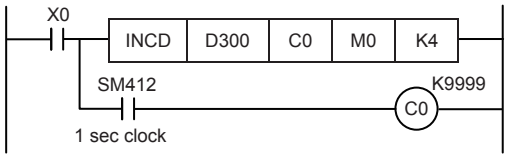
*1 Only C can be used.
*2 T, ST, and C cannot be used.

Processing details

- The current value of a counter is compared with the data table having "n" lines starting from (s1) (which occupies "n" lines × 1 device). When the value is equivalent to the table data, the current output is reset, and the next output is controlled. In this way, the ON/OFF status of specified outputs is controlled in turn.

■Operation example

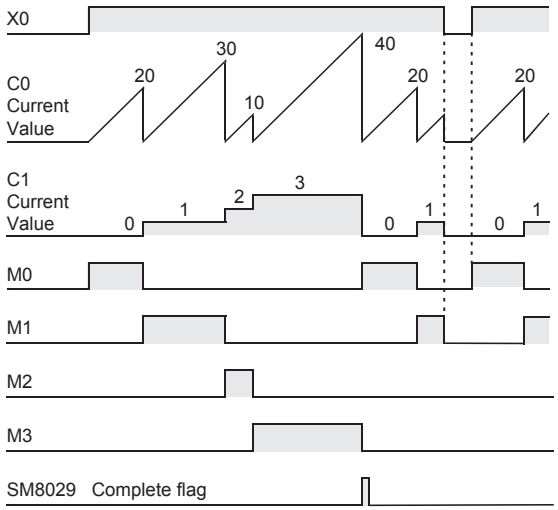
- The following ladder example shows the operation. (s2) occupies two points. In the following timing chart, C0 and C1 correspond to the two points.



- Suppose that the following data is written in advance by a transfer instruction:

Device storing data		Output	
—	Data value (example)	—	Example
(s1)	D300=20	(d)	M0
(s1)+1	D301=30	(d)+1	M1
(s1)+2	D302=10	(d)+2	M2
(s1)+3	D303=40	(d)+3	M3
⋮	⋮	⋮	⋮
(s1)+(n)-1	—	(d)+(n)-1	—

- Timing chart



- When the command contact turns on, the output M0 turns on.
- When the current value of C0 reaches the comparison value D300, the output M0 is reset, "1" is added to the count value of the process counter C1, and the current value of the counter C0 is reset.
- The next output M1 turns ON.
- When the current value of C0 reaches the comparison value D301, the output M1 is reset, "1" is added to the count value of the process counter C1, and the current value of the counter C0 is reset.
- The current value is compared for up to "n (K4)" outputs in the same way ($1 \leq (n) \leq 64$).
- When the final process specified by (n) is finished, the execution complete flag SM8029 turns on and remains on for one operation cycle. SM8029 is used for many instructions as the instruction execution complete flag. Use SM8029 as a contact just after a corresponding instruction.
- The program execution returns to the beginning, and outputs are repeated.

Precautions

When specifying the nibble of a bit device to (s1), specify a multiple of 16 (0, 16, 32, 64 ...) as a device number.

Operation error

Error code (SD0/SD8067)	Description
2820H	The number of device points specified by (s1), (s2), or (d) is insufficient.
3405H	The value specified by (n) is outside the following range. 1 to 64

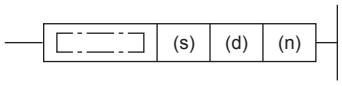
8.22 Check Code

Check code

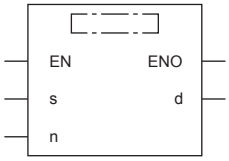
CCD(P)

FX5S FX5UJ FX5U FX5UC

These instructions calculate the horizontal parity value and sum check value in the error check methods used in communication. There is another check method, called CRC (cyclic redundancy check). For obtaining CRC value, use the CRC(P) instructions.

Ladder diagram	Structured text
	ENO:=CCD(EN,s,n,d); ENO:=CCDP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number of applicable device	—	16-bit signed binary	ANY16
(d)	Head device number storing the calculated data	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 2)
(n)	Number of data	1 to 32767	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

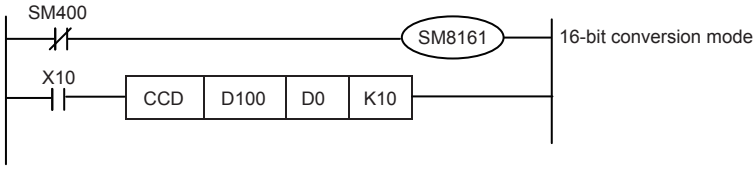
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○*1	○	—	—	—	○	—	—	—	—
(d)	○	○*1	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

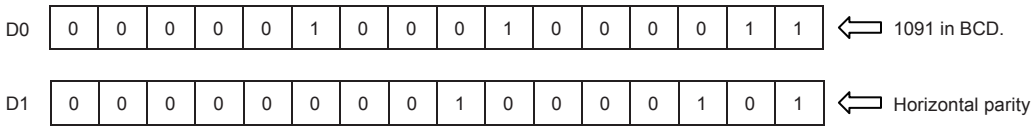
Processing details

- These instructions calculate the addition data and horizontal parity value of data stored in (s) to (s)+(n)-1. The addition data is stored to (d), and the horizontal parity value is stored to (d)+1. The 16-bit mode and 8-bit mode are available for these instructions. For the operation in each mode, refer to the proceeding pages.
  - 16-bit conversion mode (while SM8161 is OFF)

    With regard to (n) data points starting from (s), the addition data and horizontal parity data of high-order 8 bits and low-order 8 bits are stored to (d) and (d)+1 respectively.
    SM8161 is shared with the ASCI(P), HEXA(P), and CRC(P) instructions. SM8161 must always be off in the 16-bit mode.
    SM8161 is cleared when the CPU module mode is changed from RUN to STOP.
    In the following program, conversion is executed as follows:

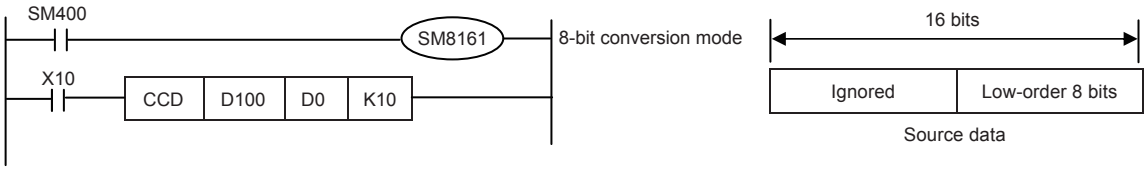


(s)	Example of data contents	
D100 lowest-order byte	K100 = 0 1 1 0 0 1 0 0	
D100 highest-order byte	K111 = 0 1 1 0 1 1 1 (1)	←
D101 lowest-order byte	K100 = 0 1 1 0 0 1 0 0	
D101 highest-order byte	K 98 = 0 1 1 0 0 0 1 0	
D102 lowest-order byte	K123 = 0 1 1 1 1 0 1 (1)	←
D102 highest-order byte	K 66 = 0 1 0 0 0 0 1 0	
D103 lowest-order byte	K100 = 0 1 1 0 0 1 0 0	
D103 highest-order byte	K 95 = 0 1 0 1 1 1 1 (1)	←
D104 lowest-order byte	K210 = 1 1 0 1 0 0 1 0	
D104 highest-order byte	K 88 = 0 1 0 1 1 0 0 0	
Total	K1091	
Horizontal parity	1 0 0 0 0 1 0 (1)	← When the number of "1" is odd, the horizontal parity is "1". When the number of "1" is even, the horizontal parity is "0".

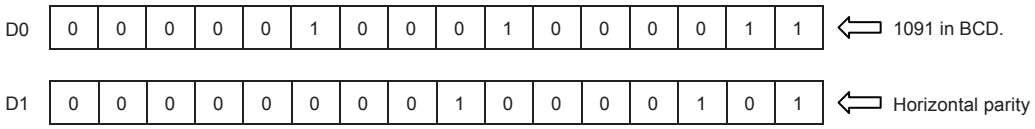


• 8-bit conversion mode (while SM8161 is ON)

With regard to (n) data points starting from (s), the addition data and horizontal parity data of only low-order 8 bits are stored to (d) and (d)+1 respectively.
SM8161 is shared with the ASCI(P), HEXA(P), and CRC(P) instructions. SM8161 must always be on in the 8-bit mode.
SM8161 is cleared when the CPU module mode is changed from RUN to STOP.
In the following program, conversion is executed as follows:



(s)	Example of data contents	
D100	K100 = 0 1 1 0 0 1 0 0	
D101	K111 = 0 1 1 0 1 1 1 (1)	←
D102	K100 = 0 1 1 0 0 1 0 0	
D103	K 98 = 0 1 1 0 0 0 1 0	
D104	K123 = 0 1 1 1 1 0 1 (1)	←
D105	K 66 = 0 1 0 0 0 0 1 0	
D106	K100 = 0 1 1 0 0 1 0 0	
D107	K 95 = 0 1 0 1 1 1 1 (1)	←
D108	K210 = 1 1 0 1 0 0 1 0	
D109	K 88 = 0 1 0 1 1 0 0 0	
Total	K1091	
Horizontal parity	1 0 0 0 0 1 0 (1)	← When the number of "1" is odd, the horizontal parity is "1". When the number of "1" is even, the horizontal parity is "0".



Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (s) or (d) exceeds the corresponding device range.
3405H	The value specified by (n) is outside the following range. 1 to 32767

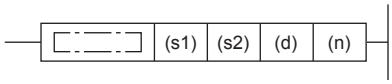
8.23 Data Operation Instruction

Searching 16-bit data

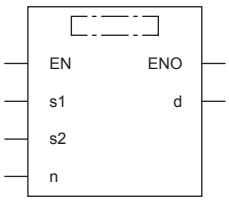
SERMM(P)

FX5S FX5UJ FX5U FX5UC

These instructions search for the same data, maximum value and minimum value in a data table.

Ladder diagram	Structured text
	ENO:=SERMM(EN,s1,s2,n,d); ENO:=SERMMP(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device number in which same data, maximum value and minimum value are searched	—	16-bit signed binary	ANY16
(s2)	Data to be searched for or device number storing data	—	16-bit signed binary	ANY16
(d)	Head device number storing number of same data, maximum value and minimum value detected by search	—	16-bit unsigned binary	ANY16_ARRAY (Number of elements: 5)
(n)	Number of data in which same data, maximum value and minimum value are searched	1 to 65535	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	—	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions search the same data as the 16-bit binary data of (s2) in (n) data starting from (s1), and store the search result in (d) to (d)+4.
- When the same data exists, five devices starting from (d) store the number of same data, first position, last position, maximum value position and minimum value position.
- When the same data does not exist, five devices starting from (d) store the number of same data, first position, last position, maximum value position and minimum value position. In this case, however, 0 is stored in three devices starting from (d) (which store the number of same data, first position and last position).

• The following table shows example of search result table configuration and data. (n=10)

Searched device (s1)	Searched data (s1) value (example)	Comparison data (s2) value (example)	Data position	Search result		
				Maximum value (d)+4	Same (d)	Minimum value (d)+3
(s1)	K100	K100	0		○ (First time)	
(s1)+1	K111		1			
(s1)+2	K100		2		○	
(s1)+3	K98		3			
(s1)+4	K123		4			
(s1)+5	K66		5			○
(s1)+6	K100		6		○ (Last)	
(s1)+7	K95		7			
(s1)+8	K210		8	○		
(s1)+9	K88		9			

• The following table shows example of search result table.

Device number	Description	Search result item
(d)	3	Number of same data
(d)+1	0	Same data position (first position)
(d)+2	6	Same data position (last position)
(d)+3	5	Minimum value position (last position)
(d)+4	8	Maximum value position (last position)

Precautions

- Comparison is executed algebraically. (-10<2)
- When there are two or more maximum or minimum values in the searched data, the last position of the max/min is stored respectively.
- When these instructions are driven, five devices ((d), (d)+1, (d)+2, (d)+3, and (d)+4) are occupied for storing the search result (d). Make sure that these devices are not used in other controls for the machine.

Operation error

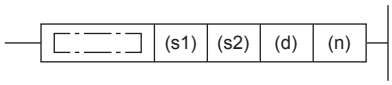
Error code (SD0/SD8067)	Description
2820H	The device range specified by (s1) or (d) exceeds the corresponding device range.
3405H	The value stored in a device specified by (n) is 0.

Searching 32-bit data

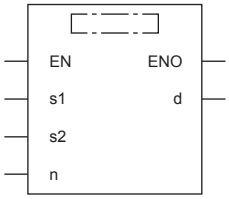
DSERMM(P)

FX5S FX5UJ FX5U FX5UC

These instructions search for the same data, maximum value and minimum value in a data table.

Ladder diagram	Structured text
	ENO:=DSERMM(EN,s1,s2,n,d); ENO:=DSERMMP(EN,s1,s2,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device number in which same data, maximum value and minimum value are searched	—	32-bit signed binary	ANY32
(s2)	Data to be searched for or device number storing data	—	32-bit signed binary	ANY32
(d)	Head device number storing number of same data, maximum value and minimum value detected by search	—	Bit/32-bit unsigned binary	ANY32_ARRAY (Number of elements: 5)
(n)	Number of data in which same data, maximum value and minimum value are searched	1 to 65535	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	—	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions search the same data as the 32-bit binary data of (s2)+1 and (s2) in (n) data starting from (s1)+1 and (s1), and store the search result in (d)+1, (d) to (d)+9, and (d)+8.
- When the same data exists, five 32-bit binary data devices starting from (d)+1 and (d) store the number of same data, first position, last position, maximum value position and minimum value position.
- When the same data does not exist, five 32-bit binary data devices starting from (d)+1 and (d) store the number of same data, first position, last position, maximum value position and minimum value position. In this case, however, 0 is stored in three 32-bit devices starting from (d)+1 and (d) (which store the number of same data, first position and last position).

• The following table shows example of search result table configuration and data. (n=10)

Searched device (s1)	Searched data (s1) value (example)	Comparison data (s2) value (example)	Data position	Search result		
				Maximum value (d)+9, (d)+8	Same (d)	Minimum value (d)+7, (d)+6
(s1)+1, (s)	K100000	K100000	0		○ (First time)	
(s1)+3, (s1)+2	K110100		1			
(s1)+5, (s1)+4	K100000		2		○	
(s1)+7, (s1)+6	K98000		3			
(s1)+9, (s1)+8	K123000		4			
(s1)+11, (s1)+10	K66000		5			○
(s1)+13, (s1)+12	K100000		6		○ (Last)	
(s1)+15, (s1)+14	K95000		7			
(s1)+17, (s1)+16	K910000		8	○		
(s1)+19, (s1)+18	K910000		9	○		

• The following table shows example of search result table.

Device number	Description	Search result item
(d)+1, (d)	3	Number of same data
(d)+3, (d)+2	0	Same data position (first position)
(d)+5, (d)+4	6	Same data position (last position)
(d)+7, (d)+6	5	Minimum value position (last position)
(d)+9, (d)+8	9	Maximum value position (last position)

Precautions

- Comparison is executed algebraically. (-10<2)
- When there are two or more maximum or minimum values in the searched data, the last position of the max/min is stored respectively.
- When these instructions are driven, five devices ([(d)+1,(d)], [(d)+3, (d)+2], [(d)+5, (d)+4], [(d)+7, (d)+6], and [(d)+9, (d)+8]) are occupied for storing the these result (d). Make sure that these devices are not used in other controls for the machine.

Operation error

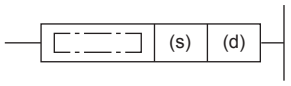
Error code (SD0/SD8067)	Description
2820H	The device range specified by (s1) or (d) exceeds the corresponding device range.
3405H	The value stored in a device specified by (n) is 0.

Bit check of 16-bit data

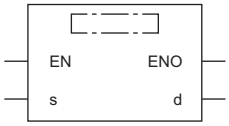
SUM(P)

FX5S FX5UJ FX5U FX5UC

These instructions store the total bits of 1 in the binary 16-bit data of the device specified by (s) to the device specified by (d).

Ladder diagram	Structured text
	ENO:=SUM(EN,s,d); ENO:=SUMP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

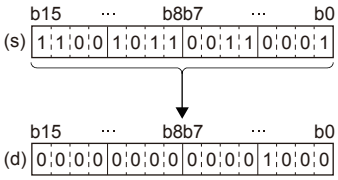
Operand	Description	Range	Data type	Data type (label)
(s)	Head device number that counts the total bits of 1	—	16-bit signed binary	ANY16
(d)	Head device number storing the total bits	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions store the total bits of 1 in the binary 16-bit data of the device specified by (s) to the device specified by (d).



The total of 1 is stored in the binary data. (In the example shown on the left, the total is 8.)

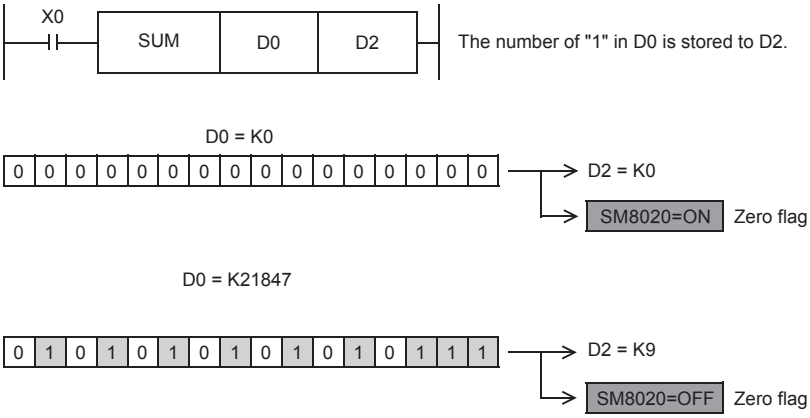
- When all binary 16-bit data of the device specified by (s) are 0 (off), the zero flag (SM8020) turns on.

Precautions

While the command input is off, the instruction is not executed. The output of the number of bits in the on status is latched in the previous status.

Program example

In the program example shown below, when X0 is ON, the number of bits in the ON status in D0 is counted, and stored to D2.



Operation error

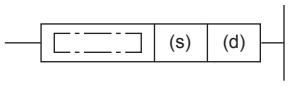
There is no operation error.

Bit check of 32-bit data

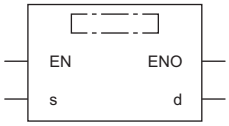
DSUM(P)

FX5S FX5UJ FX5U FX5UC

These instructions store the total bits of 1 in the binary 32-bit data of the device specified by (s) to the device specified by (d).

Ladder diagram	Structured text
	ENO:=DSUM(EN,s,d); ENO:=DSUMP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

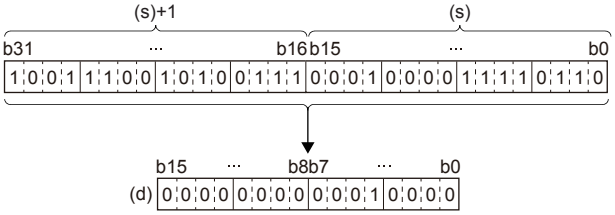
Operand	Description	Range	Data type	Data type (label)
(s)	Head device number that counts the total bits of 1	—	32-bit signed binary	ANY32
(d)	Head device number storing the total bits	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions store the total bits of 1 in the binary 32-bit data of the device specified by (s) to the device specified by (d).



The total of 1 is stored in the binary data. (In the example shown on the left, the total is 16.)

- When all binary 32-bit data of the device specified by (s) are 0 (off), the zero flag (SM8020) turns on.

Precautions

While the command input is off, the instruction is not executed. The output of the number of bits in the on status is latched in the previous status.

Operation error

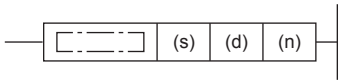
There is no operation error.

Bit judgment of 16-bit data

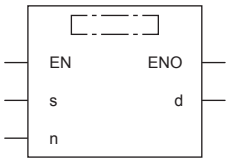
BON(P)

FX5S FX5UJ FX5U FX5UC

These instructions check whether (n) bit(s) of binary 16-bit data of the device specified by (s) are on or off, and output the result to the device specified by (d).

Ladder diagram	Structured text
	ENO:=BON(EN,s,n,d); ENO:=BONP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Word device number storing the data	—	16-bit signed binary	ANY16
(d)	Bit device number to be driven	—	Bit	ANY_BOOL
(n)	Bit position to be checked	0 to 15	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

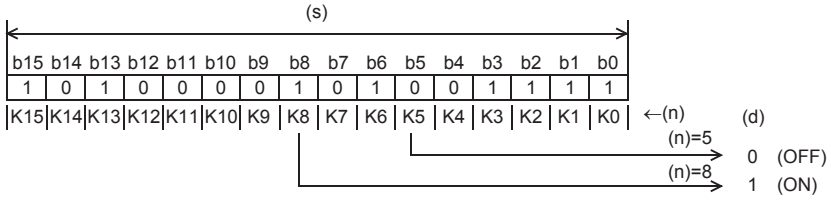
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○*1	○	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

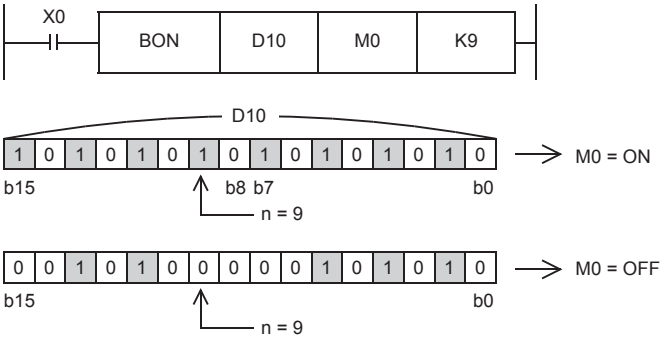
Processing details

- These instructions check whether (n) bit(s) of binary 16-bit data of the device specified by (s) are on or off, and output the result to the device specified by (d).
- When the result above is on, these instructions turn (d) on. When the result above is off, these instructions turn (d) off.
- When a constant (K) is specified in the device specified by (s), it is automatically converted into binary.



Program example

In the program example shown below, when the bit 9 (n = 9) in D10 is "1" (ON), M0 is set to "1" (ON).



Operation error

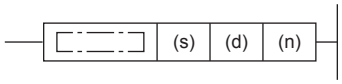
Error code (SD0/SD8067)	Description
3405H	The value specified by (n) is outside the following range. 0 to 15

Bit judgment of 32-bit data

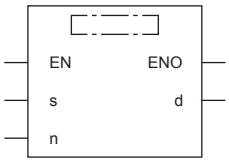
DBON(P)

FX5S FX5UJ FX5U FX5UC

These instructions check whether (n) bit(s) of binary 32-bit data of the device specified by (s) are on or off, and output the result to the device specified by (d).

Ladder diagram	Structured text
	ENO:=DBON(EN,s,n,d); ENO:=DBONP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Word device number storing the data	—	32-bit signed binary	ANY32
(d)	Bit device number to be driven	—	Bit	ANY_BOOL
(n)	Bit position to be checked	0 to 31	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

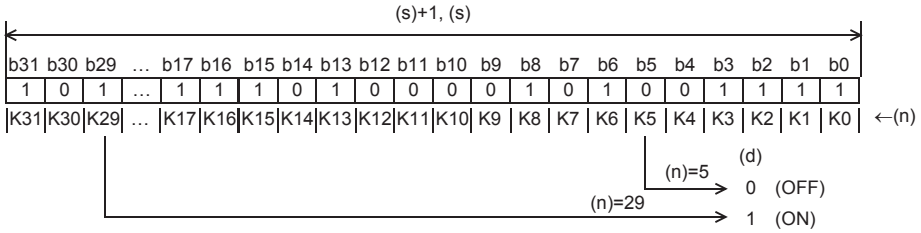
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○*1	○	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions check whether (n) bit(s) of binary 32-bit data of the device specified by (s) are on or off, and output the result to the device specified by (d).
- When the result above is on, these instructions turn (d) on. When the result above is off, these instructions turn (d) off.
- When a constant (K) is specified in the device specified by (s), it is automatically converted into binary.



Operation error

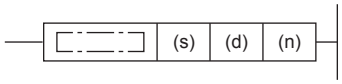
Error code (SD0/SD8067)	Description
3405H	The value specified by (n) is outside the following range. 0 to 31

Searching the maximum value of 16-bit data

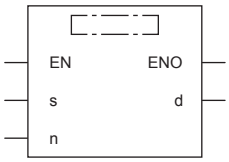
MAX(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions search the maximum value from the (n) point(s) of 16-bit binary data in the device starting from the one specified by (s), and store the maximum value in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=MAXP(EN,s,n,d);	ENO:=MAXP_U(EN,s,n,d);

FBD/LD*1



*1 The MAX and MAX_U instructions are not supported by the ST language and the FBD/LD language. Use MAX of the standard function.
☞ Page 1327 MAX(_E), MIN(_E)

Setting data

■Descriptions, ranges, and data types

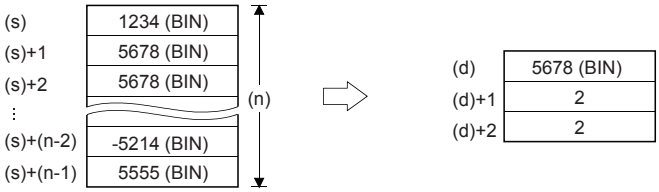
Operand		Description	Range	Data type	Data type (label)
(s)	MAX(P)	Head device number where the maximum value is searched	—	16-bit signed binary	ANY16_S
	MAX(P)_U			16-bit unsigned binary	ANY16_U
(d)	MAX(P)	Head device number for storing the maximum value	—	16-bit signed binary	ANY16_S_ARRAY (Number of elements: 3)
	MAX(P)_U			16-bit unsigned binary	ANY16_U_ARRAY (Number of elements: 3)
(n)		Number of data to be searched	0 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions search the maximum value from the (n) point(s) of 16-bit binary data in the device starting from the one specified by (s), and store the maximum value in the device specified by (d). These instructions start searching from the device specified by (s), and store the location from (s) of the first maximum value in (d)+1 and the number of maximum values in (d)+2.



(d): Maximum value
(d)+1: Position
(d)+2: Number of data

Operation error

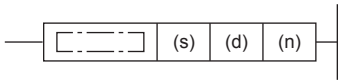
Error code (SD0/SD8067)	Description
2820H	The (n) point(s) of data in the device starting from the one specified by (s) exceed the corresponding device range.
	The device specified by (d) exceeds the corresponding device range.

Searching the maximum value of 32-bit data

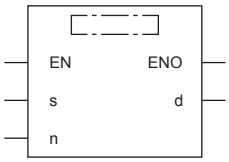
DMAX(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions search the maximum value from the (n) point(s) of 32-bit binary data in the device starting from the one specified by (s), and store the maximum value in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=DMAXP(EN,s,n,d);	ENO:=DMAXP_U(EN,s,n,d);

FBD/LD*1



*1 The DMAX and DMAX_U instructions are not supported by the ST language and the FBD/LD language. Use MAX of the standard function.
☞ Page 1327 MAX(_E), MIN(_E)

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)	DMAX(P)	Head device number where the maximum value is searched	—	32-bit signed binary	ANY32_S
	DMAX(P)_U			32-bit unsigned binary	ANY32_U
(d)	DMAX(P)	Head device number for storing the maximum value	—	32-bit signed binary	—*1 (ANY32_S_ARRAY)
	DMAX(P)_U			32-bit unsigned binary	—*1 (ANY32_S_ARRAY)
(n)		Number of data to be searched	0 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

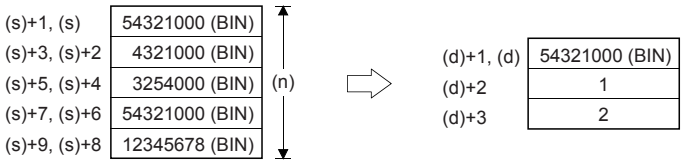
*1 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions search the maximum value from the (n) point(s) of 32-bit binary data in the device starting from the one specified by (s), and store the maximum value in the device specified by (d) and (d)+1. These instructions start searching from the device specified by (s), and store the location from (s) of the first minimum value in (d)+2 and the number of maximum values in (d)+3.



(d)+1, (d): Maximum value
(d)+2: Position
(d)+3: Number of data

Operation error

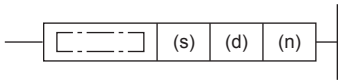
Error code (SD0/SD8067)	Description
2820H	The (n) point(s) of data in the device starting from the one specified by (s) exceed the corresponding device range.
	The device specified by (d) exceeds the corresponding device range.

Searching the minimum value of 16-bit data

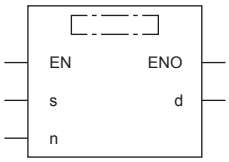
MIN(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions search the minimum value from the (n) point(s) of 16-bit binary data in the device starting from the one specified by (s), and store the minimum value in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=MINP(EN,s,n,d);	ENO:=MINP_U(EN,s,n,d);

FBD/LD*1



*1 The MIN and MIN_U instructions are not supported by the ST language and the FBD/LD language. Use MIN of the standard function.
☞ Page 1327 MAX(_E), MIN(_E)

Setting data

■Descriptions, ranges, and data types

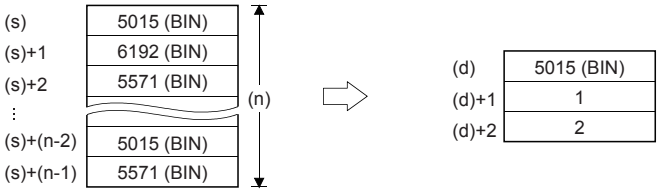
Operand		Description	Range	Data type	Data type (label)
(s)	MIN(P)	Head device number where the minimum value is searched	—	16-bit signed binary	ANY16_S
	MIN(P)_U			16-bit unsigned binary	ANY16_U
(d)	MIN(P)	Head device number for storing the minimum value	—	16-bit signed binary	ANY16_S_ARRAY (Number of elements: 3)
	MIN(P)_U			16-bit unsigned binary	ANY16_U_ARRAY (Number of elements: 3)
(n)		Number of data to be searched	0 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions search the minimum value from the (n) point(s) of 16-bit binary data in the device starting from the one specified by (s), and store the minimum value in the device specified by (d). These instructions start searching from the device specified by (s), and store the location from (s) of the first minimum value in (d)+1 and the number of minimum values in (d)+2.



(d): Minimum value
(d)+1: Position
(d)+2: Number of data

Operation error

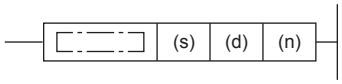
Error code (SD0/SD8067)	Description
2820H	The (n) point(s) of data in the device starting from the one specified by (s) exceed the corresponding device range.
	The device specified by (d) exceeds the corresponding device range.

Searching the minimum value of 32-bit data

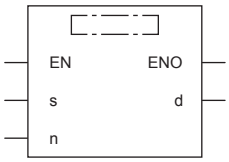
DMIN(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions search the minimum value from the (n) point(s) of 32-bit binary data in the device starting from the one specified by (s), and store the minimum value in the device specified by (d).

Ladder diagram	Structured text*1	
	ENO:=DMINP(EN,s,n,d);	ENO:=DMINP_U(EN,s,n,d);

FBD/LD*1



*1 The DMIN and DMIN_U instructions are not supported by the ST language and the FBD/LD language. Use MIN of the standard function.
☞ Page 1327 MAX(_E), MIN(_E)

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)	DMIN(P)	Head device number where the minimum value is searched	—	32-bit signed binary	ANY32_S
	DMIN(P)_U			32-bit unsigned binary	ANY32_U
(d)	DMIN(P)	Head device number for storing the minimum value	—	32-bit signed binary	—*1 (ANY32_S_ARRAY)
	DMIN(P)_U			32-bit unsigned binary	—*1 (ANY32_S_ARRAY)
(n)		Number of data to be searched	0 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

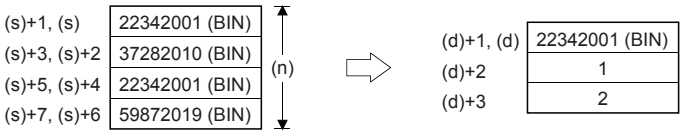
*1 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions search the minimum value from the (n) point(s) of 32-bit binary data in the device starting from the one specified by (s), and store the minimum value in the device specified by (d) and (d)+1. These instructions start searching from the device specified by (s), and store the location from (s) of the first minimum value in (d)+2 and the number of minimum values in (d)+3.



(d)+1, (d): Minimum value
(d)+2: Position
(d)+3: Number of data

Operation error

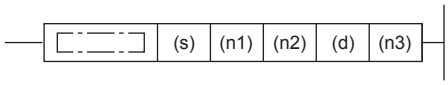
Error code (SD0/SD8067)	Description
2820H	The (n) point(s) of data in the device starting from the one specified by (s) exceed the corresponding device range.
	The device specified by (d) exceeds the setting area in the device/label memory.

Sorting 16-bit data

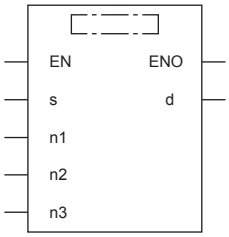
SORTTBL(_U)

FX5S FX5UJ FX5U FX5UC

These instructions sort data lines in the data table (sorting source) having ((n1)×(n2)) points specified by (s) in the ascending order based on the group data in the column number (n3), and store the result in the data table (sorting result) having ((n1)×(n2)) points specified by (d).

Ladder diagram	Structured text
	ENO:=SORTTBL(EN,s,n1,n2,n3,d); ENO:= SORTTBL_U(EN,s,n1,n2,n3,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)	SORTTBL	Head device number storing the data table	—	16-bit signed binary	ANY16
	SORTTBL_U			16-bit unsigned binary	ANY16_U
(n1)		Number of data (lines)	1 to 32	16-bit unsigned binary	ANY16_U
(n2)		Number of group data (columns)	1 to 6	16-bit unsigned binary	ANY16_U
(d)	SORTTBL	Head device number for storing the operation result	—	16-bit signed binary	ANY16
	SORTTBL_U			16-bit unsigned binary	ANY16_U
(n3)		Column number of group data (column) used as the basis of sorting	—	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n3)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions sort data lines in the data table (sorting source) having ((n1)×(n2)) points specified by (s) in the ascending order based on the group data in the column number (n3), and store the result in the data table (sorting result) having (n1×n2) points specified by (d).
- The data table configuration is explained in an example in which the sorting source data table has 3 lines and 4 columns (n1 = K3, n2 = K4). For the sorting result data table, understand (s) as (d).

		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 3	Line No. 1	(s)	(s)+3	(s)+6	(s)+9
	Line No. 2	(s)+1	(s)+4	(s)+7	(s)+10
	Line No. 3	(s)+2	(s)+5	(s)+8	(s)+11

- When the command input turns on, data sorting is started. Data sorting is completed after (n1) scans, and the instruction execution complete flag SM8029 is set to on.
- The following table shows an operation example based on the sorting source data below. It is recommended to put a serial number such as a control number in the first column so that the original line number can be estimated based on the contents.

		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 5	Line No. 1	(s)	(s)+5	(s)+10	(s)+15
		1	150	45	20
	Line No. 2	(s)+1	(s)+6	(s)+11	(s)+16
		2	180	50	40
	Line No. 3	(s)+2	(s)+7	(s)+12	(s)+17
		3	160	70	30
	Line No. 4	(s)+3	(s)+8	(s)+13	(s)+18
		4	100	20	8
	Line No. 5	(s)+4	(s)+9	(s)+14	(s)+19
		5	150	50	45

- Sorting result when the instructions are executed with (n3) = K2 (column No. 2)

		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 5	Line No. 1	(d)	(d)+5	(d)+11	(d)+15
		4	100	20	8
	Line No. 2	(d)+2	(d)+6	(d)+10	(d)+16
		1	150	45	20
	Line No. 3	(d)+1	(d)+7	(d)+12	(d)+17
		5	150	50	45
	Line No. 4	(d)+3	(d)+8	(d)+13	(d)+18
		3	160	70	30
	Line No. 5	(d)+4	(d)+9	(d)+14	(d)+19
		2	180	50	40

• Sorting result when the instructions are executed with (n3) = K3 (column No. 3)

		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 5	Line No. 1	(d)	(d)+5	(d)+10	(d)+15
		4	100	20	8
	Line No. 2	(d)+1	(d)+6	(d)+11	(d)+16
		1	150	45	20
	Line No. 3	(d)+2	(d)+7	(d)+12	(d)+17
		2	180	50	40
	Line No. 4	(d)+3	(d)+8	(d)+13	(d)+18
		5	150	50	45
	Line No. 5	(d)+4	(d)+9	(d)+14	(d)+19
		3	160	70	30

Precautions

- Do not change the contents of operands and data during operation.
- To execute these instructions again, set the command input to off once, then on again.
- These instructions can only be used once in any program.
- When specifying the same device in (s) and (d), the source data is overwritten by the data acquired by sorting. Take special care so that the contents of (s) are not changed until execution is completed.

Operation error

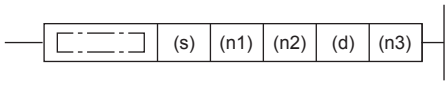
Error code (SD0/SD8067)	Description
1811H	These instructions are used more than once.
2820H	The device range specified by (s) exceeds the corresponding device range.
	The device range specified by (d) exceeds the corresponding device range.
3405H	The value specified by (n1) is outside the following range. 1 to 32
	The value specified by (n2) is outside the following range. 1 to 6
	The value specified by (n3) is outside the following range. 1 to (n2)

Sorting 16-bit data 2

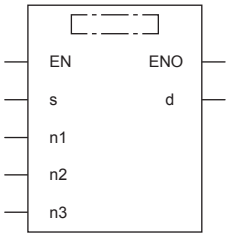
SORTTBL2(_U)

FX5S FX5UJ FX5U FX5UC

These instructions sort data lines in the data table (sorting source) of 16-bit binary data having (n1×n2) points specified by (s) in the ascending order or descending order based on the group data in the column number (n3), and store the result in the data table (sorting result) of 16-bit binary data having ((n1)×(n2)) points specified by (d).

Ladder diagram	Structured text
	ENO:=SORTTBL2(EN,s,n1,n2,n3,d); ENO:=SORTTBL2_U(EN,s,n1,n2,n3,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)	SORTTBL2	Head device number storing the data table	—	16-bit signed binary	ANY16
	SORTTBL2_U			16-bit unsigned binary	ANY16_U
(n1)		Number of data (lines)	1 to 32	16-bit unsigned binary	ANY16_U
(n2)		Number of group data (columns)	1 to 6	16-bit unsigned binary	ANY16_U
(d)	SORTTBL2	Head device number for storing the operation result	—	16-bit signed binary	ANY16
	SORTTBL2_U			16-bit unsigned binary	ANY16_U
(n3)		Column number of group data (column) used as the basis of sorting	—	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	—	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n3)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions sort data lines in the data table (sorting source) of 16-bit binary data having (n1×n2) points specified by (s) in the ascending order or descending order based on the group data in the column number (n3), and store the result in the data table (sorting result) of 16-bit binary data having ((n1)×(n2)) points specified by (d).
- The data table configuration is explained in an example in which the sorting source data table has 3 lines and 4 columns (n1 = K3, n2 = K4). For the sorting result data table, understand (s) as (d).

		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 3	Line No. 1	(s)	(s)+1	(s)+2	(s)+3
	Line No. 2	(s)+4	(s)+5	(s)+6	(s)+7
	Line No. 3	(s)+8	(s)+9	(s)+10	(s)+11

- Set the sorting order by setting SM703 to on or off.

	Sorting order
SM703 = ON	Descending order
SM703 = OFF	Ascending order

- When the command input turns on, data sorting is started. Data sorting is completed after (n1) scans, and the instruction execution complete flag SM8029 is set to on.
- The following table shows an operation example based on the sorting source data below. It is recommended to put a serial number such as a control number in the first column so that the original line number can be estimated based on the contents.

		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 5	Line No. 1	(s)	(s)+1	(s)+2	(s)+3
		1	150	45	20
	Line No. 2	(s)+4	(s)+5	(s)+6	(s)+7
		2	180	50	40
	Line No. 3	(s)+8	(s)+9	(s)+10	(s)+11
		3	160	70	30
	Line No. 4	(s)+12	(s)+13	(s)+14	(s)+15
		4	100	20	8
	Line No. 5	(s)+16	(s)+17	(s)+18	(s)+19
		5	150	50	45

- Sorting result when the instructions are executed with (n3) = K2 (column No. 2) (in the case of ascending order SM703=OFF)

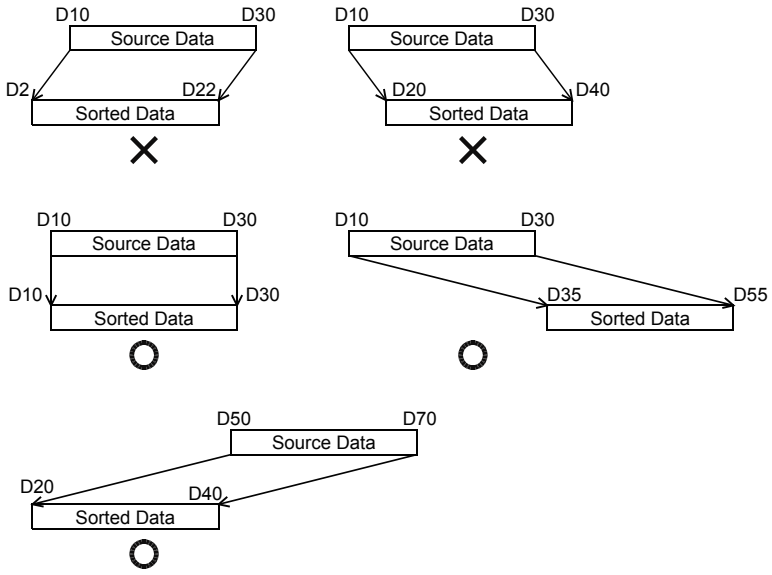
		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 5	Line No. 1	(d)	(d)+1	(d)+2	(d)+3
		4	100	20	8
	Line No. 2	(d)+4	(d)+5	(d)+6	(d)+7
		1	150	45	20
	Line No. 3	(d)+8	(d)+9	(d)+10	(d)+11
		5	150	50	45
	Line No. 4	(d)+12	(d)+13	(d)+14	(d)+15
		3	160	70	30
	Line No. 5	(d)+16	(d)+17	(d)+18	(d)+19
		2	180	50	40

• Sorting result when the instructions are executed with (n3) = K3 (column No. 3) (in the case of descending order SM703=ON)

		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 5	Line No. 1	(d)	(d)+1	(d)+2	(d)+3
		3	160	70	30
	Line No. 2	(d)+4	(d)+5	(d)+6	(d)+7
		2	180	50	40
	Line No. 3	(d)+8	(d)+9	(d)+10	(d)+11
		5	150	50	45
	Line No. 4	(d)+12	(d)+13	(d)+14	(d)+15
		1	150	45	20
	Line No. 5	(d)+16	(d)+17	(d)+18	(d)+19
		4	100	20	8

Precautions

- Do not change the contents of operands and data during operation.
- To execute these instructions again, set the command input to off once, then on again.
- These instructions can be used up to twice in any program.
- When specifying the same device in (s) and (d), the source data is overwritten by the data acquired by sorting. Take special care so that the contents of (s) are not changed until execution is completed.
- Ensure that the sorted data does not overlap with the source data.



Operation error

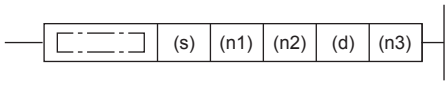
Error code (SD0/SD8067)	Description
2820H	The device range specified by (s) exceeds the corresponding device range.
	The device range specified by (d) exceeds the corresponding device range.
3405H	The value specified by (n1) is outside the following range. 1 to 32
	The value specified by (n2) is outside the following range. 1 to 6
	The value specified by (n3) is outside the following range. 1 to (n2)

Sorting 32-bit data 2

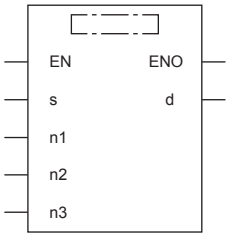
DSORTTBL2(_U)

FX5S FX5UJ FX5U FX5UC

These instructions sort data lines in the data table (sorting source) of 32-bit binary data having (n1×n2) points specified by (s) in the ascending order or descending order based on the group data in the column number (n3), and store the result in the data table (sorting result) of 32-bit binary data having ((n1)×(n2)) points specified by (d).

Ladder diagram	Structured text
	ENO:=DSORTTBL2(EN,s,n1,n2,n3,d); ENO:=DSORTTBL2_U(EN,s,n1,n2,n3,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s)	DSORTTBL2	Head device number storing the data table	—	32-bit signed binary	ANY32
	DSORTTBL2_U			32-bit unsigned binary	ANY32_U
(n1)		Number of data (lines)	1 to 32	16-bit unsigned binary	ANY16_U
(n2)		Number of group data (columns)	1 to 6	16-bit unsigned binary	ANY16_U
(d)	DSORTTBL2	Head device number for storing the operation result	—	32-bit signed binary	ANY32
	DSORTTBL2_U			32-bit unsigned binary	ANY32_U
(n3)		Column number of group data (column) used as the basis of sorting	—	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	—	○	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○	—	—	○	—	○	—	—	—	—
(n3)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions sort data lines in the data table (sorting source) of 32-bit binary data having (n1×n2) points specified by (s) in the ascending order or descending order based on the group data in the column number (n3), and store the result in the data table (sorting result) of 32-bit binary data having ((n1)×(n2)) points specified by (d).
- The data table configuration is explained in an example in which the sorting source data table has 3 lines and 4 columns (n1 = K3, n2 = K4). For the sorting result data table, understand (s) as (d).

		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 3	Line No. 1	(s)+1, (s)	(s)+3, (s)+2	(s)+5, (s)+4	(s)+7, (s)+6
	Line No. 2	(s)+9, (s)+8	(s)+11, (s)+10	(s)+13, (s)+12	(s)+15, (s)+14
	Line No. 3	(s)+17, (s)+16	(s)+19, (s)+18	(s)+21, (s)+20	(s)+23, (s)+22

- Set the sorting order by setting SM703 to on or off.

	Sorting order
SM703 = ON	Descending order
SM703 = OFF	Ascending order

- When the command input turns on, data sorting is started. Data sorting is completed after (n1) scans, and the instruction execution complete flag SM8029 is set to on.
- The following table shows an operation example based on the sorting source data below. It is recommended to put a serial number such as a control number in the first column so that the original line number can be estimated based on the contents.

		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 5	Line No. 1	(s)+1, (s)	(s)+3, (s)+2	(s)+5, (s)+4	(s)+7, (s)+6
		1	150	45	20
	Line No. 2	(s)+9, (s)+8	(s)+11, (s)+10	(s)+13, (s)+12	(s)+15, (s)+14
		2	180	50	40
	Line No. 3	(s)+17, (s)+16	(s)+19, (s)+18	(s)+21, (s)+20	(s)+23, (s)+22
		3	160	70	30
	Line No. 4	(s)+25, (s)+24	(s)+27, (s)+26	(s)+29, (s)+28	(s)+31, (s)+30
		4	100	20	8
	Line No. 5	(s)+33, (s)+32	(s)+35, (s)+34	(s)+37, (s)+36	(s)+39, (s)+38
		5	150	50	45

- Sorting result when the instructions are executed with (n3) = K2 (column No. 2) (in the case of ascending order SM703=OFF)

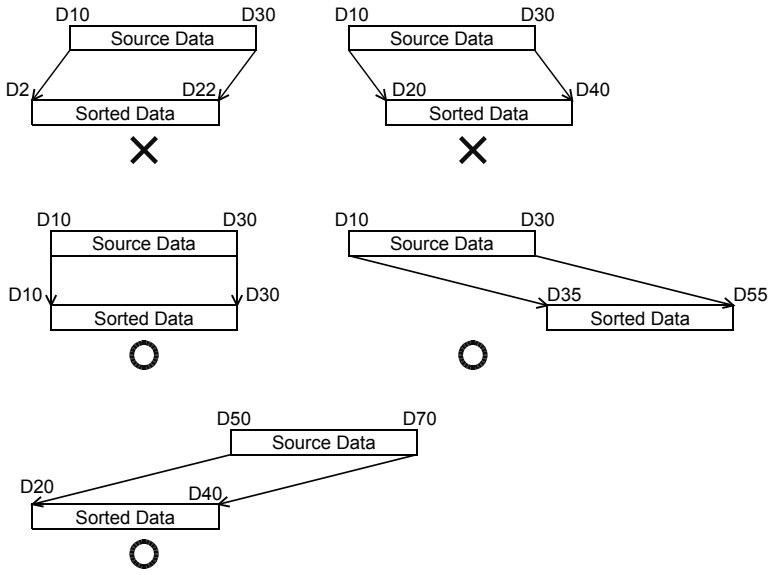
		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 5	Line No. 1	(d)+1, (d)	(d)+3, (d)+2	(d)+5, (d)+4	(d)+7, (d)+6
		4	100	20	8
	Line No. 2	(d)+9, (d)+8	(d)+11, (d)+10	(d)+13, (d)+12	(d)+15, (d)+14
		1	150	45	20
	Line No. 3	(d)+17, (d)+16	(d)+19, (d)+18	(d)+21, (d)+20	(d)+23, (d)+22
		5	150	50	45
	Line No. 4	(d)+25, (d)+24	(d)+27, (d)+26	(d)+29, (d)+28	(d)+31, (d)+30
		3	160	70	30
	Line No. 5	(d)+33, (d)+32	(d)+35, (d)+34	(d)+37, (d)+36	(d)+39, (d)+38
		2	180	50	40

- Sorting result when the instructions are executed with (n3) = K3 (column No. 3) (in the case of descending order SM703=ON)

		Number of groups (n2 = K4)			
		Column No. 1	Column No. 2	Column No. 3	Column No. 4
		Control number	Height	Weight	Age
Number of data (n1) = 5	Line No. 1	(d)+1, (d)	(d)+3, (d)+2	(d)+5, (d)+4	(d)+7, (d)+6
		3	160	70	30
	Line No. 2	(d)+9, (d)+8	(d)+11, (d)+10	(d)+13, (d)+12	(d)+15, (d)+14
		2	180	50	40
	Line No. 3	(d)+17, (d)+16	(d)+19, (d)+18	(d)+21, (d)+20	(d)+23, (d)+22
		5	150	50	45
	Line No. 4	(d)+25, (d)+24	(d)+27, (d)+26	(d)+29, (d)+28	(d)+31, (d)+30
		1	150	45	20
	Line No. 5	(d)+33, (d)+32	(d)+35, (d)+34	(d)+37, (d)+36	(d)+39, (d)+38
		4	100	20	8

Precautions

- Do not change the contents of operands and data during operation.
- To execute these instructions again, set the command input to off once, then on again.
- These instructions can be used up to or twice in any program.
- When specifying the same device in (s) and (d), the source data is overwritten by the data acquired by sorting. Take special care so that the contents of (s) are not changed until execution is completed.
- Ensure that the sorted data does not overlap with the source data.



Operation error

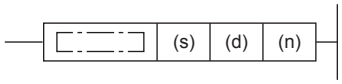
Error code (SD0/SD8067)	Description
2820H	The device range specified by (s) exceeds the corresponding device range.
	The device range specified by (d) exceeds the corresponding device range.
3405H	The value specified by (n1) is outside the following range. 1 to 32
	The value specified by (n2) is outside the following range. 1 to 6
	The value specified by (n3) is outside the following range. 1 to (n2)

Adding 16-bit data

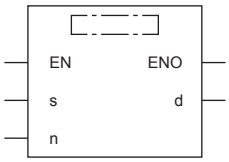
WSUM(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add the (n) point(s) of 16-bit binary data in the device starting from the one specified by (s), and store the result in the device specified by (d).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

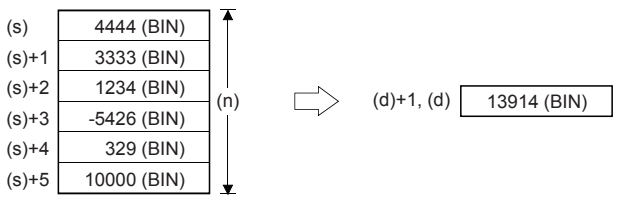
Operand		Description	Range	Data type	Data type (label)
(s)	WSUM(P)	Head device number where the addition target data are stored	—	16-bit signed binary	ANY16_S
	WSUM(P)_U			16-bit unsigned binary	ANY16_U
(d)	WSUM(P)	Head device number storing sum	—	32-bit signed binary	ANY32_S
	WSUM(P)_U			32-bit unsigned binary	ANY32_U
(n)		Number of data	—	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

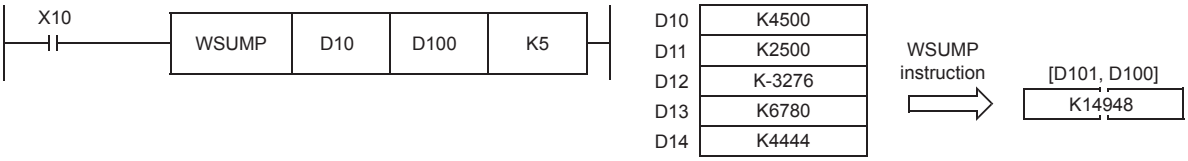
Processing details

- These instructions add the (n) point(s) of 16-bit binary data in the device starting from the one specified by (s), and store the result in the device specified by (d).



Program example

In the program shown below, the sum of 16-bit data stored in D10 to D14 is stored in [D101, D100] when X10 is set to ON.



Operation error

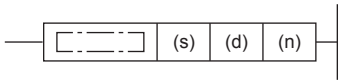
Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
	The (n) point(s) of data in the device starting from (s) exceed the corresponding device range.
3405H	The data stored in a device specified by (n) is 0.

Adding 32-bit data

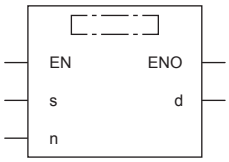
DWSUM(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions add the (n) point(s) of 32-bit binary data in the device starting from the one specified by (s), and store the result in the device specified by (d).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

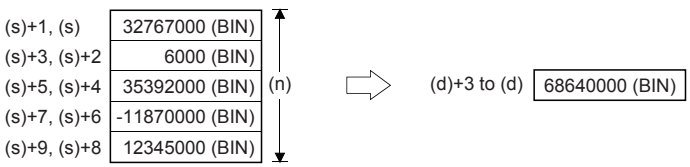
Operand		Description	Range	Data type	Data type (label)
(s)	DWSUM(P)	Head device number where the addition target data are stored	—	32-bit signed binary	ANY32_S
	DWSUM(P)_U			32-bit unsigned binary	ANY32_U
(d)	DWSUM(P)	Head device number storing sum	—	64-bit signed binary	ANY32_ARRAY
	DWSUM(P)_U			64-bit unsigned binary	
(n)		Number of data	—	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	○	—	○	—	—	—	—
(d)	—	○	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions add the (n) point(s) of 32-bit binary data in the device starting from the one specified by (s), and store the result in the device specified by (d).



Precautions

In the 32-bit operation, the acquired sum is 64-bit data. The FX5 CPU module cannot handle 64-bit data. When the sum is within the numeric range of 32-bit data (K-2147483648 to K2147483647), however, the FX5 CPU module can handle the low-order 32 bits of 32-bit data as the sum while ignoring the high-order 32 bits.

Operation error

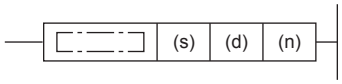
Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.
	The (n) point(s) of data in the device starting from (s) exceed the corresponding device range.
3405H	The data stored in a device specified by (n) is 0.

Calculating the mean value of 16-bit data

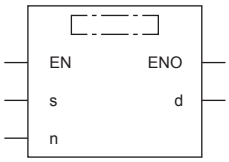
MEAN(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions calculate the mean value of the (n) point(s) of 16-bit data units starting from the one specified by (s), and store the operation result in (d).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

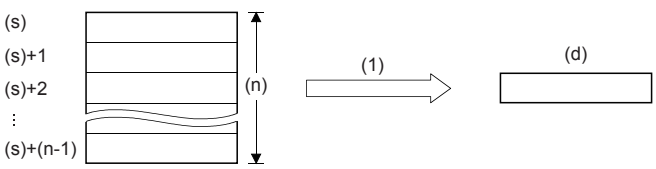
Operand		Description	Range	Data type	Data type (label)
(s)	MEAN(P)	Head device number where the mean value target data are stored	—	16-bit signed binary	ANY16_S
	MEAN(P)_U			16-bit unsigned binary	ANY16_U
(d)	MEAN(P)	Head device number storing mean value	—	16-bit signed binary	ANY16_S
	MEAN(P)_U			16-bit unsigned binary	ANY16_U
(n)		Number of data or the device number storing the number of data	1 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	—	—	○	—	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions calculate the mean value of the (n) point(s) of 16-bit data starting from the one specified by (s), and store the operation result in a device specified by (d).



(1): Mean

- The sum is obtained as algebraic sum, and divided by (n).
- The remainder is ignored.

Precautions

When a device number is exceeded, (n) is handled as a smaller value in the possible range.

Program example

In the program example shown below, the data of D0, D1 and D2 are summed, divided by "3", and then stored to D10.

$$\frac{(D0) + (D1) + (D2)}{3} \longrightarrow (D10)$$

Operation error

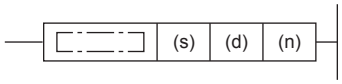
Error code (SD0/SD8067)	Description
3405H	The value stored in a device specified by (n) is 0.

Calculating the mean value of 32-bit data

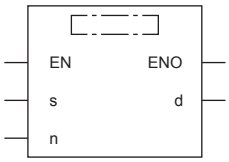
DMEAN(P)(_U)

FX5S FX5UJ FX5U FX5UC

These instructions calculate the mean value of the (n) point(s) of 32-bit data units starting from the one specified by (s), and store the operation result in (d).

Ladder diagram	Structured text	
	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



Setting data

■Descriptions, ranges, and data types

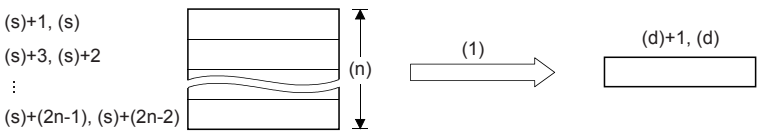
Operand		Description	Range	Data type	Data type (label)
(s)	DMEAN(P)	Head device number where the mean value target data are stored	—	32-bit signed binary	ANY32_S
	DMEAN(P)_U			32-bit unsigned binary	ANY32_U
(d)	DMEAN(P)	Head device number storing mean value	—	32-bit signed binary	ANY32_S
	DMEAN(P)_U			32-bit unsigned binary	ANY32_U
(n)		Number of data or the device number storing the number of data	1 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	○	—	○	—	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions calculate the mean value of the (n) point(s) of 32-bit data starting from the one specified by (s), and store the operation result in a device specified by (d).



(1): Mean

- The sum is obtained as algebraic sum, and divided by (n).
- The remainder is ignored.

Precautions

When a device number is exceeded, (n) is handled as a smaller value in the possible range.

Operation error

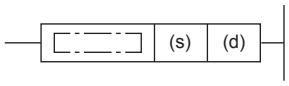
Error code (SD0/SD8067)	Description
3405H	The value stored in a device specified by (n) is 0.

Calculating the square root of 16-bit data

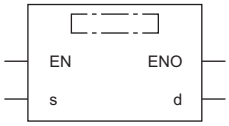
SQRT(P)

FX5S FX5UJ FX5U FX5UC

These instructions calculate the square root of binary 16-bit data specified by (s1), and store the operation result in (d).

Ladder diagram	Structured text*1
	ENO:=SQRTP(EN,s,d);

FBD/LD*1



*1 The SQRT instruction is not supported by the ST language and the FBD/LD language. Use SQRT of the standard function.
☞ Page 1285 SQRT(_E)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Device where the data whose square root is operated is calculated	—	16-bit unsigned binary	ANY16
(d)	Device for storing the calculated square root	—	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	○	—	—	○	○	—	—	—
(d)	—	○	○	○	—	—	○	—	—	—	—

Processing details

- These instructions calculate the square root of binary 16-bit data specified by (s1), and store the operation result in (d).

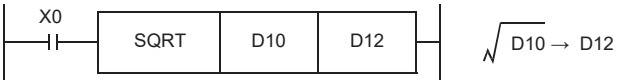
$\sqrt{(s)} \rightarrow (d)$

Precautions

- The obtained square root is an integer because the decimal point is ignored. When the calculated decimal value is ignored, SM8021 (borrow flag) turns on.
- When the operation result is true 0, SM8020 (zero flag) turns on.

Program example

In the program example shown below, the square root of D10 is stored to D12.



- When K100 is specified in D10

D10 = K100 → D12 = K10

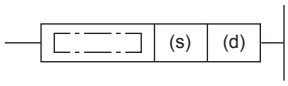
$\sqrt{100}$ → 10

Calculating the square root of 32-bit data

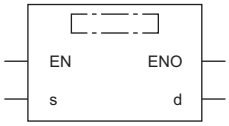
DSQRT(P)

FX5S FX5UJ FX5U FX5UC

These instructions calculate the square root of binary 32-bit data specified by (s1), and store the operation result in (d).

Ladder diagram	Structured text
	ENO:=DSQRT(EN,s,d); ENO:=DSQRTP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Device where the data whose square root is operated is calculated	—	32-bit unsigned binary	ANY32
(d)	Device for storing the calculated square root	—	32-bit unsigned binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	○	○	○	○	○	—	—	—
(d)	—	○	○	○	○	○	○	—	—	—	—

Processing details

- These instructions calculate the square root of binary 32-bit data specified by (s1), and store the operation result in (d).

$\sqrt{(s)+1, (s)} \rightarrow (d)+1, (d)$

Precautions

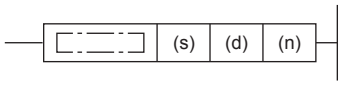
- The obtained square root is an integer because the decimal point is ignored. When the calculated decimal value is ignored, SM8021 (borrow flag) turns on.
- When the operation result is true 0, SM8020 (zero flag) turns on.

CRC calculation

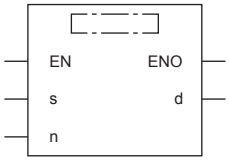
CRC(P)

FX5S FX5UJ FX5U FX5UC

These instructions calculate the CRC (cyclic redundancy check) value which is an error check method used in communication. In addition to CRC value, parity check and sum check are available. For obtaining the horizontal parity value and sum check value, the CCD(P) instruction is available. For the generation of CRC value (CRC-16), these instructions use "$X^{16} + X^{15} + X^2 + 1$" in a polynomial.

Ladder diagram	Structured text
	ENO:=CRC(EN,s,n,d); ENO:=CRCP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number storing data for which the CRC value is generated	—	16-bit signed binary	ANY16
(d)	Device number storing the generated CRC value	—	16-bit signed binary	ANY16
(n)	Number of 8-bit (1-byte) data for which the CRC value is generated or the device number storing the number of data	1 to 32767	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

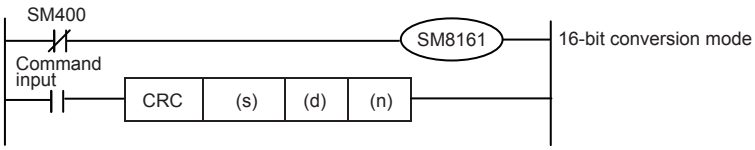
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	—	—	○	—	—	—	—
(d)	○	○	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions generate CRC value for (n) 8-bit data (unit: byte) starting from a device specified in (s), and store to (d). The 16-bit conversion mode and 8-bit conversion mode are available for these instructions. For the operation in each mode, refer to the proceeding pages.
  - 16-bit conversion mode (while SM8161 is OFF)

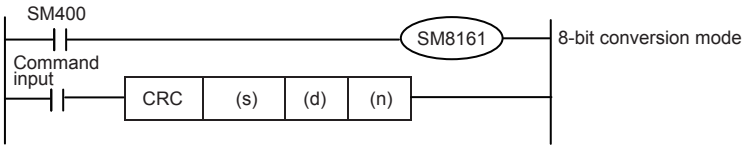
    In this mode, the operation is executed for high-order 8 bits (1 byte) and low-order 8 bits (1 byte) of a device specified in (s). The operation result is stored to one 16-bit device specified in (d).
    SM8161 is shared with the ASCI(P), HEXA(P), and CCD(P) instructions. SM8161 must always be off in the 16-bit mode.
    In the following program, conversion is executed as follows:



			Example) (s) = D100, (d) = D0, (n) = 6		
			Device	Contents of target data	
				8 bits	16 bits
Device storing data for which the CRC value is generated	(s)	Low-order byte	Low-order bits of D100	01H	0301H
		High-order byte	High-order bits of D100	03H	
	(s)+1	Low-order byte	Low-order bits of D101	03H	0203H
		High-order byte	High-order bits of D101	02H	
	(s)+2	Low-order byte	Low-order bits of D102	00H	1400H
		High-order byte	High-order bits of D102	14H	
	⋮	⋮	—		
	(s)+(n)/2-1	Low-order byte	—		
		High-order byte			
Device storing the generated CRC value	(d)	Low-order byte	Low-order bits of D0	E4H	41E4H
		High-order byte	High-order bits of D0	41H	

  - 8-bit conversion mode (while SM8161 is ON)

    In this mode, the operation is executed only for low-order 8 bits (low-order 1 byte) of a device specified by (s). With regard to the operation result, low-order 8 bits (1 byte) are stored to a device specified by (d), and high-order 8 bits (1 byte) are stored to a device specified by (d)+1.
    SM8161 is shared with the ASCI(P), HEXA(P), and CCD(P) instructions. SM8161 must always be on in the 8-bit mode.
    In the following program, conversion is executed as follows:



			Example) (s) = D100, (d) = D0, (n) = 6	
			Device	Contents of target data
Device storing data for which the CRC value is generated	(s)	Low-order byte	Low-order bits of D100	01H
	(s)+1	Low-order byte	Low-order bits of D101	03H
	(s)+2	Low-order byte	Low-order bits of D102	03H
	(s)+3	Low-order byte	Low-order bits of D103	02H
	(s)+4	Low-order byte	Low-order bits of D104	00H
	(s)+5	Low-order byte	Low-order bits of D105	14H
	⋮		—	
	(s)+(n)-1	Low-order byte	—	
Device storing the generated CRC value	(d)	Low-order byte	Low-order bits of D0	E4H
	(d)+1	High-order byte	High-order bits of D0	41H

Precautions

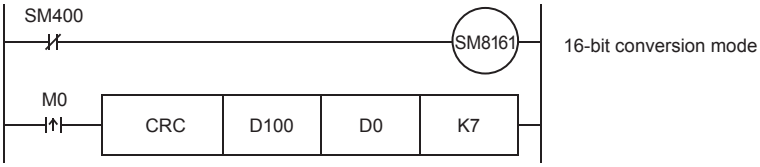
- In these instructions, "$X^{16}+X^{15}+X^2+1$" is used in a polynomial for generating the CRC value (CRC-16). There are many other standard polynomials for generating the CRC value. Note that the CRC value completely differs if an adopted polynomial is different. Major polynomials for generating the CRC value are shown below.

Name	Polynomial
CRC-12	$X^{12} + X^{11} + X^3 + X^2 + X + 1$
CRC-16	$X^{16} + X^{15} + X^2 + 1$
CRC-32	$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
CRC-CCITT	$X^{16} + X^{12} + X^5 + 1$

Program example

In the program example shown below, the CRC value of the data “0123456” stored in D100 to D106 is generated and stored to D0 when M0 turns ON.

- In the case of 16-bit conversion mode



	Contents of data		
	Device	Target data	
Device storing data for which CRC value is generated	D100: 3130H	Low-order byte	30H
		High-order byte	31H
	D101: 3332H	Low-order byte	32H
		High-order byte	33H
	D102: 3534H	Low-order byte	34H
		High-order byte	35H
	D103: 3736H	Low-order byte	36H
		—	—
Device storing generated CRC value	D0: 2ACFH	Low-order byte	CFH
		High-order byte	2AH

Operation error

There is no operation error.

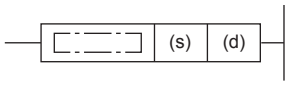
8.24 Indirect Address Read Instruction

Reading the indirect address

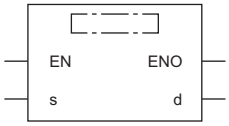
ADRSET(P)

FX5S FX5UJ FX5U FX5UC

These instructions store the indirect address of the device specified by (s) to the device specified by (d).
The addresses stored in the device specified by (d)+0 and (d)+1 are used by the program to execute the indirect address of the device.

Ladder diagram	Structured text
	ENO:=ADRSET(EN,s,d); ENO:=ADRSETP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Device number for reading the indirect address	—	Device name	ANY_ELEMENTARY*1
(d)	Device number for storing the indirect address of the device specified by (s)	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

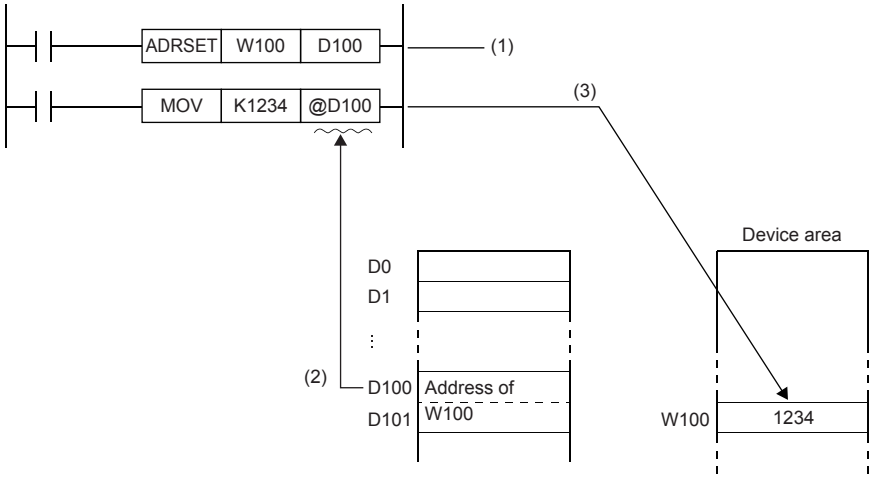
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions store the indirect address of the device specified by (s) to the device specified by (d). The addresses stored in the device specified by (d)+0 and (d)+1 are used by the program to execute the indirect address of the device.

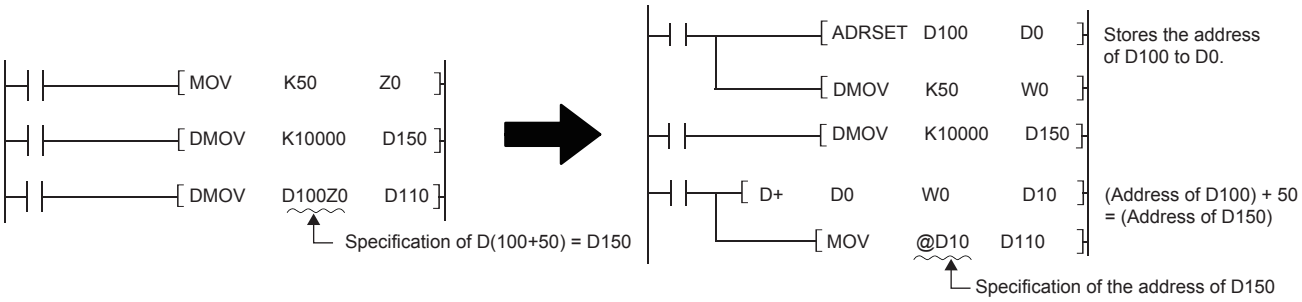


(1) Stores the address of W100 to D100 and D101.
(2) Specifies the content of D100 and D101 (address of W100).
(3) Writes "1234" to W100.

- The nibble of a bit device, and the bit of a word device cannot be specified in (s).

Precautions

- In the indirect specification, the device address used in sequence program is specified with a word device of 2 words (2-word devices). Use the indirect specification as an index when index register is insufficient.



[When the index register is used] [When the indirect specification is used]

- In the indirect specification, the device which specify the address of the specified device is specified by "@+(word device number)". For example, when "@D100" is specified, and the content of D101 and D100 becomes the device address.

Operation error

There is no operation error.

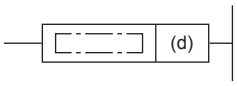
8.25 Clock Instruction

Reading clock data

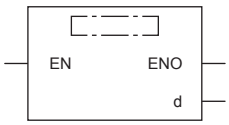
TRD(P)

FX5S FX5UJ FX5U FX5UC

These instructions read the clock data from the built-in real time clock in the CPU module.

Ladder diagram	Structured text
	ENO:=TRD(EN,d); ENO:=TRDP(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Head device number where the read clock data is stored	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 7)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○	○	—	—	—	○	—	—	—	—

Processing details

- These instructions read the clock data (SD210 to SD216) from the built-in real time clock in the CPU module to the device numbers (d) to (d)+6 in the following format.

Special register			Read data	
Device	Item	Clock data	Device	Item
SD210	Year	1980 to 2079 (year, four digits)	(d)	Year
SD211	Month	1 to 12	(d)+1	Month
SD212	Day	1 to 31	(d)+2	Day
SD213	Hour data	0 to 23	(d)+3	Hour data
SD214	Minute data	0 to 59	(d)+4	Minute data
SD215	Second data	0 to 59	(d)+5	Second data
SD216	Day-of-the-week data	0 (Sunday) to 6 (Saturday)	(d)+6	Day-of-the-week data

- The table below shows the related devices. The clock data stored in these special registers is updated during the END processing.

Device	Name	Description
Binary code		
SD210	Binary clock data (year)	The year data in the clock data is stored as a four-digit binary code.
SD211	Binary clock data (month)	The month data in the clock data is stored as a binary code.
SD212	Binary clock data (day)	The day data in the clock data is stored as a binary code.
SD213	Binary clock data (hour)	The hour data in the clock data is stored as a binary code.
SD214	Binary clock data (minute)	The minute data in the clock data is stored as a binary code.
SD215	Binary clock data (second)	The second data in the clock data is stored as a binary code.
SD216	Binary clock data (day of the week)	The day-of-a-week data in the clock data (0: Sunday, 1: Monday, ..., 6: Saturday) is stored as a binary code.
Binary code (FX3 compatible area)		
SD8013	Binary clock data (second)	The second data in the clock data is stored as a binary code.
SD8014	Binary clock data (minute)	The minute data in the clock data is stored as a binary code.
SD8015	Binary clock data (hour)	The hour data in the clock data is stored as a binary code.
SD8016	Binary clock data (day)	The day data in the clock data is stored as a binary code.
SD8017	Binary clock data (month)	The month data in the clock data is stored as a binary code.
SD8018	Binary clock data (year)	The year data in the clock data is stored as a four-digit binary code.
SD8019	Binary clock data (day of the week)	The day-of-a-week data in the clock data (0: Sunday, 1: Monday, ..., 6: Saturday) is stored as a binary code.

Precautions

- These instructions occupy seven points of device starting from device number specified by (d). Make sure that these devices are not used by other machine controls.

Operation error

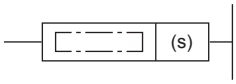
Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.

Writing clock data

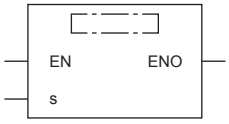
TWR(P)

FX5S FX5UJ FX5U FX5UC

This instruction writes the clock data to the built-in CPU module real time clock.

Ladder diagram	Structured text
	ENO:=TWR(EN,s); ENO:=TWRP(EN,s);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number where the clock write source data is stored	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 7)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	○	—	—	—	○	—	—	—	—

Processing details

- These instructions write the clock data stored in device numbers (s) to (s)+6 to the clock data area (SD210 to SD216 and SD8013 to SD8019) of the built-in real time clock in the CPU module.

Time setting data			Special registers	
Device	Item	Clock data	Device	Item
(s)	Year	1980 to 2079 (year, four digits)	SD210, SD8018	Year
(s)+1	Month	1 to 12	SD211, SD8017	Month
(s)+2	Day	1 to 31	SD212, SD8016	Day
(s)+3	Hour data	0 to 23	SD213, SD8015	Hour data
(s)+4	Minute data	0 to 59	SD214, SD8014	Minute data
(s)+5	Second data	0 to 59	SD215, SD8013	Second data
(s)+6	Day-of-the-week data	0 (Sunday) to 6 (Saturday)	SD216, SD8019	Day-of-the-week data

- Executing these instructions immediately changes the real time clock data. Therefore, transfer the clock data of a few minutes ahead the current time to the clock data area (s) to (s)+6 in advance. Execute the instruction when the actual time matches the clock data time.
- When using these instructions to set the clock data (i.e., performing time adjustment), control of special relay SM8015 (clock stop/adjustment) is not required.
- If incorrect values (i.e., values out of range) are set to the write source area, the clock data will not be updated. In this case, correct the clock data in the write source area and execute the instruction.

- Day of the week (SD216 and SD8019) is automatically corrected.
- The table below shows the related devices.

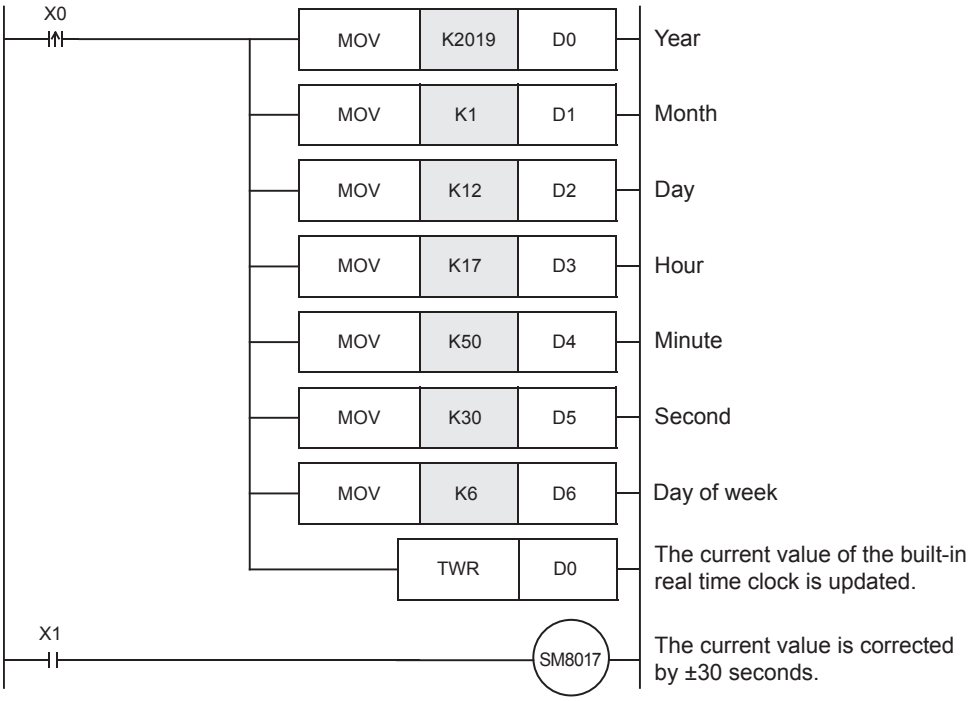
Device	Name	Description
SM8019	Real time clock error	This special data register turns on when the clock data value in the special register is exceeding the setting range.
Binary code		
SD210	Binary clock data (year)	The year data in the clock data is stored as a four-digit binary code.
SD211	Binary clock data (month)	The month data in the clock data is stored as a binary code.
SD212	Binary clock data (day)	The day data in the clock data is stored as a binary code.
SD213	Binary clock data (hour)	The hour data in the clock data is stored as a binary code.
SD214	Binary clock data (minute)	The minute data in the clock data is stored as a binary code.
SD215	Binary clock data (second)	The second data in the clock data is stored as a binary code.
SD216	Binary clock data (day of the week)	The day-of-a-week data in the clock data (0: Sunday, 1: Monday, ..., 6: Saturday) is stored as a binary code.
Binary code (FX3 compatible area)		
SD8013	Binary clock data (second)	The second data in the clock data is stored as a binary code.
SD8014	Binary clock data (minute)	The minute data in the clock data is stored as a binary code.
SD8015	Binary clock data (hour)	The hour data in the clock data is stored as a binary code.
SD8016	Binary clock data (day)	The day data in the clock data is stored as a binary code.
SD8017	Binary clock data (month)	The month data in the clock data is stored as a binary code.
SD8018	Binary clock data (year)	The year data in the clock data is stored as a four-digit binary code.
SD8019	Binary clock data (day of the week)	The day-of-a-week data in the clock data (0: Sunday, 1: Monday, ..., 6: Saturday) is stored as a binary code.

Precautions

- These instructions occupy seven points of device starting from device number specified by (s). Make sure that these devices are not used by other machine controls.

Program example

In the program example shown below, the CPU built-in real time clock is set (to 17:50:30 on Saturday, January 12, 2019).



- The shaded area indicates the set value of each item.
- When setting the time, it is recommended to set the time to several minutes ahead in advance, and then set X0 to ON when the accurate time is reached. The set time is then immediately written to the real time clock, and the clock data is updated.
- Every time X1 is set to ON, SM8017 is turned on and the current time can be corrected by ±30 seconds. For SM8017, refer to MELSEC iQ-F FX5 User's Manual (Application).

Operation error

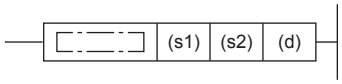
Error code (SD0/SD8067)	Description
2820H	The device range specified by (s) exceeds the corresponding device range.

Adding clock data

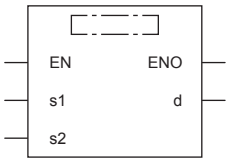
TADD(P)

FX5S FX5UJ FX5U FX5UC

These instructions add the time data stored in the device number specified by (s2) and later to the clock data stored in the device number specified by (s1) and later, and store the result to the device number specified by (d) and later.

Ladder diagram	Structured text
	ENO:=TADD(EN,s1,s2,d); ENO:=TADDP(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device number where the clock data to be added is stored.	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(s2)	Head device number where the adding time value (or clock data value) is stored.	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(d)	Head device number where the resultant clock data (or time value) is stored.	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	—	—	○	—	—	—	—
(s2)	—	○	○	—	—	—	○	—	—	—	—
(d)	—	○	○	—	—	—	○	—	—	—	—

Processing details

- These instructions add the time data stored in the device numbers starting from (s2) to the clock data stored in the device numbers starting from (s1), and store the result to the device numbers starting from (d).

(s1)	hour	(0 to 23)		(s2)	hour	(0 to 23)		(d)	hour	(0 to 23)
(s1)+1	minute	(0 to 59)	+	(s2)+1	minute	(0 to 59)	⇨	(d)+1	minute	(0 to 59)
(s1)+2	second	(0 to 59)		(s2)+2	second	(0 to 59)		(d)+2	second	(0 to 59)

Ex.

When adding 7:48:10 to 6:32:40

(s1)	6		(s2)	7		(d)	14
(s1)+1	32	+	(s2)+1	48	⇨	(d)+1	20
(s1)+2	40		(s2)+2	10		(d)+2	50

- If the sum of two values exceeds 24:00:00, the carry flag turns on, and the result will be the sum minus 24:00:00. For example, if a time value of 20:20:20 is added to another time value of 14:30:30, the sum is 34:40:50. However, the actual addition result will be 10:40:50.

(s1)	14		(s2)	20		(d)	10
(s1)+1	20	+	(s2)+1	20	⇨	(d)+1	40
(s1)+2	30		(s2)+2	20		(d)+2	50

- If the result is 0 (0:00:00), the zero flag turns on.
- If 1 second is added to 23:59:59, the result will be 0:00:00. This turns on both the carry flag and the zero flag.
- The table below shows the related devices.

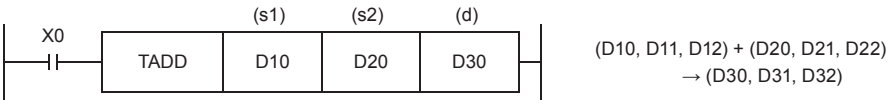
Device	Name	Description
SM700	Carry	If the result exceeds the maximum value of the time data, 23:59:59, this special relay turns on.
SM8020	Zero	If the result is 0:00:00, this special relay turns on.
SM8022	Carry	If the result exceeds the maximum value of the time data, 23:59:59, this special relay turns on.

Precautions

- These instructions occupy three points for each of three devices starting from device number specified by (s1), (s2), and (d) respectively. Make sure that these devices are not used by other machine controls.
- When using the time value (hour, minute, second) of the built-in real time clock in the CPU module for the operation, use the TRD(P) operation to read the special register values first. Then specify the word devices where the read values are stored to each operand.

Program example

In the program shown below, when X0 is set to ON, the time obtained by adding the D20 to D22 time data to the D10 to D12 clock data is stored in D30 to D32.



(D10, D11, D12) + (D20, D21, D22)
→ (D30, D31, D32)

D10	10 (hour)		D20	3 (hour)		D30	13 (hour)
D11	30 (minute)	+	D21	10 (minute)	⟶	D31	40 (minute)
D12	10 (second)		D22	5 (second)		D32	15 (second)
10:30:10			3:10:05			13:40:15	

- When the operation result exceeds 24 hours

(s1)		(s2)		(d)	
18 (hour)		10 (hour)		4 (hour)	⟵18 + 10 = 28 (≥24)
10 (minute)	+	20 (minute)	⟶	30 (minute)	
30 (second)		5 (second)		35 (second)	
18:10:30		10:20:05		4:30:35	

Operation error

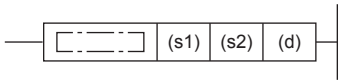
Error code (SD0/SD8067)	Description
2820H	The device range specified by (s1), (s2), and (d) exceeds the corresponding device range.
3405H	Any of values specified by (s1) and (s2) is outside the following range. 0 to 23
	Any of values specified by (s1)+1, (s2)+1, (s1)+2, and (s2)+2 is outside the following range. 0 to 59

Subtracting clock data

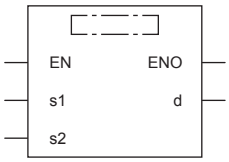
TSUB(P)

FX5S FX5UJ FX5U FX5UC

These instructions subtract the time data stored in the device numbers starting from (s2) from the clock data stored in the device numbers starting from (s1), and store the result to the device numbers starting from (d).

Ladder diagram	Structured text
	ENO:=TSUB(EN, s1, s2,d); ENO:=TSUBP(EN, s1, s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device number where the clock data that is subtracted is stored	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(s2)	Head device number where the subtracting time value (or clock data value) is stored	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(d)	Head device number where the resultant clock data (or time value) is stored	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	—	—	○	—	—	—	—
(s2)	—	○	○	—	—	—	○	—	—	—	—
(d)	—	○	○	—	—	—	○	—	—	—	—

Processing details

- These instructions subtract the time data stored in the device numbers starting from (s2) from the clock data stored in the device numbers starting from (s1), and store the result to the device numbers starting from (d).

(s1)	hour	(0 to 23)	-	(s2)	hour	(0 to 23)	⇨	(d)	hour	(0 to 23)
(s1)+1	minute	(0 to 59)		(s2)+1	minute	(0 to 59)		(d)+1	minute	(0 to 59)
(s1)+2	second	(0 to 59)		(s2)+2	second	(0 to 59)		(d)+2	second	(0 to 59)

Ex.

When subtracting 3:50:10 from 10:40:20

(s1)	10	-	(s2)	3	⇨	(d)	6
(s1)+1	40		(s2)+1	50		(d)+1	50
(s1)+2	20		(s2)+2	10		(d)+2	10

- If the remainder is a negative time value, the borrow flag turns on. The actual result will be the remainder plus 24:00:00. For example, if a time value of 10:42:12 is subtracted from another time value of 4:50:32, the remainder is -6:08:20. However, the actual subtraction result will be 18:08:20.

(s1)	4	-	(s2)	10	⇨	(d)	18
(s1)+1	50		(s2)+1	42		(d)+1	8
(s1)+2	32		(s2)+2	12		(d)+2	20

- If the result is 0 (0:00:00), the zero flag turns on.
- The table below shows the related devices.

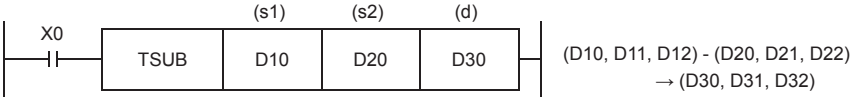
Device	Name	Description
SM8020	Zero	If the result is 0:00:00, this special relay turns on.
SM8021	Borrow	If the execution result of the TSUB(P) instruction is less than 0:00:00, this special relay turns on.

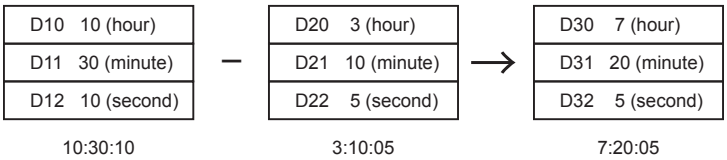
Precautions

- These instructions occupy three points for each of three devices starting from device number specified by (s1), (s2), and (d) respectively. Make sure that these devices are not used by other machine controls.
- When using the time value (hour, minute, second) of the built-in real time clock in the CPU module for the operation, use the TRD(P) operation to read the special register values first. Then specify the word devices where the read values are stored to each operand.

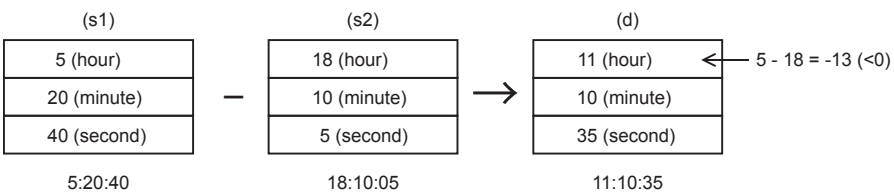
Program example

In the program shown below, when X0 is set to ON, the time obtained by subtracting the D20 to D22 time data from the D10 to D12 clock data is stored in D30 to D32.





- When the operation result is smaller than "00:00:00"



Operation error

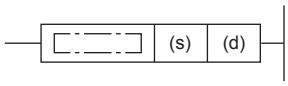
Error code (SD0/SD8067)	Description
2820H	The device range specified by (s1), (s2), and (d) exceeds the corresponding device range.
3405H	Any of values specified by (s1) and (s2) is outside the following range. 0 to 23
	Any of values specified by (s1)+1, (s2)+1, (s1)+2, and (s2)+2 is outside the following range. 0 to 59

Converting time data from hour/minute/second to seconds in 16 bits

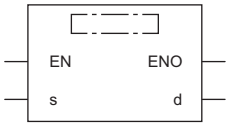
HTOS(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the time data stored in the device numbers starting from (s) to the time value in seconds, and store the converted data in the device numbers starting from (d) as 16-bit binary.

Ladder diagram	Structured text
	ENO:=HTOS(EN,s,d); ENO:=HTOSP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

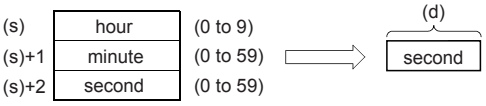
Operand	Description	Range	Data type	Data type (label)
(s)	Head device number where the clock data before conversion is stored	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(d)	Head device number where the clock data after conversion is stored	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	—	—	○	—	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—

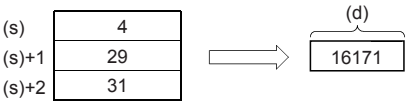
Processing details

- These instructions convert the time data stored in the device numbers starting from (s) to the time value in seconds, and store the converted data in the device numbers starting from (d).



Ex.

When specifying 4 hours 29 minutes 31 seconds in (s)



Operation error

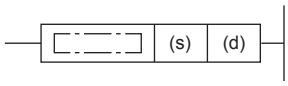
Error code (SD0/SD8067)	Description
2820H	Any of the device area ranges specified in (s) and (d) exceed the corresponding device range.
3403H	The result is outside the following range. 0 to 32767
3405H	A value specified by (s) is outside the following range. 0 to 9
	Any of values specified by (s)+1 and (s)+2 is outside the following range. 0 to 59

Converting time data from hour/minute/second to seconds in 32 bits

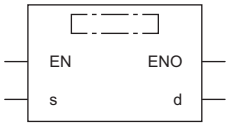
DHTOS(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the time data stored in the device numbers starting from (s) to the time value in seconds, and store the converted data in the device numbers starting from (d) as 32-bit binary.

Ladder diagram	Structured text
	ENO:=DHTOS(EN,s,d); ENO:=DHTOSP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

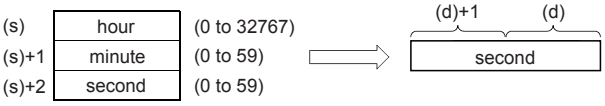
Operand	Description	Range	Data type	Data type (label)
(s)	Head device number where the clock data before conversion is stored	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(d)	Head device number where the clock data after conversion is stored	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

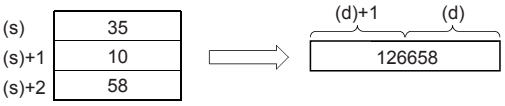
Processing details

- These instructions convert the time data stored in the device numbers starting from (s) to the time value in seconds, and store the converted data in the device numbers starting from (d).



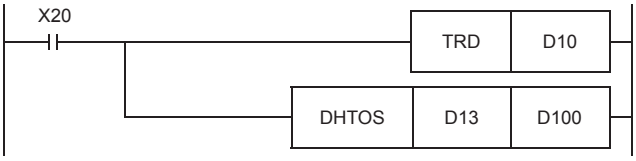
Ex.

When specifying 35 hours 10 minutes 58 seconds in (s)

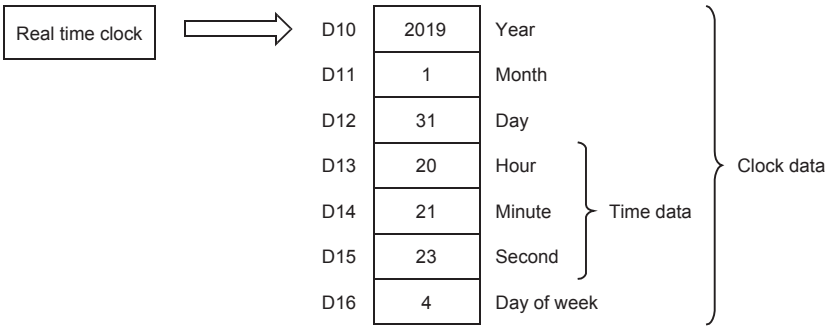


Program example

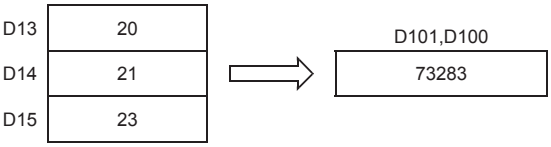
In the program shown below, the time data read from the built-in CPU module real time clock is converted into data in units of "second", and stored to D100 and D101 when X20 turns ON.



- Clock data reading operation by TRD instruction



- Conversion operation into "second" by DHTOS instruction



Operation error

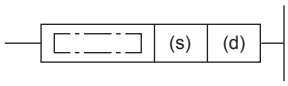
Error code (SD0/SD8067)	Description
2820H	Any of the device area ranges specified in (s) and (d) exceed the corresponding device range.
3405H	A value specified by (s) is outside the following range. 0 to 32767
	Any of values specified by (s)+1 and (s)+2 is outside the following range. 0 to 59

Converting time data from seconds to hour/minute/second in 16 bits

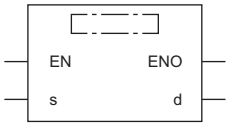
STOH(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 16-bit time value in seconds stored in the device numbers starting from (s) to the time value in the HHMMDD format, and store the converted data in the device numbers starting from (d).

Ladder diagram	Structured text
	ENO:=STOH(EN,s,d); ENO:=STOHP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

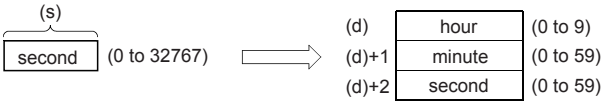
Operand	Description	Range	Data type	Data type (label)
(s)	Head device number where the clock data before conversion is stored	—	16-bit signed binary	ANY16
(d)	Head device number where the clock data after conversion is stored	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	—	—	—	○	—	—	—	—

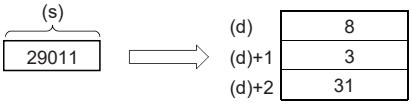
Processing details

- These instructions convert the time value in seconds stored in the device numbers starting from (s) to the time value in HHMMDD format, and store the converted data in the device numbers starting from (d).



Ex.

When specifying 29011 seconds in (s)



Operation error

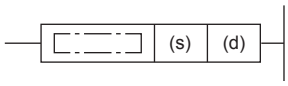
Error code (SD0/SD8067)	Description
2820H	The specified device area exceeds the corresponding device range.
3405H	The value specified by (s) is outside the range.

Converting time data from seconds to hour/minute/second in 32 bits

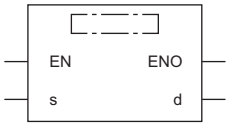
DSTOH(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert the 32-bit time value in seconds stored in the device numbers starting from (s) to the time value in the HHMMDD format, and store the converted data in the device numbers starting from (d).

Ladder diagram	Structured text
	ENO:=DSTOH(EN,s,d); ENO:=DSTOHP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

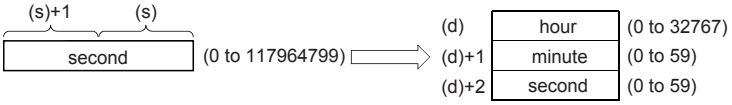
Operand	Description	Range	Data type	Data type (label)
(s)	Head device number where the clock data before conversion is stored	—	32-bit signed binary	ANY32
(d)	Head device number where the clock data after conversion is stored	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	—	—	—	○	—	—	—	—

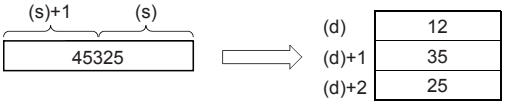
Processing details

- These instructions convert the time value in seconds stored in the device numbers starting from (s) to the time value in HHMMDD format, and store the converted data in the device numbers starting from (d).



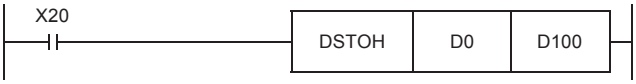
Ex.

When specifying 45325 seconds in (s)

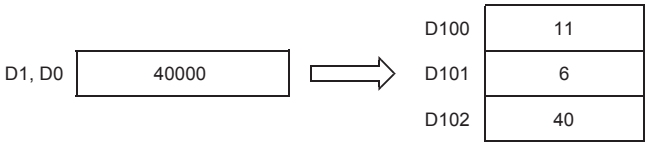


Program example

In the program shown below, the time data in units of "second" stored in D0 and D1 is converted into data in units of "hour, minute, and second", and stored to D100, D101, and D102 when X20 turns ON.



- Converting the data in second into the data in hour, minute and second using DSTOH instruction (when "40000 seconds" is specified by D1 and D0)



Operation error

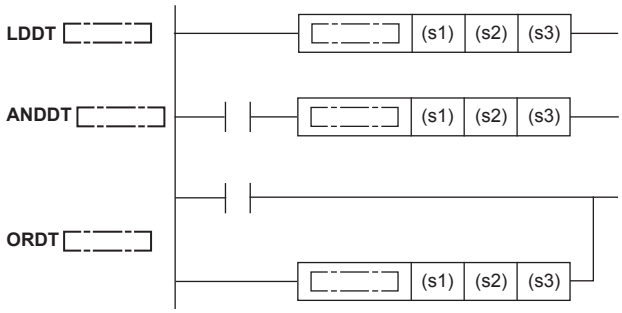
Error code (SD0/SD8067)	Description
2820H	The specified device area exceeds the corresponding device range.
3405H	The value specified by (s) is outside the range.

Comparing date data

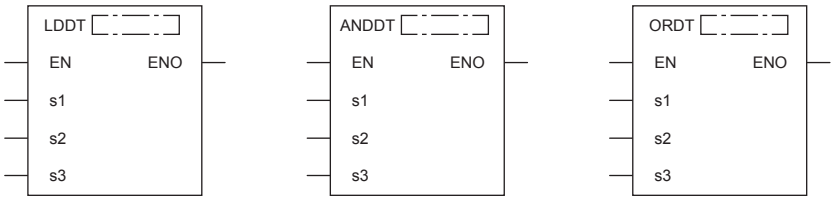
LDDT□, ANDDT□, ORDT□

FX5S FX5UJ FX5U FX5UC

These instructions compare the date data in the devices specified by (s1) and (s2). Or, these instructions compare the date data in the device specified by (s1) with the current date.
Set the comparison target by (s3).

Ladder diagram	Structured text*1
 ("=", "<>", ">", "<=", "<", ">=" enters □.)	ENO:=LDDT_□(EN,s1,s2,s3); ENO:=ANDDT_□(EN,s1,s2,s3); ENO:=ORDT_□(EN,s1,s2,s3); ("EQ", "NE", "GT", "LE", "LT", "GE" enters □.)*2

FBD/LD



("_EQ", "_NE", "_GT", "_LE", "_LT", "_GE" enters □.)*2

*1 Supported by engineering tool version "1.035M" and later.
*2 EQ is =, NE is <>, GT is >, LE is <=, LT is <, and GE is >=.

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device number where the comparison data is stored	—	16-bit signed binary	ANY_DT
(s2)	Head device number where the comparison data is stored	—	16-bit signed binary	ANY_DT
(s3)	Comparison target setting value or the number of comparison target data	0001H to 0007H, 8001H to 8007H	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

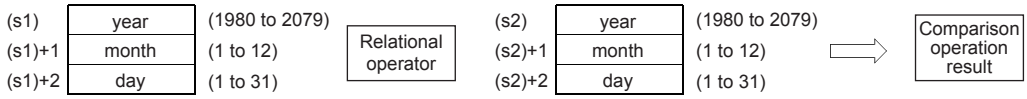
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	○	○	○	—	—	○	○	—	—	—

Processing details

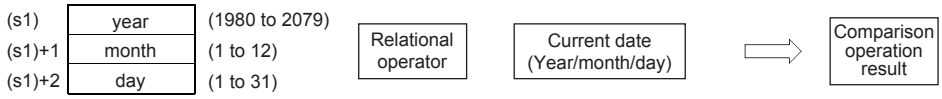
- These instructions compare the date data in the devices specified by (s1) and (s2), or compare the date data in the device specified by (s1) with the current date. Set the comparison target by (s3).
  - Comparing two specified date data

    These instructions compare the date data in the device specified by (s1) with the date data in the device specified by (s2) in accordance with the conditions set by (s3). (Devices are used as a normally open contact.)

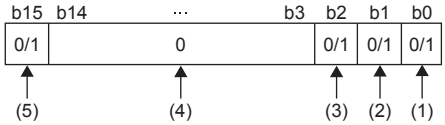


  - Comparing the specified date data with the current date

    These instructions compare the date data in the device specified by (s1) with the current date data in accordance with the conditions set by (s3). (Devices are used as a normally open contact.) The date data in the device specified by (s2) is regarded as dummy data and ignored.



- Set each data in binary.
- Set the 4 digit "year" data in the devices specified by (s1) and (s2) within the range 1980 to 2079.
- Set the "month" data in the devices specified by (s1)+1 and (s2)+1 within the range 1 to 12.
- Set the "date" data in the devices specified by (s1)+2 and (s2)+2 within the range 1 to 31.
- Set the following in (s3) as comparison target setting values. The following shows the bit configuration of (s3).



(1) Set "day" as comparison target.
(2) Set "month" as comparison target.
(3) Set "year" as comparison target.
(4) Set 0. If a value other than 0 is set, the operation result will be non-continuity.
(5) When 1 is set to the 15 bit, the data in the device specified by (s1) is compared with the current date in accordance with the conditions set in the 0 to 2 bits.

- When 0 is set to the 0 to 2 bits, the date data are not compared. When 1 is set, the entire date data (year, month, and day) are compared.
- When 0 is set to the 15 bit, the data in the device specified by (s1) and the date data in the device specified by (s2) are compared. When 1 is set, the data in the device specified by (s1) is compared with the current date. The date data in the device specified by (s2) is ignored.
- The following table lists processing details of each bit.

(s3) value when comparing two specified date data	(s3) value when comparing the specified date data with the current date	Comparison target	Contents of processing
0001H	8001H	Day	Only data in the device specified by (s1)+2 is compared.
0002H	8002H	Month	Only data in the device specified by (s1)+1 is compared.
0003H	8003H	Month, day	Data in the device areas specified by (s1)+2 and (s1)+2 are compared.
0004H	8004H	Year	Only data in the device specified by (s1) is compared.
0005H	8005H	Year, day	Data in the device areas specified by (s1) and (s1)+2 are compared.
0006H	8006H	Year, month	Data in the device areas specified by (s1) and (s1)+1 are compared.
0007H	8007H	Year, month, day	The entire date data in the device areas specified by (s1), (s1)+1, and (s1)+2 are compared.
Other than 0001H to 0007H, 8001H to 8007H		None	The entire date data in the device areas specified by (s1), (s1)+1, and (s1)+2 are not compared. (The operation result will be non-continuity.)

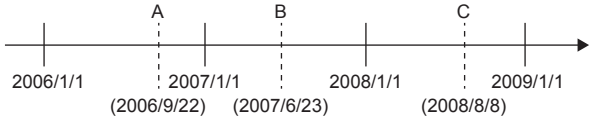
- If the comparison target data in the device are not recognized as date data, SM709 turns on after the instruction is executed and the operation result will be non-continuity. Even if the data are not recognized as date data, SM709 does not turn on if the data are within the setting range. If the device areas specified by (s1) to (s1)+2 or (s2) to (s2)+2 exceed the corresponding device range, SM709 turns on after the instruction is executed and the operation result will be non-continuity as well. Once SM709 turns on, the on state is held until the CPU module is powered off or reset. Turn off SM 709 as needed.

• The following table lists the comparison operation results of each instruction.

Instruction symbol	Condition	Result
DT=	(s1)=(s2)	Conductive state
DT<>	(s1)≠(s2)	
DT>	(s1)>(s2)	
DT<=	(s1)≤(s2)	
DT<	(s1)<(s2)	
DT>=	(s1)≥(s2)	
DT=	(s1)≠(s2)	Non-conductive state
DT<>	(s1)=(s2)	
DT>	(s1)≤(s2)	
DT<=	(s1)>(s2)	
DT<	(s1)≥(s2)	
DT>=	(s1)<(s2)	

Ex.

The date data A, B, and C are compared.



• The following table lists the comparison operation results between A, B, and C. Even when the data are compared under the same conditions, the results differ depending on the comparison target data.

○: Continuity, ×: Non-continuity

Comparison target data	Condition		
	A<B	B<C	A<C
Day	○	×	×
Month	×	○	×
Month, day	×	○	×
Year	○	○	○
Year, day	○	○	○
Year, month	○	○	○
Year, month, day	○	○	○
None	×	×	×

• Even though the specified date does not exist, the comparison operation is performed in accordance with the conditions in the following table as long as the date data are within the valid range.
  • Date A: 2006/02/30 (Even though the date does not exist, this date can be set.)
  • Date B: 2007/03/29
  • Date C: 2008/02/31 (Even though the date does not exist, this date can be set.)

○: Continuity, ×: Non-continuity

Comparison target data	Condition		
	A<B	B<C	A<C
Day	×	×	○
Month	×	×	×
Month, day	○	×	○
Year	○	○	○
Year, day	○	○	○
Year, month	○	○	○
Year, month, day	○	○	○
None	×	×	×

Operation error

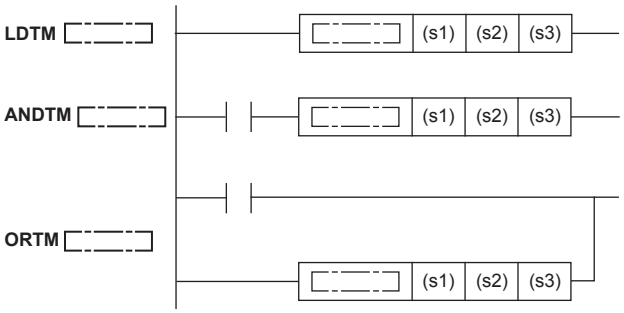
There is no operation error.

Comparing time data

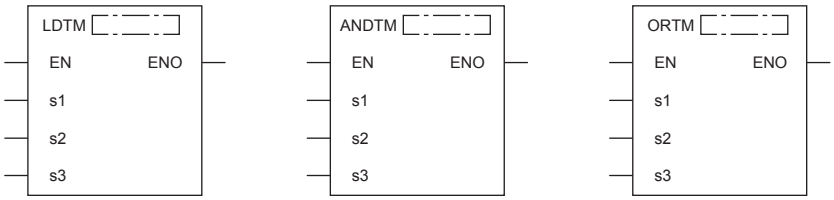
LDTM□, ANDTM□, ORTM□

FX5S FX5UJ FX5U FX5UC

These instructions compare the time data in the devices specified by (s1) and (s2). Or, these instructions compare the time data in the device specified by (s1) with the current time.
Set the comparison target by (s3).

Ladder diagram	Structured text*1
 ("=", "<>", ">", "<=", "<", ">=" enters □.)	ENO:=LDTM_□(EN,s1,s2,s3); ENO:=ANDTM_□(EN,s1,s2,s3); ENO:=ORTM_□(EN,s1,s2,s3); ("EQ", "NE", "GT", "LE", "LT", "GE" enters □.)*2

FBD/LD



("_EQ", "_NE", "_GT", "_LE", "_LT", "_GE" enters □.)*2

*1 Supported by engineering tool version "1.035M" and later.
*2 EQ is =, NE is <>, GT is >, LE is <=, LT is <, and GE is >=.

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Head device number where the comparison data is stored	—	16-bit signed binary	ANY_TM
(s2)	Head device number where the comparison data is stored	—	16-bit signed binary	ANY_TM
(s3)	Comparison target setting value or the number of comparison target data	0001H to 0007H, 8001H to 8007H	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

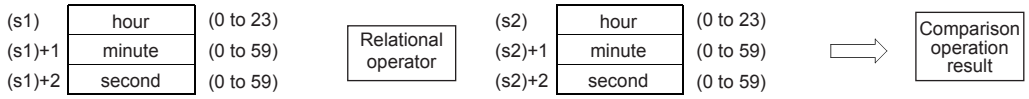
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	○	○	○	—	—	○	○	—	—	—

Processing details

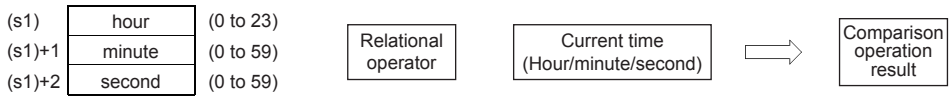
- These instructions compare the time data in the devices specified by (s1) and (s2), or compare the time data in the device specified by (s1) with the current time. Set the comparison target by (s3).
  - Comparing two specified time data

    These instructions compare the time data in the device specified by (s1) with the time data in the device specified by (s2) in accordance with the conditions set by (s3). (Devices are used as a normally open contact.)

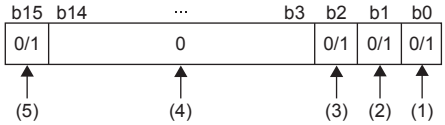


  - Comparing specified time data with current time data

    These instructions compare the time data in the device specified by (s1) with the current time data in accordance with the conditions set by (s3). (Devices are used as a normally open contact.) The time data in the device specified by (s2) is regarded as dummy data and ignored.



- Set each data in binary.
- Set the "hour" data as in the 24-hour clock in the devices specified by (s1) and (s2) within the range 0 to 23.
- Set the "minute" data in the devices specified by (s1)+1 and (s2)+1 within the range 0 to 59.
- Set the "second" data in the devices specified by (s1)+2 and (s2)+2 within the range 0 to 59.
- Set the following in (s3) as comparison target setting values. The following shows the bit configuration of (s3).



(1) Set "second" as comparison target.
(2) Set "minute" as comparison target.
(3) Set "hour" as comparison target.
(4) Set 0. If a value other than 0 is set, the operation result will be non-continuity.
(5) When 1 is set to the 15 bit, the data in the device specified by (s1) is compared with the current time in accordance with the conditions set in the 0 to 2 bits.

- When 0 is set to the 0 to 2 bits, the time data (hour, minute, and second) are not compared. When 1 is set, the entire time data (hour, minute, and second) are compared.
- When 0 is set to the 15 bit, the data in the device specified by (s1) and the time data in the device specified by (s2) are compared. When 1 is set, the data in the device specified by (s1) is compared with the current time. The time data in the device specified by (s2) is ignored.
- The following table lists processing details of each bit.

(s3) value when comparing two specified time data	(s3) value when comparing the specified time data with the current time	Comparison target	Contents of processing
0001H	8001H	Second data	Only data in the device specified by (s1)+2 is compared.
0002H	8002H	Minute data	Only data in the device specified by (s1)+1 is compared.
0003H	8003H	Minute and second data	Data in the device areas specified by (s1)+2 and (s1)+2 are compared.
0004H	8004H	Hour data	Only data in the device specified by (s1) is compared.
0005H	8005H	Hour and second data	Data in the device areas specified by (s1) and (s1)+2 are compared.
0006H	8006H	Hour and minute data	Data in the device areas specified by (s1) and (s1)+1 are compared.
0007H	8007H	Hour, minute, and second data	The entire time data in the device areas specified by (s1), (s1)+1, and (s1)+2 are compared.
Other than 0001H to 0007H, 8001H to 8007H		None	The entire time data in the device areas specified by (s1), (s1)+1, and (s1)+2 are not compared. (The operation result will be non-continuity.)

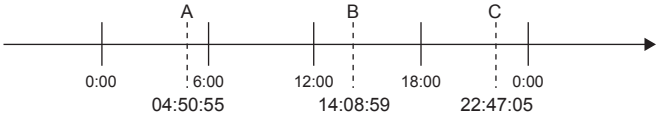
- If the comparison target data in the device are not recognized as time data, SM709 turns on after the instruction is executed and the operation result will be non-continuity. If the device areas specified by (s1) to (s1)+2 or (s2) to (s2)+2 exceed the corresponding device range, SM709 turns on after the instruction is executed and the operation result will be non-continuity as well. Once SM709 turns on, the on state is held until the CPU module is powered off or reset. Turn off SM709 as needed.

• The following table lists the comparison operation results of each instruction.

Instruction symbol	Condition	Result
TM=	(s1)=(s2)	Conductive state
TM<>	(s1)≠(s2)	
TM>	(s1)>(s2)	
TM<=	(s1)≤(s2)	
TM<	(s1)<(s2)	
TM>=	(s1)≥(s2)	
TM=	(s1)≠(s2)	Non-conductive state
TM<>	(s1)=(s2)	
TM>	(s1)≤(s2)	
TM<=	(s1)>(s2)	
TM<	(s1)≥(s2)	
TM>=	(s1)<(s2)	

Ex.

The time data A, B, and C are compared.



• The following table lists the comparison operation results between A, B, and C. Even when the data are compared under the same conditions, the results differ depending on the comparison target data.

○: Continuity, ×: Non-continuity

Comparison target data	Condition		
	A<B	B<C	A<C
Second data	○	×	×
Minute data	×	○	×
Minute and second data	×	○	×
Hour data	○	○	○
Hour and second data	○	○	○
Hour and minute data	○	○	○
Hour, minute, and second data	○	○	○
None	×	×	×

Operation error

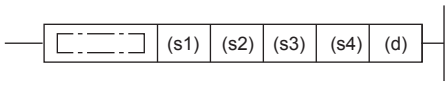
There is no operation error.

Comparing clock data

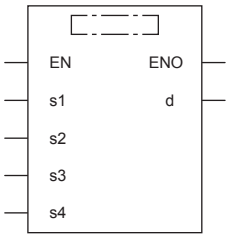
TCMP(P)

FX5S FX5UJ FX5U FX5UC

These instructions compare the time specified by (s1), (s2), and (s3) with the time data specified by (s4), and turn on/off the bit device specified by (d) depending on the size match.

Ladder diagram	Structured text
	ENO:=TCMP(EN,s1,s2,s3,s4,d); ENO:=TCMPP(EN,s1,s2,s3,s4,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Specify the "hour" of the time comparison	0 to 23	16-bit signed binary	ANY16
(s2)	Specify the "minute" of the time comparison	0 to 59	16-bit signed binary	ANY16
(s3)	Specify the "second" of the time comparison	0 to 59	16-bit signed binary	ANY16
(s4)	Specify the time data (hour, minute, and second)	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(d)	Specify the Bit device that turns on/off depending on the comparison result	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	○	—	—	○	○	—	—	—
(s4)	—	○	○	—	—	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- These instructions compare the time specified by (s1), (s2), and (s3) with the time data specified by (s4), and turn on/off the bit device specified by (d) depending on the size match.

(s1)	hour	(0 to 23)	>	(s4)	hour	(0 to 23)	⇨ (d) = ON
(s2)	minute	(0 to 59)		(s4)+1	minute	(0 to 59)	
(s3)	second	(0 to 59)		(s4)+2	second	(0 to 59)	

(s1)	hour	(0 to 23)	=	(s4)	hour	(0 to 23)	⇨ (d) +1 = ON
(s2)	minute	(0 to 59)		(s4)+1	minute	(0 to 59)	
(s3)	second	(0 to 59)		(s4)+2	second	(0 to 59)	

(s1)	hour	(0 to 23)	<	(s4)	hour	(0 to 23)	⇨ (d) +2 = ON
(s2)	minute	(0 to 59)		(s4)+1	minute	(0 to 59)	
(s3)	second	(0 to 59)		(s4)+2	second	(0 to 59)	

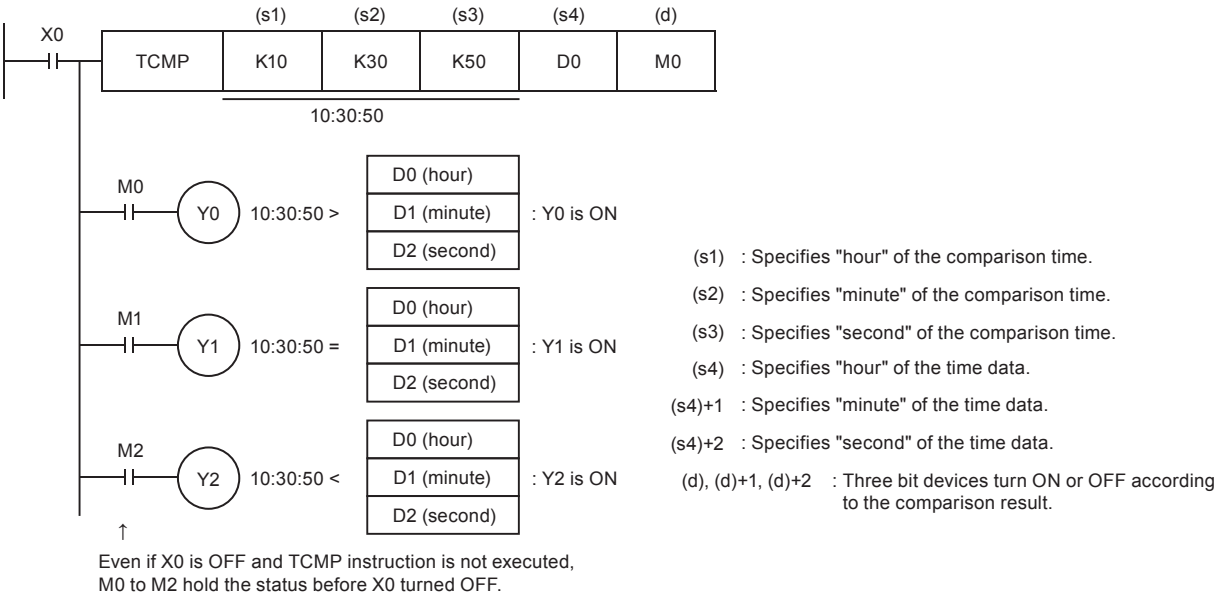
- (d), (d)+1, and (d)+2 hold the state before the command contact is turned off even if, the TCMP instruction is not executed by switching on → off the command contact.

Precautions

- Three devices are occupied by (s4) and (d). Make sure that these devices are not used by other machine controls.
- Specify each operand of the word device after reading the value of the special register used in the TRD(P) instruction when the time (hour, minute, second) of the clock data of the built-in real time clock in the CPU module is used.

Program example

In the program shown below, when X0 is set to ON, the time (10 hours 30 minutes 50 seconds) designated with (s1) to (s3) is compared with the D0 to D2 clock data, and the results are stored in M0 to M2.



Operation error

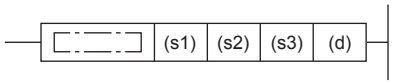
Error code (SD0/SD8067)	Description
2820H	The device range specified exceeds the corresponding device range.
3405H	The value specified by (s1) and (s4) is outside the following range. 0 to 23
	The value specified by (s2), (s3), (s4)+1, and (s4)+2 is outside the following range. 0 to 59

Comparing clock data zones

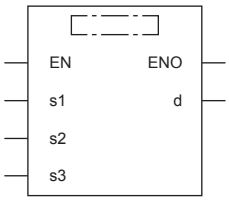
TZCP(P)

FX5S FX5UJ FX5U FX5UC

This instruction compares two comparison time (comparison time zone) specified by (s1) and (s2) with the time data specified by (s3), and turns on or off the specified bit devices (d) according to the comparison results.

Ladder diagram	Structured text
	ENO:=TZCP(EN,s1,s2,s3,d); ENO:=TZCPP(EN,s1,s2,s3,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Specify the lower limit of time comparison (hour, minute, and second).	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(s2)	Specify the upper limit of time comparison (hour, minute, and second).	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(s3)	Specify the time data (hour, minute, and second).	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(d)	Specify the Bit device that turns on/off depending on the comparison result	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

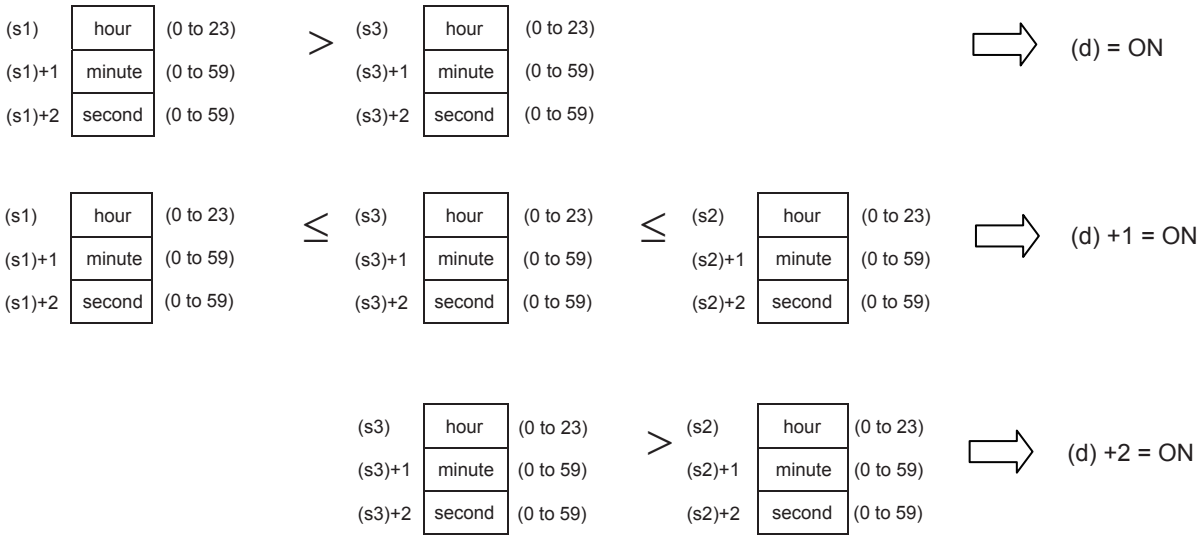
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	—	—	○	—	—	—	—
(s2)	—	○	○	—	—	—	○	—	—	—	—
(s3)	—	○	○	—	—	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- This instruction compares two comparison time (comparison time zone) specified by (s1) and (s2) with the time data specified by (s3), and turns on or off the specified bit devices (d) according to the comparison results.



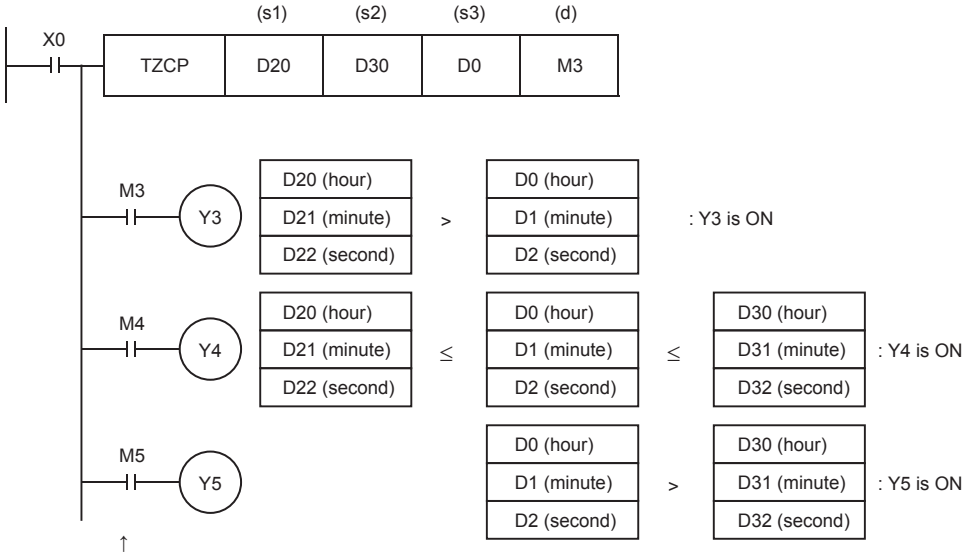
- Even if the command contact turns off from on and the TZCP instruction is not executed, (d), (d)+1, and (d)+2 hold the status before the command contact turned off.

Precautions

- Three devices are occupied by (s1), (s2), (s3), and (d). Make sure that these devices are not used by other machine controls.
- When the time (hour, minute, second) of the clock data of the real time clock built in the CPU module is used, read the values of special registers by the TRD instruction, and then specify those word devices as the operands.
- Make (s1) ≤ (s2).

Program example

In the program shown below, when X0 is set to ON, the range of the D20 to D22 and D30 to D32 clock data is compared with the D0 to D2 clock data, and the results are stored in M3 to M5.



Even if X0 is OFF and TZCP instruction is not executed,
M3 to M5 hold the status before X0 turned OFF.

(s1), (s1)+1, (s1)+2: Specify the lower limit of the comparison time zone in "hour", "minute" and "second."

(s2), (s2)+1, (s2)+2: Specify the upper limit of the comparison time zone in "hour", "minute" and "second."

(s3), (s3)+1, (s3)+2: Specify the time data in "hour", "minute" and "second."

(d), (d)+1, (d)+2: Turn ON or OFF according to the comparison result.

The setting range of "hour" is from 0 to 23.
The setting range of "minute" is from 0 to 59.
The setting range of "second" is from 0 to 59.

Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified exceeds the corresponding device range.
3405H	The value specified by (s1), (s2), and (s3) is outside the following range. 0 to 23
	The value specified by (s1)+1, (s2)+1, (s3)+1, (s1)+2, (s2)+2, and (s3)+2 is outside the following range. 0 to 59

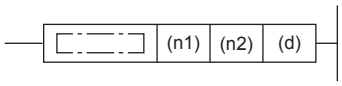
8.26 Timing Check Instruction

Generating timing pulses

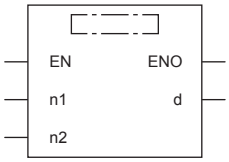
DUTY

FX5S FX5UJ FX5U FX5UC

This instruction sets user timing clock output destinations (SM420 to SM424 and SM8330 to SM8334) specified by (d) to on for the number of scans specified by (n1) and to off for the number of scans specified by (n2).

Ladder diagram	Structured text
	ENO:=DUTY(EN,n1,n2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(n1)	Number of scans to be turned on	0 to 32767	16-bit unsigned binary	ANY16
(n2)	Number of scans to be turned off	0 to 32767	16-bit unsigned binary	ANY16
(d)	Special relay of the timing clock output destination	(SM420 to SM424, SM8330 to SM8334)	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

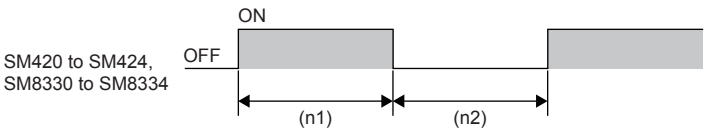
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○*1	—	—	—	—	—	—	—	—	—	—

*1 Only SM can be used.

Processing details

- This instruction sets user timing clock output destinations (SM420 to SM424 and SM8330 to SM8334) specified by (d) to on for the number of scans specified by (n1) and to off for the number of scans specified by (n2).



(n1): (n1) scans
(n2): (n2) scans

- Specify SM420 to SM424 (SM8330 to SM8334) in the special relay of the timing clock output destination specified by (d).
- In SM420 to SM424 (SM8330 to SM8334), when one device is turned on, another device is also turned on at the same time.
- The counted number of scans is stored among SD8330 to SD8334 in accordance with the special relay of the timing clock output destination specified by (d).
- The counted number of scans stored among SD8330 to SD8334 is reset when the counted value reaches "(n1)+(n2)" or when the command input (instruction) is set to on.

Special relay (d) for outputting the timing clock	Scan counting device
SM420(SM8330)	SD8330
SM421(SM8331)	SD8331
SM422(SM8332)	SD8332
SM423(SM8333)	SD8333
SM424(SM8334)	SD8334

- When the command input is set to ON, the operation is started. The special relay of the timing clock output destination is set to ON or OFF by the END instruction. Even if the command input is set to OFF, the operation is not stopped. In the STOP mode, the operation is stopped. When the power to the CPU module is turned OFF, the operation is stopped.
- When (n1) and (n2) are set to "0", the status is as shown below:

Status of (n1) and (n2)	ON/OFF status of (d)
(n1)=0, (n2)≥0	(d)= Fixed to OFF
(n1)>0, (n2)=0	(d)= Fixed to ON

- The table below shows the related devices (special relay).

Special relay	Name	Description
SM420(SM8330)	Timing clock output 1	Timing clock output in the DUTY instruction
SM421(SM8331)	Timing clock output 2	
SM422(SM8332)	Timing clock output 3	
SM423(SM8333)	Timing clock output 4	
SM424(SM8334)	Timing clock output 5	

- The table below shows the related devices (special register).

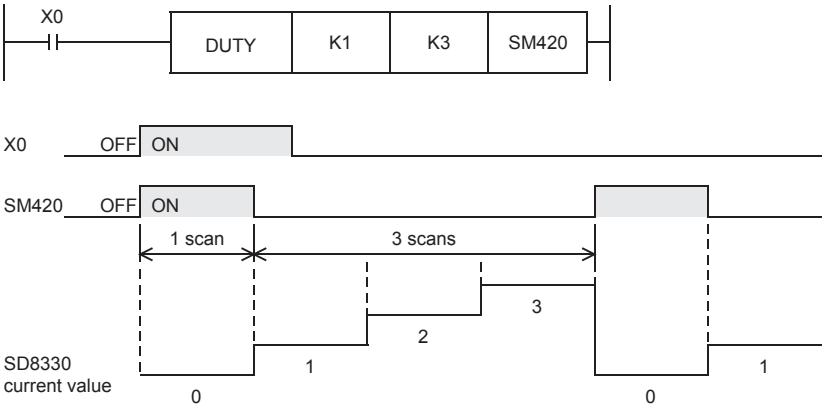
Special register	Name	Description
SD8330	Counted number of scans for timing clock output 1	Counted number of scans for timing clock output 1 in the DUTY instruction
SD8331	Counted number of scans for timing clock output 2	Counted number of scans for timing clock output 2 in the DUTY instruction
SD8332	Counted number of scans for timing clock output 3	Counted number of scans for timing clock output 3 in the DUTY instruction
SD8333	Counted number of scans for timing clock output 4	Counted number of scans for timing clock output 4 in the DUTY instruction
SD8334	Counted number of scans for timing clock output 5	Counted number of scans for timing clock output 5 in the DUTY instruction

Precautions

- The DUTY instruction can be used up to 5 times (points). It is not permitted, however, to use the same timing clock output destination device for two or more DUTY instructions.

Program example

In the program shown below, when X0 is set to ON, SM420 is set to ON for 1 scan, and is set to OFF for 3 scans.



Operation error

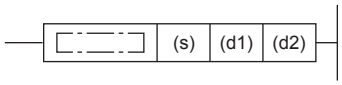
Error code (SD0/SD8067)	Description
2820H	The device specified by (d) is out of the range from SM420 to SM424 (SM8330 to SM8334).
3405H	The value specified by (n1), (n2) is other than 0 to 32767.

Hour meter

HOURM

FX5S FX5UJ FX5U FX5UC

This instruction measures the on time of the input contact in units of hour.

Ladder diagram	Structured text
	ENO:=HOURM(EN,s,d1,d2);

FBD/LD

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Time after which the alarm (d2) is set to on (unit: hour)	—	16-bit signed binary	ANY16
(d1)	Device for storing the measured current value (latched (battery backed) type data register)	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 2)
(d2)	Device to be turned on when timeout occurs (alarm output)	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

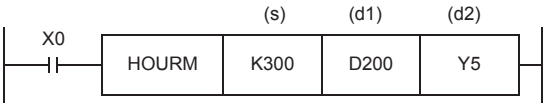
- This instruction measures the period of time for which the input contact is on in units of hour, and turns on the device specified by (d2) when the accumulated ON time exceeds the time (16-bit binary data) specified in (s).
- In (s), specify the period of time until the device specified by (d2) is turned on in units of hour.
- The measured current value in units of hour is stored in (d1).
- The measured current value of less than one hour (in units of second) is stored in (d1)+1.
- (d2) is set to on when the current value in (d1) exceeds the time specified by (s).
- Specify a latched (battery backed) type data register as (d1) so that the current value data can be continuously used even after the power to CPU module turns off. If a general data type register is used, the current value data is cleared when the power to the CPU module is turned OFF or when the controller mode switches from STOP to RUN.
- Even after the alarm output specified by (d2) turns ON, the measurement is continued.
- When the current value reaches the maximum value of 16-bit data, the measurement is stopped. For continuing the measurement, clear the current value stored in (d1) to (d1)+1.

Precautions

- Two devices are occupied by (d1). Make sure that these devices are not used by other machine controls.

Program example

In the program example shown below, when the accumulated X0 ON time exceeds 300 hours, Y5 turns ON.
The current value less than one hour is stored in D201 in units of second.



(s) : Time after which (d2) is set to ON
Specify a value in units of hour.
(d1) : Current value in units of hour
(d1)+1 : Current value less than one hour (unit: second)
(d2) : Alarm output destination
Turns ON when the current value (d1) reaches or exceeds the time specified in (s).
(In this example, the alarm output destination turns ON when the current value becomes 300 hours.)

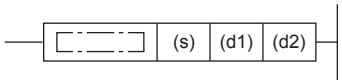
Operation error

Error code (SD0/SD8067)	Description
2820H	The device areas specified by (d1) exceed the corresponding device range.
3405H	The value of (s) is negative.

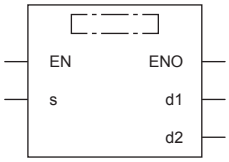
DHOURM

FX5S FX5UJ FX5U FX5UC

This instruction measures the on time of the input contact in units of hour.

Ladder diagram	Structured text
	ENO:=DHOURM(EN,s,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Time after which the alarm (d2) is set to on (unit: hour)	—	32-bit signed binary	ANY32
(d1)	Device for storing the measured current value (latched (battery backed) type data register)	—	32-bit signed binary	ANY32_ARRAY (Number of elements: 2)
(d2)	Device to be turned on when timeout occurs (alarm output)	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- This instruction measures the period of time for which the input contact is on in units of hour, and turns on the device specified by (d2) when the accumulated ON time exceeds the time (32-bit binary data) specified in (s).
- In (s)+1 and (s), specify the period of time until the device specified by (d2) is turned on in units of hour.
- The measured current value in units of hour is stored in (d1)+1 and (d1). ((d1)+1: highest-order, (d1): lowest-order)
- The measured current value of less than one hour (in units of second) is stored in (d1)+2.
- (d2) is set to on when the current value in (d1)+1 and (d1) exceeds the time specified by (s).
- Specify a latched (battery backed) type data register as (d1) so that the current value data can be continuously used even after the power to CPU module turns off. If a general data type register is used, the current value data is cleared when the power to the CPU module is turned OFF or when the controller mode switches from STOP to RUN.
- Even after the alarm output specified by (d2) turns ON, the measurement is continued.
- When the current value reaches the maximum value of 32-bit data, the measurement is stopped. For continuing the measurement, clear the current value stored in (d1) to (d1)+2.

Precautions

- Three devices are occupied by (d1). Make sure that these devices are not used by other machine controls.

Operation error

Error code (SD0/SD8067)	Description
2820H	The device areas specified by (d1) exceed the corresponding device range.
3405H	The value of (s) is negative.

8.27 Module Access Instruction

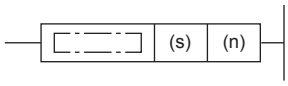
I/O refresh

REF(P)/RFS(P)

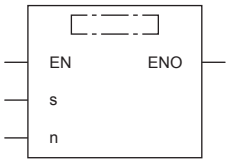
FX5S FX5UJ FX5U FX5UC

These instructions refresh the (n) points of devices starting from the device specified by (s), and receive an external input or generate an output.

The REF(P) instructions can also be used as RFS(P).

Ladder diagram	Structured text
	ENO:=REF(EN,s,n); ENO:=REFP(EN,s,n); ENO:=RFS(EN,s,n); ENO:=RFSP(EN,s,n);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number to be refreshed	—	Bit	ANY_BOOL
(n)	Number of devices to be refreshed	0 to 65535	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○*1	—	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

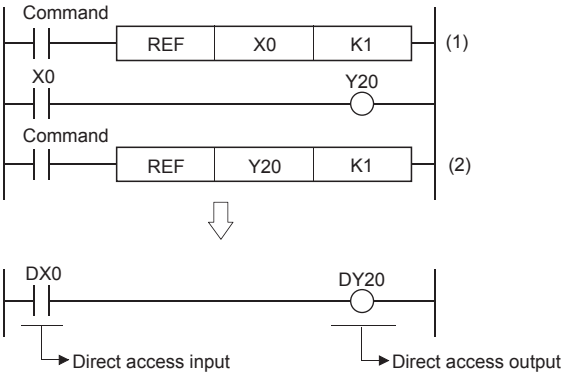
*1 Only X and Y can be used.

Processing details

- This function refreshes only the corresponding devices in the middle of a scan and receives an external input or generates an output.
- Since the input receptions and external outputs are performed at one time only after the END instruction is executed in the program, a pulse signal cannot be output externally in the middle of a scan. The execution of the I/O refresh instruction forcibly refreshes the corresponding input (X) or output (Y) in the middle of program execution, and then a pulse signal can be output externally in the middle of a scan.
- To refresh an input (X) or an output (Y) in 1 point units, use the direct access input (DX) or the direct access output (DY).

Ex.

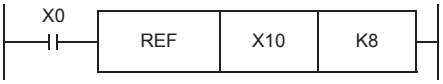
When a program using the RFS instruction is changed to a program using direct access input/output



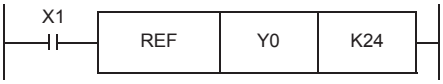
(1): Refreshes X0.
(2): Refreshes Y20.

Program example

In the program example shown below, when X0 is set to ON, X10 to X17 (8 points in total) are refreshed.



In the program example shown below, when X1 is set to ON, Y0 to Y7, Y10 to Y17 and Y20 to Y27 (24 points in total) are refreshed.



Operation error

Error code (SD0/SD8067)	Description
2820H	The (n) points of device range starting from the device specified by (s) exceed the range of proximal I/O.

Reading 1-word/2-word data from another module

FROM(P), DFROM(P)

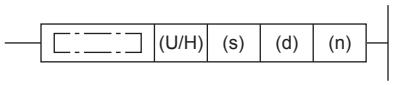
~~FX5S~~ FX5UJ FX5U FX5UC

• FROM(P)

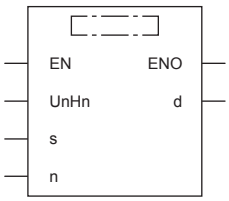
These instructions read (n) words of data from the buffer memory specified by (s) in intelligent function module specified by (U/H), and store the data to the device specified by (d) and later (d).

• DFROM(P)

These instructions read the (n) × 2 words of data from the buffer memory specified by (s) intelligent function module specified by (U/H), and store the data to the device specified by (d) and later.

Ladder diagram	Structured text
	ENO:=FROM(EN,UnHn,s,n,d); ENO:=FROMP(EN,UnHn,s,n,d); ENO:=DFROM(EN,UnHn,s,n,d); ENO:=DFROMP(EN,UnHn,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(U/H)*1		Unit number	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s)		Start address of the buffer memory where the read-target data is stored	0 to 65535	16-bit unsigned binary	ANY16
(d)	FROM(P)	Head device number for storing the read data	—	16-bit signed binary	ANY16
	DFROM(P)			32-bit signed binary	ANY32
(n)		Number of read data	1 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U/H displays as UnHn.

■Applicable devices

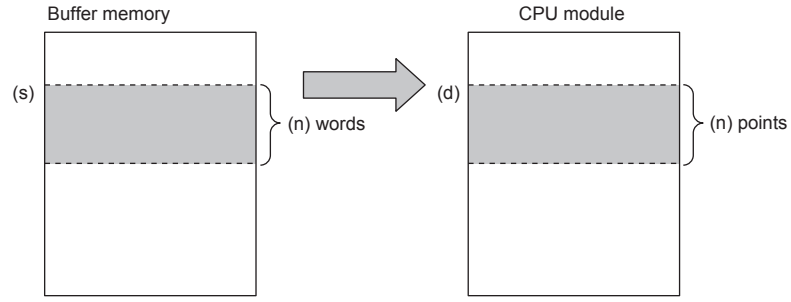
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U/H)	○	○	○	○	—	—	○	○	—	—	○
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	—	○	○*1	○*1	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 Only the DFROM(P) instruction can be used.

Processing details

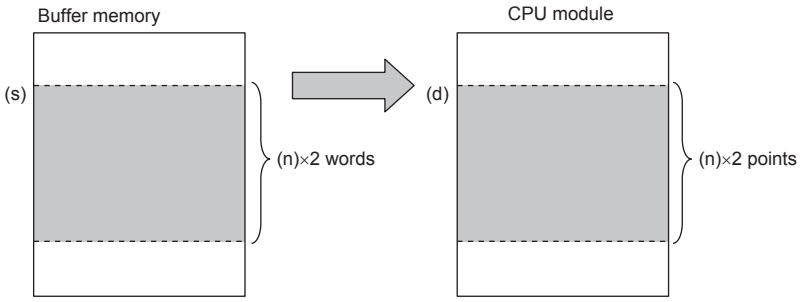
■FROM(P)

- These instructions read (n) words of data from the buffer memory specified by (s) in intelligent function module specified by (U/H), and store the data to the device specified by (d) and later.



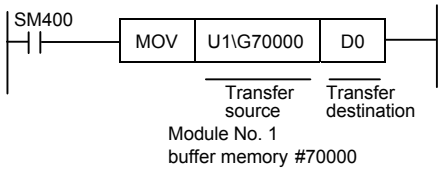
■DFROM(P)

- These instructions read the (n) × 2 words of data from the buffer memory specified by (s) in intelligent function module specified by (U/H), and store the data to the device specified by (d) and later.



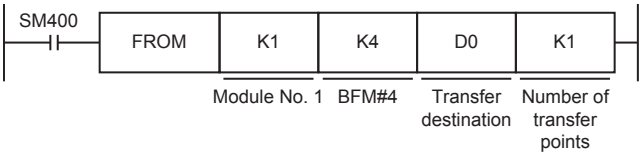
Precautions

- For the nibble of a bit device specified by (d), specify K1 to K4 in the FROM(P) instruction and K1 to K8 in the DFROM(P) instruction.
- When a number greater than 65535 is specified as the buffer memory specified by (s), use the FROMD(P) instruction or use U□\G□ in the MOV(P) instruction. The following shows the program to transfer the buffer memory #70000 in the intelligent function module No. 1 to D0.

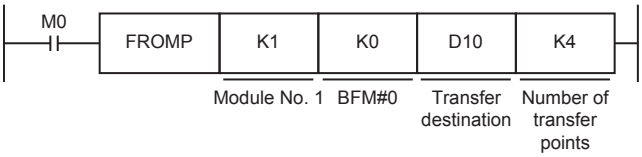


Program example

When the buffer memory #4 of the module No. 1 is read to D0.



When the buffer memories #0 to #3 of the module No. 1 are read to D10 to D13.



Operation error

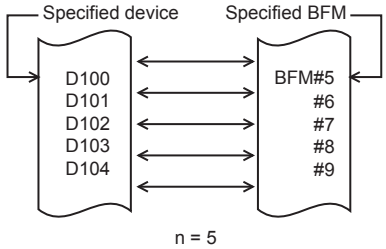
Error code (SD0/SD8067)	Description
2441H	Updating procedure with the unit was not properly completed during the execution of the instruction.
2801H	The unit number specified by (U/H) does not exist.
2823H	The buffer memory number specified by (s) exceeds the buffer memory area.
	The buffer memory number specified by (s) + the number of transfer points specified by (n) exceeds the buffer memory area.
2820H	The device number specified by (d) + the number of read data specified by (n) exceeds the corresponding device range.
3056H	Timeout occurred while communicating with the connected units during the execution of the instruction.
3060H	Signal error is detected while accessing the connected units during the execution of the instruction.
3580H	An instruction that cannot be used in an interrupt program is used.

Common items among the FROM(P), DFROM(P), TO(P), and DTO(P) (details)

- Use the module number to specify which Intelligent function module the instruction works for. The setting range of each CPU module is shown below.
  - FX5UJ CPU module: 1H to 8H (K1 to K8)
  - FX5U/FX5UC CPU module: 1H to 10H (K1 to K16)

		Module No. 1		Module No. 2		Module No. 3	Module No. 4	Module No. 5
CPU module	I/O module	Intelligent function module	Extension power supply unit	Intelligent function module	I/O module	Intelligent function module	Bus conversion module	Intelligent function module

- A module number is automatically assigned to each intelligent function module connected to a CPU module. The module number is assigned in the way "No. 1 → No. 2 → No. 3 ..." starting from the equipment nearest the CPU module.
- 16-bit RAM memories are built in an intelligent function module, and they are called buffer memories. The contents of buffer memories vary depending on the purpose of control of each Intelligent function module, and the setting range is from K0 to K65535.
- The number of read data is specified by (n), and the setting range is from K1 to K65535.



- If this instruction is executed in an interrupt program with the priority 1, operation error (3580H) occurs. This instruction operates in an interrupt program with the priority 2 or 3.
- If this instruction is executed to an FX3 intelligent function module within an interrupt program, operation error (3580H) occurs.

Writing 1-word/2-word data to another module

TO(P), DTO(P)

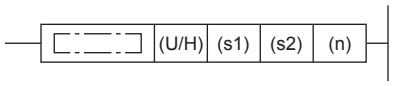
FX5S FX5UJ FX5U FX5UC

• TO(P)

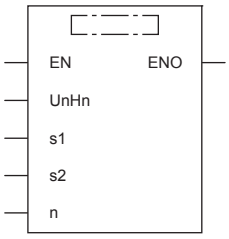
These instructions write the (n) points of data in the device starting from the one specified by (s2) to the buffer memory address specified by (s1) in intelligent function module specified by (U/H).

• DTO(P)

These instructions write the (n) × 2 points of data in the device starting from the one specified by (s2) to the buffer memory address specified by (s1) in intelligent function module specified by (U/H).

Ladder diagram	Structured text
	ENO:=TO(EN,UnHn,s1,s2,n); ENO:=TOP(EN,UnHn,s1,s2,n); ENO:=DTO(EN,UnHn,s1,s2,n); ENO:=DTOP(EN,UnHn,s1,s2,n);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(U/H)*1		Unit number	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)		Start address of the buffer memory for writing the data	0 to 65535	16-bit unsigned binary	ANY16
(s2)	TO(P)	Write data, or head device number which stores the write data	—	16-bit signed binary	ANY16
	DTO(P)		—	32-bit signed binary	ANY32
(n)		Number of write data	1 to 65535	16-bit unsigned binary	ANY16
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U/H displays as UnHn.

■Applicable devices

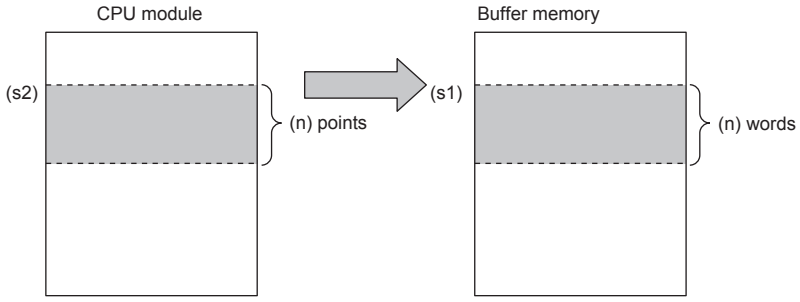
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U/H)	○	○	○	○	—	—	○	○	—	—	○
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	—	○	○*1	○*1	○	○	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 Only the DTO(P) instruction can be used.

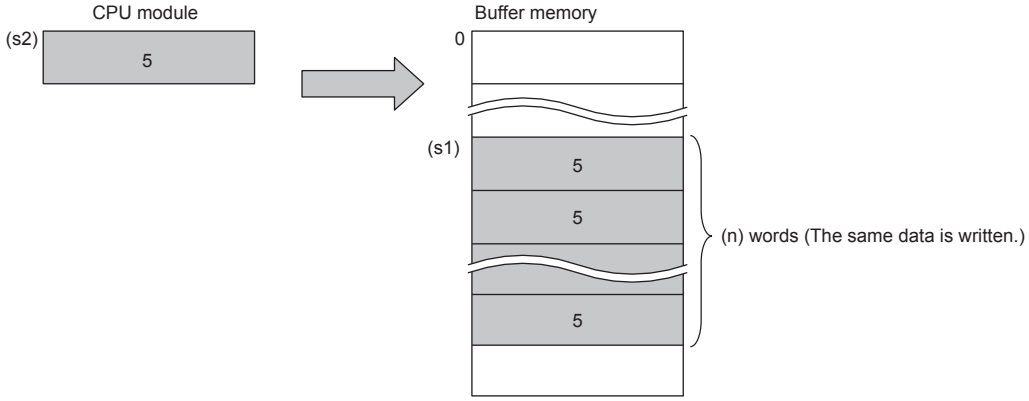
Processing details

■TO(P)

- These instructions write the (n) points of data in the device starting from the one specified by (s2) to the buffer memory address specified by (s1) in intelligent function module specified by (U/H).

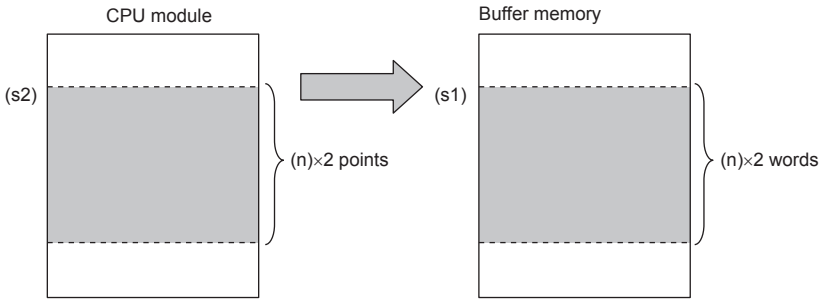


- When a constant is specified in (s2), (n) words of the same data (the value specified by (s2)) is written starting from the specified buffer memory address.

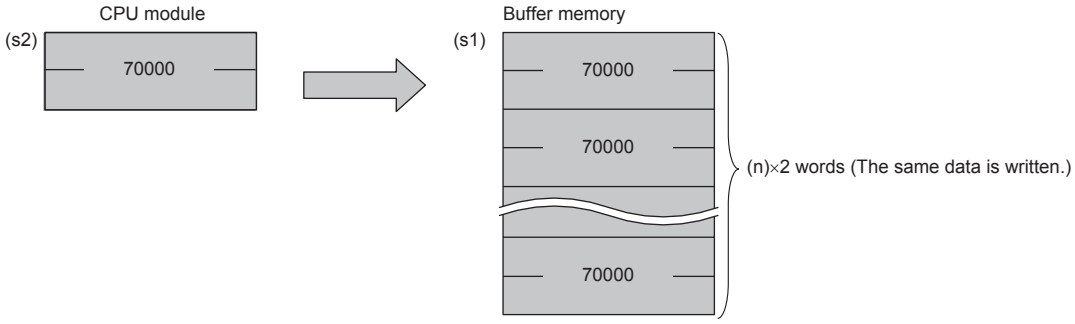


■DTO(P)

- These instructions write the (n) × 2 points of data in the device starting from the one specified by (s2) to the buffer memory address specified by (s1) in intelligent function module specified by (U/H).



- When a constant is specified in (s2), (n) x 2 words of the same data (the value specified by (s2)) is written starting from the specified buffer memory address.



Precautions

- For the nibble of a bit device specified by (s2), specify K1 to K4 in the TO(P) instruction and K1 to K8 in the DTO(P) instruction.
- When a number greater than 65535 is specified as the buffer memory specified by (s1), use the TOD(P) instruction or use U□\G□ in the MOV(P) instruction.

Program example

When writing "0" to the buffer memory #27 of the module No. 1.

Operation error

Error code (SD0/SD8067)	Description
2441H	Updating procedure with the unit was not properly completed during the execution of the instruction.
2801H	The unit number specified by (U/H) does not exist.
2823H	The buffer memory number specified by (s1) exceeds the buffer memory area.
	The buffer memory number specified by (s1) + the number of transfer points specified by (n) exceeds the buffer memory area.
2820H	The device number specified by (s2) + the number of write data specified by (n) exceeds the corresponding device range.
3056H	Timeout occurred while communicating with the connected units during the execution of the instruction.
3060H	Signal error is detected while accessing the connected units during the execution of the instruction.
3580H	An instruction that cannot be used in an interrupt program is used.

Reading 1-word/2-word data from another module

FROMD(P), DFROMD(P)

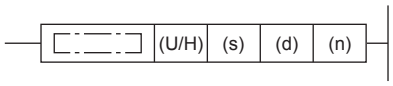
FX5S FX5UJ FX5U FX5UC

• FROMD(P)

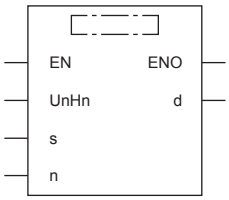
These instructions read (n) words of data from the buffer memory specified by (s) in intelligent function module specified by (U/H), and store the data to the device specified by (d) and later.

• DFROMD(P)

These instructions read the (n) × 2 words of data from the buffer memory specified by (s) in intelligent function module specified by (U/H), and store the data to the device specified by (d) and later.

Ladder diagram	Structured text
	ENO:=FROMD(EN,UnHn,s,n,d); ENO:=FROMDP(EN,UnHn,s,n,d); ENO:=DFROMD(EN,UnHn,s,n,d); ENO:=DFROMDP(EN,UnHn,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(U/H)*1		Unit number	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s)		Start address of the buffer memory where the read-target data is stored	0 to 4294967295	32-bit unsigned binary	ANY32
(d)	FROMD(P)	Head device number for storing the read data	—	16-bit signed binary	ANY16
	DFROMD(P)			32-bit signed binary	ANY32
(n)		Number of read data	1 to 65535	32-bit unsigned binary	ANY32
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U/H displays as UnHn.

■Applicable devices

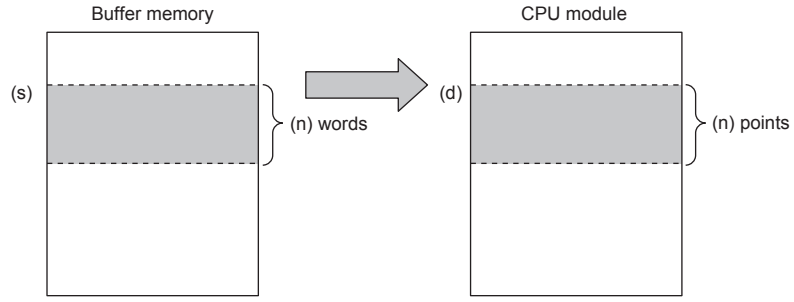
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U/H)	○	○	○	○	—	—	○	○	—	—	○
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	—	○	○*1	○*1	○	—	—	—	—
(n)	○	○	○	○	○	○	○	○	—	—	—

*1 Only the DFROMD(P) instruction can be used.

Processing details

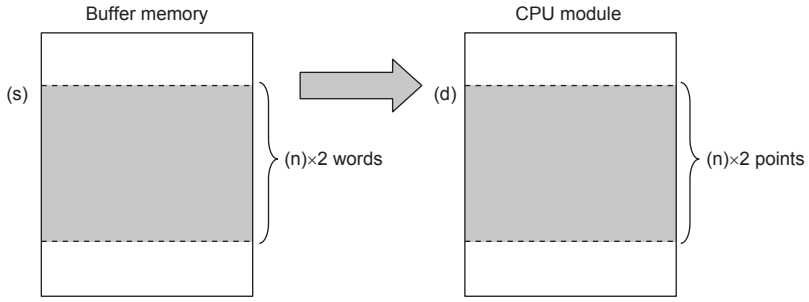
■FROMD(P)

- These instructions read (n) words of data from the buffer memory specified by (s) in intelligent function module specified by (U/H), and store the data to the device specified by (d) and later.



■DFROMD(P)

- These instructions read the (n) × 2 words of data from the buffer memory specified by (s) in intelligent function module specified by (U/H), and store the data to the device specified by (d) and later.



Precautions

- For the nibble of a bit device specified by (d), specify K1 to K4 in the FROMD(P) instruction and K1 to K8 in the DFROMD(P) instruction.

Operation error

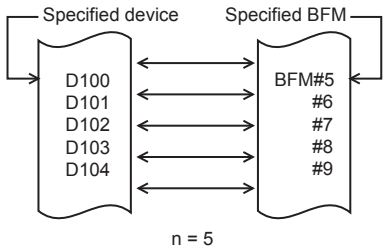
Error code (SD0/SD8067)	Description
2441H	Updating procedure with the unit was not properly completed during the execution of the instruction.
2801H	The unit number specified by (U/H) does not exist.
2823H	The buffer memory number specified by (s) exceeds the buffer memory area.
	The buffer memory number specified by (s) + the number of transfer points specified by (n) exceeds the buffer memory area.
2820H	The device number specified by (d) + the number of read data specified by (n) exceeds the corresponding device range.
3056H	Timeout occurred while communicating with the connected units during the execution of the instruction.
3060H	Signal error is detected while accessing the connected units during the execution of the instruction.
3580H	An instruction that cannot be used in an interrupt program is used.

Common items among the FROMD(P), DFROMD(P), TOD(P), and DTOD(P) (details)

- Use the module number to specify which intelligent function module the instruction works for. The setting range of each CPU module is shown below.
  - FX5UJ CPU module: 1H to 8H (K1 to K8)
  - FX5U/FX5UC CPU module: 1H to 10H (K1 to K16)

		Module No. 1		Module No. 2		Module No. 3	Module No. 4	Module No. 5
CPU module	I/O module	Intelligent function module	Extension power supply unit	Intelligent function module	I/O module	Intelligent function module	Bus conversion module	Intelligent function module

- A module number is automatically assigned to each intelligent function module connected to a CPU module. The module number is assigned in the way "No. 1 → No. 2 → No. 3 ..." starting from the equipment nearest the CPU module.
- 16-bit RAM memories are built in an intelligent function module, and they are called buffer memories. The contents of buffer memories vary depending on the purpose of control of each intelligent function module, and the setting range is from K0 to K4294967295.
- The number of read data is specified by (n), and the setting range is from K1 to K65535.



- If this instruction is executed in an interrupt program with the priority 1, operation error (3580H) occurs. This instruction operates in an interrupt program with the priority 2 or 3.
- If this instruction is executed to an FX3 intelligent function module within an interrupt program, operation error (3580H) occurs.

Writing 1-word/2-word data to another module (32-bit specification)

TOD(P), DTOD(P)

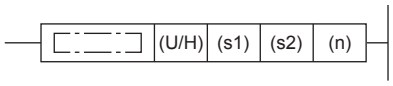
FX5S FX5UJ FX5U FX5UC

• TOD(P)

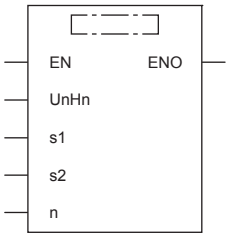
These instructions write the (n) points of data in the device starting from the one specified by (s2) to the buffer memory address specified by (s1) in intelligent function module specified by (U/H).

• DTOD(P)

These instructions write the (n) × 2 points of data in the device starting from the one specified by (s2) to the buffer memory address specified by (s1) in intelligent function module specified by (U/H).

Ladder diagram	Structured text
	ENO:=TOD(EN,UnHn,s1,s2,n); ENO:=TODP(EN,UnHn,s1,s2,n); ENO:=DTOD(EN,UnHn,s1,s2,n); ENO:=DTODP(EN,UnHn,s1,s2,n);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(U/H)*1		Unit number	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)		Start address of the buffer memory for writing the data	0 to 4294967295	32-bit unsigned binary	ANY32
(s2)	TOD(P)	Write data, or head device number which stores the write data	—	16-bit signed binary	ANY16
	DTOD(P)		—	32-bit signed binary	ANY32
(n)		Number of write data	1 to 65535	32-bit unsigned binary	ANY32
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U/H displays as UnHn.

■Applicable devices

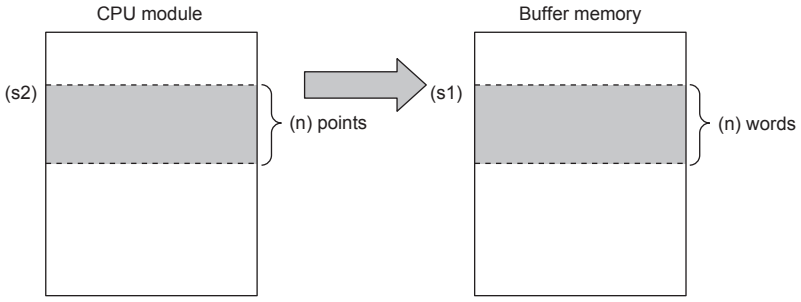
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U/H)	○	○	○	○	—	—	○	○	—	—	○
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	—	○	○*1	○*1	○	○	—	—	—
(n)	○	○	○	○	○	○	○	○	—	—	—

*1 Only the DTOD(P) instruction can be used.

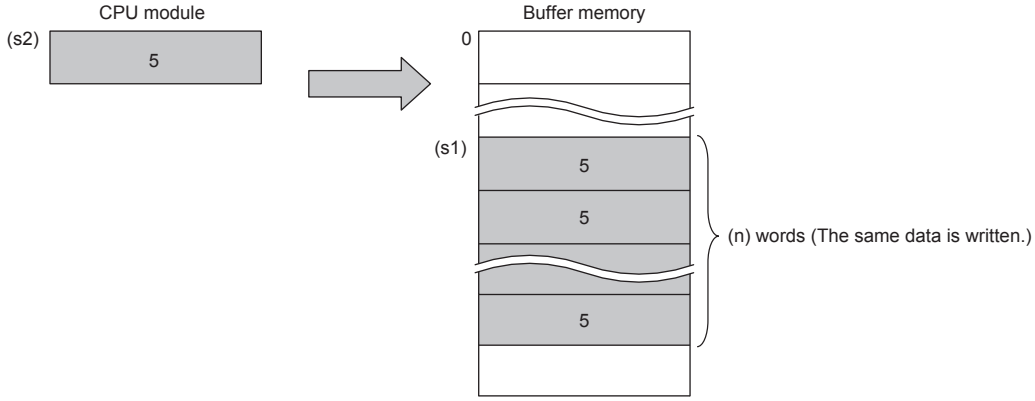
Processing details

■TOD(P)

- These instructions write the (n) points of data in the device starting from the one specified by (s2) to the buffer memory address specified by (s1) in intelligent function module specified by (U/H).



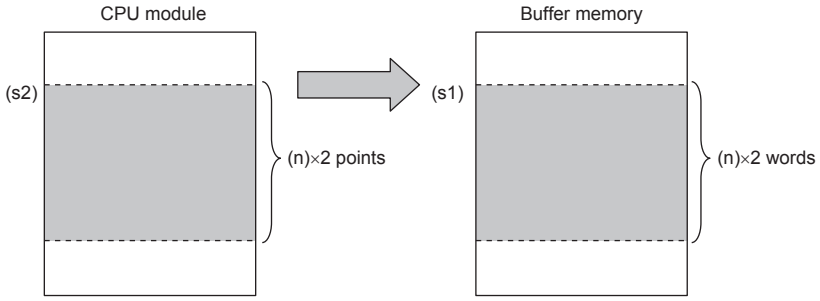
- When a constant is specified in (s2), (n) words of the same data (the value specified by (s2)) is written starting from the specified buffer memory address.



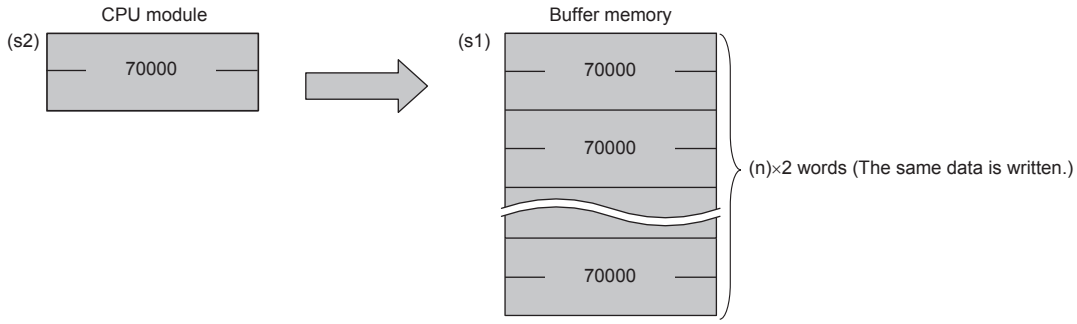
8

■DTOD(P)

- These instructions write the (n) × 2 points of data in the device starting from the one specified by (s2) to the buffer memory address specified by (s1) in intelligent function module specified by (U/H).



- When a constant is specified in (s2), (n) × 2 words of the same data (the value specified by (s2)) is written starting from the specified buffer memory address.



Precautions

- For the nibble of a bit device specified by (s2), specify K1 to K4 in the TOD(P) instruction and K1 to K8 in the DTOD(P) instruction.

Operation error

Error code (SD0/SD8067)	Description
2441H	Updating procedure with the unit was not properly completed during the execution of the instruction.
2801H	The unit number specified by (U/H) does not exist.
2823H	The buffer memory number specified by (s1) exceeds the buffer memory area.
	The buffer memory number specified by (s1) + the number of transfer points specified by (n) exceeds the buffer memory area.
2820H	The device number specified by (s2) + the number of write data specified by (n) exceeds the corresponding device range.
3056H	Timeout occurred while communicating with the connected units during the execution of the instruction.
3060H	Signal error is detected while accessing the connected units during the execution of the instruction.
3580H	An instruction that cannot be used in an interrupt program is used.

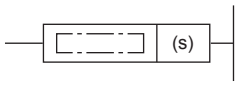
8.28 Logging Instructions

Setting trigger logging

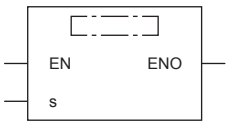
LOGTRG

FX5S FX5UJ FX5U FX5UC

This instruction generates a trigger condition for the specified logging setting number in trigger logging.

Ladder diagram	Structured text
	ENO:=LOGTRG(EN,s);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Logging setting number	1 to 4	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	○	—	—	○	○	—	—	—

Processing details

- This instruction generates a trigger for trigger logging with the logging setting number specified by (s).
- Specify a value 1 to 4 in (s).
- The LOGTRG instruction turns on the special relay (logging trigger) with the logging setting number in the device specified by (s) and executes trigger logging for the specified number of records.
- The instruction is enabled when "When trigger instruction executed" is selected in the "Trigger condition".
- Even if the LOGTRG instruction is executed, no processing is performed in the following cases.
  - A logging setting number in which an item other than "When trigger instruction executed" is selected in the "Trigger condition" is specified.
  - A logging setting number with no setting is specified.
  - A logging setting number specifying the execution of continuous logging is specified.
  - Another LOGTRG instruction is executed without executing the LOGTRGR instruction after a LOGTRG instruction was executed once.

Operation error

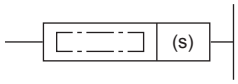
Error code (SD0/SD8067)	Description
3405H	The value set in (s) is other than the following. 1≤(s)≤4

Resetting trigger logging

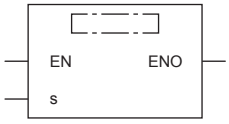
LOGTRGR

FX5S FX5UJ FX5U FX5UC

This instruction resets the trigger condition of the specified logging setting number.

Ladder diagram	Structured text
	ENO:=LOGTRGR(EN,s);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Logging setting number	1 to 4	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○	—	○	—	—	○	○	—	—	—

Processing details

- This instruction resets the LOGTRG instruction of the logging setting number specified by (s). The instruction disables the LOGTRG instruction of the specified trigger logging setting number.
- The LOGTRGR instruction turns off the special relay (logging completion, logging trigger, and after logging trigger) with the logging setting number in the device specified by (s).

Precautions

When the LOGTRGR instruction is executed after the LOGTRG instruction is executed in the same scan, the LOGTRG instruction becomes invalid.

Operation error

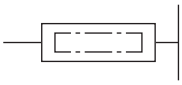
Error code (SD0/8067)	Description
3405H	The value set in (s) is other than the following. 1≤(s)≤4

8.29 Real-time Monitor Function Instruction

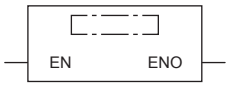
RTM

FX5S FX5UJ FX5U FX5UC

The real-time monitor function monitors the data when the instruction is executed.

Ladder diagram	Structured text
	ENO:=RTM(EN);

FBD/LD



Processing details

Depending on the monitor condition of the real-time monitor function, the data collection timing is when the instruction is executed. When the command input turns on as shown below, the data can be monitored at any timing.

■Related devices

Device	Name	Description
SD1484	Real-time monitor internal buffer free space	The amount of free space of the internal buffer is stored in K bytes.

Precautions

- More than one RTM instruction must not be issued during one scan. The data missing occurs.
- In the following case, even if the RTM instruction is executed, the procession is not performed.
  - When the collection timing of the real time monitor is not set to "RTM Instruction Specification" by GX LogViewer.
  - When the RTM instruction is executed in the interrupt program.

Operation error

There is no operation error.

MEMO

9 STEP LADDER INSTRUCTIONS

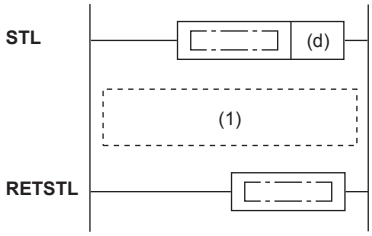
9.1 Starts/Ends Step Ladder

STL, RETSTL

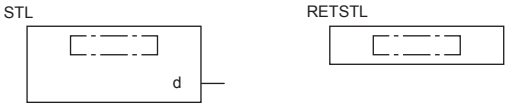
FX5S FX5UJ FX5U FX5UC

STL: This instruction starts step ladder.

RETSTL: This instruction ends step ladder.

Ladder diagram	Structured text*1
 (1): Step ladder program	STL(d); RETSTL();

FBD/LD*1



*1 For supported version, refer to ☞ Page 1452 Added and Changed Functions.

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	The step relay number which assigns the State	0 to 4095	Bit	ANY_BOOL

■Applicable devices

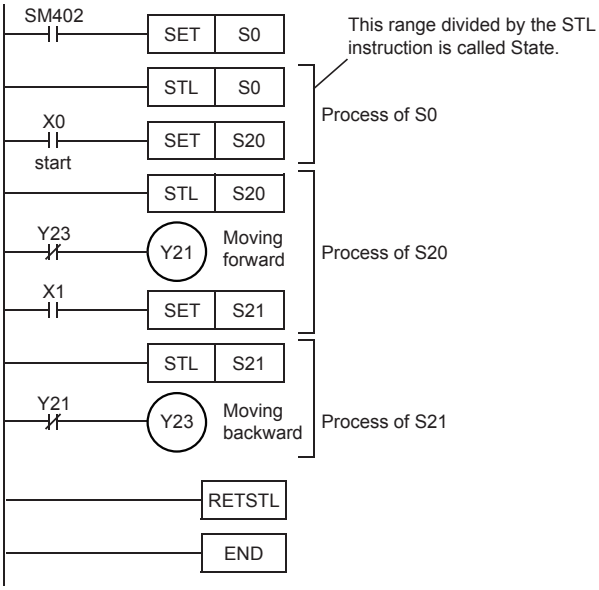
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1	—	—	—	—	—	—	—	—	—	—

*1 Only S can be used.

Processing details

- In programs using step ladder instructions, a step relay S is assigned to each process based on machine operations, and sequences of input condition and output control are programmed as circuits connected to contacts (STL contacts) of state relays.
- In a step ladder program, a step relay S is regarded as one control process, and a sequence of input conditions and output controls are programmed in a state relay. Because the preceding process is stopped when the program execution proceeds to the next process, a machine can be controlled using simple sequences for each process.
- The step relay number specified by STL instruction is assigned to the State. The start and completion of the State use SET instruction, OUT instruction, RST instruction, and ZRST instruction.
- Program a step ladder program starting from the initial state relay in the order of state relay ON status transfer. Make sure to put the RETSTL instruction at the end of a step ladder program. The RETSTL instruction is omissible anywhere other than the last Step Ladder.

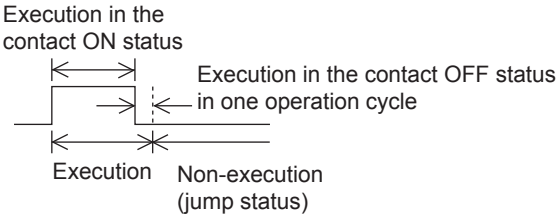
The actual step ladder program is as follows.



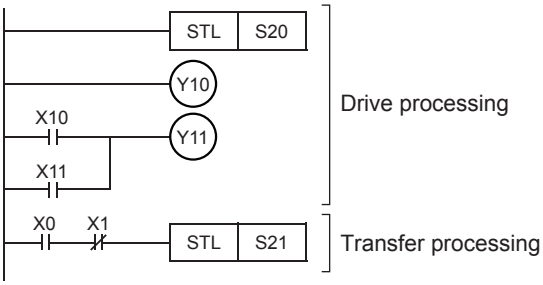
- A step ladder program is expressed as a relay ladder, but it can be created according to the machine control flow using state relays. A state relay consists of a drive coil and contact (STL contact) in the same way as other relays. Use SET or OUT instructions to drive a coil, and use STL instructions for a contact.
- The table below shows operations of internal circuits connected to the state relays,

Internal circuit diagram operation	
Execution in the contact ON status	When a state relay turns ON, a connected circuit (internal circuit) is activated with a STL contact.
Execution in the contact OFF status (for one operation cycle)	When a condition (transfer condition) provided between state relays is satisfied, the next state relay turns ON, and the state relay which was been ON before hand turns OFF (transfer operation). In the state relay ON status transfer process, both state relays are ON for one operation cycle. In the next operation cycle after the ON status is transferred to the next state relay, the former state relay is reset to OFF. A drive instruction connected to the bus line of the reset state relay is executed in the contact OFF status for one operation cycle regardless of the actual contact status before the drive instruction. When the transfer state relay is used in a contact instruction, however, the contact image is executed in the OFF status immediately after the transfer condition is satisfied.
Non-execution	An instruction is not executed in the contact OFF status after the operation cycle where the instruction was executed in the contact OFF status (jump status).

- The figure below shows the timing chart of the state relay (internal circuit) activation status.

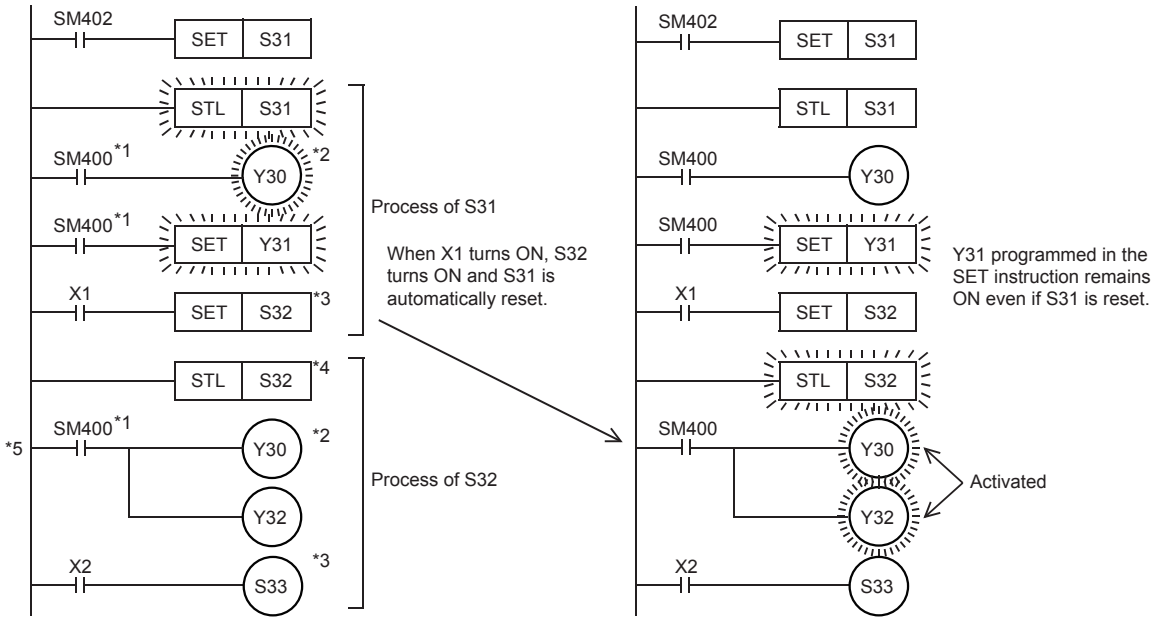


- Each state relay has three functions, driving a load, specifying a transfer destination and specifying a transfer condition. Such as the following program, driving a load and then transition processing. In a state relay without any load, the drive processing is not required.



■Operation of program

The step ladder program operates as follows.



*1 It is always necessary to program to drive an output.
*2 Output coils can be used again in different state relays.
*3 Each OUT and SET instruction for step relays automatically resets the transfer source, and has the self-holding function.
*4 One step relay number can only be used once.
*5 It is not possible to place a pointer immediately after the STL instruction. If a pointer is placed, a program error (33E2H) occurs.

Point

When the step relay is latched, the ON/OFF status is backed up by nonvolatile memory against power failure. Use this type of state relay if the operation should be restarted from the last point before power failure. These relays hold the operation status also at the time of RUN to STOP. If it RUN(s) again, the operation will be resumed from the State before STOP.

■Related devices

Device	Name	Description
SM8040	STL transfer disable	When SM8040 is set to ON, transfer of the ON status is disabled among all state relays.
SM8046	STL state ON	When the step relay turns on, SM8046 turns on automatically.
SM8047	Enable STL monitoring	When SM8047 is turned on, it stores in SD8040 to SD8047 sequentially from the young number of the step relay which operates in the step relay.
SD8040 to SD8047	ON step relay numbers	The turned-on step relay number is stored in SD8040 to SD8047 (Max. 8 points) sequentially from the smaller number.

Precautions

- When the step relay is not latched, step relay clears at power supply ON→OFF or RUN→STOP. If the State is valid and power supply ON→OFF or PLC RUN→STOP, operation can not be restarted from the last point before power-supply ON→OFF or RUN→STOP.
- The STL/RETSTL instruction cannot be used in a project including an SFC program.

Operation error

Error code (SD0/SD8067)	Description
33E3H	The STL instruction or RETSTL instruction is programmed between FOR-NEXT.
33E5H	The STL instruction or RETSTL instruction is programmed in an interrupt program, subroutine program or function block.
	The frequency of use of the STL instruction is not consistent with that of the RETSTL instruction.
33F3H	Three or more STL instructions with the same step relay No. are programmed.

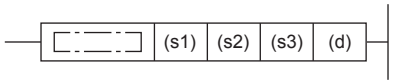
10 PID CONTROL INSTRUCTION

10.1 PID Control Loop

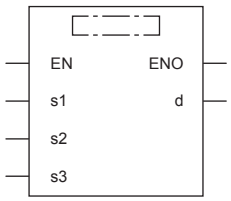
PID

FX5S FX5UJ FX5U FX5UC

This instruction executes PID control which changes the output value according to the input variation.

Ladder diagram	Structured text
	ENO:=PID(EN,s1,s2,s3,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Device number storing the target value (SV)	-32768 to +32767	16-bit signed binary	ANY16*1
(s2)	Device number storing the measured value (PV)	-32768 to +32767	16-bit signed binary	ANY16*1
(s3)	Device number storing a parameter	1 to 32767	16-bit signed binary	ANY16*1
(d)	Device number storing the output value (MV)	-32768 to +32767	16-bit signed binary	ANY16*1
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 When setting using a label, use the global label assigned to the device.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	○	—	—	—	—	—	—	—	—
(s2)	—	○*1	○	—	—	—	—	—	—	—	—
(s3)	—	○*1	—	—	—	—	—	—	—	—	—
(d)	—	○*1	○	—	—	—	—	—	—	—	—

*1 Only D, SD, R can be used.

Processing details

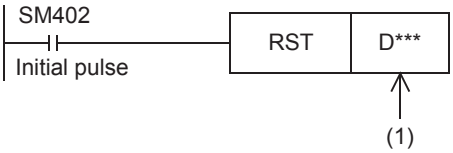
When the target value (s1), measured value (s2), and parameters (s3) to (s3)+6 are set and a program is executed, the operation result (MV) is stored to the output value (d) at every sampling time (s3). For details, refer to the MELSEC iQ-F FX5 User's Manual (Application).

Precautions

- When auto tuning is not used, twenty-five devices are occupied from the head device specified in (s3).
- When auto tuning (limit cycle method) is used, twenty-nine devices are occupied from the head device specified in (s3).
- When auto tuning (step response method) is used, the number of devices occupied from the devices specified in (s3) differs depending on the ON/OFF status of b8 of (s3)+1.

Operation of b8	Occupied device points
ON	Twenty-eight devices are occupied from the device specified in (s3).
OFF	Twenty-five devices are occupied from the device specified in (s3).

- Two or more PID instructions can be executed at the same time. (There is no limitation in the number of loops.) However, make sure that (s3), (d) and other operands specified in each instruction are different to each other.
- For the output value (MV) in the PID instruction, specify an unlatched data register. If specifying a latched data register, make sure to clear the backup contents when the PLC mode is set to RUN using the following program.



(1): Latched data register number specified to (d)

Operation error

Error code (SD0/SD8067)	Description
2820H	The connection number specified by (s1) is other than 1 to 8.
2822H	The device number specified by (s2), (d), or (n) is outside the range of the number of device points.
3500H	Incorrect sampling time (TS) (TS ≤ 0)
3502H	Incompatible input filter constant (α) (α < 0 or 100 ≤ α)
3503H	Incompatible proportional gain (KP) (KP ≤ 0)
3504H	Incompatible integral time (TI) (TI < 0)
3505H	Incompatible derivative gain (KD) (KD < 0 or 100 ≤ KD)
3506H	Incompatible derivative time (TD) (TD < 0)
350AH	Sampling time (TS) ≤ Scan time
350CH	Variation of measured value exceeds limit. (△PV < -32768 or +32767 < △PV)
350DH	Deviation exceeds limit. (EV < -32768 or +32767 < EV)
350EH	Integral result exceeds limit. (Outside range from -32768 to +32767)
350FH	Derivative value exceeds limit due to derivative gain (KD).
3510H	Derivative result exceeds limit. (Outside range from -32768 to +32767)
3511H	PID operation result exceeds limit. (Outside range from -32768 to +32767)
3512H	PID output upper limit set value < PID output lower limit set value.
3513H	Abnormal PID input variation alarm set value or output variation alarm set value. (Set value < 0)
3514H	Step response method Improper auto tuning result.
3515H	Step response method Auto tuning operation direction mismatch.
3516H	Step response method Improper auto tuning operation.
3517H	Limit cycle method Abnormal output set value for auto tuning [ULV (upper limit) ≤ LLV (lower limit)]
3518H	Limit cycle method Abnormal PV threshold (hysteresis) set value for auto tuning (SHPV < 0)
3519H	Limit cycle method Abnormal auto tuning transfer status (Data of device controlling transfer status is abnormally overwritten.)
351AH	Limit cycle method Abnormal result due to excessive auto tuning measurement time (τon > τ, τon < 0, τ < 0)

Error code (SD0/SD8067)	Description
351BH	Limit cycle method Auto tuning result exceeds proportional gain. (KP = outside range from 1 to 32767)
351CH	Limit cycle method Auto tuning result exceeds integral time. (TI = outside range from 0 to 32767)
351DH	Limit cycle method Auto tuning result exceeds derivative time. (TD = outside range from 0 to 32767)
351EH	Step response method Abnormal timeout time set value after maximum ramp for auto tuning

11 SFC PROGRAM INSTRUCTIONS

Point

This chapter describes the instructions used in SFC programs. For details on SFC programs, refer to the following.

MELSEC iQ-F FX5 Programming Manual (Program Design)

11.1 SFC Control Instructions

Checking the status of a step

LD, LDI, AND, ANI, OR, ORI [S□/BL□\S□]

FX5S FX5UJ FX5U FX5UC

• LD: Normally open contact, LDI: Normally closed contact

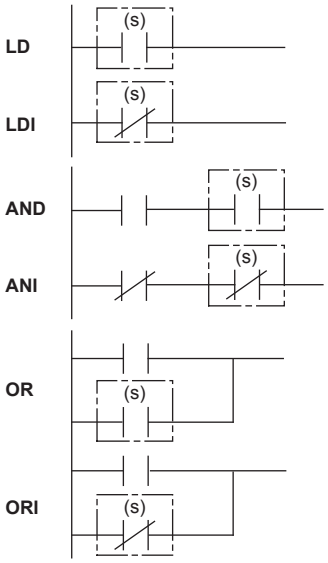
These instructions output the status (active or inactive) of the specified step as the operation result.

• AND: Normally open contact series connection, ANI: Normally closed contact series connection

These instructions perform an AND operation between the status (active or inactive) of the specified step and the previous operation result(s), and output the operation result.

• OR: Single normally open contact parallel connection, ORI: Single normally closed contact parallel connection

These instructions perform an OR operation between the status (active or inactive) of the specified step and the previous operation result(s), and output the operation result.

Ladder diagram	Structured text
	Not supported.

FBD/LD
Not supported.

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Device used as a contact	—	Bit	ANY_BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□\S□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○*1*2	—	—	—	—	—	—	—	—	—	○*2

*1 Only S can be used.
*2 Indexing is not available.

Processing details

- These instructions check whether the specified step in the specified block is active or not.
- The status (on or off) of each contact will be as follows depending on the status (active or inactive) of the specified step.

Status of the specified step	Contact of the normally open contact instruction	Contact of the normally closed contact instruction
Active	ON	OFF
Inactive*1	OFF	ON

*1 Including the cases where no corresponding step exists in an SFC program.

- The following table summarizes specification methods of steps.

Program		Specification method
SFC program	Specifying a step in current block	Use S□.
	Specifying a step in another block	Use BL□\S□.
Sequence program		• Use BL□\S□. • If the block No. is not specified, the target block will be the block 0.

- If the block No. or the step No. specified is out of range, both of the normally open contact and normally closed contact turn off.
- If the instruction is executed while no SFC program exists (SM320 (Presence/absence of SFC program) is off) or SM321 (Start/stop SFC program) is off and both of the specified block No. and step No. are within the range, the normally open contact instruction turns off and the normally closed contact instruction turns on.

Operation error

Error code (SD0/SD8067)	Description
2820H	When a block No. is specified, the specified block No. is out of the range of 0 to 31.
	When a block No. is specified, the specified step No. is out of the range of 0 to 511.
	When a block No. is specified, the step relay No. assigned the specified step is out of the range of 0 to 4095.
	When a block No. is not specified, the specified step No. is out of the range of 0 to 4095.
3582H	The SFC control instruction is used in the interrupt routine program.

Checking the status of a block

LD, LDI, AND, ANI, OR, ORI [BL☐]

FX5S FX5UJ FX5U FX5UC

• LD: Normally open contact, LDI: Normally closed contact

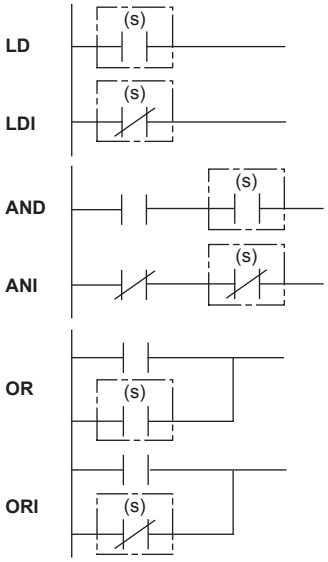
These instructions output the status (active or inactive) of the specified block as the operation result.

• AND: Normally open contact series connection, ANI: Normally closed contact series connection

These instructions perform an AND operation between the status (active or inactive) of the specified block and the previous operation result(s), and output the operation result.

• OR: Single normally open contact parallel connection, ORI: Single normally closed contact parallel connection

These instructions perform an OR operation between the status (active or inactive) of the specified block and the previous operation result(s), and output the operation result.

Ladder diagram	Structured text
	Not supported.

FBD/LD
Not supported.

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Device used as a contact	—	Bit	ANY_BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL☐)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U☐\G☐	Z	LC	LZ		K, H	E	$	
(s)	—	—	—	—	—	—	—	—	—	—	○

Processing details

- These instructions check whether the specified block is active or not.
- The status (on or off) of each contact will be as follows depending on the status (active or inactive) of the specified block.

Status of the specified block	Contact of the normally open contact instruction	Contact of the normally closed contact instruction
Active	ON	OFF
Inactive*1	OFF	ON

*1 Including the cases where no corresponding block exists in an SFC program.

- If the block No. specified is out of range, both of the normally open contact and normally closed contact turn off.
- If the instruction is executed while no SFC program exists (SM320 (Presence/absence of SFC program) is off) or SM321 (Start/stop SFC program) is off and the specified block No. is within the range, the normally open contact instruction turns off and the normally closed contact instruction turns on.

Operation error

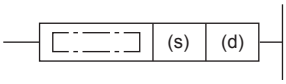
Error code (SD0/SD8067)	Description
2820H	When the specified block No. is out of the range of 0 to 31.
3582H	The SFC control instruction is used in the interrupt routine program.

Batch-reading the status of steps

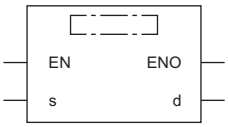
MOV(P) [KnS□/BL□\KnS□]

FX5S FX5UJ FX5U FX5UC

These instructions batch-read (in units of 16-bit binary data) the status (active or inactive) of steps in the specified block, and store the read data in the specified device. (Kn: K1 to K4)

Ladder diagram	Structured text
	ENO:=MOV(EN,s,d); ENO:=MOVP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Device number where the transfer source data is stored	—	16-bit signed binary	ANY16
(d)	Transfer destination device number	—	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□\S□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○*1	—	—	—	—	—	—	—	—	—	○
(d)	○*2	○	○	○	—	—	○	—	—	—	—

*1 Only S can be used.
*2 Other than S can be used.

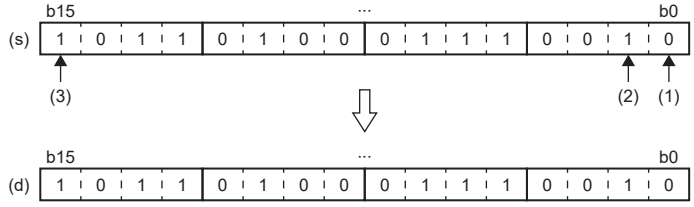
Processing details

- These instructions batch-read (in units of 16-bit binary data) the status (active or inactive) of steps in the specified block. K1 to K4 can be specified to the nibble specification of the step relay (S).
- When a block is not specified, the status (active or inactive) of the following block is read.

- Sequence program: Block 0
- SFC program (within the action): Block where the instruction is executed (current block)

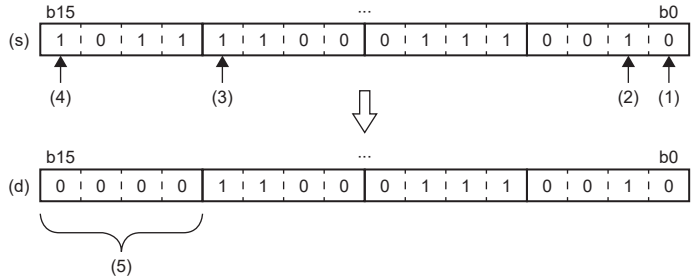
- The read data are stored in the device specified by (d). When the step is active, 1 is stored. When the step is inactive, 0 is stored.

[Example: when K4 is specified to the nibble specification]



(1) Device specified by S☐
(2) S☐+1
(3) S☐+15

[Example: when K3 is specified to the nibble specification]

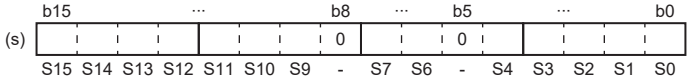


(1) The device specified by S☐
(2) S☐+1
(3) S☐+11
(4) S☐+15
(5) 0 is stored.

- When there is a missing step No., 0 is stored in the corresponding bit.

Ex.

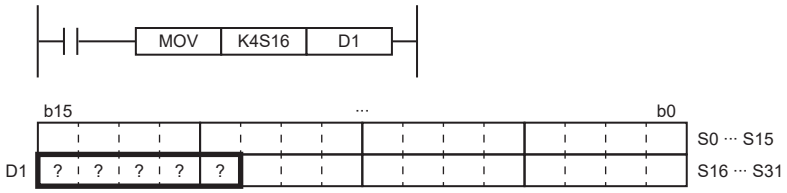
When the step No.5 and No.8 are missing in the specified block (The status of each step is stored in other bits.)



- If no block is specified, when the read target range exceeds the maximum step No. in the block, undefined values will be stored.

Ex.

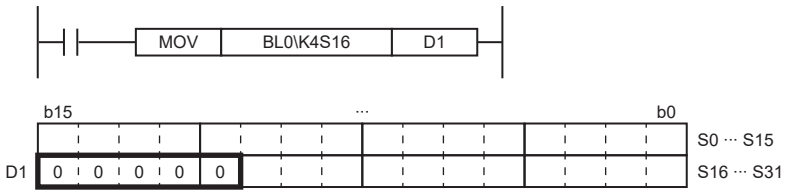
When the last step No. in the specified block is S26 and the status of steps (S16 to S31) are read to D1. (The status of each step is stored in other bits.)



- If the read target range exceeds the number of steps in the specified block, 0 is stored in the bits exceeding the existing step No.

Ex.

When the last step No. in the specified block is S26 and the status of steps (S16 to S31) are read to D1. (The status of each step is stored in other bits.)



- If the block 0 does not exist, when the instruction is executed in the sequence program, undefined values are stored in all bits.
- In the case of "with block specification," if a non-existent block is specified, the bit data to be read will be 0.
- If a non-existent step is at the beginning of the data to be read, the bit data to be read will be 0.
- If readout is performed in the absence of the SFC program, the following operation will be performed depending on the presence/absence of the SFC program setting and block setting in the CPU parameters.

SFC program setting	Block specification	Instruction operation
Not use	None	The device value of the step relay will be transferred to (d).
	Present	The bit data to be read becomes "0".
Use	None	The bit data to be read becomes "0".
	Present	The bit data to be read becomes "0".

Operation error

Error code (SD0/SD8067)	Description
2820H	When a block No. is specified, the specified block No. is out of the range of 0 to 31.
	When a block No. is specified, the specified step No. is out of the range of 0 to 511.
	When the step relay No. where the specified step is assigned is out of the range of 0 to 4095.
	When a block No. is not specified, the specified step No. is out of the range of 0 to 4095.
2821H	When a step relay (s) is specified to (d).
3582H	The SFC control instruction is used in the interrupt routine program.

Point

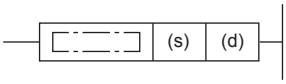
Use a digit specification to specify a step. (Kn: K1 to K4)

- To specify a step in the current block of an SFC program, use KnS☐.
- To specify a step in another block of an SFC program, use BL☐\KnS☐.
- To specify a step of a sequence program, use BL☐\KnS☐.

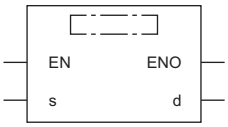
DMOV(P) [KnS□/BL□\KnS□]

FX5S FX5UJ FX5U FX5UC

These instructions batch-read (in units of 32-bit binary data) the status (active or inactive) of steps in the specified block, and store the read data in the specified device. (Kn: K1 to K8)

Ladder diagram	Structured text
	ENO:=DMOV(EN,s,d); ENO:=DMOVP(EN,s,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Device number where the transfer source data is stored	—	32-bit signed binary	ANY32
(d)	Transfer destination device number	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□\S□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○*1	—	—	—	—	—	—	—	—	—	○
(d)	○*2	○	○	○	○	○	○	—	—	—	—

*1 Only S can be used.
*2 Other than S can be used.

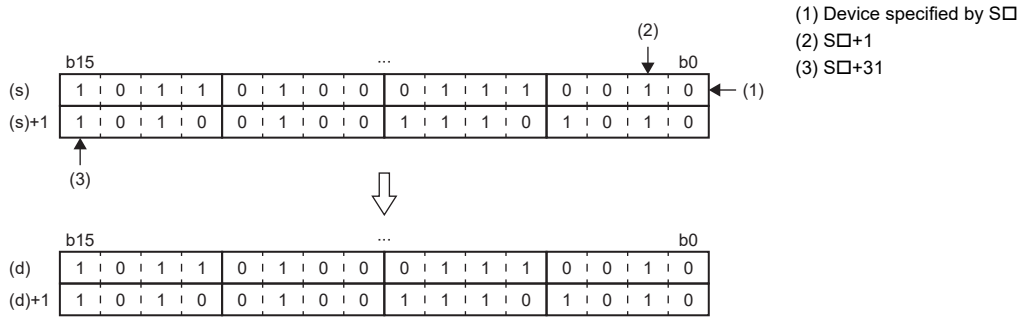
Processing details

- These instructions batch-read (in units of 32-bit binary data) the status (active or inactive) of steps in the specified block. K1 to K8 can be specified to the nibble specification of the step relay (S).
- When a block is not specified, the status (active or inactive) of the following block is read.

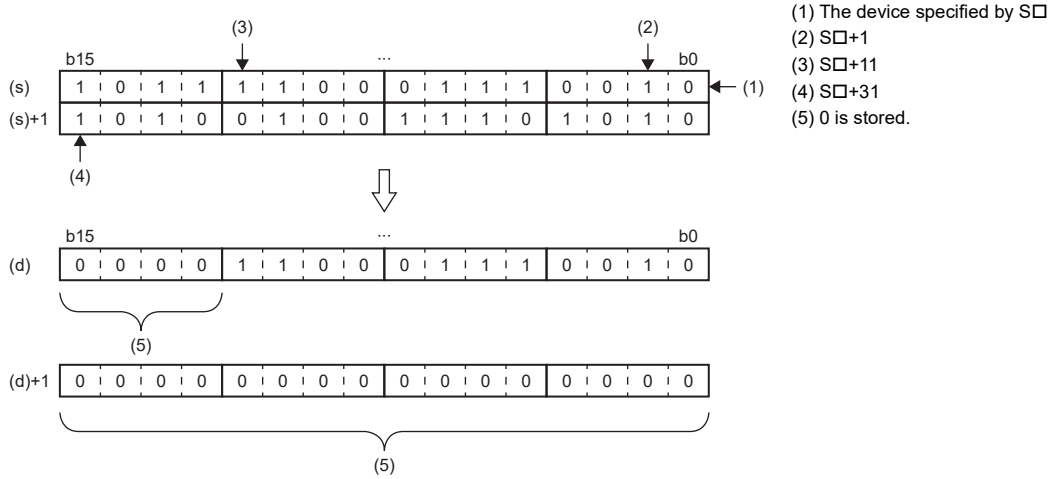
- Sequence program: Block 0
- SFC program (within the action): Block where the instruction is executed (current block)

- The read data are stored in the device specified by (d). When the step is active, 1 is stored. When the step is inactive, 0 is stored.

[Example: when K8 is specified to the nibble specification]



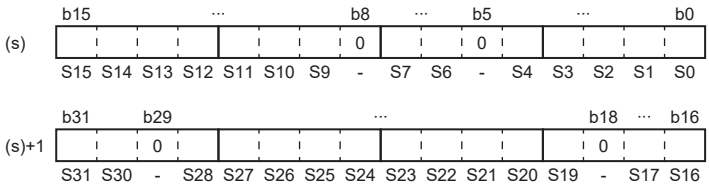
[Example: when K3 is specified to the nibble specification]



- When there is a missing step No., 0 is stored in the corresponding bit.

Ex.

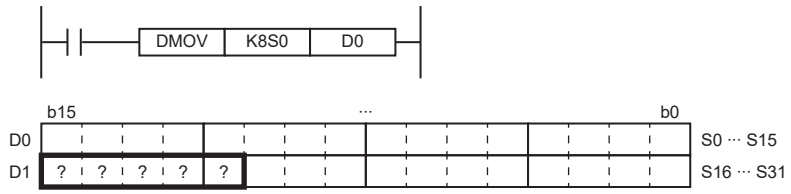
When the step No.5, 8, 18, and 29 are missing in the specified block. (The status of each step is stored in other bits.)



- If no block is specified, when the read target range exceeds the maximum step No. in the block, undefined values will be stored.

Ex.

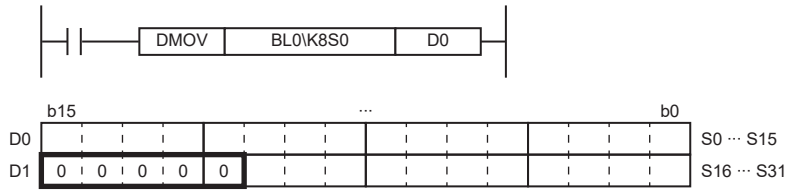
When the last step No. in the specified block is S26 and the status of steps (S0 to S31) are read to D0 and D1. (The status of each step is stored in other bits.)



- If the read target range exceeds the number of steps in the specified block, 0 is stored in the bits exceeding the existing step No.

Ex.

When the last step No. in the specified block is S26 and the status of steps (S0 to S31) are read to D0 and D1. (The status of each step is stored in other bits.)



- If block 0 does not exist, when the instruction is executed in the sequence program, undefined values are stored in all bits.
- In the case of "with block specification," if a non-existent block is specified, the bit data to be read will be 0.
- If a non-existent step is at the beginning of the data to be read, the bit data to be read will be 0.
- If readout is performed in the absence of the SFC program, the following operation will be performed depending on the presence/absence of the SFC program setting and block setting in the CPU parameters.

SFC program setting	Block specification	Instruction operation
Not use	None	The device value of the step relay will be transferred to (d).
	Present	The bit data to be read becomes "0".
Use	None	The bit data to be read becomes "0".
	Present	The bit data to be read becomes "0".

Operation error

Error code (SD0/SD8067)	Description
2820H	When a block No. is specified, the specified block No. is out of the range of 0 to 31.
	When a block No. is specified, the specified step No. is out of the range of 0 to 511.
	When the step relay No. where the specified step is assigned is out of the range of 0 to 4095.
	When a block No. is not specified, the specified step No. is out of the range of 0 to 4095.
2821H	When a step relay (s) is specified to (d).
3582H	The SFC control instruction is used in the interrupt routine program.

Point

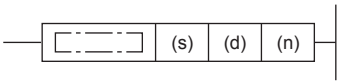
Use a digit specification to specify a step. (Kn: K1 to K8)

- To specify a step in the current block of an SFC program, use KnS☐.
- To specify a step in another block of an SFC program, use BL☐\KnS☐.
- To specify a step of a sequence program, use BL☐\KnS☐.

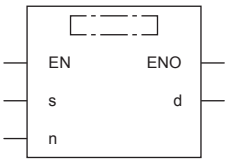
BMOV(P) [KnS□/BL□\KnS□]

FX5S FX5UJ FX5U FX5UC

These instructions batch-read (in units of the specified number of words starting from the specified step) the status (active or inactive) of steps in the specified block. (Kn: K1 to K4)

Ladder diagram	Structured text
	ENO:=BMOV(EN,s,n,d); ENO:=BMOVP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Start number of a device where the transfer target data is stored	—	16-bit signed binary	ANY16
(d)	Transfer destination device start number	—	16-bit signed binary	ANY16
(n)	Number of transfer data points	0 to 65535	16-bit unsigned binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□\S□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○*1	—	—	—	—	—	—	—	—	—	○
(d)	○*2	○	○	—	○	—	○	—	—	—	—
(n)	○*2	○	○	○	—	—	○	○	—	—	—

*1 Only S can be used.
*2 Other than S can be used.

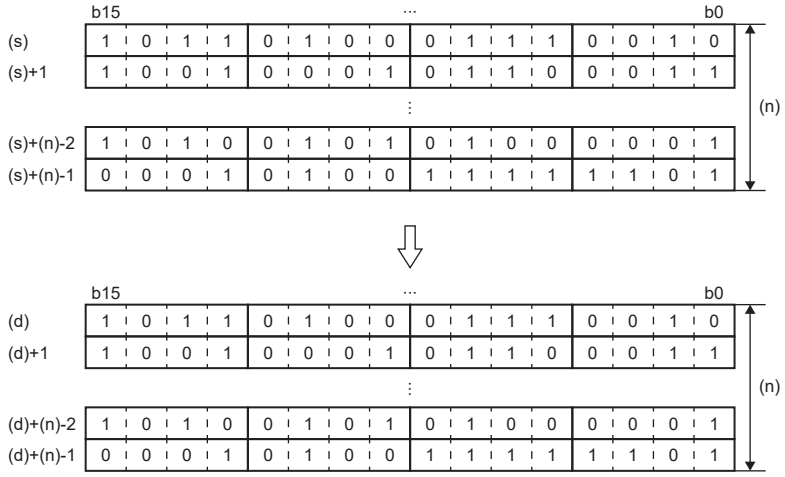
Processing details

- These instructions batch-read (in units of the specified number of words starting from the specified step) the status (active or inactive) of steps in the specified block. K1 to K4 can be specified to the nibble specification of the step relay (S).
- When a block is not specified, the status (active or inactive) of the following block is read.

- Sequence program: Block 0
- SFC program (within the action): Block where the instruction is executed (current block)

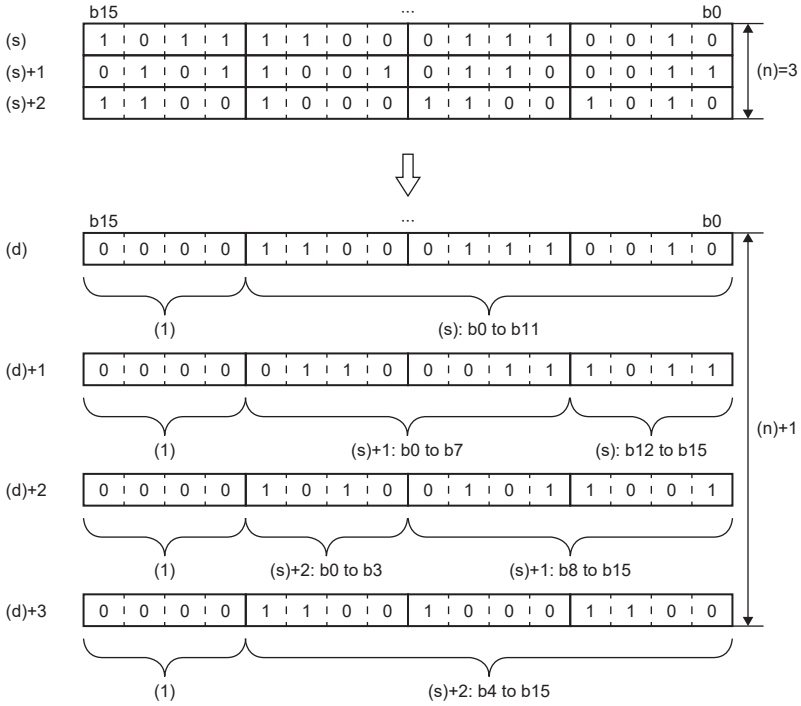
- The read data are stored in the device specified by (d).

[Example: when K4 is specified to the nibble specification]



[Example: when K3 (the numbers of transfers: 03) is specified to the nibble specification]

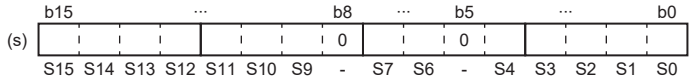
(1) 0 is stored.



• When there is a missing step No., 0 is stored in the corresponding bit.

Ex.

When the step No.5 and No.8 are missing in the specified block (The status of each step is stored in other bits.)

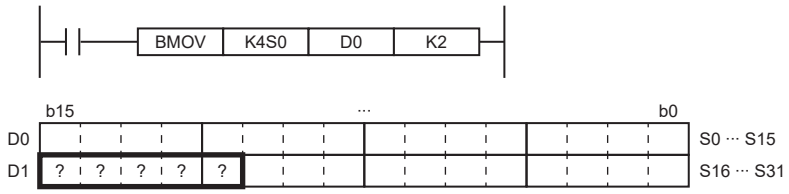


• In the case of "without block specification," if the read target range is as shown below, undefined data will be stored.

- When the maximum step No. in the block is exceeded
- If block 0 does not exist, when the instruction is executed in the sequence program

Ex.

When the last step No. in the block is 26 and the status of the steps (two words from S0) is read to D0 and D1. (The status of each step is stored in other bits.)



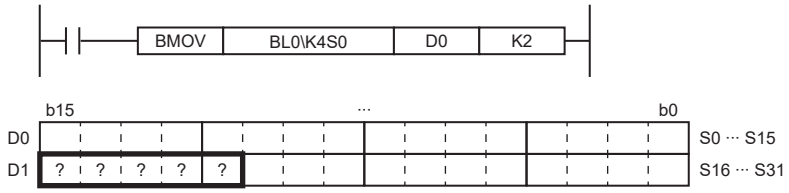
• In the case of "with block specification," if the read target range is as shown below, undefined data will be stored.

- When the maximum step No. in the block is exceeded
- When a non-existent step is at the beginning of the read target range

• If a block is specified and block 0 does not exist, when the instruction that does not specify a block is executed in the sequence program, 0 is stored.

Ex.

When the last step No. in the block is 26 and the status of the steps (two words from S0) is read to D0 and D1. (The status of each step is stored in other bits.)



• If a block is specified, when the block that does not exist is specified, 0 is read and stored in all bits.
• If no block is specified, when the read target range exceeds the maximum step No. (S4095), the data from the first step No. to the maximum step No. will be read.
• When the read target range exceeds the maximum device No. (such as D7999), the data from the first device No. to the maximum device No. are transferred.
• When the first device is out of the range of the device No., an error occurs and data is not read.
• If readout is performed in the absence of the SFC program, the following operation will be performed depending on the presence/absence of the SFC program setting and block setting in the CPU parameters.

SFC program setting	Block specification	Instruction operation
Not use	None	The device value of the step relay will be transferred to (d).
	Present	The bit data to be read becomes "0".
Use	None	The bit data to be read becomes "0".
	Present	The bit data to be read becomes "0".

Operation error

Error code (SD0/SD8067)	Description
2820H	When a block No. is specified, the specified block No. is out of the range of 0 to 31.
	When a block No. is specified, the specified step No. is out of the range of 0 to 511.
	When the step relay No. where the specified step is assigned is out of the range of 0 to 4095.
	When a block No. is not specified, the specified step No. is out of the range of 0 to 4095.
	When the first device of (d) is out of the range of the device No.
2821H	When a step relay (s) is specified to (d) and (n)
3405H	The number of nibbles of the nibble specification of bit device (s) and (d) is different.
3582H	The SFC control instruction is used in the interrupt routine program.

Point

Use a digit specification to specify a step. (Kn: K1 to K4)

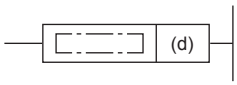
- To specify a step in the current block of an SFC program, use KnS☐.
- To specify a step in another block of an SFC program, use BL☐\KnS☐.
- To specify a step of a sequence program, use BL☐\KnS☐.

Starting a block

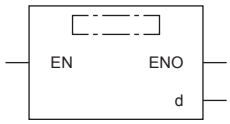
SET [BL□]

FX5S FX5UJ FX5U FX5UC

This instruction activates the specified block, and executes a step sequence starting from an initial step.

Ladder diagram	Structured text
	ENO:=SET(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Block No. to be activated (Target bit device number to be set (on))	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	—	—	—	—	—	—	—	—	—	○

Processing details

- This instruction activates the specified block, and executes a step sequence starting from an initial step.
- If the instruction is executed to an active block, the instruction is ignored and processing will continue.

Operation error

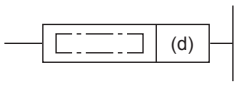
Error code (SD0/SD8067)	Description
31A1H	The specified block does not exist.
	The instruction is executed when no SFC program (scan execution type) exists or SM321 is off.
31A2H	When the specified block No. is out of the range of 0 to 31.
3582H	The SFC control instruction is used in the interrupt routine program.

Ending a block

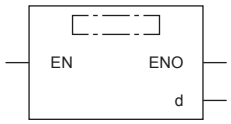
RST [BL□]

FX5S FX5UJ FX5U FX5UC

This instruction deactivates the specified block.

Ladder diagram	Structured text
	ENO:=RST(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Block No. to be deactivated (Target bit device number to be reset (off))	—	Bit	ANY_ELEMENTARY
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K,H	E	$	
(d)	—	—	—	—	—	—	—	—	—	—	○

Processing details

- This instruction deactivates the specified block.
- All the active steps are deactivated and coil outputs are turned off.
- If the instruction is executed to an inactive block, the instruction is ignored and processing will continue.

Operation error

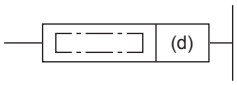
Error code (SD0/SD8067)	Description
31A1H	The specified block does not exist.
	The instruction is executed when no SFC program (scan execution type) exists or SM321 is off.
31A2H	When the specified block No. is out of the range of 0 to 31.
3582H	The SFC control instruction is used in the interrupt routine program.

Pausing a block

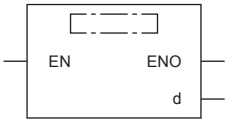
PAUSE [BL□]

FX5S FX5UJ FX5U FX5UC

This instruction temporarily stops the step sequence in the specified block.

Ladder diagram	Structured text
	ENO:=PAUSE(EN,d);

FBD/LD



■Execution condition

Instruction	Execution condition
PAUSE	

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Block No. where the sequence is temporarily stopped	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	—	—	—	—	—	—	—	—	—	○

Processing details

- This instruction temporarily stops the step sequence in the specified block.
- All the steps become inactive immediately after execution of the instruction.
- The coil output status caused by the OUT instruction at the temporary stop depends on the status of SM325 (output mode at pausing a block).

Status of SM325	Coil output status by using the OUT instruction
On	During the stop, the coil output remains the state immediately before stop.
Off	The coil output turns off at stop.

Precautions

If the sequence is stopped while SM325 is off, coil HOLD steps become inactive. The sequence cannot be restarted with the hold status. If the sequence is stopped while SM325 is on, the sequence can be restarted with the hold status.

Operation error

Error code (SD0)	Description
31A1H	The specified block does not exist.
	The instruction is executed when no SFC program (scan execution type) exists or SM321 (start/stop SFC program) is off.
31A2H	The specified block No. is out of the range.
3582H	The SFC control instruction is used in the interrupt routine program.

Point

Operation of the PAUSE and RESTART instructions depends on the combination of the SM325 status and step hold status. For details, refer to the following.

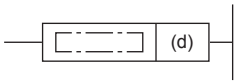
MELSEC iQ-F FX5 Programming Manual (Program Design)

Restarting a block

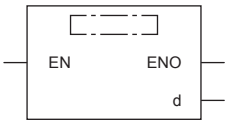
RSTART [BL□]

FX5S FX5UJ FX5U FX5UC

This instruction releases the temporary stop, and restarts the sequence from the step where the sequence was stopped in the specified block.

Ladder diagram	Structured text
	ENO:=RSTART(EN, d);

FBD/LD



■Execution condition

Instruction	Execution condition
RSTART	

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Block No. where the temporary stop is released	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	—	—	—	—	—	—	—	—	—	○

Processing details

- This instruction restarts the sequence from the step where the sequence was stopped in the specified block. Operation HOLD steps (with or without transition check), which have been stopped with operation hold status, restart operation with the hold status.
- After the step sequence is restarted, the operation of the PLS instruction and the instructions which are executed only on the rising edge depends on the status of SM325 (output mode at pausing a block).

Status of SM325	Operation of the PLS instruction and the instructions executed on the rising edge
On (Coil output is held.)	The instruction is not executed.
Off (Coil output is off.)	The instruction is executed again.

Precautions

If the sequence is stopped while SM325 is off, coil HOLD steps become inactive. The sequence cannot be restarted with the hold status. If the sequence is stopped while SM325 is on, the sequence can be restarted with the hold status.

Operation error

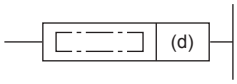
Error code (SD0)	Description
31A1H	The specified block does not exist.
	The instruction is executed when no SFC program (scan execution type) exists or SM321 (Start/stop SFC program) is off.
31A2H	The specified block No. is out of the range.
3582H	The SFC control instruction is used in the interrupt routine program.

Activating a step

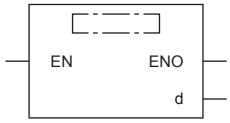
SET [S□/BL□\S□]

FX5S FX5UJ FX5U FX5UC

This instruction activates the specified step.

Ladder diagram	Structured text
	ENO:=SET(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Step No. to be activated (Target bit device number to be set (on))	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□\S□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1*2	—	—	—	—	—	—	—	—	—	○*2

*1 Only S can be used.
*2 Indexing is not available.

Processing details

- This instruction activates the specified step in the specified block. The operation of the specified block will be as follows depending on the status (active or inactive) of the specified block.

Status of the specified block	Operation
Inactive	The specified block is activated at execution of the instruction, and starts the processing from the specified step.
Active	The specified step is newly activated while the step which has already been active continues processing.

- If the instruction is executed to an active step, the instruction is ignored and processing will continue. Note that if the specified step is holding the operating status, the hold status is reset. The step becomes a normal step, and executes the action and transition.
- If no block is specified, the following block is targeted depending on the execution program type.

- Sequence program: Block 0
- SFC program (within the action): Block where the instruction is executed (current block)

Precautions

- In a simultaneous sequence, specify and activate all the steps by using the SET instruction (Activating a step). If there is any inactive step left, a convergence is not performed. Likewise, if the RST instruction (Deactivating a step) is executed to a single step in a simultaneous sequence, a convergence condition is not satisfied.
- The own step cannot be specified as the step No. within the action of the SFC program. If specified, an error occurs.

Operation error

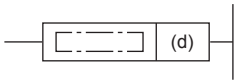
Error code (SD0/SD8067)	Description
31A1H	The specified block does not exist.
	The control target block does not exist. (SET/RST/OUT S□)
31A2H	When a block No. is specified, the specified block No. is out of the range of 0 to 31.
31B1H	The instruction is executed when no SFC program (Start/stop SFC program) exists or SM321 is off.
	No specified step exists in the specified block. (SET/RST/OUT BL□\S□/S□)
31B2H	The specified step No. is out of the range (0 to 511).
31B5H	The current step is specified within the action.
3582H	The SFC control instruction is used in the interrupt routine program.

Deactivating a step

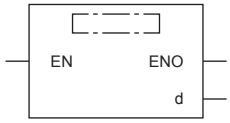
RST [S□/BL□\S□]

FX5S FX5UJ FX5U FX5UC

This instruction deactivates the specified step.

Ladder diagram	Structured text
	ENO:=RST(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Step No. to be deactivated (Target bit device number to be reset (off))	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□\S□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1*2	—	—	—	—	—	—	—	—	—	○*2

*1 Only S can be used.
*2 Indexing is not available.

Processing details

- This instruction deactivates the specified step in the specified block.
- When the number of active steps in the specified becomes 0, the specified block will be inactive.
- If the instruction is executed to an inactive step, the instruction is ignored and processing will continue.
- If no block is specified, the following block will be deactivated.

- Sequence program: Block 0
- SFC program (within the action): Block where the instruction is executed (current block)

Precautions

- If the step termination instruction is executed for a step in a parallel branch, the parallel convergence condition will not be met.
- The own step cannot be specified as the step No. within the action of the SFC program. If the own step No. is specified as the specified step No., an error will occur.

Operation error

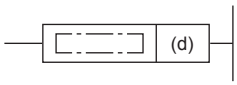
Error code (SD0/SD8067)	Description
31A1H	The specified block does not exist.
	The control target block does not exist. (SET/RST/OUT S□)
31A2H	When a block No. is specified, the specified block No. is out of the range of 0 to 31.
31B1H	The instruction is executed when no SFC program (Start/stop SFC program) exists or SM321 is off.
	No specified step exists in the specified block. (SET/RST/OUT BL□\S□/S□)
31B2H	The specified step No. is out of the range (0 to 511).
31B5H	The current step is specified within the action.
3582H	The SFC control instruction is used in the interrupt routine program.

Activating/deactivating a step

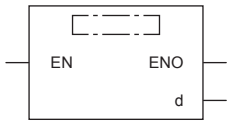
OUT [S□/BL□\S□]

FX5S FX5UJ FX5U FX5UC

This instruction activates or deactivates the specified step.

Ladder diagram	Structured text
	ENO:=OUT(EN,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Specified step No. to be activated or deactivated (Bit device No. to be set (on)/reset (off))	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□\S□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1*2	—	—	—	—	—	—	—	—	—	○*2

*1 Only S can be used.
*2 Indexing is not available.

Processing details

- When the drive contact is on, this instruction activates the specified step in the specified block. Same as the operation by the SET instruction (SET S□/BL□\S□). (☞ Page 976 Activating a step)
- When the drive contact is off, this instruction deactivates the specified step in the specified block. That is the same operation executed by the RST instruction (RST S□/BL□\S□). (☞ Page 978 Deactivating a step) (However, when the instruction is executed within the action, the instruction is ignored. That is the different operation executed by the RST instruction.)

Operation error

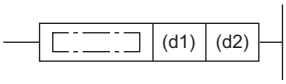
Error code (SD0/SD8067)	Description
31A1H	The specified block does not exist.
	The control target block does not exist. (SET/RST/OUT S□)
31A2H	When a block No. is specified, the specified block No. is out of the range of 0 to 31.
31B1H	The instruction is executed when no SFC program (Start/stop SFC program) exists or SM321 is off.
	No specified step exists in the specified block. (SET/RST/OUT BL□\S□/S□)
31B2H	The specified step No. is out of the range (0 to 511).
31B5H	The current step is specified within the action.
3582H	The SFC control instruction is used in the interrupt routine program.

Batch-deactivating a step

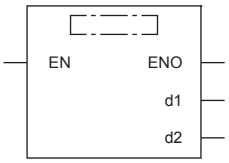
ZRST(P) [S□/BL□\S□]

FX5S FX5UJ FX5U FX5UC

Batch-deactivates the steps in the specified range.

Ladder diagram	Structured text
	ENO:=ZRST(EN,d1,d2); ENO:=ZRSTP(EN,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d1)	A first step No. to be batch-deactivated	—	Bit	ANY_BOOL
(d2)	A last step No. to be batch-deactivated	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (BL□\S□)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d1)	○*1	—	—	—	—	—	—	—	—	—	○
(d2)	○*1	—	—	—	—	—	—	—	—	—	○

*1 Only S can be used.

Processing details

- This instruction batch-deactivates the activated steps within the range specified by (d1) and (d2). When the instruction is executed and the number of activated steps in the corresponding block becomes 0, the block will be deactivated.
- Unify the device specified to (d1) or (d2) into S□ or BL□\S□. When devices are not unified (Example: [ZRST S2 BL0\S5]), an operation error occurs and the status of the step is not changed.
- If no block is specified, make sure to specify the step Nos. so as to be (d1) < (d2). When (d1) ≥ (d2), only one step No. that is specified to (d1) will be deactivated. When a step that does not exist is specified to (d1) or (d2), the steps that exist in the specified range will be deactivated.

- When a block is specified, specification is possible over blocks. Make sure to specify the block Nos. so as to be (d1) < (d2). When (d1) ≥ (d2), only one step No. that is specified to (d1) will be deactivated. When a block that does not exist is specified to (d1) or (d2), the steps that exist in the specified range will be deactivated.

Ex.

When executing [ZRST BL1\S20 BL5\S10] to SFC programs having the following configuration.

- Block 1, 3, 5: No block exists.
- Block 2: S0 to S5 exist.
- Block 4: S0 to S3 are exist.

Block No.	Step No.	Deactivating target
1 (No target)	21 to 511 (No target)	Excluded from target
2	0 to 5	Deactivates
	6 to 511 (No target)	Excluded from target
3 (No target)	0 to 511 (No target)	Excluded from target
4	0 to 3	Deactivates
	4 to 511 (No target)	Excluded from target
5 (No target)	0 to 10 (No target)	Excluded from target

- If the current step is specified so as to be in the deactivating target range within the action, the step will not be deactivated.
- If no block is specified, the following specified block step will be deactivated. Specification over blocks is not possible.

- When the instruction is executed in a sequence program: Block 0
- When the instruction is executed in an SFC program (within the action): Block where the instruction is executed (current block)

Ex.

When executing [ZRST S0 S511] in an within the action of block1 to SFC programs having the following configuration.

- Block 1: S0 to S255 exist.
- Block 2: S0 to S255 exist.

Block No.	Step No.	Deactivating target
1	0 to 255	Deactivates
	256 to 511 (Not target)	Excluded from target
2	0 to 255	Excluded from target
	256 to 511 (No target)	Excluded from target

- When one of the following conditions are satisfied, the instruction will be ignored.

- No steps in the specified range exist.
- If a block is specified, when no SFC program exists.
- If an SFC program exists, the instruction is executed when SM321 (Start/stop SFC program) is off.

Operation error

Error code (SD0/SD8067)	Description
2820H	When a block No. is specified, the specified block No. of (d1) or (d2) is out of the range of 0 to 31.
	When a block No. is specified, the specified step No. of (d1) or (d2) is out of the range of 0 to 511.
	When a block No. is not specified, the specified step No. of (d1) or (d2) is out of the range of 0 to 4095.
3405H	When S□ and BL□\S□ are mixed in the specified devices to (d1) and (d2).
3582H	The SFC control instruction is used in the interrupt routine program.

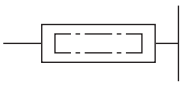
11.2 SFC Dedicated Instruction

Creating a dummy transition condition

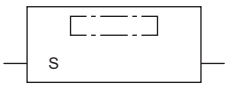
TRAN

FX5S FX5UJ FX5U FX5UC

This instruction is a dummy output which satisfies a transition condition.

Ladder diagram	Structured text
	TRAN(s);

FBD/LD



Point

For details on transitions, refer to the following.

MELSEC iQ-F FX5 Programming Manual (Program Design)

PART 4 MODULE DEDICATED INSTRUCTION

This part consists of the following chapters.

12 NETWORK COMMON INSTRUCTION

13 ETHERNET INSTRUCTION

14 CC-LINK IE TSN INSTRUCTION

15 CC-LINK IE FIELD NETWORK INSTRUCTION

16 HIGH-SPEED COUNTER INSTRUCTION

17 EXTERNAL DEVICE COMMUNICATION INSTRUCTION

18 POSITIONING INSTRUCTION

19 DIVIDED DATA READ/WRITE FROM/TO BFM INSTRUCTION

12 NETWORK COMMON INSTRUCTION

Target networks and target station types

Each link dedicated instruction allows access to a station on a network other than the own station network.

The following table lists the target stations of each instruction.

Instruction symbol	Target station (another station)	
	Target network	Target station type
GP.READ GP.SREAD	• Ethernet • CC-Link IE TSN • CC-Link IE Controller Network • CC-Link IE Field Network • MELSECNET/H • MELSECNET/10	• RCPU • QCPU • LCPU • QSCPU • QnACPU • FX5CPU • Intelligent device station
GP.WRITE GP.SWRITE	• Ethernet • CC-Link IE TSN • CC-Link IE Controller Network • CC-Link IE Field Network • MELSECNET/H • MELSECNET/10	• RCPU • QCPU • LCPU • QnACPU • FX5CPU • Intelligent device station
GP.SEND GP.RECV	• Ethernet • CC-Link IE TSN • CC-Link IE Controller Network • CC-Link IE Field Network • MELSECNET/H • MELSECNET/10	• RCPU • QCPU • LCPU • QnACPU • FX5CPU • Interface board for personal computer*1

*1 These instructions can access the following personal computer interface boards having the SEND/RECV functions.
CC-Link IE Field Network interface board, CC-Link IE Controller Network interface board, MELSECNET/H interface board, and MELSECNET/10 interface board

Specifications of character string data specified by link dedicated instructions

Some operands specified in link dedicated instructions are specified in character strings. The following table summarizes the specifications of character string data specified.

Instruction symbol	Applicable operand	Specifications of character string data
GP.READ	(s2)	■Specification method Specify a character string by enclosing it in single quotes ('). ■Specifiable device types • Word device: D, W, SW, SD, T, C, ST, ZR, R • Digit-specified bit device: X, Y, M, L, B, F, SB, SM ■ASCII string specifications • An ASCII string can consist of a maximum of 32 characters. (Specify NULL as the termination character.) • ASCII strings are zero-suppressed. (Example: D1 and D001 are handled the same.) • ASCII strings are case-sensitive. ■Precautions Digit-specified bit devices can be specified only when the following conditions are all satisfied. • The device number is a multiple of 16 (10H). • The number of specified nibbles is 4 (K4). An instruction specifying bit devices by nibble can be executed only when the target station is FX5CPU, RCPU, basic model QCPU, universal model QCPU, or LCPU. If an instruction uses digit specification for other types of CPU modules, an abnormal response is returned from the target station and the instruction is completed with an error. (The network module does not detect an error during execution of the instruction.) The target device of an instruction executed in index modification mode is determined by the value of the index register of the own station. When specifying X and Y devices, specify the device number in hexadecimal.
GP.SREAD	(s2)	
GP.WRITE	(d1)	
GP.SWRITE	(d1)	
GP.SREAD	(d3)	■Specification method Specify a character string by enclosing it in single quotes ('). ■Specifiable device types • Bit specification of word device: D, W, SW, SD, ZR, R • Bit device: X, Y, M, L, B, F, SB, SM • Bit device index modification: Z ■ASCII string specifications • An ASCII string can consist of a maximum of 32 characters. (Specify NULL as the termination character.) • ASCII strings are zero-suppressed. (Example: D1 and D001 are handled the same.) • ASCII strings are case-sensitive. ■Precautions The target device of an instruction executed in index modification mode is determined by the value of the index register of the own station. When specifying X and Y devices, specify the device number in hexadecimal.
GP.SWRITE	(d3)	

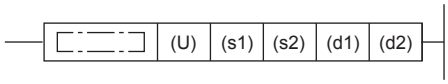
12.1 Link Dedicated Instructions

Reading data from another station programmable controller

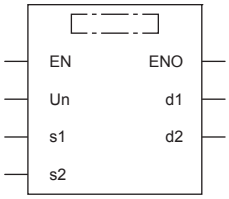
GP.READ

FX5S FX5UJ FX5U FX5UC

This instruction reads data from a device in another station programmable controller. (In units of words)
CC-Link IE TSN is compatible only with FX5U/FX5UC CPU module.

Ladder diagram	Structured text
	ENO:=GP_READ(EN,Un,s1,s2,d1,d2);

FBD/LD



("GP_READ" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)	Own station start device where control data is stored	☞ Page 989 Control dataRefer to	Device name	ANY16*4
(s2)	Target station head device where the data to be read is stored	—	Character string*2	ANYSTRING_SINGLE*2
(d1)	Own station head device for storing the read data*3	—	Device name	ANY16*4
(d2)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.
*2 For the specifications of the string data to be specified, refer to the following.
(☞ Page 987 Specifications of character string data specified by link dedicated instructions)
*3 The continuous area specified by the read data length (s1)+9 is required.
*4 Digit specified bit type label cannot be used.

■Applicable devices

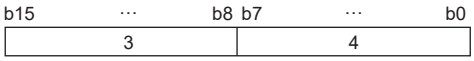
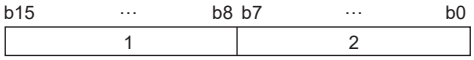
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	—	—	—	—	—	—	—	○	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(d2)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

*2 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s1)+0	Error completion type	b15 b14 ··· b9 b8 b7 b6 ··· b0 (3) 0 (2) (1) 0 (1) Error completion type (b7) Specify whether to set data at completion with an error. • 0: Do not set data in (s1)+11 and later at completion with an error. • 1: Set data in (s1)+11 and later at completion with an error. (2)Setting of arrival acknowledgment time (b8) (compatible only with CC-Link IE TSN) • 0: By second • 1: By 100 ms (3)Method for specifying the target station address (b15) (compatible only with CC-Link IE TSN) • 0: Specify the network No. in (s1)+4 and the station No. in (s1)+5. • 1: Specify the IP address in (s1)+4 and 5.	0000H 0080H 0100H 0180H 8000H 8080H 8100H 8180H	User
(s1)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System
(s1)+2	Own station channel	Specify the channel to be used by the own station. [CC-Link IE Field Network] • Channels 1 and 2 [CC-Link IE TSN] • Channels 1 to 8	1 to 8	User
(s1)+3	Target station CPU type	Specify the CPU type of the target station. • 0000H: Addressed to target station CPU (control CPU) • 03D0H: Addressed to control system CPU • 03D1H: Addressed to standby system CPU • 03D2H: Addressed to system A CPU • 03D3H: Addressed to system B CPU • 03E0H: Addressed to multiple CPU No. 1 • 03E1H: Addressed to multiple CPU No. 2 • 03E2H: Addressed to multiple CPU No. 3 • 03E3H: Addressed to multiple CPU No. 4 • 03FFH: Addressed to target station CPU (control CPU)	0000H 03D0H to 03D3H 03E0H to 03E3H 03FFH	User

Device	Item	Description	Setting range	Set by
(s1)+4	Target network number	[CC-Link IE Field Network] Specify the network No. of the target station. • 1 to 239 (network No.)	■(s1)+4 • Network No.: 1 to 239 ■(s1)+5 • Station No.: 1 to 120, 125 and 126 ■(s1)+4, 5 • IP address: 00000001H to FFFFFFFEH	User
		[CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 Specify the network No. of the target station. • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 Specify the IP address (third and fourth octets) of the target station.  • b8 to b15: Third octet • b0 to b7: Fourth octet		
(s1)+5	Target station number	[CC-Link IE Field Network] Specify the station number of the target station. • 125: Master station • 126: Master operating station • 1 to 120: Local station, intelligent device station or sub-master station		User
		[CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 Specify the station number of the target station. • 1 to 120: Local station • 125: Master station ■When "1" is specified in b15 in (s1)+0 Specify the IP address (first and second octets) of the target station.  • b8 to b15: First octet • b0 to b7: Second octet		
(s1)+6	Not used	—	—	—
(s1)+7	Number of resends	■At instruction execution Specify the number of resends to be performed if the instruction is not completed within the monitoring time specified by (s1)+8. ■At completion of instruction The number of resends performed (result) is stored.	0 to 15	User/ system
(s1)+8	Arrival monitoring time	Specify the monitoring time until completion of processing. If the processing is not completed within the monitoring time, the instruction will be resent the number of times specified in (s1)+7. ■When "0" is specified in b8 in (s1)+0 • 0: 10 seconds • 1 to 32767: 1 to 32767 seconds ■When "1" is specified in b8 in (s1)+0 • 0: 10000 ms (10 seconds) • 1 to 65535: 1 to 65535 × 100 ms	0 to 65535	User
(s1)+9	Read data length	Specify the number of words to be read. • 1 to 960 (words) (When reading data from QnACPU: 1 to 480 (words))	1 to 960	User
(s1)+10	Not used	—	—	—
(s1)+11	Clock setting flag	The validity status (valid or invalid) of the data in (s1)+12 and later is stored. Note that the data in (s1)+12 and later is not cleared even when the instruction is completed successfully. • 0: Invalid • 1: Valid	—	System
(s1)+12	Clock data (Set only in an abnormal state)	Upper 8 bits: Month (01H to 12H) Lower 8 bits: Year (00H to 99H: Lower two digits of the year)	—	System
(s1)+13		Upper 8 bits: Hour (00H to 23H) Lower 8 bits: Day (01H to 31H)	—	System
(s1)+14		Upper 8 bits: Second (00H to 59H) Lower 8 bits: Minute (00H to 59H)	—	System
(s1)+15		Upper 8 bits: Year (00H to 99H: Upper two digits of the year) Lower 8 bits: Day of the week (00H (Sun.) to 06H (Sat.))	—	System

Device	Item	Description	Setting range	Set by
(s1)+16	Error detection network number*1	■When "0" is specified in b15 in (s1)+0 The network number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (third and fourth octets) of the station where an error was detected is stored. b15 … b8 b7 … b0 3 | 4 • b8 to b15: Third octet • b0 to b7: Fourth octet	—	System
(s1)+17	Error-detected station number*1	■When "0" is specified in b15 in (s1)+0 The number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) [CC-Link IE Field Network] • 125: Master station • 1 to 120: Local station, intelligent device station or sub-master station [CC-Link IE TSN] • 125: Master station • 1 to 120: Device station ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (first and second octets) of the station where an error was detected is stored. b15 … b8 b7 … b0 1 | 2 • b8 to b15: First octet • b0 to b7: Second octet	—	System

*1 If completion status ((s1)+1) is "Channel in use (dedicated instruction) (error codes: D25AH to D25BH)", data is not stored.

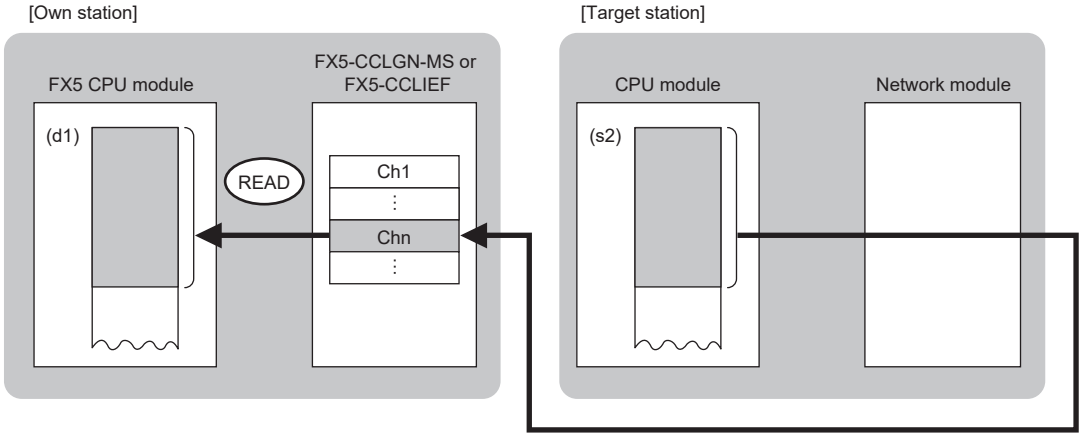
Point

- The continuous area (a maximum of 960 words) specified by the read data length ((s1)+9) is required in the read data storage device (d1).
- The number of resends (s1)+7 must be set every time the instruction is executed.

Processing details

- The instruction reads data by the specified number of words ((s1)+9) from the target station head device (s2) into the own station head device (after (d1)). Specify the target stations in control data ((s1)+4) and ((s1)+5). Upon completion of reading the device data specified by (s2), the completion device specified by (d2) turns on.
- For the target stations that can be specified, refer to the following.

☞ Page 986 Target networks and target station types



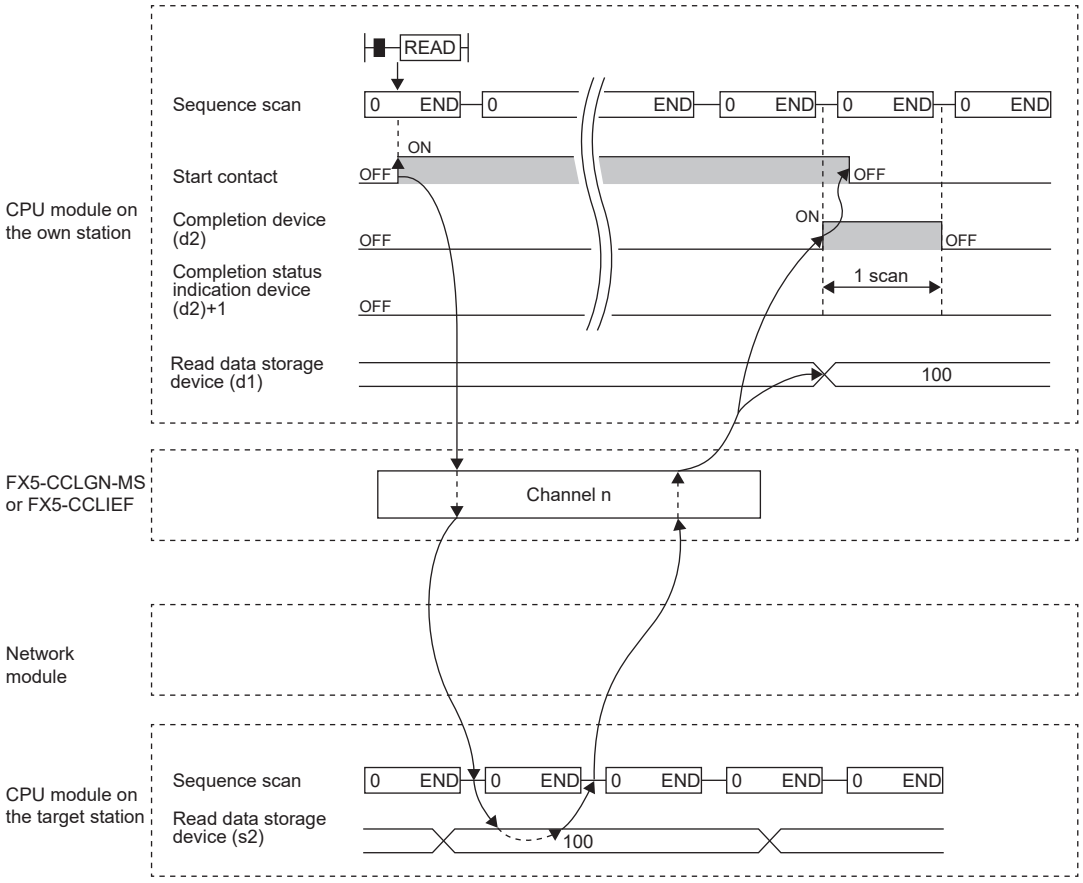
- When executing multiple link dedicated instructions concurrently, be careful not to overlap the channels of the link dedicated instructions. Multiple link dedicated instructions specifying the same channel cannot be used concurrently.

• The execution of the GP.READ instruction and whether it has been completed normally or completed with an error can be checked with the completion device (d2) or completion status indication device (d2)+1.

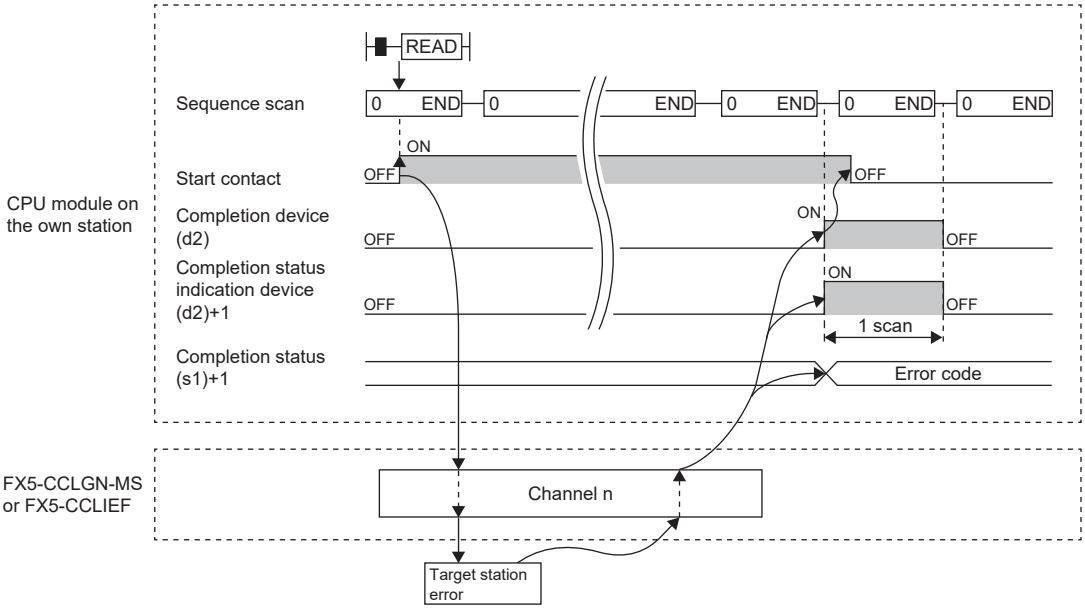
Device	Operation
Completion device (d2)	The device turns on during the END processing for the scan in which the GP.READ instruction is completed, and turns off during the next END processing.
Completion status indication device (d2)+1	The device turns on or off depending on the completion status of the GP.READ instruction. When completed normally: The device does not change (remains off). When completed with an error: The device turns on during the END processing for the scan in which the GP.READ instruction is completed, and turns off during the next END processing.

• The following figure shows the execution timing of the GP.READ instruction.

When completed normally



When completed with an error



• Read processing is performed only once on the rising edge when the read command turns on.

Operation error

Error code ((s1)+1)	Description
C000H to CFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual.
D000H to DFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual. Refer to MELSEC iQ-F FX5 CC-Link IE Field Network Module User's Manual.

Reading data from another station programmable controller (with notification)

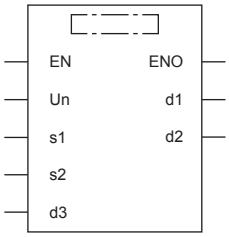
GP.SREAD

FX5S FX5UJ FX5U FX5UC

This instruction reads data from a device in another station programmable controller. (In units of words)
After the data reading is completed by the GP.SREAD instruction, the device of the other station is turned on. The other station can recognize that data has been read by the GP.SREAD instruction.
CC-Link IE TSN is compatible only with FX5U/FX5UC CPU module.

Ladder diagram	Structured text
	ENO:=GP_SREAD(EN,Un,s1,s2,d1,d2,d3);

FBD/LD



("GP_SREAD" enters ☐.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)	Own station start device where control data is stored	☞ Page 995 Control dataRefer to	Device name	ANY16*4
(s2)	Target station head device where the data to be read is stored	—	Character string*2	ANYSTRING_SINGLE*2
(d1)	Own station head device for storing the read data*3	—	Device name	ANY16*4
(d2)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
(d3)	Target station device to be turned on for one scan when the instruction completes (The target station can recognize that data has been read from another station.)	—	Character string*2	ANYSTRING_SINGLE*2
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.
*2 For the specifications of the string data to be specified, refer to the following.
(☞ Page 987 Specifications of character string data specified by link dedicated instructions)
*3 The continuous area specified by the read data length (s1)+9 is required.
*4 Digit specified bit type label cannot be used.

■Applicable devices

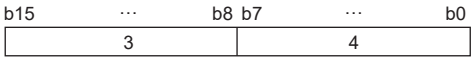
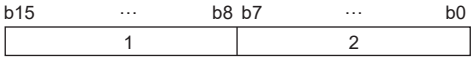
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	—	—	—	—	—	—	—	○	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(d2)	○*1	○*2	—	—	—	—	—	—	—	—	—
(d3)	—	—	—	—	—	—	—	—	—	○	—

*1 S cannot be used.

*2 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s1)+0	Error completion type	b15 b14 ··· b9 b8 b7 b6 ··· b0 (3) 0 (2) (1) 0 (1) Error completion type (b7) Specify whether to set data at completion with an error. • 0: Do not set data in (s1)+11 and later at completion with an error. • 1: Set data in (s1)+11 and later at completion with an error. (2)Setting of arrival acknowledgment time (b8) (compatible only with CC-Link IE TSN) • 0: By second • 1: By 100 ms (3)Method for specifying the target station address (b15) (compatible only with CC-Link IE TSN) • 0: Specify the network No. in (s1)+4 and the station No. in (s1)+5. • 1: Specify the IP address in (s1)+4 and 5.	0000H 0080H 0100H 0180H 8000H 8080H 8100H 8180H	User
(s1)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System
(s1)+2	Own station channel	Specify the channel to be used by the own station. [CC-Link IE Field Network] • Channels 1 and 2 [CC-Link IE TSN] • Channels 1 to 8	1 to 8	User
(s1)+3	Target station CPU type	Specify the CPU type of the target station. • 0000H: Addressed to target station CPU (control CPU) • 03D0H: Addressed to control system CPU • 03D1H: Addressed to standby system CPU • 03D2H: Addressed to system A CPU • 03D3H: Addressed to system B CPU • 03E0H: Addressed to multiple CPU No. 1 • 03E1H: Addressed to multiple CPU No. 2 • 03E2H: Addressed to multiple CPU No. 3 • 03E3H: Addressed to multiple CPU No. 4 • 03FFH: Addressed to target station CPU (control CPU)	0000H 03D0H to 03D3H 03E0H to 03E3H 03FFH	User

Device	Item	Description	Setting range	Set by
(s1)+4	Target network number	[CC-Link IE Field Network] Specify the network No. of the target station. • 1 to 239 (network No.)	■(s1)+4 • Network No.: 1 to 239 ■(s1)+5 • Station No.: 1 to 120, 125 and 126 ■(s1)+4, 5 • IP address: 00000001H to FFFFFFFEH	User
		[CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 Specify the network No. of the target station. • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 Specify the IP address (third and fourth octets) of the target station.  • b8 to b15: Third octet • b0 to b7: Fourth octet		
(s1)+5	Target station number	[CC-Link IE Field Network] Specify the station number of the target station. • 125: Master station • 126: Master operating station • 1 to 120: Local station, intelligent device station or sub-master station		User
		[CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 Specify the station number of the target station. • 1 to 120: Local station • 125: Master station ■When "1" is specified in b15 in (s1)+0 Specify the IP address (first and second octets) of the target station.  • b8 to b15: First octet • b0 to b7: Second octet		
(s1)+6	Not used	—	—	—
(s1)+7	Number of resends	■At instruction execution Specify the number of resends to be performed if the instruction is not completed within the monitoring time specified by (s1)+8. ■At completion of instruction The number of resends performed (result) is stored.	0 to 15	User/ system
(s1)+8	Arrival monitoring time	Specify the monitoring time until completion of processing. If the processing is not completed within the monitoring time, the instruction will be resent the number of times specified in (s1)+7. ■When "0" is specified in b8 in (s1)+0 • 0: 10 seconds • 1 to 32767: 1 to 32767 seconds ■When "1" is specified in b8 in (s1)+0 • 0: 10000 ms (10 seconds) • 1 to 65535: 1 to 65535 × 100 ms	0 to 65535	User
(s1)+9	Read data length	Specify the number of words to be read. • 1 to 960 (words) (When reading data from QnACPU: 1 to 480 (words))	1 to 960	User
(s1)+10	Not used	—	—	—
(s1)+11	Clock setting flag	The validity status (valid or invalid) of the data in (s1)+12 and later is stored. Note that the data in (s1)+12 and later is not cleared even when the instruction is completed successfully. • 0: Invalid • 1: Valid	—	System
(s1)+12	Clock data (Set only in an abnormal state)	Upper 8 bits: Month (01H to 12H) Lower 8 bits: Year (00H to 99H: Lower two digits of the year)	—	System
(s1)+13		Upper 8 bits: Hour (00H to 23H) Lower 8 bits: Day (01H to 31H)	—	System
(s1)+14		Upper 8 bits: Second (00H to 59H) Lower 8 bits: Minute (00H to 59H)	—	System
(s1)+15		Upper 8 bits: Year (00H to 99H: Upper two digits of the year) Lower 8 bits: Day of the week (00H (Sun.) to 06H (Sat.))	—	System

Device	Item	Description	Setting range	Set by
(s1)+16	Error detection network number*1	■When "0" is specified in b15 in (s1)+0 The network number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (third and fourth octets) of the station where an error was detected is stored. b15 … b8 b7 … b0 3 | 4 • b8 to b15: Third octet • b0 to b7: Fourth octet	—	System
(s1)+17	Error-detected station number*1	■When "0" is specified in b15 in (s1)+0 The number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) [CC-Link IE Field Network] • 125: Master station • 1 to 120: Local station, intelligent device station or sub-master station [CC-Link IE TSN] • 125: Master station • 1 to 120: Device station ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (first and second octets) of the station where an error was detected is stored. b15 … b8 b7 … b0 1 | 2 • b8 to b15: First octet • b0 to b7: Second octet	—	System

*1 If completion status ((s1)+1) is "Channel in use (dedicated instruction) (error codes: D25AH to D25BH)", data is not stored.

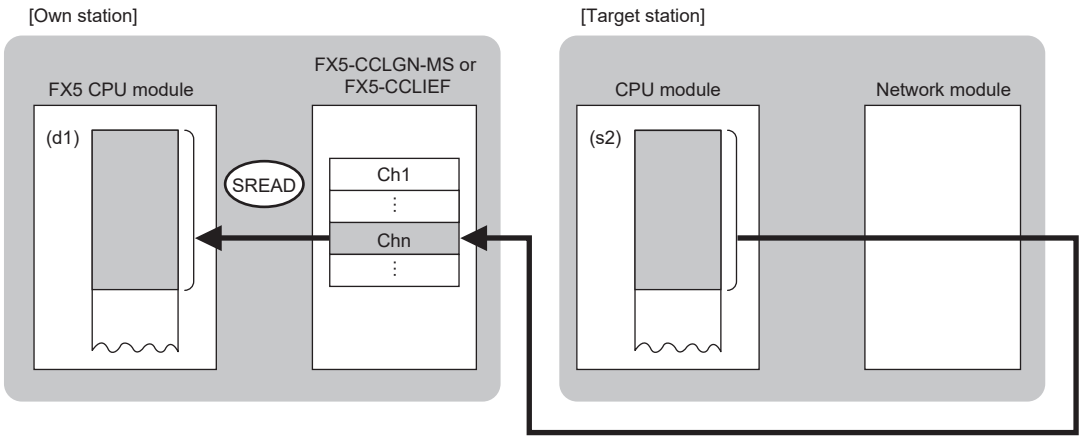
Point

- The continuous area (a maximum of 960 words) specified by the read data length ((s1)+9) is required in the read data storage device (d1).
- The number of resends (s1)+7 must be set every time the instruction is executed.

Processing details

- The instruction reads data by the specified number of words ((s1)+9) from the target station head device (s2) into the own station word device (after (d1)). Specify the target stations in control data ((s1)+4) and ((s1)+5). Upon completion of reading the device data specified by (s2), the completion device specified by (d2) turns on. In target station, upon completion of sending the device data specified by (s2), the device specified by (d3) turns on.
- For the target stations that can be specified, refer to the following.

☞ Page 986 Target networks and target station types



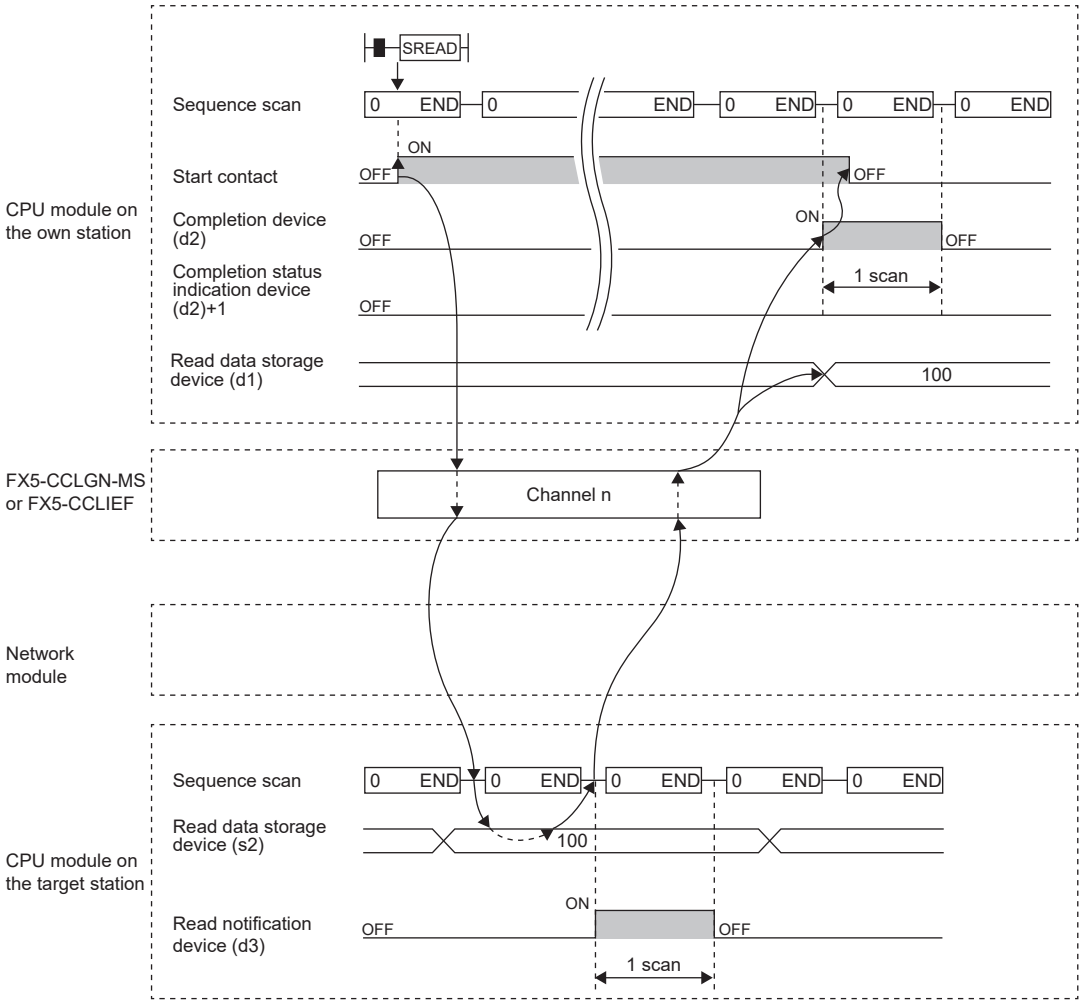
- When executing multiple link dedicated instructions concurrently, be careful not to overlap the channels of the link dedicated instructions. Multiple link dedicated instructions specifying the same channel cannot be used concurrently.

• The execution of the GP.SREAD instruction and whether it has been completed normally or completed with an error can be checked with the completion device (d2) or completion status indication device (d2)+1.

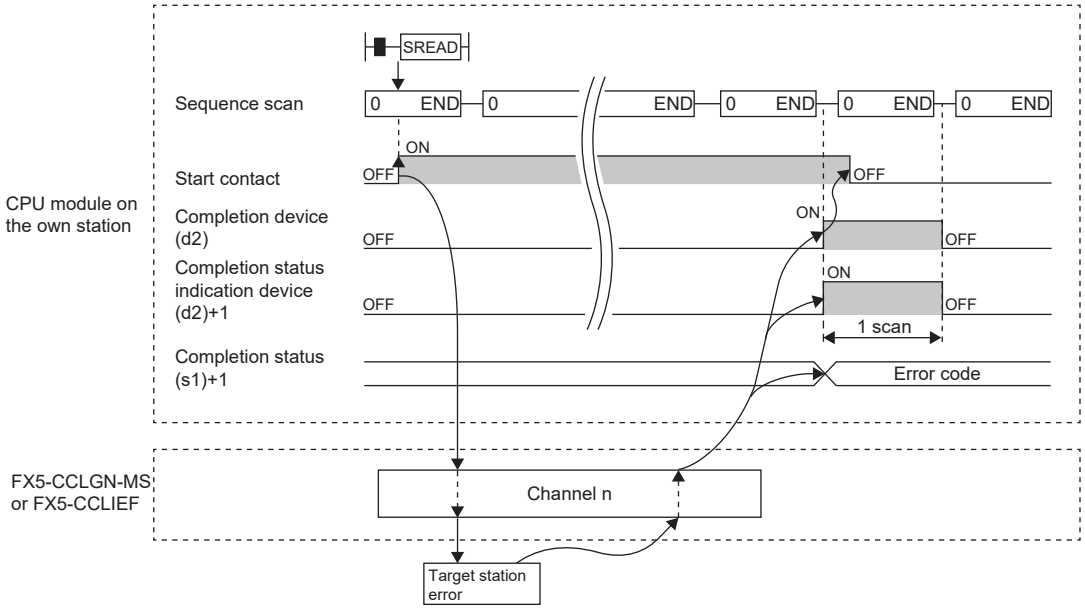
Device	Operation
Completion device (d2)	The device turns on during the END processing for the scan in which the GP.SREAD instruction is completed, and turns off during the next END processing.
Completion status indication device (d2)+1	The device turns on or off depending on the completion status of the GP.SREAD instruction. When completed normally: The device does not change (remains off). When completed with an error: The device turns on during the END processing for the scan in which the GP.SREAD instruction is completed, and turns off during the next END processing.

• The following figure shows the execution timing of the GP.SREAD instruction.

When completed normally



When completed with an error



- Read processing is performed only once on the rising edge when the read command turns on.

Operation error

Error code ((s1)+1)	Description
C000H to CFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual.
D000H to DFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual. Refer to MELSEC iQ-F FX5 CC-Link IE Field Network Module User's Manual.

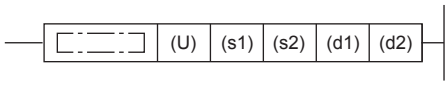
Writing data to another station programmable controller

GP.WRITE

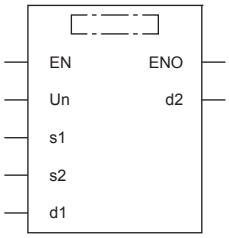
FX5S FX5UJ FX5U FX5UC

This instruction writes data to the device in another station programmable controller. (In units of words)

CC-Link IE TSN is compatible only with FX5U/FX5UC CPU module.

Ladder diagram	Structured text
	ENO:=GP_WRITE(EN,Un,s1,s2,d1,d2);

FBD/LD



("GP_WRITE" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)	Own station start device where control data is stored	☞ Page 1001 Control dataRefer to	Device name	ANY16*4
(s2)	Own station head device where the data to be written is stored	—	Device name	ANY16*4
(d1)	Target station head device where data is to be written*2	—	Character string*3	ANYSTRING_SINGLE*3
(d2)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.

*2 The continuous area specified by the write data length (s1)+9 is required.

*3 For the specifications of the string data to be specified, refer to the following.
(☞ Page 987 Specifications of character string data specified by link dedicated instructions)

*4 Digit specified bit type label cannot be used.

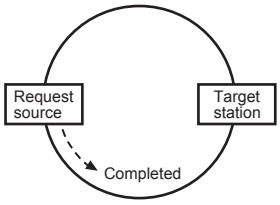
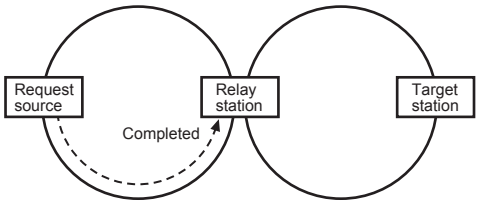
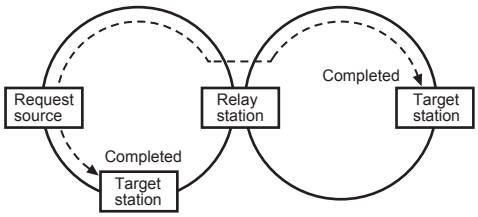
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	—	—	—	—	—	—	—	○	—
(d2)	○*1	○*2	—	—	—	—	—	—	—	—	—

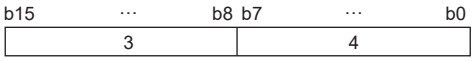
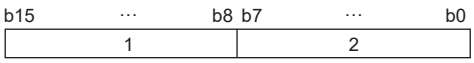
*1 S cannot be used.

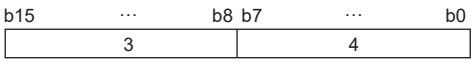
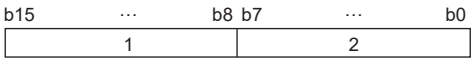
*2 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s1)+0	Execution/error completion type	b15 b14 ··· b9 b8 b7 b6 ··· b0 (4) 0 (3) (2) 0 (1) (1) Execution type (b0) • 0: No arrival acknowledgment When the target station is in the own network: Completed when data has been sent from the own station.  When the target station is in another network: Completed when data has arrived the relay station of the own network. (compatible only with CC-Link IE Field Network)  • 1: Arrival acknowledgment used Completed when data has been written in the target station.  (2) Error completion type (b7) Specify whether to set data at completion with an error. • 0: Do not set data in (s1)+11 and later at completion with an error. • 1: Set data in (s1)+11 and later at completion with an error. (3)Setting of arrival acknowledgment time (b8) (compatible only with CC-Link IE TSN) • 0: By second • 1: By 100 ms (4)Method for specifying the target station address (b15) (compatible only with CC-Link IE TSN) • 0: Specify the network No. in (s1)+4 and the station No. in (s1)+5. • 1: Specify the IP address in (s1)+4 and 5.	0000H 0001H 0080H 0081H 0100H 0101H 0180H 0181H 8000H 8001H 8080H 8081H 8100H 8101H 8180H 8181H	User
(s1)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System

Device	Item	Description	Setting range	Set by
(s1)+2	Own station channel	Specify the channel to be used by the own station. [CC-Link IE Field Network] • Channels 1 and 2 [CC-Link IE TSN] • Channels 1 to 8	1 to 8	User
(s1)+3	Target station CPU type	Specify the CPU type of the target station. • 0000H: Addressed to target station CPU (control CPU) • 03D0H: Addressed to control system CPU • 03D1H: Addressed to standby system CPU • 03D2H: Addressed to system A CPU • 03D3H: Addressed to system B CPU • 03E0H: Addressed to multiple CPU No. 1 • 03E1H: Addressed to multiple CPU No. 2 • 03E2H: Addressed to multiple CPU No. 3 • 03E3H: Addressed to multiple CPU No. 4 • 03FFH: Addressed to target station CPU (control CPU)	0000H 03D0H to 03D3H 03E0H to 03E3H 03FFH	User

Device	Item	Description	Setting range	Set by
(s1)+4	Target network number	[CC-Link IE Field Network] Specify the network No. of the target station. • 1 to 239 (network No.)	■(s1)+4 • Network No.: 1 to 239 ■(s1)+5 • Station No.: 1 to 120, 125 and 126 • Group number: 0081H to 00A0H • Specification of all stations: 00FFH ■(s1)+4, 5 • IP address: 00000001H to FFFFFFFEH	User
		[CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 Specify the network No. of the target station. • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 Specify the IP address (third and fourth octets) of the target station.  • b8 to b15: Third octet • b0 to b7: Fourth octet		
(s1)+5	Target station number	[CC-Link IE Field Network] Specify the station number of the target station. (1)Station number specification • 125: Master station • 126: Master operating station • 1 to 120: Local station, intelligent device station or sub-master station (2)Group number specification • 0081H to 00A0H: All stations in group numbers (0001H to 0020H) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) (3)Specification of all stations • 00FFH: All stations in the target network No. (broadcast (excluding the own station)) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) If the instruction is executed by specifying the group number or all stations, specify "0000H" or "03FFH" for the CPU type of the target station ((s1)+3).		User
		[CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 Specify the station number of the target station. (1)Station number specification • 1 to 120: Local station • 125: Master station (2)Group number specification • 0081H to 00A0H: All stations in group numbers (0001H to 0020H) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) (3)Specification of all stations • 00FFH: All stations in the target network No. (broadcast (excluding the own station)) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) If the instruction is executed by specifying the group number or all stations, specify "0000H" or "03FFH" for the CPU type of the target station ((s1)+3). ■When "1" is specified in b15 in (s1)+0 Specify the IP address (first and second octets) of the target station.  • b8 to b15: First octet • b0 to b7: Second octet		
(s1)+6	Not used	—	—	—
(s1)+7	Number of resends	Effective when the execution type specified by (s1)+0 is "1: Arrival acknowledgment used". ■At instruction execution Specify the number of resends to be performed if the instruction is not completed within the monitoring time specified by (s1)+8. ■At completion of instruction The number of resends performed (result) is stored.	0 to 15	User/ system

Device	Item	Description	Setting range	Set by
(s1)+8	Arrival monitoring time	Effective when the execution type specified by (s1)+0 is "1: Arrival acknowledgment used". Specify the monitoring time until completion of processing. If the processing is not completed within the monitoring time, the instruction will be resent the number of times specified in (s1)+7. ■When "0" is specified in b8 in (s1)+0 • 0: 10 seconds • 1 to 32767: 1 to 32767 seconds ■When "1" is specified in b8 in (s1)+0 • 0: 10000 ms (10 seconds) • 1 to 65535: 1 to 65535 × 100 ms	0 to 65535	User
(s1)+9	Write data length	Specify the number of words to be written. • 1 to 960 (words) (When writing data to QnACPU: 1 to 480 (words))	1 to 960	User
(s1)+10	Not used	—	—	—
(s1)+11	Clock setting flag	The validity status (valid or invalid) of the data in (s1)+12 and later is stored. Note that the data in (s1)+12 and later is not cleared even when the instruction is completed successfully. • 0: Invalid • 1: Valid	—	System
(s1)+12	Clock data (Set only in an abnormal state)	Upper 8 bits: Month (01H to 12H) Lower 8 bits: Year (00H to 99H: Lower two digits of the year)	—	System
(s1)+13		Upper 8 bits: Hour (00H to 23H) Lower 8 bits: Day (01H to 31H)	—	System
(s1)+14		Upper 8 bits: Second (00H to 59H) Lower 8 bits: Minute (00H to 59H)	—	System
(s1)+15		Upper 8 bits: Year (00H to 99H: Upper two digits of the year) Lower 8 bits: Day of the week (00H (Sun.) to 06H (Sat.))	—	System
(s1)+16	Error detection network number*1	■When "0" is specified in b15 in (s1)+0 The network number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (third and fourth octets) of the station where an error was detected is stored.  • b8 to b15: Third octet • b0 to b7: Fourth octet	—	System
(s1)+17	Error-detected station number*1	■When "0" is specified in b15 in (s1)+0 The number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) [CC-Link IE Field Network] • 125: Master station • 1 to 120: Local station, intelligent device station or sub-master station [CC-Link IE TSN] • 125: Master station • 1 to 120: Device station ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (first and second octets) of the station where an error was detected is stored.  • b8 to b15: First octet • b0 to b7: Second octet	—	System

*1 If completion status ((s1)+1) is "Channel in use (dedicated instruction) (error codes: D25AH to D25BH)", data is not stored.

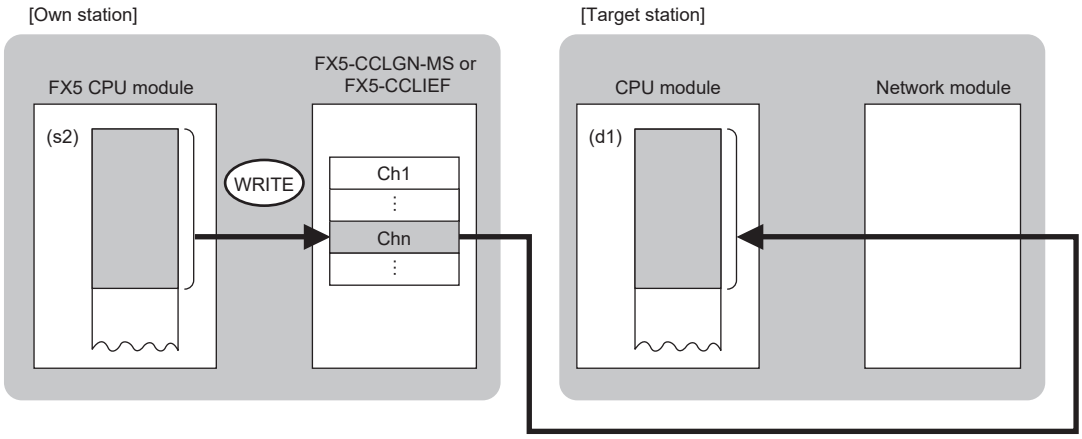
Point

- The continuous area (a maximum of 960 words) specified by the write data length ((s1)+9) is required in the write data storage device (d1).
- When a number from 1 to 120 is specified for the target station number, the GP.WRITE instruction should be executed with the execution type set to "Arrival acknowledgment used" to improve data reliability. When a number from 81H to A0H or FFH is specified for the target station number, the GP.WRITE instruction should be executed with the execution type set to "No arrival acknowledgment".
- When performing device writing to the same station from multiple stations, do not overlap the write timing. When the execution type is set to "No arrival acknowledgment", successful completion results in the write source station if communication is completed successfully even when the sending data contains an error. Also, even when the sending data is normal, a timeout results in the write source station if the GP.WRITE instruction is executed for the same station from multiple stations.
- The number of resends (s1)+7 must be set every time the instruction is executed.

Processing details

- The instruction writes data by the specified number of words ((s1)+9) from the own station head device (s2) into the target station word devices. Specify the target stations in control data ((s1)+4) and ((s1)+5). Upon completion of writing device data to target station, the completion device specified by (d2) turns on.
- For the target stations that can be specified, refer to the following.

☞ Page 986 Target networks and target station types

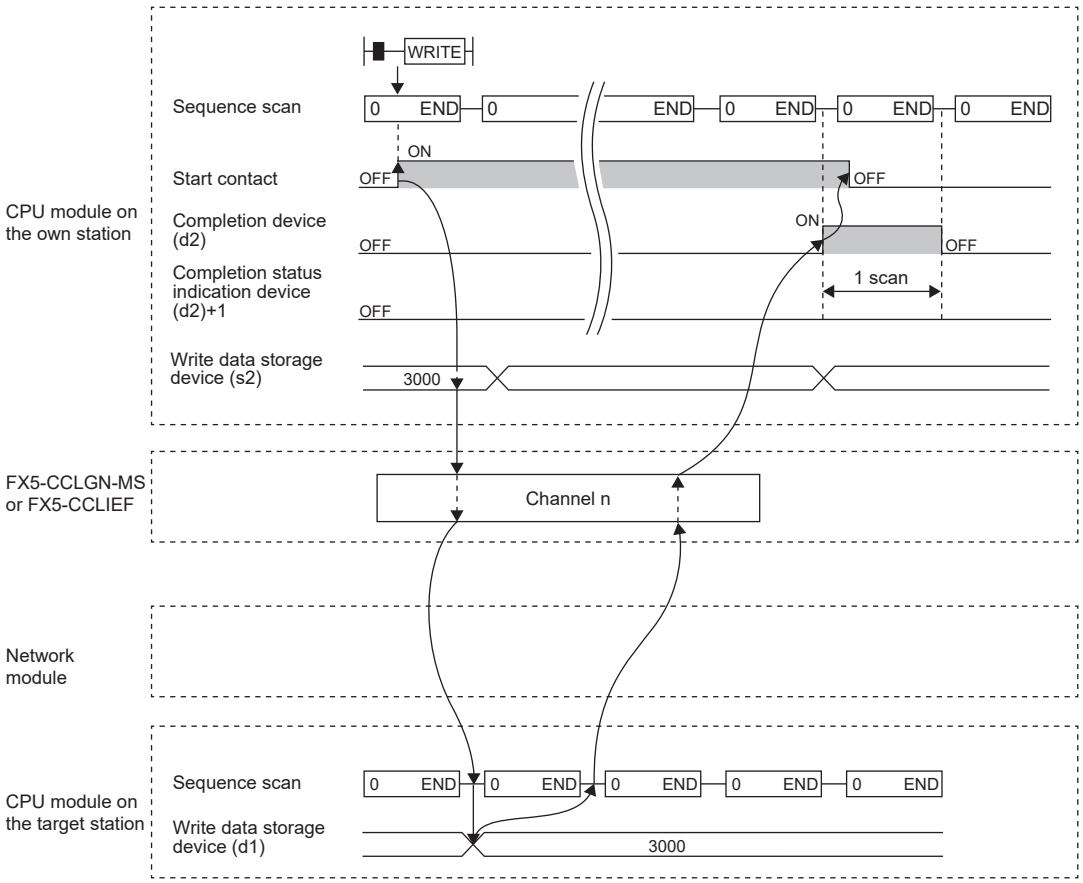


- When executing multiple link dedicated instructions concurrently, be careful not to overlap the channels of the link dedicated instructions. Multiple link dedicated instructions specifying the same channel cannot be used concurrently.
- The execution of the GP.WRITE instruction and whether it has been completed normally or completed with an error can be checked with the completion device (d2) or completion status indication device (d2)+1.

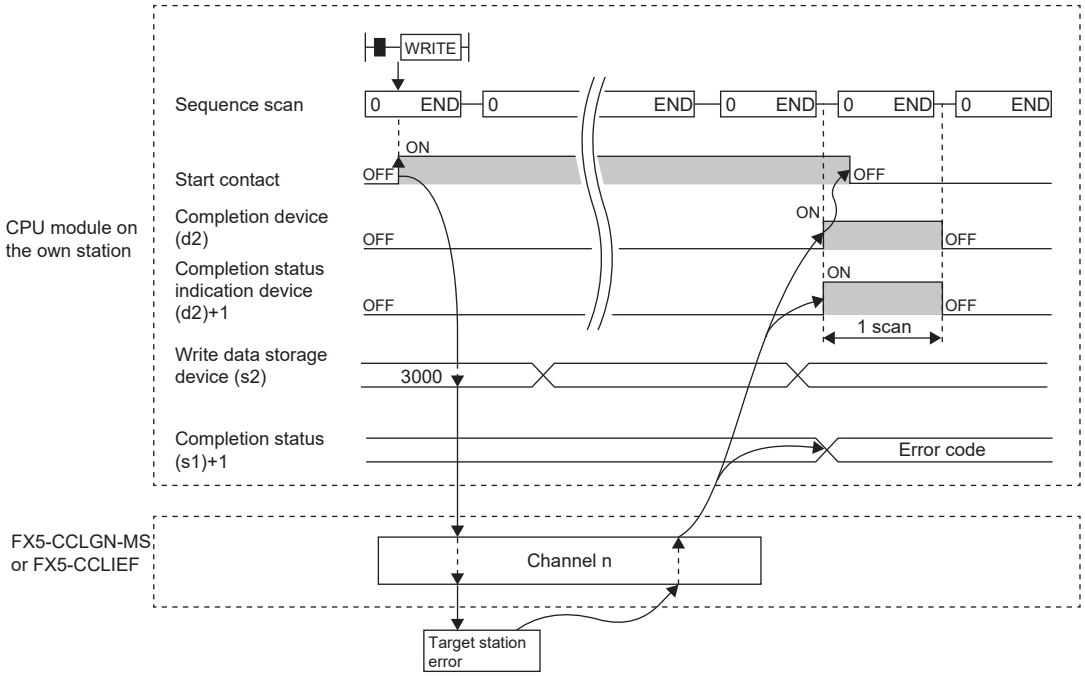
Device	Operation
Completion device (d2)	The device turns on during the END processing for the scan in which the GP.WRITE instruction is completed, and turns off during the next END processing.
Completion status indication device (d2)+1	The device turns on or off depending on the completion status of the GP.WRITE instruction. When completed normally: The device does not change (remains off). When completed with an error: The device turns on during the END processing for the scan in which the GP.WRITE instruction is completed, and turns off during the next END processing.

• The following figure shows the execution timing of the GP.WRITE instruction.

When completed normally



When completed with an error



• Write processing is performed only once on the rising edge when the write command turns on.

Operation error

Error code ((s1)+1)	Description
C000H to CFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual.
D000H to DFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual. Refer to MELSEC iQ-F FX5 CC-Link IE Field Network Module User's Manual.

Writing data to another station programmable controller (with notification)

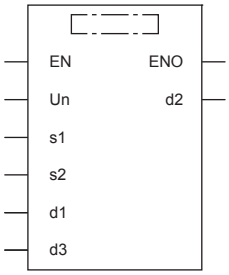
GP.SWRITE

FX5S FX5UJ FX5U FX5UC

This instruction writes data to the device in another station programmable controller. (In units of words)
After the data writing is completed by the GP.SWRITE instruction, the device of the other station is turned on. The other station can recognize that data has been written by the GP.SWRITE instruction.
CC-Link IE TSN is compatible only with FX5U/FX5UC CPU module.

Ladder diagram	Structured text
	ENO:=GP_SWRITE(EN,Un,s1,s2,d1,d2,d3);

FBD/LD



("GP_SWRITE" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)	Own station start device where control data is stored	☞ Page 1009 Control dataRefer to	Device name	ANY16*4
(s2)	Own station head device where the data to be written is stored	—	Device name	ANY16*4
(d1)	Target station head device where data is to be written*2	—	Character string*3	ANYSTRING_SINGLE*3
(d2)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
(d3)	Target station device to be turned on for one scan when the instruction completes (enabling the target station to recognize that data has been written from another station)	—	Character string*3	ANYSTRING_SINGLE*3
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

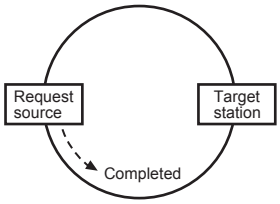
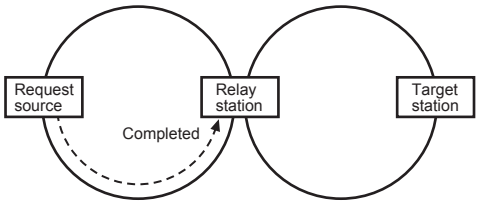
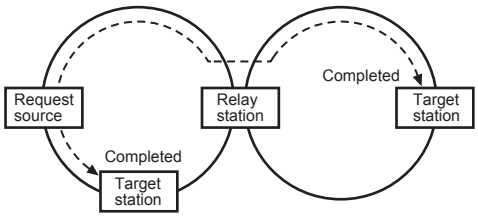
*1 In the case of the ST language and the FBD/LD language, U displays as Un.
*2 The continuous area specified by the write data length (s1)+9 is required.
*3 For the specifications of the string data to be specified, refer to the following.
(☞ Page 987 Specifications of character string data specified by link dedicated instructions)
*4 Digit specified bit type label cannot be used.

■Applicable devices

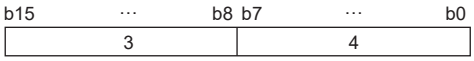
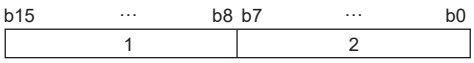
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	—	—	—	—	—	—	—	○	—
(d2)	○*1	○*2	—	—	—	—	—	—	—	—	—
(d3)	—	—	—	—	—	—	—	—	—	○	—

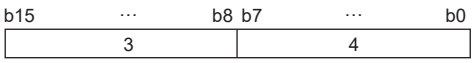
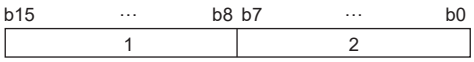
*1 S cannot be used.
*2 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s1)+0	Execution/error completion type	b15 b14 ··· b9 b8 b7 b6 ··· b0 (4) 0 (3) (2) 0 (1) (1) Execution type (b0) • 0: No arrival acknowledgment When the target station is in the own network: Completed when data has been sent from the own station.  When the target station is in another network: Completed when data has arrived the relay station of the own network. (compatible only with CC-Link IE Field Network)  • 1: Arrival acknowledgment used Completed when data has been written in the target station.  (2) Error completion type (b7) Specify whether to set data at completion with an error. • 0: Do not set data in (s1)+11 and later at completion with an error. • 1: Set data in (s1)+11 and later at completion with an error. (3)Setting of arrival acknowledgment time (b8) (compatible only with CC-Link IE TSN) • 0: By second • 1: By 100 ms (4)Method for specifying the target station address (b15) (compatible only with CC-Link IE TSN) • 0: Specify the network No. in (s1)+4 and the station No. in (s1)+5. • 1: Specify the IP address in (s1)+4 and 5.	0000H 0001H 0080H 0081H 0100H 0101H 0180H 0181H 8000H 8001H 8080H 8081H 8100H 8101H 8180H 8181H	User
(s1)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System

Device	Item	Description	Setting range	Set by
(s1)+2	Own station channel	Specify the channel to be used by the own station. [CC-Link IE Field Network] • Channels 1 and 2 [CC-Link IE TSN] • Channels 1 to 8	1 to 8	User
(s1)+3	Target station CPU type	Specify the CPU type of the target station. • 0000H: Addressed to target station CPU (control CPU) • 03D0H: Addressed to control system CPU • 03D1H: Addressed to standby system CPU • 03D2H: Addressed to system A CPU • 03D3H: Addressed to system B CPU • 03E0H: Addressed to multiple CPU No. 1 • 03E1H: Addressed to multiple CPU No. 2 • 03E2H: Addressed to multiple CPU No. 3 • 03E3H: Addressed to multiple CPU No. 4 • 03FFH: Addressed to target station CPU (control CPU)	0000H 03D0H to 03D3H 03E0H to 03E3H 03FFH	User

Device	Item	Description	Setting range	Set by
(s1)+4	Target network number	[CC-Link IE Field Network] Specify the network No. of the target station. • 1 to 239 (network No.)	■(s1)+4 • Network No.: 1 to 239 ■(s1)+5 • Station No.: 1 to 120, 125 and 126 • Group number: 0081H to 00A0H • Specification of all stations: 00FFH ■(s1)+4, 5 • IP address: 00000001H to FFFFFFFEH	User
		[CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 Specify the network No. of the target station. • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 Specify the IP address (third and fourth octets) of the target station.  • b8 to b15: Third octet • b0 to b7: Fourth octet		
(s1)+5	Target station number	[CC-Link IE Field Network] Specify the station number of the target station. (1)Station number specification • 125: Master station • 126: Master operating station • 1 to 120: Local station, intelligent device station or sub-master station (2)Group number specification • 0081H to 00A0H: All stations in group numbers (0001H to 0020H) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) (3)Specification of all stations • 00FFH: All stations in the target network No. (broadcast (excluding the own station)) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) If the instruction is executed by specifying the group number or all stations, specify "0000H" or "03FFH" for the CPU type of the target station ((s1)+3).		User
		[CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 Specify the station number of the target station. (1)Station number specification • 1 to 120: Local station • 125: Master station (2)Group number specification • 0081H to 00A0H: All stations in group numbers (0001H to 0020H) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) (3)Specification of all stations • 00FFH: All stations in the target network No. (broadcast (excluding the own station)) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) If the instruction is executed by specifying the group number or all stations, specify "0000H" or "03FFH" for the CPU type of the target station ((s1)+3). ■When "1" is specified in b15 in (s1)+0 Specify the IP address (first and second octets) of the target station.  • b8 to b15: First octet • b0 to b7: Second octet		
(s1)+6	Not used	—	—	—
(s1)+7	Number of resends	Effective when the execution type specified by (s1)+0 is "1: Arrival acknowledgment used". ■At instruction execution Specify the number of resends to be performed if the instruction is not completed within the monitoring time specified by (s1)+8. ■At completion of instruction The number of resends performed (result) is stored.	0 to 15	User/ system

Device	Item	Description	Setting range	Set by
(s1)+8	Arrival monitoring time	Effective when the execution type specified by (s1)+0 is "1: Arrival acknowledgment used". Specify the monitoring time until completion of processing. If the processing is not completed within the monitoring time, the instruction will be resent the number of times specified in (s1)+7. ■When "0" is specified in b8 in (s1)+0 • 0: 10 seconds • 1 to 32767: 1 to 32767 seconds ■When "1" is specified in b8 in (s1)+0 • 0: 10000 ms (10 seconds) • 1 to 65535: 1 to 65535 × 100 ms	0 to 65535	User
(s1)+9	Write data length	Specify the number of words to be written. • 1 to 960 (words) (When writing data to QnACPU: 1 to 480 (words))	1 to 960	User
(s1)+10	Not used	—	—	—
(s1)+11	Clock setting flag	The validity status (valid or invalid) of the data in (s1)+12 and later is stored. Note that the data in (s1)+12 and later is not cleared even when the instruction is completed successfully. • 0: Invalid • 1: Valid	—	System
(s1)+12	Clock data (Set only in an abnormal state)	Upper 8 bits: Month (01H to 12H) Lower 8 bits: Year (00H to 99H: Lower two digits of the year)	—	System
(s1)+13		Upper 8 bits: Hour (00H to 23H) Lower 8 bits: Day (01H to 31H)	—	System
(s1)+14		Upper 8 bits: Second (00H to 59H) Lower 8 bits: Minute (00H to 59H)	—	System
(s1)+15		Upper 8 bits: Year (00H to 99H: Upper two digits of the year) Lower 8 bits: Day of the week (00H (Sun.) to 06H (Sat.))	—	System
(s1)+16	Error detection network number*1	■When "0" is specified in b15 in (s1)+0 The network number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (third and fourth octets) of the station where an error was detected is stored.  • b8 to b15: Third octet • b0 to b7: Fourth octet	—	System
(s1)+17	Error-detected station number*1	■When "0" is specified in b15 in (s1)+0 The number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) [CC-Link IE Field Network] • 125: Master station • 1 to 120: Local station, intelligent device station or sub-master station [CC-Link IE TSN] • 125: Master station • 1 to 120: Device station ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (first and second octets) of the station where an error was detected is stored.  • b8 to b15: First octet • b0 to b7: Second octet	—	System

*1 If completion status ((s1)+1) is "Channel in use (dedicated instruction) (error codes: D25AH to D25BH)", data is not stored.

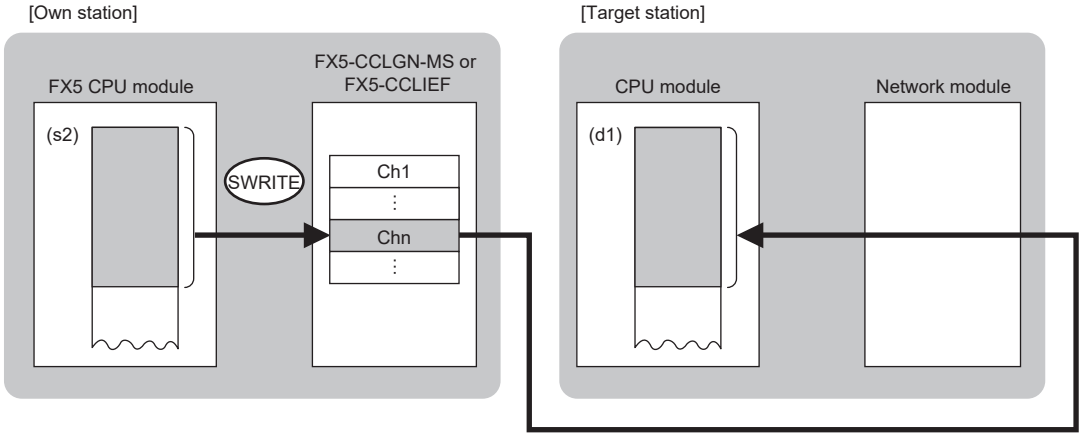
Point

- The continuous area (a maximum of 960 words) specified by the write data length ((s1)+9) is required in the write data storage device (d1).
- When a number from 1 to 120 is specified for the target station number, the GP.SWRITE instruction should be executed with the execution type set to "Arrival acknowledgment used" to improve data reliability. When a number from 81H to A0H or FFH is specified for the target station number, the GP.SWRITE instruction should be executed with the execution type set to "No arrival acknowledgment".
- When performing device writing to the same station from multiple stations, do not overlap the write timing. When the execution type is set to "No arrival acknowledgment", successful completion results in the write source station if communication is completed successfully even when the sending data contains an error. Also, even when the sending data is normal, a timeout results in the write source station if the GP.SWRITE instruction is executed for the same station from multiple stations.
- The number of resends (s1)+7 must be set every time the instruction is executed.

Processing details

- The instruction writes data by the specified number of words ((s1)+9) from the own station head device (s2) into the target station word devices. Specify the target stations in control data ((s1)+4) and ((s1)+5). Upon completion of writing device data to target station, the completion device specified by (d2) turns on. In the other station, upon completion of writing the device data specified by (s2), the device specified by (d3) turns on.
- For the target stations that can be specified, refer to the following.

☞ Page 986 Target networks and target station types

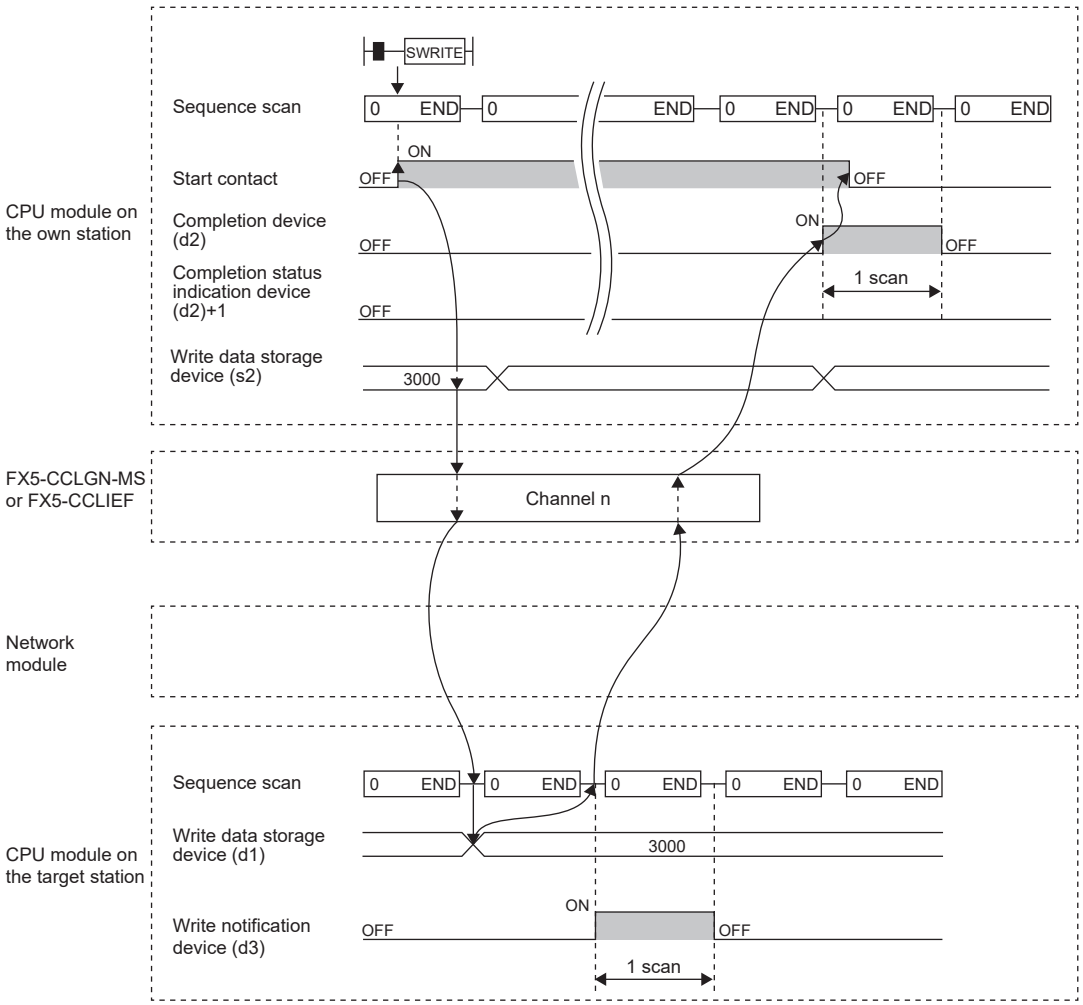


- When executing multiple link dedicated instructions concurrently, be careful not to overlap the channels of the link dedicated instructions. Multiple link dedicated instructions specifying the same channel cannot be used concurrently.
- The execution of the GP.SWRITE instruction and whether it has been completed normally or completed with an error can be checked with the completion device (d2) or completion status indication device (d2)+1.

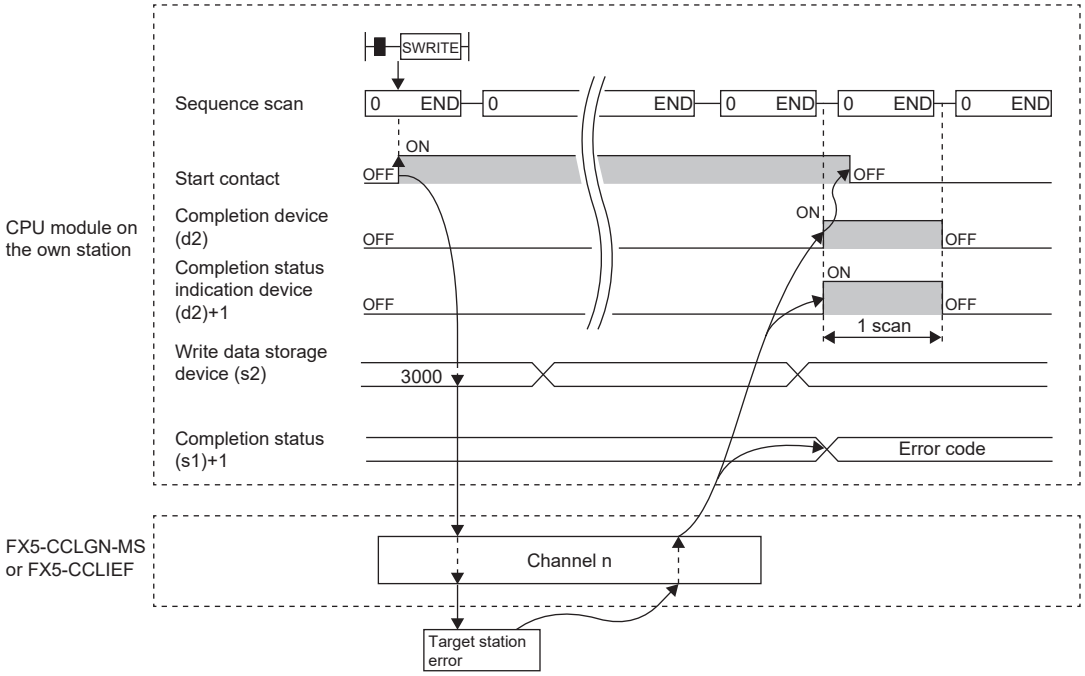
Device	Operation
Completion device (d2)	The device turns on during the END processing for the scan in which the GP.SWRITE instruction is completed, and turns off during the next END processing.
Completion status indication device (d2)+1	The device turns on or off depending on the completion status of the GP.SWRITE instruction. When completed normally: The device does not change (remains off). When completed with an error: The device turns on during the END processing for the scan in which the GP.SWRITE instruction is completed, and turns off during the next END processing.

• The following figure shows the execution timing of the GP.SWRITE instruction.

When completed normally



When completed with an error



• Write processing is performed only once on the rising edge when the write command turns on.

Operation error

Error code ((s1)+1)	Description
C000H to CFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual.
D000H to DFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual. Refer to MELSEC iQ-F FX5 CC-Link IE Field Network Module User's Manual.

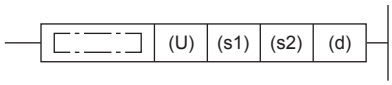
Sending data to another station programmable controller

GP.SEND

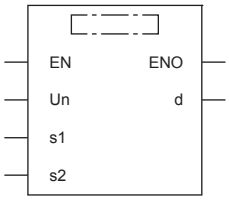
FX5S FX5UJ FX5U FX5UC

This instruction sends data to another station programmable controller.

CC-Link IE TSN is compatible only with FX5U/FX5UC CPU module.

Ladder diagram	Structured text
	ENO:=GP_SEND(EN,Un,s1,s2,d);

FBD/LD



("GP_SEND" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)	Own station start device where control data is stored	☞ Page 1017 Control dataRefer to	Device name	ANY16*3
(s2)	Own station head device where sending data is stored*2	—	Device name	ANY16*3
(d)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.

*2 The continuous area specified by the send data length (s1)+9 is required.

*3 Digit specified bit type label cannot be used.

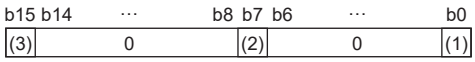
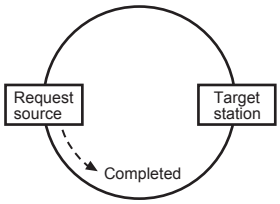
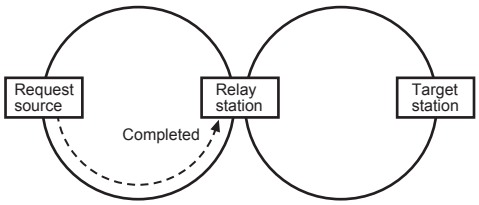
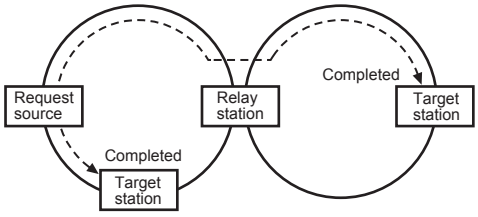
■Applicable devices

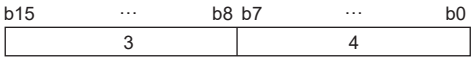
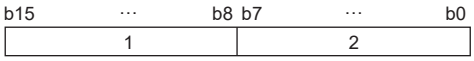
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

*2 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s1)+0	Execution/error completion type	 (1) Execution type (b0) • 0: No arrival acknowledgment When the target station is in the same network: Completed when data has been sent from the own station.  When the target station is in another network: Completed when data has arrived the relay station of the own network. (compatible only with CC-Link IE Field Network)  • 1: Arrival acknowledgment used Completed when data has been stored in the specified channel of the target station.  (2) Error completion type (b7) Specify whether to set data at completion with an error. • 0: Clock data at error occurrence is not stored in (s1)+11 and later. • 1: Clock data at error occurrence is stored in (s1)+11 and later. (3)Method for specifying the target station address (b15) (compatible only with CC-Link IE TSN) • 0: Specify the network No. in (s1)+4 and the station No. in (s1)+5. • 1: Specify the IP address in (s1)+4 and 5.	0000H 0001H 0080H 0081H 8000H 8001H 8080H 8081H	User
(s1)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System
(s1)+2	Own station channel	Specify the channel to be used by the own station. [CC-Link IE Field Network] • Channels 1 and 2 [CC-Link IE TSN] • Channels 1 to 8	1 to 8	User
(s1)+3	Target station storage channel	Specify the channel of the target station for storing data.	1 to 8	User

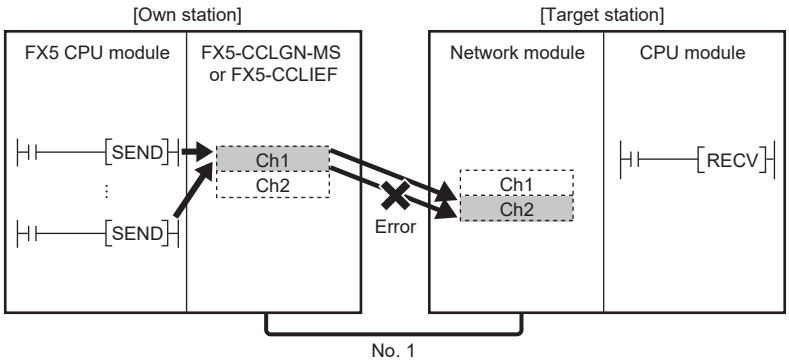
Device	Item	Description	Setting range	Set by
(s1)+4	Target network number	[CC-Link IE Field Network] Specify the network No. of the target station. • 1 to 239 (network No.)	■(s1)+4 • Network No.: 1 to 239 ■(s1)+5 • Station No.: 1 to 120, 125 and 126 • Group number: 0081H to 00A0H • Specification of all stations: 00FFH ■(s1)+4, 5 • IP address: 00000001H to FFFFFFFEH	User
		[CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 Specify the network No. of the target station. • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 Specify the IP address (third and fourth octets) of the target station.  • b8 to b15: Third octet • b0 to b7: Fourth octet		
(s1)+5	Target station number	[CC-Link IE Field Network] Specify the station number of the target station. (1)Station number specification • 125: Master station • 126: Master operating station • 1 to 120: Local station, intelligent device station or sub-master station (2)Group number specification • 0081H to 00A0H: All stations in group numbers (0001H to 0020H) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) (3)Specification of all stations • 00FFH: All stations in the target network No. (broadcast (excluding the own station)) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) If the instruction is executed by specifying the group number or all stations, specify "0000H" or "03FFH" for the CPU type of the target station ((s1)+3).		User
		[CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 Specify the station number of the target station. (1)Station number specification • 1 to 120: Local station • 125: Master station (2)Group number specification • 0081H to 00A0H: All stations in group numbers (0001H to 0020H) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) (3)Specification of all stations • 00FFH: All stations in the target network No. (broadcast (excluding the own station)) (This can be set when the execution type specified in (s1)+0 is "0: No arrival acknowledgment".) If the instruction is executed by specifying the group number or all stations, specify "0000H" or "03FFH" for the CPU type of the target station ((s1)+3). ■When "1" is specified in b15 in (s1)+0 Specify the IP address (first and second octets) of the target station.  • b8 to b15: First octet • b0 to b7: Second octet		
(s1)+6	Not used	—	—	—
(s1)+7	Number of resends (retries)	Effective when the execution type specified by (s1)+0 is "1: Arrival acknowledgment used". ■At instruction execution Specify the number of resends to be performed if the instruction is not completed within the monitoring time specified by (s1)+8. ■At completion of instruction The number of resends performed (result) is stored.	0 to 15	User/ system

Device	Item	Description	Setting range	Set by
(s1)+8	Arrival monitoring time	Effective when the execution type specified by (s1)+0 is "1: Arrival acknowledgment used". Specify the monitoring time until completion of processing. If the processing is not completed within the monitoring time, the instruction will be resent the number of times specified in (s1)+7. • 0: 10 seconds • 1 to 32767: 1 to 32767 seconds	0 to 32767	User
(s1)+9	Send data length	Specify the number of sending data from (s2) to (s2)+n. • 1 to 960 (words) (When sending data to QnACPU: 1 to 480 (words))	1 to 960	User
(s1)+10	Not used	—	—	—
(s1)+11	Clock setting flag	The validity status (valid or invalid) of the data in (s1)+12 and later is stored. Note that the data in (s1)+12 and later is not cleared even when the instruction is completed successfully. • 0: Invalid • 1: Valid	—	System
(s1)+12	Clock data (Set only in an abnormal state)	Upper 8 bits: Month (01H to 12H) Lower 8 bits: Year (00H to 99H: Lower two digits of the year)	—	System
(s1)+13		Upper 8 bits: Hour (00H to 23H) Lower 8 bits: Day (01H to 31H)	—	System
(s1)+14		Upper 8 bits: Second (00H to 59H) Lower 8 bits: Minute (00H to 59H)	—	System
(s1)+15		Upper 8 bits: Year (00H to 99H: Upper two digits of the year) Lower 8 bits: Day of the week (00H (Sun.) to 06H (Sat.))	—	System
(s1)+16	Error detection network number*1	■When "0" is specified in b15 in (s1)+0 The network number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (third and fourth octets) of the station where an error was detected is stored. b15 … b8 b7 … b0 3 | 4 • b8 to b15: Third octet • b0 to b7: Fourth octet	—	System
(s1)+17	Error-detected station number*1	■When "0" is specified in b15 in (s1)+0 The number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) [CC-Link IE Field Network] • 125: Master station • 1 to 120: Local station, intelligent device station or sub-master station [CC-Link IE TSN] • 125: Master station • 1 to 120: Device station ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (first and second octets) of the station where an error was detected is stored. b15 … b8 b7 … b0 1 | 2 • b8 to b15: First octet • b0 to b7: Second octet	—	System

*1 If completion status ((s1)+1) is "Channel in use (dedicated instruction) (error codes: D25AH to D25BH)", data is not stored.

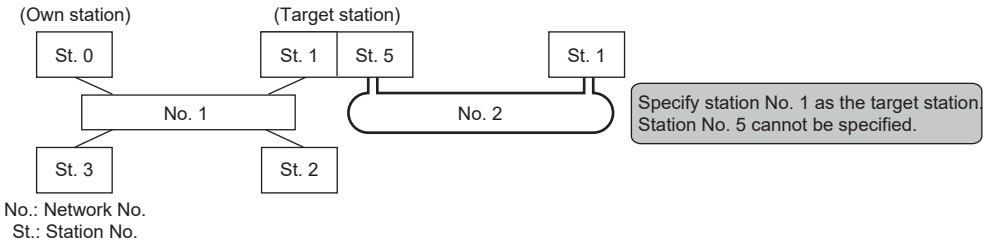
Point

- The continuous area (a maximum of 960 words) specified by the send data length ((s1)+9) is required in the send data storage device (s2).
- When a number from 1 to 120 is specified for the target station number, the instruction should be executed with the execution type set to "Arrival acknowledgment used" to improve data reliability. When a group number or all stations are specified, the instruction should be executed with the execution type set to "No arrival acknowledgment".
- When sending data to the same channel in the receiving station, execute the instruction after data has been read by the GP.RECV instruction in the receiving station. When the execution type is set to "No arrival acknowledgment", successful completion results in the sending station if communication is completed successfully even when the sending data contains an error. Also, even when the sending data is normal, a timeout results in the sending station if the instructions are executed for the same station from multiple stations.
- With the execution type set to "Arrival acknowledgment used", if the sending station sends data to the same channel in the receiving station before the receiving station reads data using the GP.RECV instruction, a buffer full error results in the sending station.



- When multiple network modules are mounted in the target station, specify the network number and station number of the network module that receives a request from the own station.

(Example: In the following figure, specify station No. 1 of network No. 1. (Station No. 5 of network No. 2. cannot be specified.))



No.: Network No.

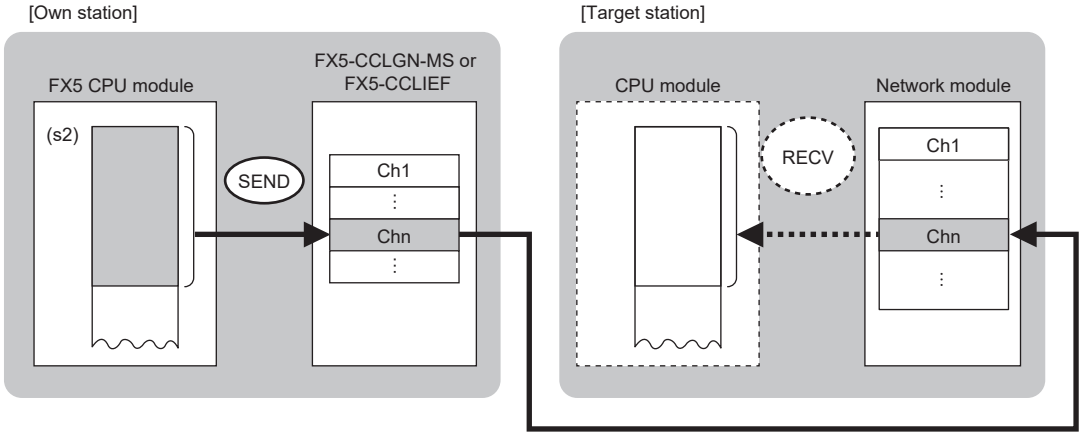
St.: Station No.

- The number of resends (s1)+7 must be set every time the instruction is executed.

Processing details

- The instruction sends data by the specified number of words ((s1)+9) from the own station head device (s2) into the specified channel of a target station. The data that has been sent is stored in the channel specified by (s2) in the target station. To read the sent data in the target station, use the GP.RECV instruction. Upon completion of sending data to the target station number, the completion device specified by (d) turns on.
- For the target stations that can be specified, refer to the following.

☞ Page 986 Target networks and target station types

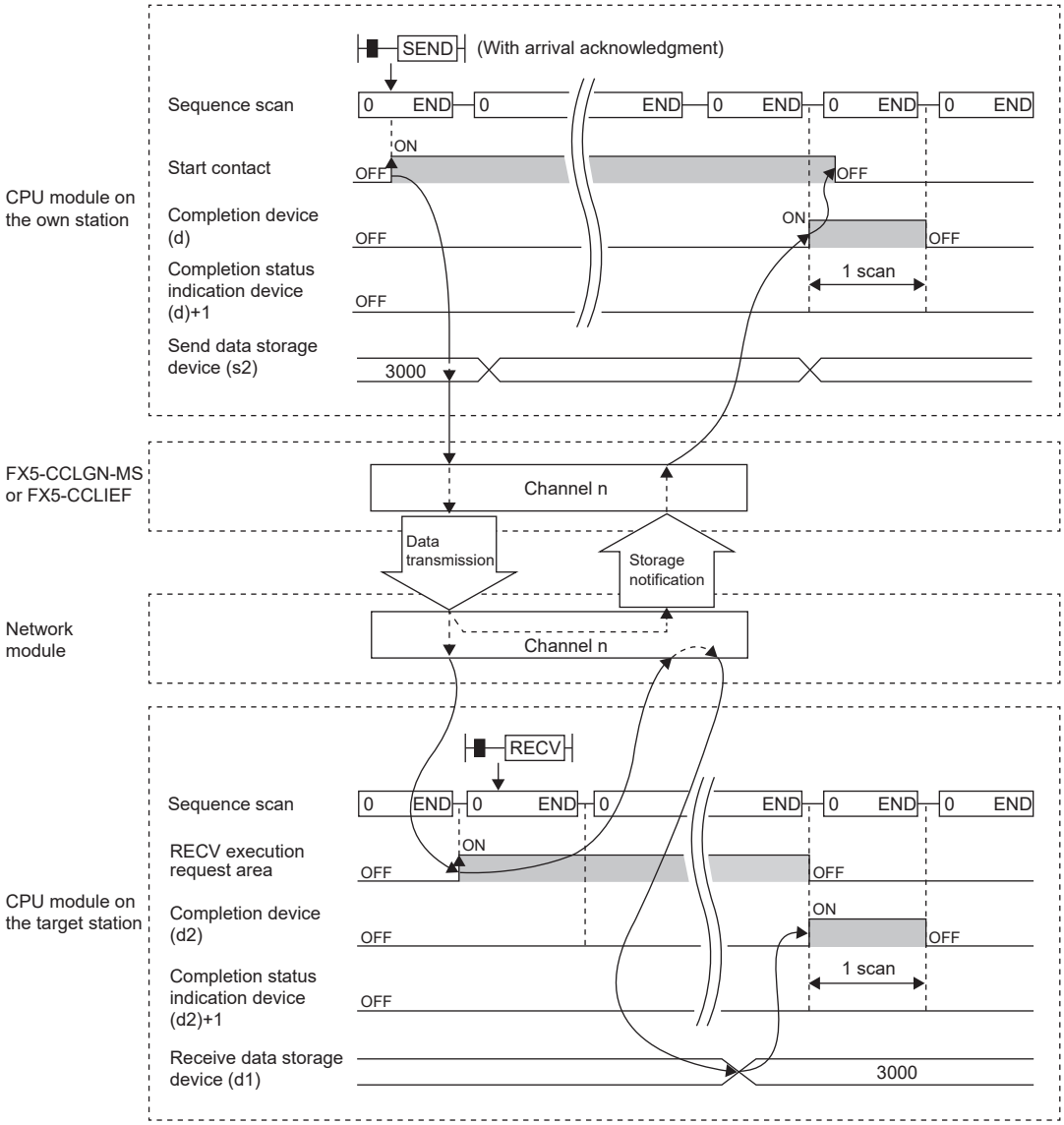


- When executing multiple link dedicated instructions concurrently, be careful not to overlap the channels of the link dedicated instructions. Multiple link dedicated instructions specifying the same channel cannot be used concurrently.
- The execution of the GP.SEND instruction and whether it has been completed normally or completed with an error can be checked with the completion device (d) or completion status indication device (d)+1.

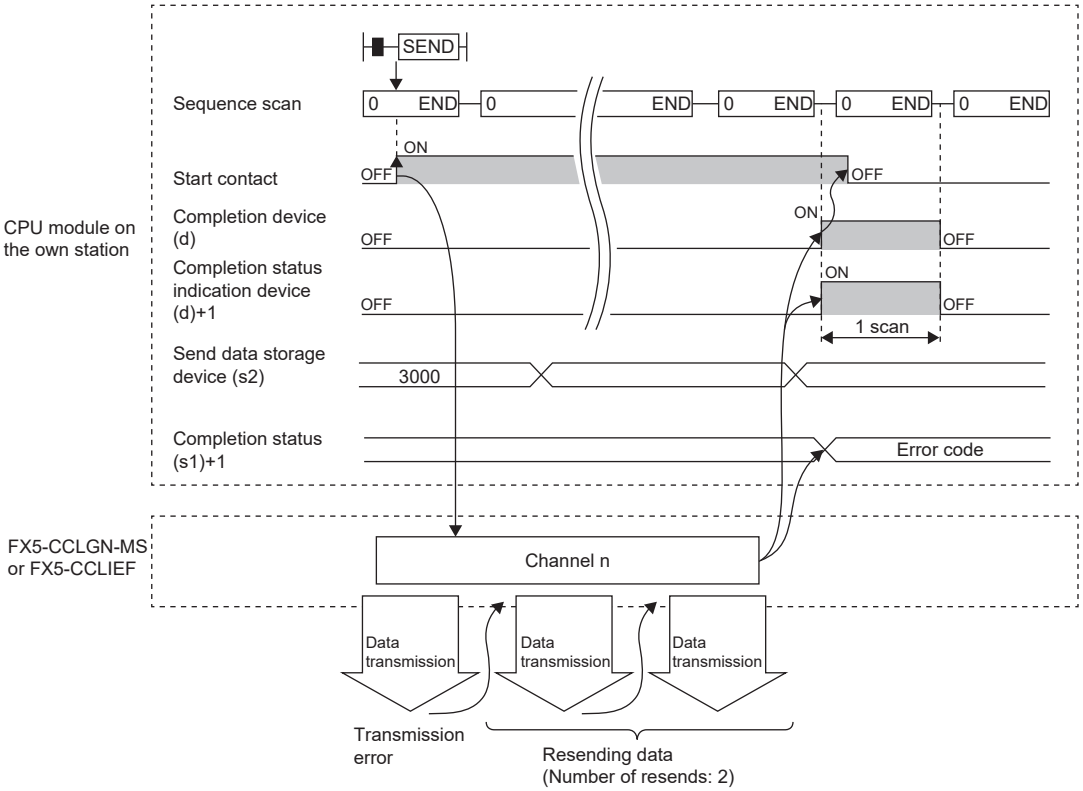
Device	Operation
Completion device (d)	The device turns on during the END processing for the scan in which the GP.SEND instruction is completed, and turns off during the next END processing.
Completion status indication device (d)+1	The device turns on or off depending on the completion status of the GP.SEND instruction. When completed normally: The device does not change (remains off). When completed with an error: The device turns on during the END processing for the scan in which the GP.SEND instruction is completed, and turns off during the next END processing.

• The following figure shows the execution timing of the SEND instruction.

When completed normally



When completed with an error



- Send processing is performed only once on the rising edge when the send command turns on.

Operation error

Error code ((s1)+1)	Description
C000H to CFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual.
D000H to DFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual. Refer to MELSEC iQ-F FX5 CC-Link IE Field Network Module User's Manual.

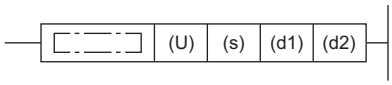
Receiving data from another station programmable controller

GP.RECV

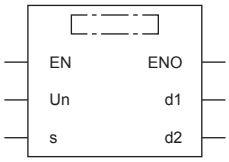
FX5S FX5UJ FX5U FX5UC

This instruction reads data received from another station programmable controller. (For the main routine program)

CC-Link IE TSN is compatible only with FX5U/FX5UC CPU module.

Ladder diagram	Structured text
	ENO:=GP_RECV(EN,Un,s,d1,d2);

FBD/LD



("GP_RECV" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s)	Own station start device where control data is stored	☞ Page 1025 Control dataRefer to	Device name	ANY16*3
(d1)	Own station head device for storing the receive data*2	—	Device name	ANY16*3
(d2)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.

*2 The continuous area (a maximum of 960 words) specified by the receive data length (s)+9 is required.

*3 Digit specified bit type label cannot be used.

■Applicable devices

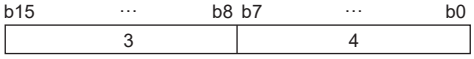
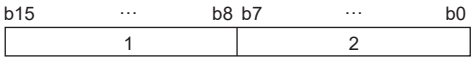
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s)	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(d2)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

*2 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s)+0	Error completion type	b15 b14 ··· b8 b7 ··· b0 (2) | 0 | (1) | 0 (1) Error completion type (b7) Specify whether to set data at completion with an error. • 0: Clock data at error occurrence is not stored in (s1)+11 and later. • 1: Clock data at error occurrence is stored in (s1)+11 and later. (2)Method for indicating the sending station address (b15) (compatible only with CC-Link IE TSN) • 0: Specify the network No. in (s1)+4 and the station No. in (s1)+5. • 1: Specify the IP address in (s1)+4 and 5.	0000H 0080H 8000H 8080H	User
(s)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System
(s)+2	Own station storage channel	Specify the channel to be used by the own station. [CC-Link IE Field Network] • Channels 1 and 2 [CC-Link IE TSN] • Channels 1 to 8	1 to 8	User
(s)+3	Channel used by sending station	The channel number (1 to 8) used by the sending station is stored.	—	System
(s)+4	Sending station network number	[CC-Link IE Field Network] The network number of the sending station is stored. • 1 to 239 (network No.) [CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 The network number of the sending station is stored. • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 The IP address (third and fourth octets) of the sending station is stored. b15 ··· b8 b7 ··· b0 3 | 4 • b8 to b15: Third octet • b0 to b7: Fourth octet	—	System
(s)+5	Sending station number	[CC-Link IE Field Network] The sending station number is stored. • 125: Master station or sub-master station (operating as master station) • 1 to 120: Local station, intelligent device station or sub-master station (operating as device station) [CC-Link IE TSN] ■When "0" is specified in b15 in (s1)+0 The sending station number is stored. • 1 to 120: Local station • 125: Master station ■When "1" is specified in b15 in (s1)+0 The IP address (first and second octets) of the sending station is stored. b15 ··· b8 b7 ··· b0 1 | 2 • b8 to b15: First octet • b0 to b7: Second octet	—	System
(s)+6	Not used	—	—	—
(s)+7	Not used	—	—	—
(s)+8	Arrival monitoring time	Specify the monitoring time until completion of processing. The instruction is completed with an error if it fails to complete within the monitoring time. • 0: 10 seconds • 1 to 32767: 1 to 32767 seconds	0 to 32767	User
(s)+9	Receive data length	The number of data received and stored in (d1) to (d1)+n is stored. • 0: No receive data • 1 to 960: Number of words of received data	—	System
(s)+10	Not used	—	—	—

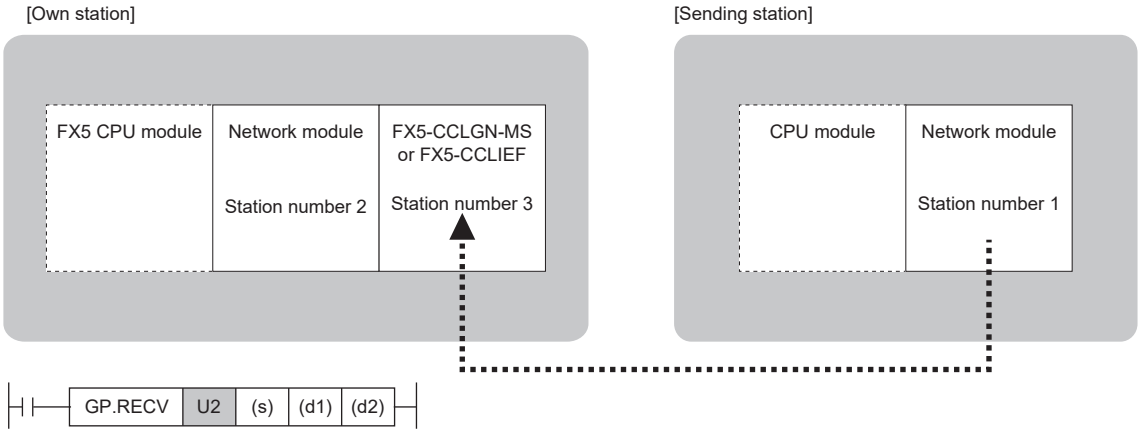
Device	Item	Description	Setting range	Set by
(s)+11	Clock setting flag	The validity status (valid or invalid) of the data in (s)+12 and later is stored. Note that the data in (s)+12 and later is not cleared even when the instruction is completed successfully. • 0: Invalid • 1: Valid	—	System
(s)+12	Clock data (Set only in an abnormal state)	Upper 8 bits: Month (01H to 12H) Lower 8 bits: Year (00H to 99H: Lower two digits of the year)	—	System
(s)+13		Upper 8 bits: Hour (00H to 23H) Lower 8 bits: Day (01H to 31H)	—	System
(s)+14		Upper 8 bits: Second (00H to 59H) Lower 8 bits: Minute (00H to 59H)	—	System
(s)+15		Upper 8 bits: Year (00H to 99H: Upper two digits of the year) Lower 8 bits: Day of the week (00H (Sun.) to 06H (Sat.))	—	System
(s)+16	Error detection network number*1	■When "0" is specified in b15 in (s1)+0 The network number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) • 1 to 239 (network No.) ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (third and fourth octets) of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.)  • b8 to b15: Third octet • b0 to b7: Fourth octet	—	System
(s)+17	Error-detected station number*1	■When "0" is specified in b15 in (s1)+0 The number of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.) [CC-Link IE Field Network] • 125: Master station • 1 to 120: Local station, intelligent device station or sub-master station [CC-Link IE TSN] • 125: Master station • 1 to 120: Device station ■When "1" is specified in b15 in (s1)+0 (compatible only with CC-Link IE TSN) The IP address (first and second octets) of the station where an error was detected is stored. (The number is not stored when an error was detected in the own station.)  • b8 to b15: First octet • b0 to b7: Second octet	—	System

*1 If completion status ((s1)+1) is "Channel in use (dedicated instruction) (error codes: D25AH to D25BH)", data is not stored.

Point

- The continuous area (a maximum of 960 words) specified by the receive data length ((s1)+9) is required in the receive data storage device (d1).
- The GP.RECV instruction is executed every time the bit of the corresponding channel in the GP.RECV instruction execution request area (SB0030 to SB0037) in the buffer memory turns on (with receive data).
- If multiple network modules having the same network number are installed in the receiving station, specify “U” for the network module that will receive data according to the GP.RECV instruction.

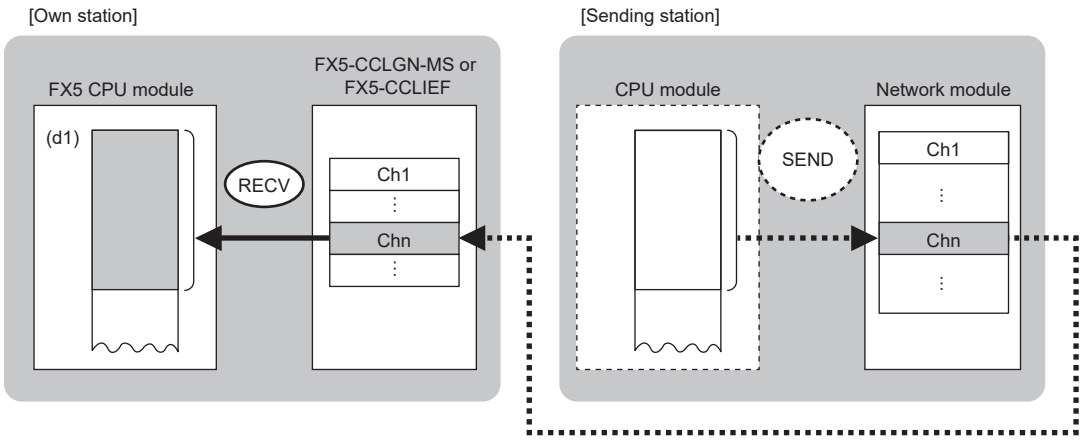
(Example: To execute the GP.RECV instruction in the station number 3 in response to the GP.SEND instruction from the station number 1, specify "U2".)



Processing details

- This instruction reads data, which were sent from another station by the SEND instruction, from the specified channel of the own station module to the own station word device (after (d1)). When the data reading from the specified channel of the own station is completed, the completion device (d2) turns on.
- For the target stations that can be specified, refer to the following.

☞ Page 986 Target networks and target station types



- The data received from the sending station is stored in the own station channel specified by the sending station and the corresponding bit in the GP.RECV instruction execution request area corresponding to each channel is set to on. When the corresponding bit in the GP.RECV instruction execution request area is set to on, the received data is read from the receive data storage channel. The following table lists the GP.RECV instruction execution request areas.

Name of GP.RECV instruction execution request area	Device number/address of corresponding bit
GP.RECV execution request flag Channel 1	SB0030
GP.RECV execution request flag Channel 2	SB0031
GP.RECV execution request flag Channel 3*1	SB0032
GP.RECV execution request flag Channel 4*1	SB0033
GP.RECV execution request flag Channel 5*1	SB0034
GP.RECV execution request flag Channel 6*1	SB0035
GP.RECV execution request flag Channel 7*1	SB0036
GP.RECV execution request flag Channel 8*1	SB0037

*1 Compatible only with FX5-CCLGN-MS.

- When executing multiple link dedicated instructions concurrently, be careful not to overlap the channels of the link dedicated instructions. Multiple link dedicated instructions specifying the same channel cannot be used concurrently.
- The execution of the GP.RECV instruction and whether it has been completed normally or completed with an error can be checked with the completion device (d2) or completion status indication device (d2)+1.

Device	Operation
Completion device (d2)	The device turns on during the END processing for the scan in which the GP.RECV instruction is completed, and turns off during the next END processing.
Completion status indication device (d2)+1	The device turns on or off depending on the completion status of the GP.RECV instruction. When completed normally: The device does not change (remains off). When completed with an error: The device turns on during the END processing for the scan in which the GP.RECV instruction is completed, and turns off during the next END processing.

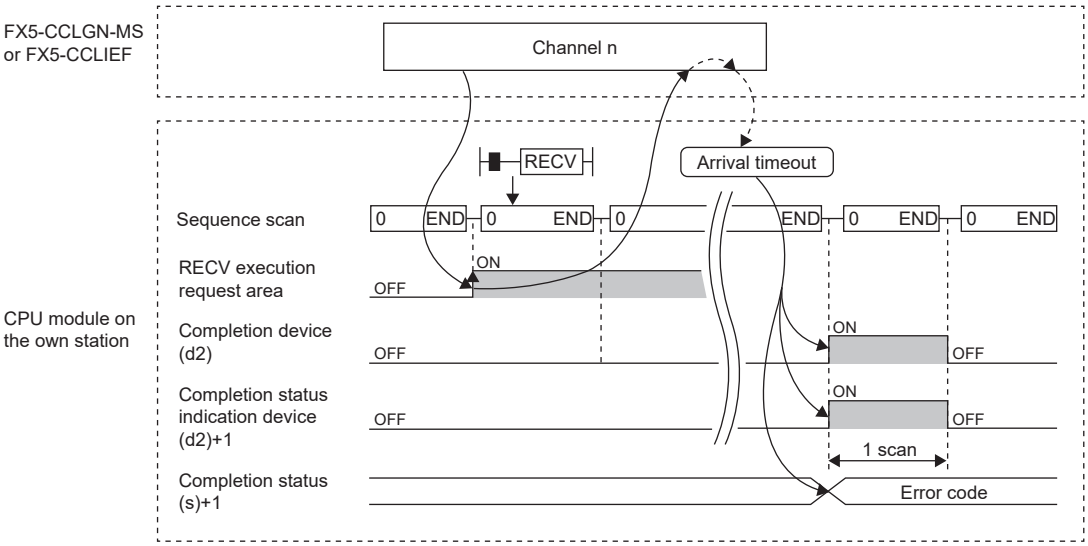
• The following figure shows the execution timing of the GP.RECV instruction.

When completed normally

For the execution timing, refer to the following the instruction.

(☞ Page 1016 GP.SEND)

When completed with an error



• Read processing is performed only once on the rising edge when the read command turns on.

Operation error

Error code ((s)+1)	Description
C000H to CFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual.
D000H to DFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual. Refer to MELSEC iQ-F FX5 CC-Link IE Field Network Module User's Manual.

13 ETHERNET INSTRUCTION

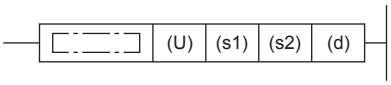
13.1 Built-in Ethernet Function Instruction

Opening a connection

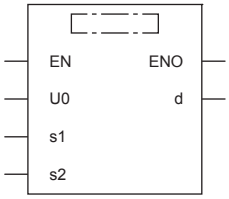
SP.SOCOPEN

FX5S FX5UJ FX5U FX5UC

This instruction opens a connection.

Ladder diagram	Structured text
	ENO:=SP_SOCOPEN(EN,U0,s1,s2,d);

FBD/LD



("SP_SOCOPEN" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy (Input the character string ['U0'].)	—	Character string	—*2 (ANYSTRING_SINGLE)
(s1)	Connection number	1 to 8	16-bit unsigned binary	ANY16
(s2)	Head device number for storing the control data	Refer to Control data (☞ Page 1031)	Word	ANY16_ARRAY (Number of elements: 10)
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. If the instruction is completed with an error, (d)+1 is also turned on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

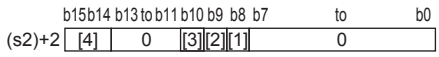
*1 In the case of the ST language and the FBD/LD language, U displays as U0.
*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	○	—
(s1)	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by[*1]
(s2)+0	Execution type/ completion type	Specify whether to use the parameter value set using the engineering tool or to use the set values of the control data (s2)+2 to (s2)+6 during the open processing of the connection. 0000H: The open processing is performed with the settings configured using "External Device Configuration" of the engineering tool. 8000H: The open processing is performed with the set values of the control data (s2)+2 to (s2)+6.	0000H 8000H	User
(s2)+1	Completion status	The status at the completion of the instruction is stored. 0000H: Completed successfully Other than 0000H: Completed with an error (error code) For error codes, refer to MELSEC iQ-F FX5 User's Manual (Communication).	—	System
(s2)+2	Application setting area	 [1] Communication method (protocol) 0: TCP/IP 1: UDP/IP [2] Socket communications function procedure 0: Communication protocol 1: Socket communications (No procedure) [3] Communication protocol setting 0: Do not use the communication protocol support function (use the socket communications function) 1: Use the protocol support function [4] Open method 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open	As shown on the left	User
(s2)+3	Host station port number	Specify the host station port number.	1 to 5548, 5570 to 65534 (0001H to 15ACH, 15C2H to FFFEH)[*3]	
(s2)+4 (s2)+5	Target device IP address[*2]	Specify the IP address of the target device.	1 to 3758096382 (00000001H to DFFFFFFEH)	
(s2)+6	Target device port number[*2]	Specify the port number of the target device.	1 to 65534 (0001H to FFFEH)	
(s2)+7 to (s2)+9	—	Use prohibited	—	System

*1 User: Data to be set before the execution of the instruction. System: The CPU module stores the execution result of the instruction.

*2 When Unpassive open is selected, the target device IP address and target device port number are ignored.

*3 Of the host station port numbers, 1 to 1023 (0001H to 03FFH) are generally reserved port numbers and 61440 to 65534 (F000H to FFFEH) are used by other communication functions. Thus, using 1024 to 5548 and 5570 to 61439 (0400H to 15ACH and 15C2H to EFFFH) as the port numbers is recommended. Do not specify 5549 to 5569 (15ADH to 15C1H) since they are used by the system.

Processing details

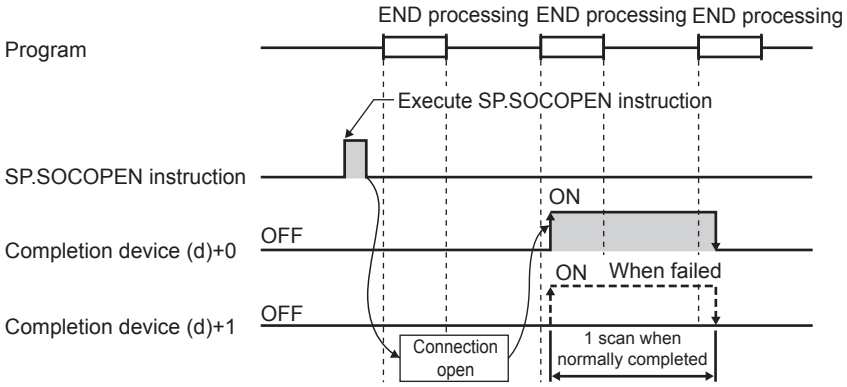
This instruction opens the connection specified by (s1).

The setting value used by the open processing is selected by (s2)+0.

The completion of the SP.SOCOPEN instruction can be checked using the completion devices (d)+0 and (d)+1.

- Completion device (d)+0: Turns ON during the END processing for the scan in which the SP.SOCOPEN instruction is completed, and turns OFF during the next END processing.
- Completion device (d)+1: Turns ON or OFF depending on the status of when the SP.SOCOPEN instruction is completed.

Status	Description
When completed normally	The device does not change (remains OFF).
When completed with an error	The device turns ON during the END processing for the scan in which the SP.SOCOPEN instruction is completed, and turns OFF during the next END processing.



The connection in which no protocol is set with the parameter can be opened and used. In this case, specify 8000H in (s2)+0 and the contents of the open processing in (s2)+2 to (s2)+6.

For details, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Operation error

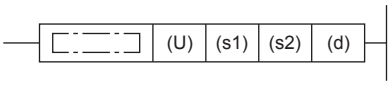
Error code (SD0/SD8067)	Description
3405H	The connection number specified by (s1) is other than 1 to 8.
2820H	The device number specified by (s2) or (d) is outside the range of the number of device points.
2822H	Device that cannot be specified is specified.
3582H	When an instruction which cannot be used in interruption routine program is used.

Closing a connection

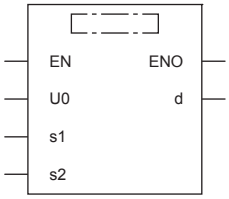
SP.SOCCLOSE

FX5S FX5UJ FX5U FX5UC

This instruction closes a connection.

Ladder diagram	Structured text
	ENO:=SP_SOCCLOSE(EN,U0,s1,s2,d);

FBD/LD



("SP_SOCCLOSE" enters □.)

13

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy (Input the character string ['U0'].)	—	String	—*2 (ANYSTRING_SINGLE)
(s1)	Connection number	1 to 8	16-bit unsigned binary	ANY16
(s2)	Head device number for storing the control data	Refer to Control data (☞ Page 1034)	Word	ANY16_ARRAY (Number of elements: 2)
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. If the instruction is completed with an error, (d)+1 is also turned on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as U0.
*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	○	—
(s1)	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by*1
(s2)+0	System area	—	—	—
(s2)+1	Completion status	The status at the completion of the instruction is stored. 0000H: Completed successfully Other than 0000H: Completed with an error (error code) For error codes, refer to MELSEC iQ-F FX5 User's Manual (Communication).	—	System

*1 System: The CPU module stores the execution result of the instruction.

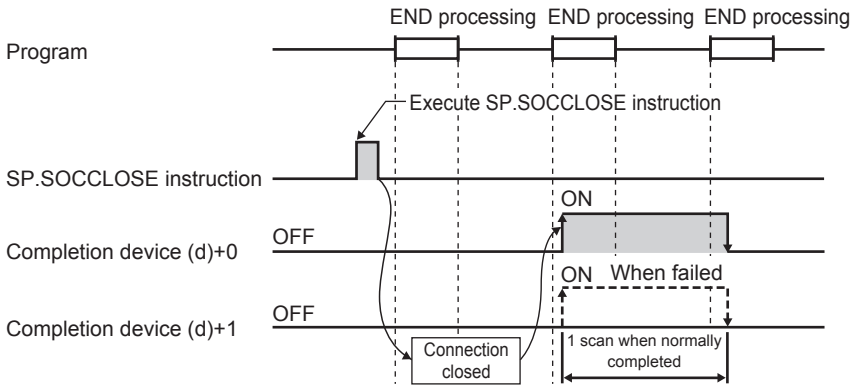
Processing details

This instruction closes the connection specified by (s1). (Connection disconnection)

The completion of the SP.SOCCLOSE instruction can be checked using the completion devices (d)+0 and (d)+1.

- Completion device (d)+0: Turns ON during the END processing for the scan in which the SP.SOCCLOSE instruction is completed, and turns OFF during the next END processing.
- Completion device (d)+1: Turns ON or OFF depending on the status when the SP.SOCCLOSE instruction is completed.

Status	Description
When completed normally	The device does not change (remains OFF).
When completed with an error	The device turns ON during the END processing for the scan in which the SP.SOCCLOSE instruction is completed, and turns OFF during the next END processing.



For details, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Operation error

Error code (SD0/SD8067)	Description
3405H	The connection number specified by (s1) is other than 1 to 8.
2820H	The device number specified by (s2) or (d) is outside the range of the number of device points.
2822H	Device that cannot be specified is specified.
3582H	When an instruction which cannot be used in interruption routine program is used.

Point

Do not execute the SP.SOCCLOSE instruction when Passive open is selected. Since the open completion signal and open request signal of the corresponding connection turn OFF and close processing is executed, the communication is disabled.

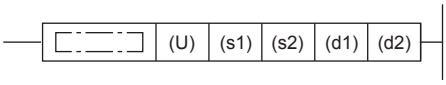
13.2 Socket Communications Function Instruction

Reading receive data during the END processing

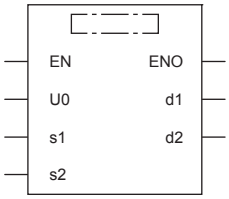
SP.SOCRCV

FX5S FX5UJ FX5U FX5UC

This instruction reads the receive data. (Reading during END processing)

Ladder diagram	Structured text
	ENO:=SP_SOCRCV(EN,U0,s1,s2,d1,d2);

FBD/LD



("SP_SOCRCV" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy (Input the character string ['U0'].)	—	Character string	—*2 (ANYSTRING_SINGLE)
(s1)	Connection number	1 to 8	16-bit unsigned binary	ANY16
(s2)	Head device number for specifying the control data	Refer to Control data (☞ Page 1036)	Word	ANY16_ARRAY (Number of elements: 2)
(d1)	Head device number for storing the receive data	—	Word	ANY16
(d2)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. If the instruction is completed with an error, (d2)+1 is also turned on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as U0.

*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	○	—
(s1)	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by*1
(s2)+0	System area	—	—	—
(s2)+1	Completion status	The status at the completion of the instruction is stored. 0000H: Completed successfully Other than 0000H: Completed with an error (error code) For error codes, refer to MELSEC iQ-F FX5 User's Manual (Communication).	—	System
(d1)+0	Receive data length	The data length of the data read from the socket communication receive data area is stored. (Number of bytes)	0 to 2046	System
(d1)+1 to (d1)+n	Receive data	The data read from the socket communication receive data area is sequentially stored.	—	System

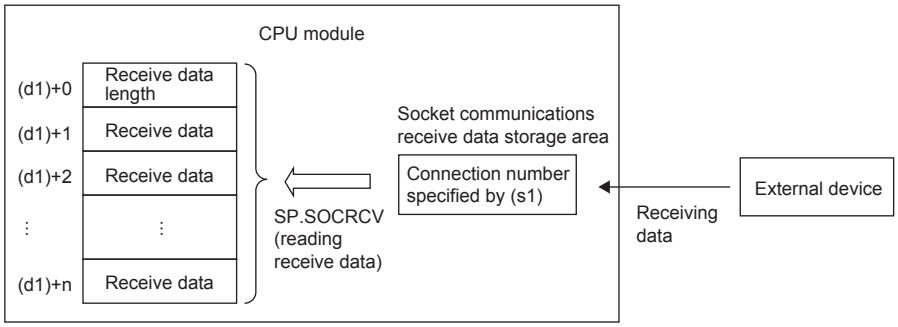
*1 System: The CPU module stores the execution result of the instruction.

Point

- When the SP.SOCRCV instruction is executed, reading data from the socket communication receive data area is executed with the END processing. Thus, executing the SP.SOCRCV instruction extends the scan time.
- When the data of odd-number of bytes is received, invalid data is stored in the higher byte of the device where the last receive data is stored.

Processing details

In the END processing after the execution of the SP.SOCRCV instruction, the receive data of the connection specified by (s1) is read from the socket communication receive data area.

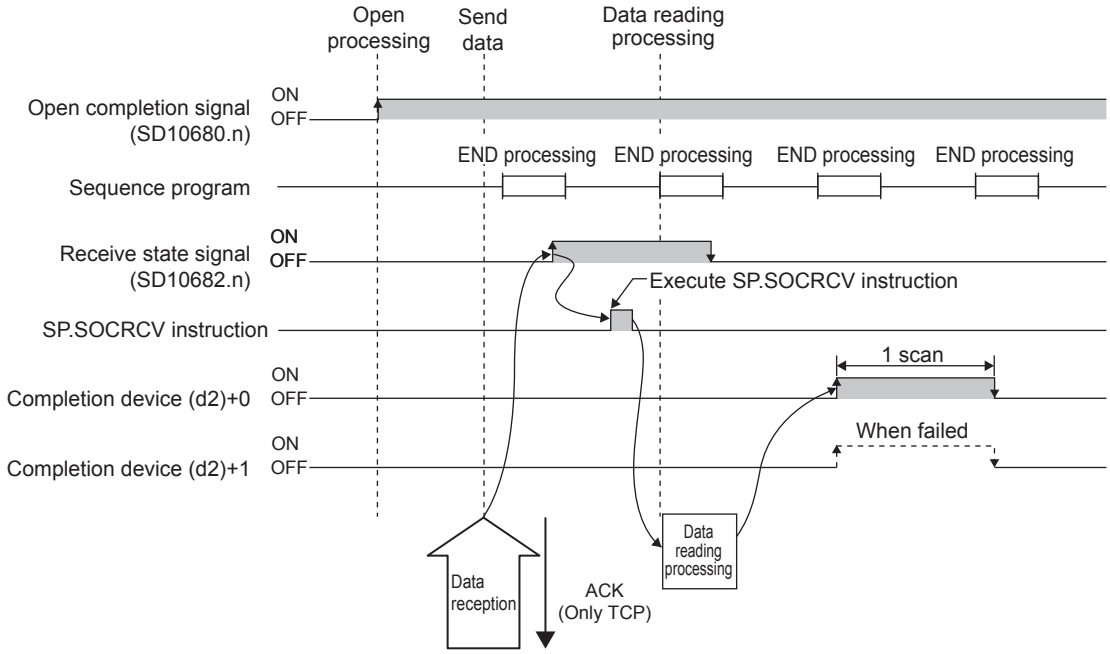


The completion of the SP.SOCRCV instruction can be checked using the completion devices (d2)+0 and (d2)+1.

- Completion device (d2)+0: Turns ON during the END processing for the scan in which the SP.SOCRCV instruction is completed, and turns OFF during the next END processing.
- Completion device (d2)+1: Turns ON or OFF depending on the status when the SP.SOCRCV instruction is completed.

Status	Description
When completed normally	The device does not change (remains OFF).
When completed with an error	The device turns ON during the END processing for the scan in which the SP.SOCRCV instruction is completed, and turns OFF during the next END processing.

The following figure shows the timing of the receive processing with the SP.SOCRCV instruction.



For details, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Operation error

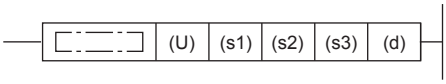
Error code (SD0/SD8067)	Description
3405H	The connection number specified by (s1) is other than 1 to 8.
2820H	The size of the receive data exceeds the size of the receive data storage device.
	The device number specified by (s2), (d1), or (d2) is outside the range of the number of device points.
2822H	Device that cannot be specified is specified.
3582H	When an instruction which cannot be used in interruption routine program is used.

Sending data

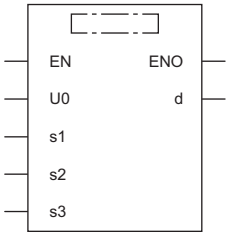
SP.SOCSND

FX5S FX5UJ FX5U FX5UC

This instruction sends data.

Ladder diagram	Structured text
	ENO:=SP_SOCSND(EN,U0,s1,s2,s3,d);

FBD/LD



("SP_SOCSND" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy (Input the character string ['U0'].)	—	Character string	—*2 (ANYSTRING_SINGLE)
(s1)	Connection number	1 to 8	16-bit unsigned binary	ANY16
(s2)	Head device number for specifying the control data	Refer to Control data (☞ Page 1039)	Word	ANY16_ARRAY (Number of elements: 2)
(s3)	Head device number for storing the send data	—	Word	ANY16
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. If the instruction is completed with an error, (d)+1 is also turned on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as U0.

*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	○	—
(s1)	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by*1
(s2)+0	System area	—	—	—
(s2)+1	Completion status	The status at the completion of the instruction is stored. 0000H: Completed successfully Other than 0000H: Completed with an error (error code) For error codes, refer to MELSEC iQ-F FX5 User's Manual (Communication).	—	System
(s3)+0	Send data length	Specifies the send data length. (Number of bytes)	1 to 2046	User
(s3)+1 to (s3)+n	Send data	Specifies the send data.	—	User

*1 User: Data to be set before the execution of the instruction. System: The CPU module stores the execution result of the instruction.

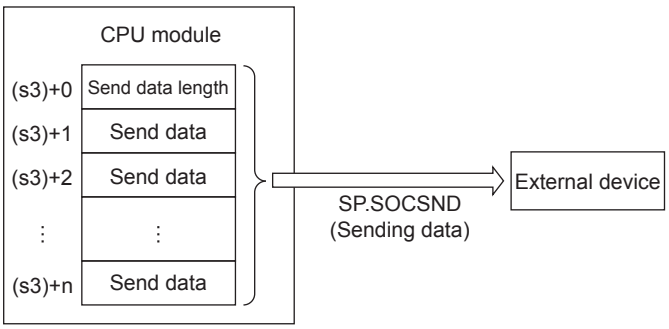
Point

When TCP is used, specify send data length that is smaller than the maximum window size of the target device (Receive data buffer of TCP). Data whose size exceeds the maximum window size of the target device cannot be sent.

13

Processing details

This instruction send the data set by (s3) to the target device of the connection specified by (s1).



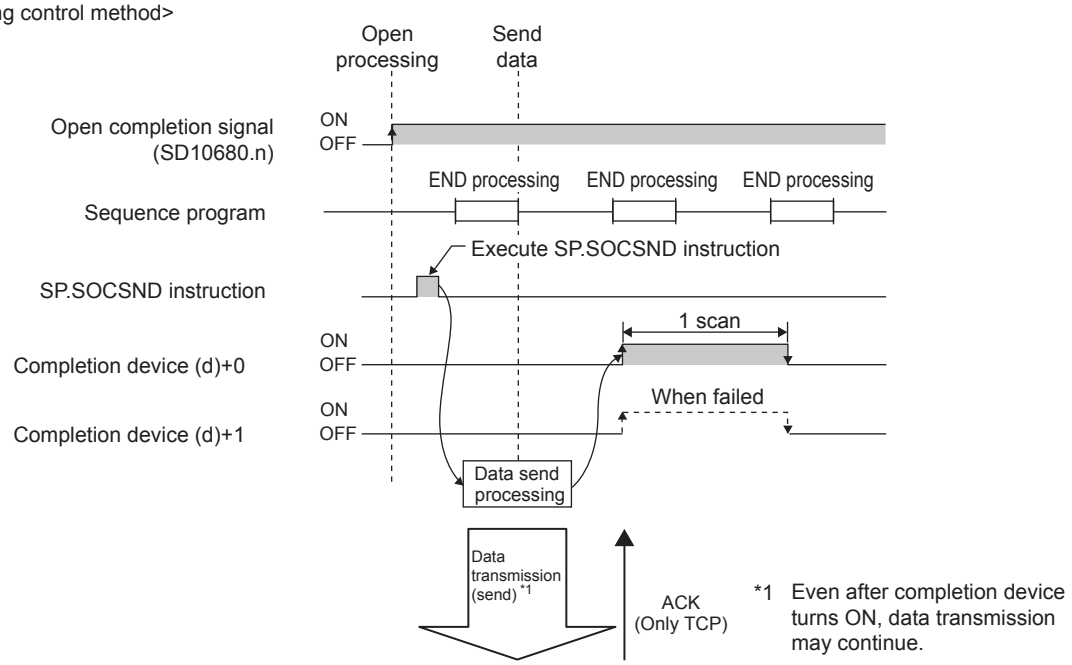
The completion of the SP.SOCSND instruction can be checked using the completion devices (d)+0 and (d)+1.

- Completion device (d)+0: Turns ON during the END processing for the scan in which the SP.SOCSND instruction is completed, and turns OFF during the next END processing.
- Completion device (d)+1: Turns ON or OFF depending on the status when the SP.SOCSND instruction is completed.

Status	Description
When completed normally	The device does not change (remains OFF).
When completed with an error	The device turns ON during the END processing for the scan in which the SP.SOCSND instruction is completed, and turns OFF during the next END processing.

The following figure shows the timing of the send processing with the SP.SOCSND instruction.

<Sending control method>



For details, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Operation error

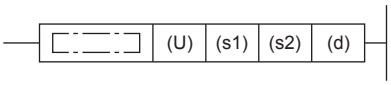
Error code (SD0/SD8067)	Description
3405H	The connection number specified by (s1) is other than 1 to 8.
2820H	The device number specified by (s2), (s3), or (d) is outside the range of the number of device points.
2822H	Device that cannot be specified is specified.
3582H	When an instruction which cannot be used in interruption routine program is used.

Reading connection information

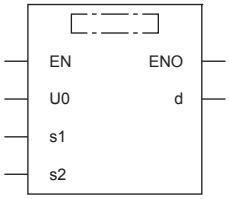
SP.SOCCINF

FX5S FX5UJ FX5U FX5UC

This instruction reads the connection information.

Ladder diagram	Structured text
	ENO:=SP_SOCCINF(EN,U0,s1,s2,d);

FBD/LD



("SP_SOCCINF" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy (Input the character string ['U0'].)	—	Character string	—*2 (ANYSTRING_SINGLE)
(s1)	Connection number	1 to 8	16-bit unsigned binary	ANY16
(s2)	Head device number for storing the control data	Refer to Control data (☞ Page 1042)	Word	ANY16_ARRAY (Number of elements: 2)
(d)	Head device number for storing the connection information	—	Word	ANY16_ARRAY (Number of elements: 5)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

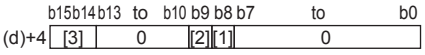
*1 In the case of the ST language and the FBD/LD language, U displays as U0.

*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	○	—
(s1)	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—

■Control data

Device	Item	Description	Setting range	Set by[*1]
(s2)+0	System area	—	—	—
(s2)+1	Completion status	The status at the completion of the instruction is stored. 0000H: Completed successfully Other than 0000H: Completed with an error (error code) For error codes, refer to 📖MELSEC iQ-F FX5 User's Manual (Communication).	—	System
(d)+0 (d)+1	Target device IP address	The IP address of the target device is stored.	1 to 3758096382 (00000001H to DFFFFFFEH)[*2]	System
(d)+2	Target device port number	The port number of the target device is stored.	1 to 65534 (0001H to FFFEH)[*2]	
(d)+3	Host station port number	The host station port number is stored.	1 to 5548, 5570 to 65534 (0001H to 15ACH, 15C2H to FFFEH)[*2*3]	
(d)+4	Application setting area	 [1] Communication method (protocol) 0: TCP/IP 1: UDP/IP [2] Procedure of the socket communication function 1: Non-protocol method [3] Open method 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open	As shown on the left[*2]	

*1 System: The CPU module stores the execution result of the instruction.

*2 When the instruction is executed for a connection that is not open, 0H is returned.

*3 Of the host station port numbers, 1 to 1023 (0001H to 03FFH) are generally reserved port numbers and 61440 to 65534 (F000H to FFFEH) are used by other communication functions. Thus, using 1024 to 5548 and 5570 to 61439 (0400H to 15ACH and 15C2H to EFFFH) as the port numbers is recommended. Do not specify 5549 to 5569 (15ADH to 15C1H) since they are used by the system.

Processing details

This instruction reads the connection information of the connection specified by (s1).

For details, refer to 📖MELSEC iQ-F FX5 User's Manual (Communication).

Operation error

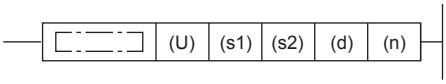
Error code (SD0/SD8067)	Description
3405H	The connection number specified by (s1) is other than 1 to 8.
2820H	The device number specified by (s2) or (d) is outside the range of the number of device points.
2822H	Device that cannot be specified is specified.

Reading socket communications receive data

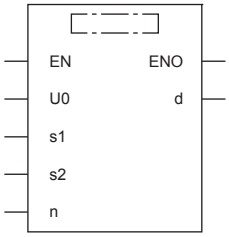
S(P).SOCRDATA

FX5S FX5UJ FX5U FX5UC

This instruction reads the data in the socket communication receive data area.

Ladder diagram	Structured text
	ENO:=S_SOCRDATA(EN,U0,s1,s2,n,d); ENO:=SP_SOCRDATA(EN,U0,s1,s2,n,d);

FBD/LD



("S_SOCRDATA", "SP_SOCRDATA" enters ☐.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy (Input the character string ['U0'].)	—	Character string	—*2 (ANYSTRING_SINGLE)
(s1)	Connection number	1 to 8	16-bit unsigned binary	ANY16
(s2)	Head device number for storing the control data	Refer to Control data (☞ Page 1044)	Word	ANY16_ARRAY (Number of elements: 2)
(d)	Head device number for storing the read data	—	Word	ANY16
(n)	Number of the read data (1 to 1024 words)	1 to 1024	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as U0.
*2 Regardless of the program language to be used, the data type is specified by a device. Do not specify a label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U☐\G☐	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	○	—
(s1)	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	—	○	—	—	—	—	○	—	—	—	—
(n)	—	○	—	—	—	—	○	○	—	—	—

■Control data

Device	Item	Description	Setting range	Set by*1
(s2)+0	System area	—	—	—
(s2)+1	Completion status	The status at the completion of the instruction is stored. 0000H: Completed successfully Other than 0000H: Completed with an error (error code) For error codes, refer to MELSEC iQ-F FX5 User's Manual (Communication).	—	System

*1 System: The CPU module stores the execution result of the instruction.

Processing details

These instructions read the data for the number of words specified by (n) from the socket communication receive data area of the connection specified by (s1) to the devices from the device specified by (d) onwards. No processing is performed when (n) is 0.

Point

- When (n) is 1, the receive data length can be read. By doing this, the device for storing the receive data can be changed when the SP.SOCRCV instruction is executed.

For details, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Precautions

- Even when the S(P).SOCRDATA instructions are executed, the socket communication receive data area is not cleared and the receiving status signal does not change. Therefore, the next receive data is not stored in the socket communication receive data area.
- To update receive data, use the SP.SOCRCV instruction to read the receive data.

Operation error

Error code (SD0/SD8067)	Description
3405H	The connection number specified by (s1) is other than 1 to 8.
2820H	The device number specified by (s2), (d), or (n) is outside the range of the number of device points.
2822H	Device that cannot be specified is specified.

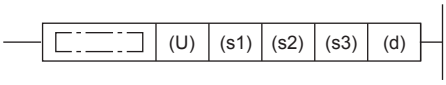
13.3 Predefined Protocol Support Function Instruction

Executing the registered protocols

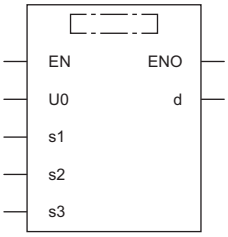
SP.ECPRTCL

FX5S FX5UJ FX5U FX5UC

This instruction executes the communication protocol registered using the engineering tool.

Ladder diagram	Structured text
	ENO:=SP_ECPRTCL(EN,U0,s1,s2,s3,d);

FBD/LD



("SP_ECPRTCL" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy (Input the character string ['U0'].)	—	Character string	ANYSTRING_SINGLE
(s1)	Connection number	1 to 8	16-bit unsigned binary	ANY16
(s2)	Number of protocols to be executed continuously	1 to 8	16-bit unsigned binary	ANY16
(s3)	Head device number for storing the control data	Refer to Control data (☞ Page 1046)	Word	ANY16_ARRAY (Number of elements: 18)
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. If the instruction is completed with an error, (d)+1 is also turned on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as U0.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	—	—	○	—
(s1)	○	○	—	—	—	—	○	○	—	—	—
(s2)	○	○	—	—	—	—	○	○	—	—	—
(s3)	○	○	—	—	—	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by*1
(s3)+0	Resulting number of executed protocols	The number of protocols executed by the SP.ECPRTCL instruction is stored. Any protocol where an error occurred is also included in the execution number. If the setting of setting data or control data contains an error, "0" is stored.	0, 1 to 8	System
(s3)+1	Completion status	The completion status is stored upon completion of the instruction. When two or more protocols are executed, the execution result of the protocol executed last is stored. 0: Normal completion Other than 0: Error completion (error code)	—	System
(s3)+2	Execution protocol number 1	Specify the number of the protocol to be executed first.	1 to 64	User
(s3)+3	Execution protocol number 2	Specify the number of the protocol to be executed second.	0, 1 to 64	
(s3)+4	Execution protocol number 3	Specify the number of the protocol to be executed third.	0, 1 to 64	
(s3)+5	Execution protocol number 4	Specify the number of the protocol to be executed fourth.	0, 1 to 64	
(s3)+6	Execution protocol number 5	Specify the number of the protocol to be executed fifth.	0, 1 to 64	
(s3)+7	Execution protocol number 6	Specify the number of the protocol to be executed sixth.	0, 1 to 64	
(s3)+8	Execution protocol number 7	Specify the number of the protocol to be executed seventh.	0, 1 to 64	
(s3)+9	Execution protocol number 8	Specify the number of the protocol to be executed eighth.	0, 1 to 64	
(s3)+10	Collation match Receive packet number 1	If receiving is included in the communication type of the protocol that has been executed first, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the first protocol, "0" is stored.	0, 1 to 16	System
(s3)+11	Collation match Receive packet number 2	If receiving is included in the communication type of the protocol that has been executed second, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the second protocol, "0" is stored. If the number of protocols executed is less than 2, "0" is stored.	0, 1 to 16	
(s3)+12	Collation match Receive packet number 3	If receiving is included in the communication type of the protocol that has been executed third, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the third protocol, "0" is stored. If the number of protocols executed is less than 3, "0" is stored.	0, 1 to 16	
(s3)+13	Collation match Receive packet number 4	If receiving is included in the communication type of the protocol that has been executed fourth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the fourth protocol, "0" is stored. If the number of protocols executed is less than 4, "0" is stored.	0, 1 to 16	
(s3)+14	Collation match Receive packet number 5	If receiving is included in the communication type of the protocol that has been executed fifth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the fifth protocol, "0" is stored. If the number of protocols executed is less than 5, "0" is stored.	0, 1 to 16	
(s3)+15	Collation match Receive packet number 6	If receiving is included in the communication type of the protocol that has been executed sixth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the sixth protocol, "0" is stored. If the number of protocols executed is less than 6, "0" is stored.	0, 1 to 16	
(s3)+16	Collation match Receive packet number 7	If receiving is included in the communication type of the protocol that has been executed seventh, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the seventh protocol, "0" is stored. If the number of protocols executed is less than 7, "0" is stored.	0, 1 to 16	
(s3)+17	Collation match Receive packet number 8	If receiving is included in the communication type of the protocol that has been executed eighth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the eighth protocol, "0" is stored. If the number of protocols executed is less than 8, "0" is stored.	0, 1 to 16	

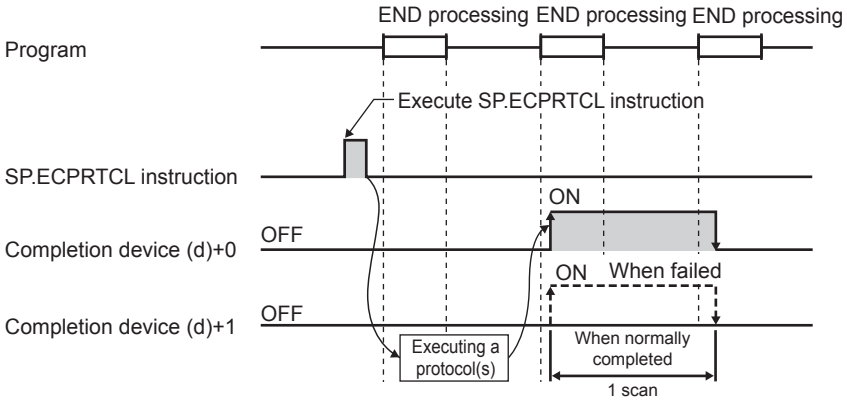
*1 User: Data to be set before the execution of the instruction. System: The CPU module stores the execution result of the instruction.

Processing details

This instruction executes the protocol registered using the engineering tool. Using the connection specified by (s1), the instruction executes the protocol in accordance with the control data stored in the device specified by (s3) and later. The instruction continuously executes as many protocols as specified by (s2) (a maximum of 8 protocols) at one time.
The number of executed protocols is stored in the device specified by (s3)+0.
The completion of the SP.ECPRTCL instruction can be checked using the completion devices (d)+0 and (d)+1.

- Completion device (d)+0: Turns ON during the END processing for the scan in which the SP.ECPRTCL instruction is completed, and turns OFF during the next END processing.
- Completion device (d)+1: Turns ON or OFF depending on the status when the SP.ECPRTCL instruction is completed.

Status	Description
When completed normally	The device does not change (remains OFF).
When completed with an error	The device turns ON during the END processing for the scan in which the SP.ECPRTCL instruction is completed, and turns OFF during the next END processing.



For details, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Precautions

- If an error occurs in the mth protocol while multiple protocols are being executed, the instruction does not execute the "m+1"th protocol and after and is completed with an error.
- The connections for which the SP.ECPRTCL instruction can be executed are only those for which "Communication protocol" is specified for the communication means.
- If a cancel request is received during execution of the mth protocol while multiple protocols are executed continuously, the following is stored in (s3).

Device	Item	Description
(s3)+0	Resulting number of executed protocols	The executed protocol number.
(s3)+1	Completion status	The error codes.
(s3)+10	Collation match Receive packet number 1	The receive packet number successful in collation match for the already executed protocol.
⋮	⋮	
(s3)+m+8	Collation match Receive packet number m-1	

- If same instructions are executed for the same connection, the subsequent instruction is ignored and is not executed until the preceding instruction is completed.
- The SP.ECPRTCL instruction itself does not open/close a connection and therefore the SP.SOCOPEN/SP.SOCCLOSE instructions need to be used to open/close the connection.

☞ Page 1030 SP.SOCOPENRefer to the Page 1033 SP.SOCCLOSE

Operation error

Error code (SD0/SD8067)	Description
2820H	The device used exceeded the specified range.
2821H	The device used to store data are overlapping.
2822H	Device that cannot be specified is specified.
3405H	The input data was out of range.

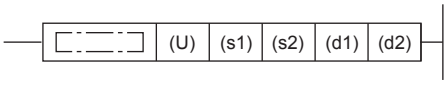
13.4 SLMP Frame Send Instruction

Sending the SLMP frame

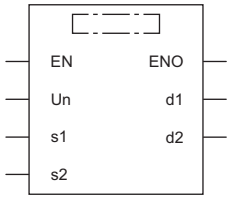
SP.SLMPSND

FX5S FX5UJ FX5U FX5UC

This instruction sends SLMP messages to an SLMP-compatible device.

Ladder diagram	Structured text
	ENO:=SP_SLMPSND(EN,Un,s1,s2,d1,d2);

FBD/LD



("SP_SLMPSND" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy	—	String	ANYSTRING_SINGLE
(s1)	Head device where control data is stored	☞ Page 1050 Control data	Word	ANY16_ARRAY (Number of elements: 19)
(s2)	Head device where request frame is stored	☞ Page 1051 Request frame	Word	ANY16_ARRAY*2
(d1)	Head device for storing response frame	☞ Page 1051 Response frame	Word	ANY16_ARRAY*2
(d2)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

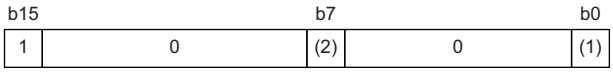
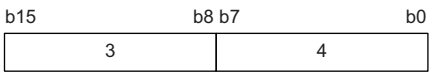
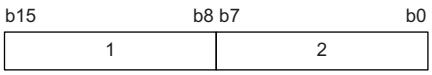
*1 In the case of ST language and FBD/LD language, U displays as Un.
*2 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	○	—
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s1)+0	Execution/error completion type	 (1) Execution type (b0) • 0: Without arrival check (The instruction is regarded as completed when a request message is sent from own station.)*1 • 1: With arrival check (The instruction is regarded as completed when a response message is received from the target device.) (2) Error completion type (b7) Specify whether to set data at completion with an error. • 0: Do not set data in (s1)+13 and later at completion with an error. (Clear (s1)+13 and later.) • 1: Set data at completion with an error in (s1)+13 and later.	—	User
(s1)+1	Completion status	The completion status is stored upon completion of the instruction. 0000H: Completed successfully Other than 0000H: Error (error code)	—	System
(s1)+2	Channel numbers to be used by own station	Specify the channel to be used by own station. 1: No serial number is given	1	User
(s1)+3	IP address of target device (third and fourth octets)	Specify the IP address (third and fourth octets) of target device.*2  3, 4: Indicates the octets of the IP address.	00000001H to DFFFFFFEH (both (s1)+3 and (s1)+4 together) (1 to 3758096382)	User
(s1)+4	IP address of target device (first and second octets)	Specify the IP address (first and second octets) of target device.  1, 2: Indicates the octets of the IP address.		User
(s1)+5	Destination port number	Specify the destination port number.	1 to 65534 (1 to FFFEH)	User
(s1)+6	Request destination network number	0000H (fixed)	0000H	User
(s1)+7	Request destination station number	00FFH (fixed)	00FFH	User
(s1)+8	Request destination module I/O number	Specify request destination module I/O number. • 03D0H: Addressed to control system CPU • 03D1H: Addressed to standby system CPU • 03D2H: Addressed to system A CPU • 03D3H: Addressed to system B CPU • 03E0H: Addressed to multiple CPU No. 1 • 03E1H: Addressed to multiple CPU No. 2 • 03E2H: Addressed to multiple CPU No. 3 • 03E3H: Addressed to multiple CPU No. 4 • 03FFH: To own station (control CPU) When executing the SP.SLMPSND instruction to FX5CPU, specify “03FFH”.	03D0H to 03D3H 03E0H to 03E3H 03FFH	User
(s1)+9	Request destination multidrop station number	0000H (fixed)	0000H	User
(s1)+10	Number of resends	The device becomes effective when the execution type specified by (s1)+0 is "1: With arrival check". ■Before instruction execution Specify the number of resends to be performed if the instruction is not completed within the monitoring time specified by (s1)+11. • 0 to 15 (times) ■At completion of instruction The number of resends performed (result) is stored. • 0 to 15 (times)	0 to 15	User/ system
(s1)+11	Arrival monitoring time	Set the monitoring time until completion of processing. If processing is not completed within the monitoring time, the request is resent the number of times specified in (s1)+10. • 0: 10 s • 1 to 32767: 1 to 32767 s	0 to 32767	User

Device	Item	Description	Setting range	Set by
(s1)+12	Clock setting flag	The validity status (valid or invalid) of the data in (s)+13 and later is stored. • 0: Invalid • 1: Valid	—	System
(s1)+13	Clock data (Set only in an abnormal state)	Upper 8 bits: Month (01H to 12H) Lower 8 bits: Year (00H to 99H: Lower two digits of the year)	—	System
(s1)+14		Upper 8 bits: Hour (00H to 23H) Lower 8 bits: Day (01H to 31H)		
(s1)+15		Upper 8 bits: Second (00H to 59H) Lower 8 bits: Minute (00H to 59H)		
(s1)+16		Upper 8 bits: Year (00H to 99H: Upper two digits of the year) Lower 8 bits: Day of the week (00H (Sun.) to 06H (Sat.))		
(s1)+17	Error-detected device IP address (third and fourth octets)	The IP address (third and fourth octets) of the station where an error was detected is stored. b15 b8 b7 b0 3 | 4 3, 4: Indicates the octets of the IP address.	—	System
(s1)+18	IP address of error detected device (first and second octets)	The IP address (first and second octets) of the station where an error was detected is stored. b15 b8 b7 b0 1 | 2 1, 2: Indicates the octets of the IP address.	—	System

*1 If (s1)+0 is set to "0: Without arrival check", receive data is not set. Set 0 in (s1)+0 in the following cases:
· When a command that does not return a response message is used
· When a response message is not referred to

*2 When “0” or “255” is specified to the fourth octet of the IP address of the target device, a send error occurs and the SP.SLMPSND instruction is completed with an error.
In addition, the error code (C1CDH (SP.SLMPSND instruction message send error)) is stored in the completion status.

• When data of the error completion type (the 7th bit of (s1)+0) is set to "1" and "0", the operations are shown below.

Error completion type (The 7th bit of (s1)+0)	Completion status (s1)+1	Clock setting flag (s1)+12	Clock data (s1)+13 to 16	IP address of error detected device (s1)+17, 18
0: Do not set the data at completion with an error	Normal completion	0: Invalid	0 (Clear)	0 (Clear)
	Error completion	0: Invalid	0 (Clear)	0 (Clear)
1: Set the data at completion with an error	Normal completion	0: Invalid	0 (Clear)	0 (Clear)
	Error completion	1: Valid	Set the information at completion with an error.	Set the information at completion with an error.

■Request frame

Device	Item	Description	Setting range	Set by
(s2)+0	Request data length	Specify the data length from the monitoring timer to the request data. (In units of bytes)	1 to 2000	User
(s2)+1	Monitoring timer	This timer sets the waiting time for the target device that received a request message to wait for response to processing request issued to the request destination. (Unit: Increments of 250 ms) • 0: Infinite wait • 1 to 65535: 1 to 65535 ×250 ms	0 to 65535	User
(s2)+2 to (s2)+n	Request data	The request data of the SLMP message is stored.	—	User

■Response frame

Device	Item	Description	Setting range	Set by
(d1)+0	Response data length	The data length from the end code to the response data is stored. (In units of bytes)	2 to 2000	System
(d1)+1	End code	The result of command processing is stored. In normal end, 0 is stored. In abnormal end, an error code set by the target device is stored.*1	—	System
(d1)+2 to (d1)+n	Response data	Execution results for the request data are set. (Some commands do not return response data.)	—*2	System

*1 For the set error code and the corresponding error description, check the specifications of the target device.

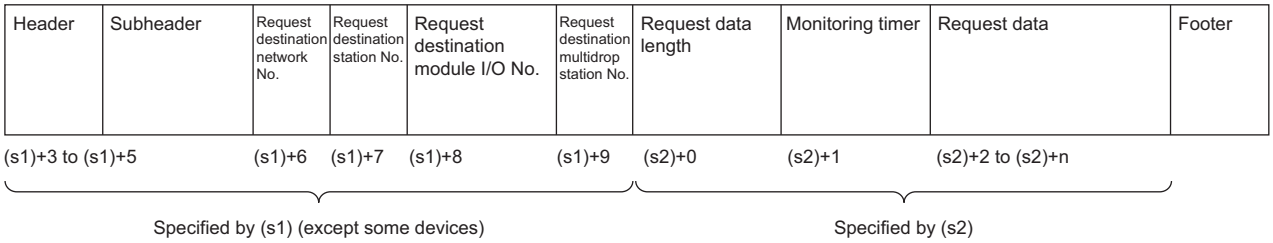
*2 For details of the response data, refer to the MELSEC iQ-F FX5 User's Manual (Communication).

Processing details

- This instruction sends the request frame in the device specified by (s2) and later to the target device specified by the target device IP address in the control data. When a response message is received from the target device, it is stored in the device specified by (d1).

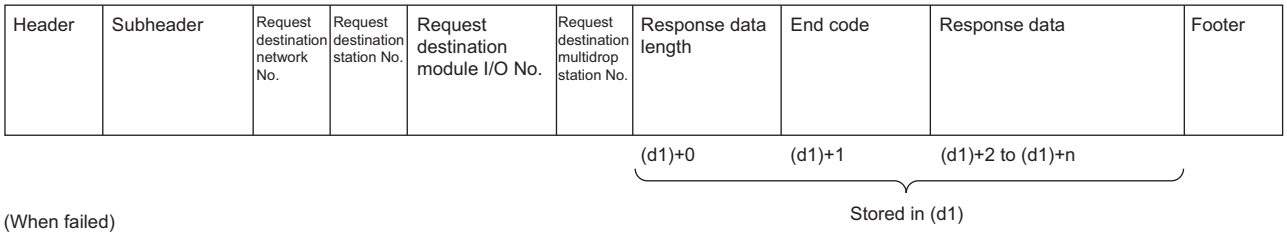
The following figures show the request data and the response data in normal/abnormal end.

●Request data

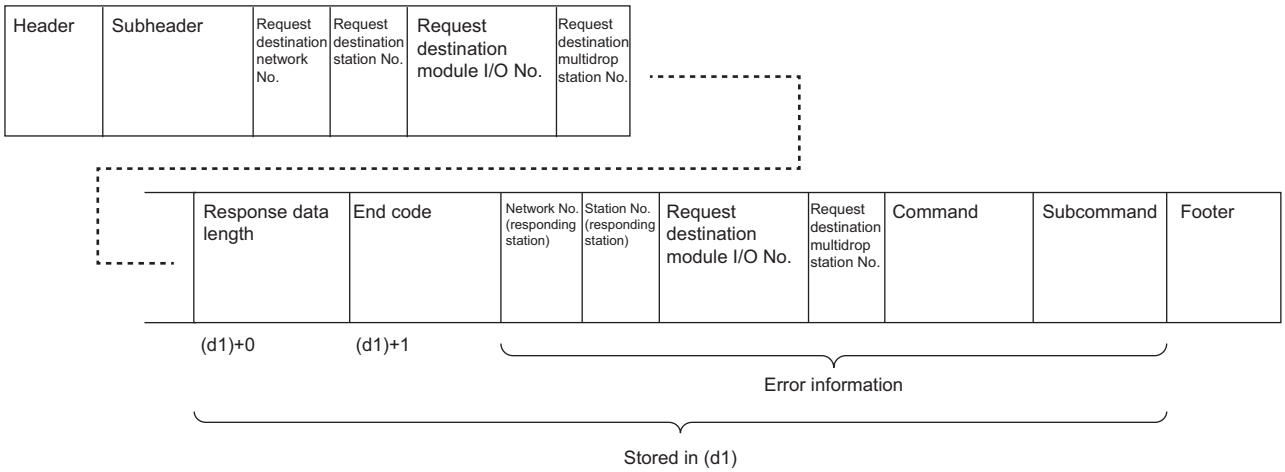


●Response data

(When completed)



(When failed)



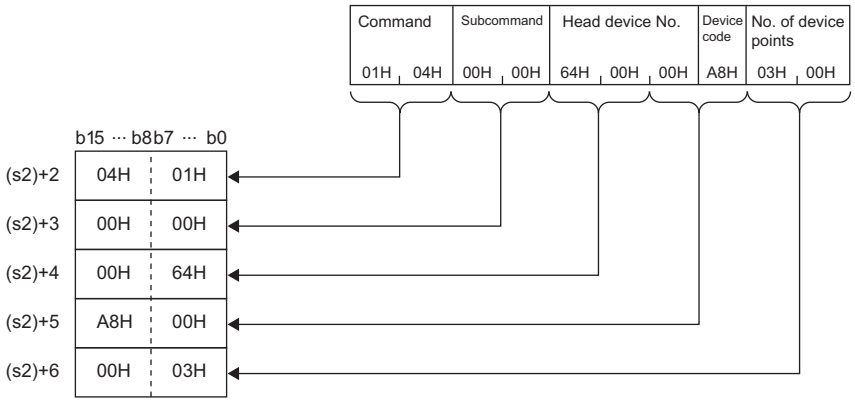
- The SP.SLMPSND instruction communicates using UDP. Set the target device to use UDP.
- The SP.SLMPSND instruction communicates in binary code. Match the setting of the target device also with the binary code.
- The normal or abnormal completion of the SP.SLMPSND instruction can be confirmed with the completion device (d2) specified in the setting data and the completion status indication device (d2)+1.

Device	Operation
Completion device (d2)	The device is turned on by END processing for the scan in which the SP.SLMPSND instruction is completed and turned off by next END processing.
Completion status indication device (d2)+1	The device is turned on or off depending on the status when the SP.SLMPSND instruction is completed. When completed normally: The device remains off. When completed abnormally: The device is turned on by END processing for the scan in which the SP.SLMPSND instruction is completed and turned off by next END processing.

Ex.

Sending "Device Read (command: 0401H)" which reads the value in D100 to D102

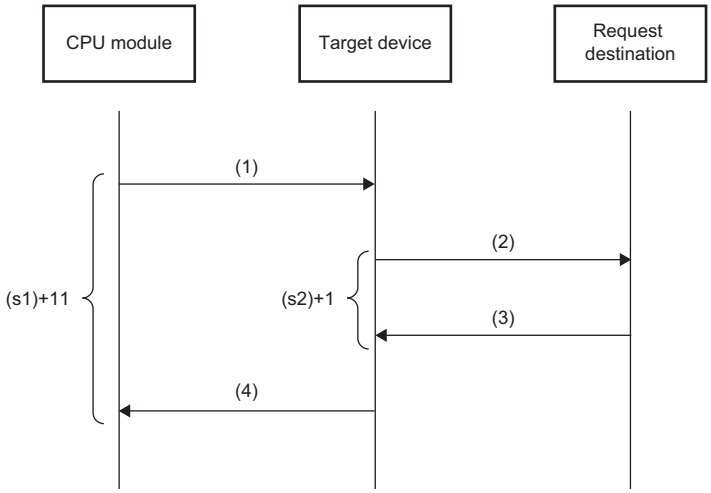
●Request data



13

Precautions

Specify the arrival monitoring time ((s1)+11) of the control data and monitoring timer ((s2)+1) of the request frame so that the arrival monitoring time ≥ monitoring timer.



(1) Request message
(2) Processing request from target device to request destination
(3) Processing response from request destination to target device
(4) Response message

Point

The SP.SLMPSND instruction is successfully completed even if the target device returns an abnormal response. When the SP.SLMPSND instruction is completed successfully, whether the response is normal or abnormal can be identified by the end code of the response frame. When an abnormal response is returned, check the manual of the SLMP-compatible device being used and take corrective action.

Operation error

Error code (SD0/SD8067)	Description
3405H	The value set to (s1)+2 as own station channel is out of the range, 1 to 9.
	The value set to (s2)+0 as the request data length is out of the range, 1 to 2000.

Upon completion with an error, the completion status indication device (d2)+1 is turned on and an error code is stored in the completion status (s1)+1. For the error code stored in the completion status (s1)+1, refer to the following.

MELSEC iQ-F FX5 User's Manual (Communication)

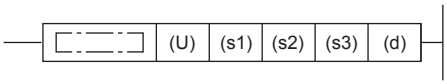
13.5 File Transfer Function Instruction

Sending FTP client files

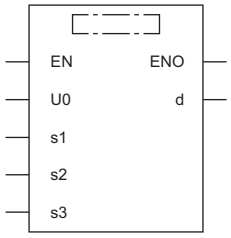
SP.FTPPUT

FX5S FX5UJ FX5U FX5UC

This instruction sends files in the CPU module, which are specified by (s2), to the folder path of the FTP server, which is specified by (s3).

Ladder	ST
	ENO:=SP_FTPPUT(EN,U0,s1,s2,s3,d);

FBD/LD



("SP.FTPPUT" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy (Input the character string ['U0'].)	—	String	—*2
(s1)	Start device where control data is stored	Refer to the control data.(☞ Page 1055)	Word	ANY16_ARRAY (Number of elements: 4)
(s2)	Name of files stored in the CPU module (transfer source)*3	—	Unicode string*4	ANYSTRING_DOUBLE
(s3)	Folder path of the FTP server (transfer destination)*3	—	Unicode string*4	ANYSTRING_DOUBLE
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. In the case of abnormal completion, also (d)+1 will be turned on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Displayed as U0 in the ST language and FBD/LD language.
*2 Specify the dummy character string with its character string constant.
*3 Unicode string or the start device where the Unicode string is stored.
*4 Even though the data type is Unicode string, only one-byte alphanumeric characters, symbols, and kana characters; and two-byte characters (Shift JIS codes) can be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	○	—
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	○	—
(s3)	—	○	—	—	—	—	○	—	—	○	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.
*2 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s1)+0	Application setting area	b15 … b4 b3 b2 b1 b0 0 (2) (1) 0 (1) Transfer completion file delete setting (b2) Specify whether to delete transfer completion files or not. • 0: Do not delete • 1: Delete (2) Temporary file create setting (b3) Specify whether to create a temporary file (FTPCLI_I.TMP) during the file transfer processing or not. Setting this bit to 0 prevents files from becoming undefined in the transfer destination when a cable is disconnected or power is shut off during the file transfer processing. • 0: Create • 1: Do not create	As shown on the left	User
(s1)+1	Completion status	The completion status is stored upon completion of the instruction. • 0000H: Completed successfully • Other than 0000H: Completed with an error (error code)	—	System
(s1)+2	Total number of files to be transferred	The total number of files to be transferred by the SP.FTPPUT instruction is stored.	—	System
(s1)+3	Number of transferred files	The number of transferred files is stored.	—	System

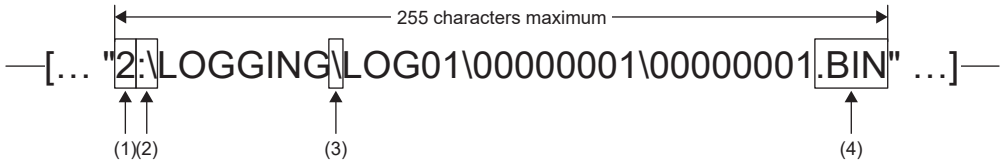
Processing details

- This instruction sends files in the CPU module, which are specified by (s2), to the folder path of the FTP server, which is specified by (s3). The CPU module opens a connection with the FTP server set in the module parameters (FTP Client Settings) at execution of the instruction, and closes a connection after sending files. For details on the parameter setting, refer to the following.

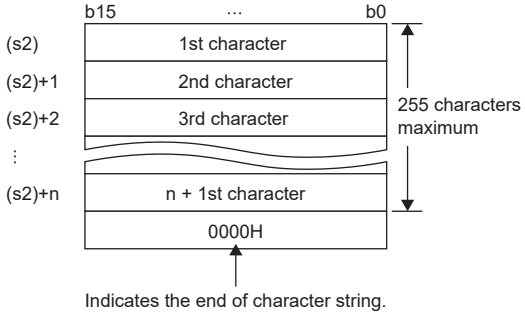
MELSEC iQ-F FX5 User's Manual (Communication)

- The total number of files to be transferred by the SP.FTPPUT instruction is stored in (s1)+2, and the number of transferred files is stored in (s1)+3.

• Specify the transfer source drive number (2) of the CPU module, the folder path where the files are stored, and the file name (including an extension) in (s2) in Unicode string. The only files which can be specified as a path are those stored under the LOGGING or DEBUG folder in the SD memory card (Drive 2). The maximum number of characters used in a file path is 255. The maximum number of characters used in a path, excluding the file name, is 246 (not including a delimiter). Use one-byte '\' or '/' as a delimiter to specify the boundaries between the elements in a file path.



(1)Drive numbers that can be specified are 2. (MELSEC iQ-F FX5 User's Manual (Application))
(2)Use one-byte ':\' or ':/' as a delimiter of the drive number.
(3)Use one-byte '\' or '/' as a delimiter of the folder path and file.
(4)The specified file name should include an extension.



• Wild card characters (*, ?) can be used in the file name or the extension specified in (s2).

Symbol	Description
*	An asterisk '*' is replaced with any character or string (including none) in a file name.
?	A question mark '?' is replaced with a character (excluding none) in a file name. ('?' can be used multiple times.)

Wild card characters do not recognize periods.

Using wild card characters in the following ways results in an error.

• Two or more asterisks '*' are used in a file name (before the period) or an extension. (Example: *abc*.txt)
• An asterisk '*' and a question mark '?' are used in a file name (before the period) or an extension. (Example: *ab?. txt)

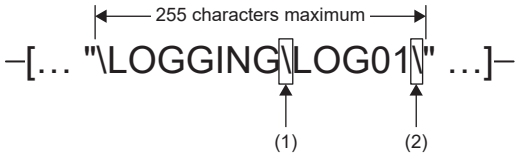
When any wild card character is used, the number of files that can be transferred is determined by the total size of the file names of the specified files. The specified files can be transferred when the number of these files and the total size of the file names satisfy the following condition. If a file transfer function instruction is executed without satisfying the following condition, the instruction completes with an error.

(Fi + NM) + 1 < 32768 [bytes]

N: Total number of files that match the wild card specification
Fi: Total size of the file names that match the wild card specification
M: Specific information size (Fixed value: 6 bytes)

• If an error occurs in any one of the files to be transferred during execution of the SP.FTPPUT instruction, the transfer processing will be stopped upon detection of the error, and the rest of target files will not be transferred.
• If the number of characters in the file path which includes a file name and an extension exceeds 255, files will not be transferred even though the wild card specification conditions are satisfied.

- Specify the folder path of the transfer destination FTP server in (s3) in Unicode string. The specified folder path shall be a relative path from home directory of the FTP server. Use one-byte '\' or '/' as a delimiter to specify the boundary of the folder path.*1 The maximum number of characters used in the folder path is 255. Note that the total number of characters in a folder path (including the delimiter at the end) and the file name specified in (s2) must be within the maximum path length supported by the transfer destination FTP server. The delimiter at the end of a string can be omitted. When omitted, '\' is assumed to be set at the end. If a nonexistent folder path is specified, a folder is automatically created by the system at execution of the instruction, and then the processing is performed.



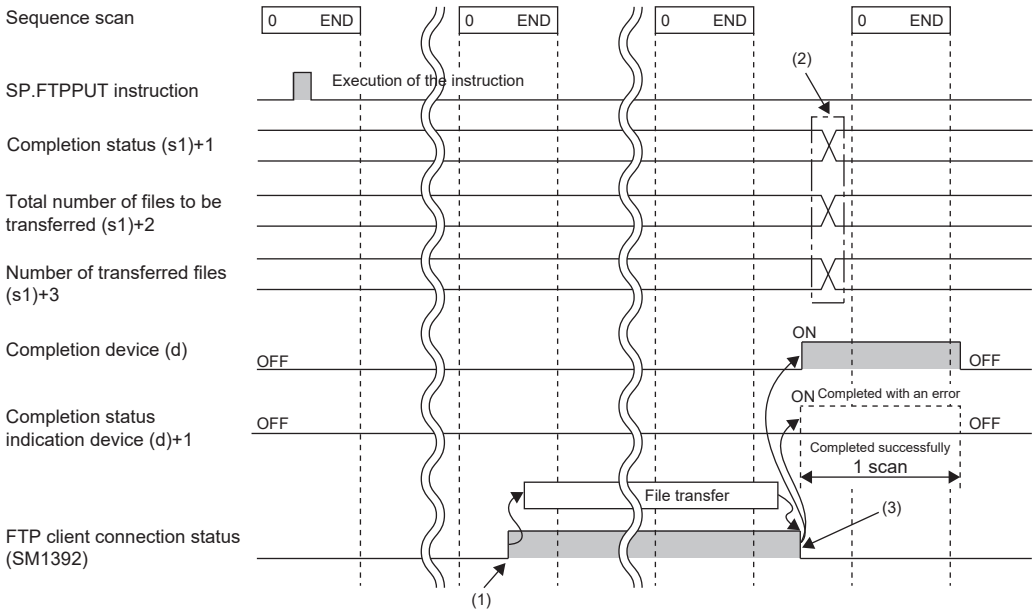
(1)Use one-byte '\' or '/' as a delimiter to specify the boundary of the folder path.*1
(2)The delimiter at the end of string can be omitted.

*1 Note that '\' cannot be used as a delimiter for some FTP servers.

- If a NULL character is specified by (s3) or only "0000H" is specified for the device, the CPU module directly accesses under the home directory of the FTP server. For details, follow the FTP server specifications.
- If a file with the same name exists in the transfer destination, the file will be overwritten.
- If a large file to be transferred is specified in the FTP client function, it may take long to transfer the file. For example: When the scan time is 5 ms, the time required for the SP.FTPPUT instruction to transfer a 16 MB file is approx. 548 seconds.
- The normal or abnormal completion of the SP.FTPPUT instruction can be confirmed with the completion device (d) specified in the setting data and the completion status indication device (d)+1.

Device	Operation
Completion device (d)	The device is turned on by END processing for the scan in which the SP.FTPPUT instruction is completed and turned off by next END processing.
Completion status indication device (d)+1	The device is turned on or off depending on the status when the SP.FTPPUT instruction is completed. When completed normally: The device remains off. When completed abnormally: The device is turned on by END processing for the scan in which the SP.FTPPUT instruction is completed and turned off by next END processing.

- The following figure shows the execution timing of the SP.FTPPUT instruction.



(1)SM1392 turns on during the END processing after the CPU module is connected to the FTP server.
(2)Values are stored upon completion of the instruction.
(3)When all files have been transferred, SM1392 turns off.

- SM1392 (FTP client connection status) is on while the CPU module is connected to the FTP server, and SM1392 turns off when disconnected.
- SM753 (File access in progress) turns on while the SP.FTPPUT instruction is being executed. While SM753 is on, the SP.FTPPUT instruction cannot be executed. If executed, no processing is performed.
- If the SP.FTPPUT instruction is executed while the SP.FTPPUT instruction is being executed, the instruction is ignored and not executed until the currently executing instruction completes. When the instruction is ignored, SM699 (Dedicated instruction skip flag) turns on.
- In the following cases, the instruction completes with an error: when there is no free space in the transfer destination; or when b3 (Temporary file create setting) of (s1)+0 is set to 0 (Create), but there is not enough free space for storing transfer-target files and a temporary file (same size as transfer-target files) in the transfer destination.
- When the number of files matching the wild card exceeds the upper limit number of transferable files, the instruction will be completed abnormally.
- When the wild card is designated in the file name, the file name in the transfer destination will be in all capital letters.

Precautions

- If a cable is disconnected, power is shut off, or the CPU module is reset during the file transfer processing, delete unnecessary files (such as a temporary file and undefined files) on the FTP server as needed. Then, transfer files again.
- When b2 (Transfer completion file delete setting) of (s1)+0 is set to 1 (Delete), note the following.

Item	Description
When files in the CPU module are transferred	Files required for the CPU module to operate are also deleted. If deleted, operations of the CPU module cannot be guaranteed.
When wild card characters are used to specify a file name	Required files may be deleted unintentionally.

- When b3 (Temporary file create setting) of (s1)+0 is set to 0 (Create), a temporary file of 12 characters (FTPCLI_I.TMP) will be created in the transfer destination. Therefore, set the folder path so that the total number of characters in the folder path and the temporary file does not exceed the maximum path length supported by the FTP server.
- Do not use any unsupported character(s) in the operand of the SP.FTPPUT instruction. Furthermore, if a file includes any unsupported character(s) in its file name, do not store the file in a folder which is a transfer source of the SP.FTPPUT instruction.

Operation error

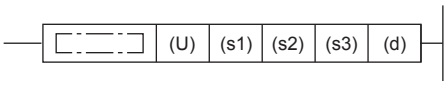
Error code (SD0/SD8067)	Description
2820H	There is no NULL code (0000H) in each setting area in the device/label memory in device specified by (s2), (s3) and later.
3405H	The number of characters in the string specified by (s2) exceeds 255.
	The number of characters in the path specified by (s2), excluding the file name, exceeds 246 (not including a delimiter).
	The total number of characters in the strings specified by (s2) (only the file name part, excluding the drive number and folder path) and (s3) exceeds 255.
	The drive number specified by (s2) is out of range.
3426H	A file name is not specified by (s2).
	The file name that cannot be transferred is specified by (s2).
	The delimiter used to separate the drive number in (s2) is neither ':\' nor ':/'.
	Two or more asterisks '*' are used in the file name (before the period) or the extension specified by (s2).
	An asterisk '*' and a question mark '?' are used in the file name (before the period) or the extension specified by (s2).
	Wild card characters, '*' and '?', are used in the string specified by (s3).
3430H	The SP.FTPPUT instruction was executed without setting FTP client parameters.
3582H	The SP.FTPPUT instruction is executed in an interrupt program.

Retrieving FTP client files

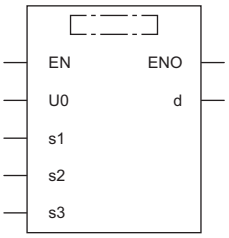
SP.FTPGET

FX5S FX5UJ FX5U FX5UC

This instruction retrieves files on the FTP server, which are specified by (s2), to the folder path of the CPU module, which is specified by (s3).

Ladder	ST
	ENO:=SP_FTPGET(EN,U0,s1,s2,s3,d);

FBD/LD



("SP.FTPGET" enters □.)

13

Setting data

■Description, range, data type

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy (Input the character string ['U0'].)	—	String	—*2
(s1)	Start device where control data is stored	Refer to the control data. (☞ Page 1060)	Word	ANY16_ARRAY (Number of elements: 4)
(s2)	Name of files stored in the FTP server (transfer source)*3	—	Unicode string*4	ANYSTRING_DOUBLE
(s3)	Folder path of the CPU module (transfer destination)*3	—	Unicode string*4	ANYSTRING_DOUBLE
(d)	Device that turns on for one scan upon completion of the instruction When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

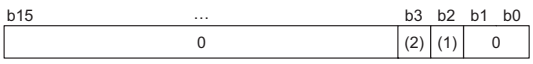
*1 Displayed as U0 in the ST language and FBD/LD language.
*2 Specify the dummy character string with its character string constant.
*3 Unicode string or the start device where the Unicode string is stored
*4 Even though the data type is Unicode string, only one-byte alphanumeric characters, symbols, and kana characters; and two-byte characters (Shift JIS codes) can be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	—	—	—	—	—	—	—	—	○	○
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	○	—
(s3)	—	○	—	—	—	—	○	—	—	○	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.
*2 T, ST, C, and FD cannot be used.

■Control data

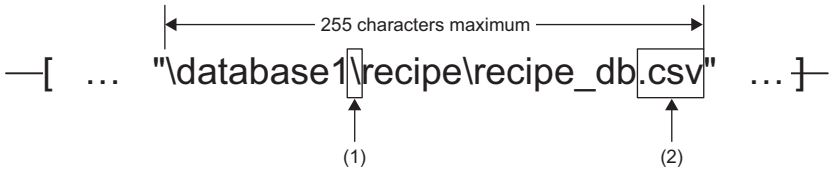
Operand: (s1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Application setting area	 (1) Transfer completion file delete setting (b2) Specify whether to delete transfer completion files or not. • 0: Do not delete • 1: Delete (2) Temporary file create setting (b3) Specify whether to create a temporary file (FTPCLI_I.TMP) during the file transfer processing or not. Setting this bit to 0 prevents files from becoming undefined in the transfer destination when a cable is disconnected or power is shut off during the file transfer processing. • 0: Create • 1: Do not create	Refer to the "Description" column.	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0000H: Completed successfully • Other than 0000H: Completed with an error (error code)	—	System
+2	Total number of files to be transferred	The total number of files to be transferred by the SP.FTPGET instruction is stored.	—	System
+3	Number of transferred files	The number of transferred files is stored.	—	System

Processing details

- This instruction retrieves files on the FTP server, which are specified by (s2), to the folder path of the CPU module, which is specified by (s3). The CPU module opens a connection with the FTP server set in the module parameters ("FTP Client Settings") at execution of the instruction, and closes a connection after retrieving files. For details on the parameter setting, refer to the following.

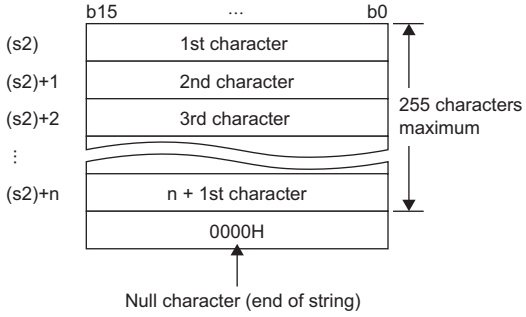
MELSEC iQ-F FX5 User's Manual (Communication)

- The total number of files to be transferred by the SP.FTPGET instruction is stored in (s1)+2, and the number of transferred files is stored in (s1)+3.
- Specify the folder path where the transfer source files are stored on the FTP server, and the file name (including an extension) in (s2) in Unicode string. The maximum number of characters used in a file path is 255. The specified file path shall be a relative path from home directory of the FTP server. Use one-byte '\' or '/' as a delimiter to specify the boundary of the folder path or the file name.*1



(1)Use one-byte '\' or '/' as a delimiter of the folder path or file.*1
(2)The specified file name should include an extension.

*1 Note that '\' cannot be used as a delimiter for some FTP servers.



• Wild card characters (*, ?) can be used in the file name or the extension specified in (s2).

Symbol	Description
*	An asterisk '*' is replaced with any character or string (including none) in a file name.
?	A question mark '?' is replaced with a character (excluding none) in a file name. ('?' can be used multiple times.)

Wild card characters do not recognize periods.

Using wild card characters in the following ways results in an error.

• Two or more asterisks '*' are used in a file name (before the period) or an extension. (Example: *abc*.csv)

• An asterisk '*' and a question mark '?' are used in a file name (before the period) or an extension. (Example: *ab?. csv)

When any wild card character is used, the number of files that can be transferred is determined by the total size of the file names of the specified files. The specified files can be transferred when the number of these files and the total size of the file names satisfy the following condition. If a file transfer function instruction is executed without satisfying the following condition, the instruction completes with an error.

(Fi + NM) + 1 < 32768 [bytes]

N: Total number of files that match the wild card specification
Fi: Total size of the file names that match the wild card specification
M: Specific information size (Fixed value: 6 bytes)

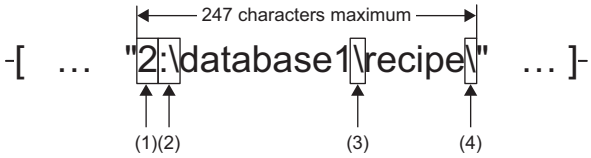
• If an error occurs in any one of the files to be retrieved during execution of the SP.FTPGET instruction, the transfer processing will be stopped upon detection of the error, and the rest of target files will not be transferred.

• Untransferable files will not be transferred even though the wild card specification conditions are satisfied.

• If the number of characters in the file path which includes a file name and an extension exceeds 255, files will not be transferred even though the wild card specification conditions are satisfied.

• If only a file name is specified by (s2), the CPU module directly accesses under the home directory of the FTP server. When specifying only a file name, start with a delimiter. For details, follow the FTP server specifications.

• Specify the folder path of the transfer destination CPU module in (s3) in Unicode string. Use one-byte '\' or '/' as a delimiter to specify the boundary of the folder path. The maximum number of characters in the folder path is 247 (246 when a delimiter at the end of the string is omitted). Note that the total number of characters in a folder path (including the delimiter at the end) and the file name specified in (s2) must be within the maximum path length (255 characters) supported by the CPU module. (When the delimiter at the end is omitted, it must not exceed 254 characters.) The delimiter at the end of a string can be omitted. When omitted, '\' is assumed to be set at the end. If a nonexistent folder path is specified, a folder is automatically created by the system at execution of the instruction, and then the processing is performed.



(1)Drive numbers that can be specified are 2. (MELSEC iQ-F FX5 User's Manual (Application))

(2)Use one-byte ':\' or ':/' as a delimiter of the drive number.

(3)Use one-byte '\' or '/' as a delimiter to specify the boundary of the folder path.

(4)The delimiter at the end of string can be omitted.

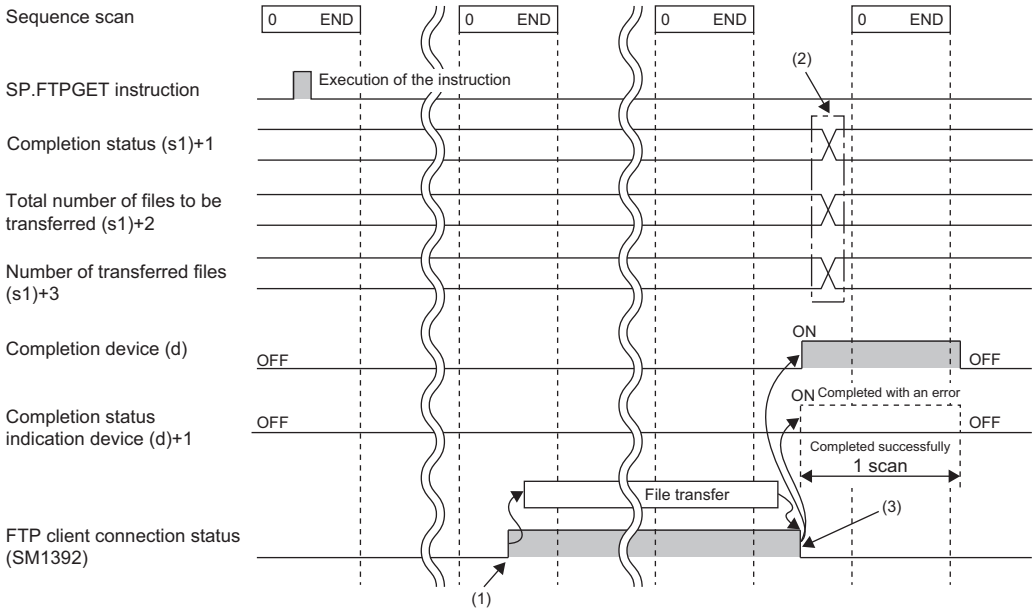
• If a file with the same name exists in the transfer destination, the file will be overwritten.

• When a file with large size is transferred, the transfer time may be longer. ((Example) Transfer time of 16-MB file with scan time of 5 ms: Approx. 450 sec)

• The execution status and the completion status of the SP.FTPGET instruction can be checked with the completion device (d) and the completion status indication device (d)+1.

Device	Operation
Completion device (d)	This device turns on during END processing of the scan where the SP.FTPGET instruction completes, and turns off during the next END processing.
Completion status indication device (d)+1	This device turns on or off depending on the completion status of the SP.FTPGET instruction. • When completed successfully: The device remains off. • When completed with an error: The device turns on during END processing of the scan where the SP.FTPGET instruction completes, and turns off during the next END processing. In addition, an error code is stored in the device specified by (s1)+1.

• The following figure shows the execution timing of the SP.FTPGET instruction.



(1)SM1392 turns on during the END processing after the CPU module is connected to the FTP server.
(2)Values are stored upon completion of the instruction.
(3)When all files have been transferred, SM1392 turns off.

• SM1392 (FTP client connection status) is on while the CPU module is connected to the FTP server, and SM1392 turns off when disconnected.
• SM753 (File access in progress) turns on while the SP.FTPGET instruction is being executed. While SM753 is on, the SP.FTPGET instruction cannot be executed. If executed, no processing is performed.
• If the SP.FTPGET instruction is executed while the SP.FTPPUT or SP.FTPGET instruction is being executed, the instruction is ignored and not executed until the currently executing instruction completes. When the instruction is ignored, SM699 (Dedicated instruction skip flag) turns on.
• In the following cases, the instruction completes with an error: when there is no free space in the transfer destination; or when b3 (Temporary file create setting) of (s1)+0 is set to 0 (Create), but there is not enough free space for storing transfer-target files and a temporary file (same size as transfer-target files) in the transfer destination.
• Even though the operating status of the CPU module is switched from RUN to STOP during the file transfer processing, the CPU module continues the processing until completed.

Precautions

• If a cable is disconnected, power is shut off, or the CPU module is reset during the file transfer processing, transfer files again. In those cases, unnecessary files will be deleted in the following ways.

Unnecessary file	Delete operation
Temporary file (FTPCLI_I.TMP)	A temporary file left in the FTP client will be automatically deleted at the start of next file transfer processing to the same folder.
Undefined transfer target file	An undefined transfer target file left in the FTP client will be overwritten during next file transfer processing to the same file.

• When a file with the same name has already existed in the transfer destination and its access is restricted by the file password function, the SP.FTPGET instruction completes with an error.
• When b3 (Temporary file create setting) of (s1)+0 is set to 0 (Create), set the maximum number of characters in the folder path of the CPU module to 243 (242 when a delimiter at the end of the string is omitted). Since a temporary file of 12 characters (FTPCLI_I.TMP) is created in the transfer destination, set the folder path so that the total number of characters in the folder path and the temporary file does not exceed the maximum path length (255 characters) supported by the CPU module.
• Do not use unsupported characters for the operands of the SP.FTPGET instruction. Do not store any file with a file name containing unsupported characters in the source folder to transfer the SP.FTPGET instruction.

Operation error

Error code (SD0)	Description
2820H	There is no NULL code (0000H) in each setting area in the device/label memory in device specified by (s2), (s3) and later.
3405H	The number of characters in the string specified by (s2) exceeds 255.
	The number of characters in the path specified by (s2), excluding the file name, exceeds 246 (not including a delimiter).
	The total number of characters in the strings specified by (s3) and (s2) (only the file name part) exceeds 255.
	The drive number specified by (s3) is out of range.
3426H	A file name is not specified by (s2).
	The file name that cannot be transferred is specified by (s2).
	The delimiter used to separate the drive number in (s3) is neither ':\' nor ':/'.
	Two or more asterisks '*' are used in the file name (before the period) or the extension specified by (s2).
	An asterisk '*' and a question mark '?' are used in the file name (before the period) or the extension specified by (s2).
	Wild card characters, '*' and '?', are used in the string specified by (s3).
3430H	The SP.FTPGET instruction was executed without setting FTP client parameters.
3582H	The SP.FTPGET instruction is executed in an interrupt program.

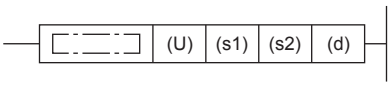
13.6 Ethernet Module

Opening a connection

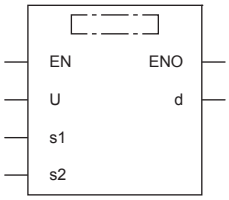
GP.OPEN

FX5S FX5UJ FX5U FX5UC

This instruction opens a connection.

Ladder diagram	Structured text
	ENO:=GP_OPEN(EN,U,s1,s2,d);

FBD/LD



("GP_OPEN" enters □.)

Setting data

■Descriptions, ranges, and data types

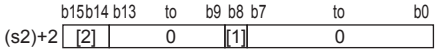
Operand	Description	Range	Data type	Data type (label)
(U)	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)	Connection number	1 to 32	16-bit unsigned binary	ANY16
(s2)	Head device number for storing the control data	Refer to Control data (☞ Page 1065)	Word	ANY16_ARRAY (Number of elements: 10)
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. If the instruction is completed with an error, (d)+1 is also turned on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	○	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by[*1]
(s2)+0	Execution type/ completion type	Specify whether to use the parameter value set using the engineering tool or to use the set values of the control data (s2)+2 to (s2)+6 during the open processing of the connection. 0000H: The open processing is performed with the settings configured using "External Device Configuration" of the engineering tool. 8000H: The open processing is performed with the set values of the control data (s2)+2 to (s2)+6.	0000H 8000H	User
(s2)+1	Completion status	The status at the completion of the instruction is stored. 0000H: Completed successfully Other than 0000H: Completed with an error (error code) For error codes, refer to the manual below. MELSEC iQ-F FX5 Ethernet Module User's Manual MELSEC iQ-F FX5 EtherNet/IP Module User's Manual	—	System
(s2)+2	Application setting area	 [1] Communication method (protocol) 0: TCP/IP 1: UDP/IP [2] Open method 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open	As shown on the left	User
(s2)+3	Host station port number	Specify the host station port number.	1 to 5548, 5570 to 65534 (0001H to 15ACH, 15C2H to FFFEH)[*3]	
(s2)+4 (s2)+5	Target device IP address[*2]	Specify the IP address of the target device.	1 to 3758096382 (00000001H to DFFFFFFEH)	
(s2)+6	Target device port number[*2]	Specify the port number of the target device.	1 to 65534 (0001H to FFFEH)	
(s2)+7 to (s2)+9	—	Use prohibited	—	System

*1 User: Data to be set before the execution of the instruction. System: The Ethernet module stores the execution result of the instruction.

*2 When Unpassive open is selected, the target device IP address and target device port number are ignored.

*3 Of the host station port numbers, 1 to 1023 (0001H to 03FFH) are generally reserved port numbers and 61440 to 65534 (F000H to FFFEH) are used by other communication functions. Thus, using 1024 to 5548 and 5570 to 61439 (0400H to 15ACH and 15C2H to EFFFH) as the port numbers is recommended. Do not specify 5549 to 5569 (15ADH to 15C1H) since they are used by the system.

Processing details

This instruction opens the connection specified by (s1).

The setting value used by the open processing is selected by (s2)+0.

The completion of the GP.OPEN instruction can be checked using the completion devices (d)+0 and (d)+1.

- Completion device (d)+0: Turns ON during the END processing for the scan in which the GP.OPEN instruction is completed, and turns OFF during the next END processing.
- Completion device (d)+1: Turns ON or OFF depending on the status of when the GP.OPEN instruction is completed.

Status	Description
When completed normally	The device does not change (remains OFF).
When completed with an error	The device turns ON during the END processing for the scan in which the GP.OPEN instruction is completed, and turns OFF during the next END processing.

The connection in which no protocol is set with the parameter can be opened and used. In this case, specify 8000H in (s2)+0 and the contents of the open processing in (s2)+2 to (s2)+6.

Operation error

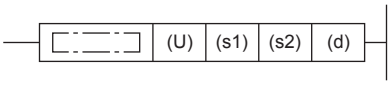
Error code ((s2)+1)	Description
C000H to CFFFH	Refer to MELSEC iQ-F FX5 Ethernet Module User's Manual. Refer to MELSEC iQ-F FX5 EtherNet/IP Module User's Manual.

Closing a connection

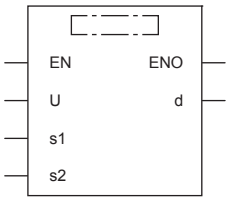
GP.CLOSE

FX5S FX5UJ FX5U FX5UC

This instruction closes a connection.

Ladder diagram	Structured text
	ENO:=GP_CLOSE(EN,U,s1,s2,d);

FBD/LD



("GP_CLOSE" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)	Connection number	1 to 32	16-bit unsigned binary	ANY16
(s2)	Head device number for storing the control data	Refer to Control data (☞ Page 1068)	Word	ANY16_ARRAY (Number of elements: 2)
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. If the instruction is completed with an error, (d)+1 is also turned on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	○	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by*1
(s2)+0	System area	—	—	—
(s2)+1	Completion status	The status at the completion of the instruction is stored. 0000H: Completed successfully Other than 0000H: Completed with an error (error code) For error codes, refer to the manual below. MELSEC iQ-F FX5 Ethernet Module User's Manual MELSEC iQ-F FX5 EtherNet/IP Module User's Manual	—	System

*1 System: The Ethernet module stores the execution result of the instruction.

Processing details

This instruction closes the connection specified by (s1). (Connection disconnection)

The completion of the GP.CLOSE instruction can be checked using the completion devices (d)+0 and (d)+1.

- Completion device (d)+0: Turns ON during the END processing for the scan in which the GP.CLOSE instruction is completed, and turns OFF during the next END processing.
- Completion device (d)+1: Turns ON or OFF depending on the status when the GP.CLOSE instruction is completed.

Status	Description
When completed normally	The device does not change (remains OFF).
When completed with an error	The device turns ON during the END processing for the scan in which the GP.CLOSE instruction is completed, and turns OFF during the next END processing.

Operation error

Error code ((s2)+1)	Description
C000H to CFFFH	Refer to MELSEC iQ-F FX5 Ethernet Module User's Manual. Refer to MELSEC iQ-F FX5 EtherNet/IP Module User's Manual.

Point

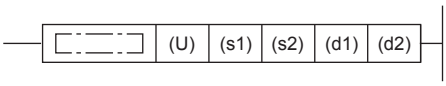
Do not execute the GP.CLOSE instruction when Passive open is selected. Since the open completion signal and open request signal of the corresponding connection turn OFF and close processing is executed, the communication is disabled.

Reading receive data

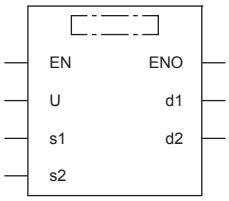
GP.SOCRCV

FX5S FX5UJ FX5U FX5UC

This instruction reads the receive data.

Ladder diagram	Structured text
	ENO:=GP_SOCRCV(EN,U,s1,s2,d1,d2);

FBD/LD



("GP_SOCRCV" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)	Connection number	1 to 32	16-bit unsigned binary	ANY16
(s2)	Head device number for specifying the control data	Refer to Control data (☞ Page 1070)	Word	ANY16_ARRAY (Number of elements: 2)
(d1)	Head device number for storing the receive data	—	Word	ANY16*1
(d2)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	○	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by*1
(s2)+0	System area	—	—	—
(s2)+1	Completion status	The status at the completion of the instruction is stored. 0000H: Completed successfully Other than 0000H: Completed with an error (error code) For error codes, refer to the manual below. MELSEC iQ-F FX5 Ethernet Module User's Manual MELSEC iQ-F FX5 EtherNet/IP Module User's Manual	—	System
(d1)+0	Receive data length	The data length of the data read from the socket communication receive data area is stored. (Number of bytes)	0 to 2046	System
(d1)+1 to (d1)+n	Receive data	The data read from the socket communication receive data area is sequentially stored.	—	System

*1 System: The Ethernet module stores the execution result of the instruction.

Point

- When the GP.SOCRCV instruction is executed, reading data from the socket communication receive data area is executed with the END processing. Thus, executing the GP.SOCRCV instruction extends the scan time.
- When the data of odd-number of bytes is received, invalid data is stored in the higher byte of the device where the last receive data is stored.

Processing details

In the END processing after the execution of the GP.SOCRCV instruction, the receive data of the connection specified by (s1) is read from the socket communication receive data area.

The completion of the GP.SOCRCV instruction can be checked using the completion devices (d2)+0 and (d2)+1.

- Completion device (d2)+0: Turns ON during the END processing for the scan in which the GP.SOCRCV instruction is completed, and turns OFF during the next END processing.
- Completion device (d2)+1: Turns ON or OFF depending on the status when the GP.SOCRCV instruction is completed.

Status	Description
When completed normally	The device does not change (remains OFF).
When completed with an error	The device turns ON during the END processing for the scan in which the GP.SOCRCV instruction is completed, and turns OFF during the next END processing.

Operation error

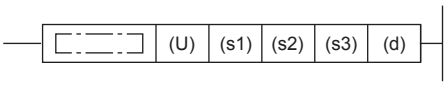
Error code ((s2)+1)	Description
C000H to CFFFH	Refer to MELSEC iQ-F FX5 Ethernet Module User's Manual. Refer to MELSEC iQ-F FX5 EtherNet/IP Module User's Manual.

Sending data

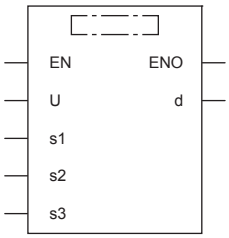
GP.SOCSND

FX5S FX5UJ FX5U FX5UC

This instruction sends data.

Ladder diagram	Structured text
	ENO:=GP_SOCSND(EN,U,s1,s2,s3,d);

FBD/LD



("GP_SOCSND" enters ☐.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)	Connection number	1 to 32	16-bit unsigned binary	ANY16
(s2)	Head device number for specifying the control data	Refer to Control data (☞ Page 1072)	Word	ANY16_ARRAY (Number of elements: 2)
(s3)	Head device number for storing the send data	—	Word	ANY16*1
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. If the instruction is completed with an error, (d)+1 is also turned on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U☐\G☐	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	○	○	—	—	—	—	○	○	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by*1
(s2)+0	System area	—	—	—
(s2)+1	Completion status	The status at the completion of the instruction is stored. 0000H: Completed successfully Other than 0000H: Completed with an error (error code) For error codes, refer to the manual below. MELSEC iQ-F FX5 Ethernet Module User's Manual MELSEC iQ-F FX5 EtherNet/IP Module User's Manual	—	System
(s3)+0	Send data length	Specifies the send data length. (Number of bytes)	1 to 2046	User
(s3)+1 to (s3)+n	Send data	Specifies the send data.	—	User

*1 User: Data to be set before the execution of the instruction. System: The Ethernet module stores the execution result of the instruction.

Point

When TCP is used, specify send data length that is smaller than the maximum window size of the target device (Receive data buffer of TCP). Data whose size exceeds the maximum window size of the target device cannot be sent.

Processing details

This instruction send the data set by (s3) to the target device of the connection specified by (s1).

The completion of the GP.SOCSND instruction can be checked using the completion devices (d)+0 and (d)+1.

- Completion device (d)+0: Turns ON during the END processing for the scan in which the GP.SOCSND instruction is completed, and turns OFF during the next END processing.
- Completion device (d)+1: Turns ON or OFF depending on the status when the GP.SOCSND instruction is completed.

Status	Description
When completed normally	The device does not change (remains OFF).
When completed with an error	The device turns ON during the END processing for the scan in which the GP.SOCSND instruction is completed, and turns OFF during the next END processing.

Operation error

Error code ((s2)+1)	Description
C000H to CFFFH	Refer to MELSEC iQ-F FX5 Ethernet Module User's Manual. Refer to MELSEC iQ-F FX5 EtherNet/IP Module User's Manual.

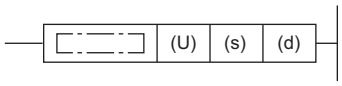
14 CC-LINK IE TSN INSTRUCTION

14.1 Own Station Number/IP Address Setting

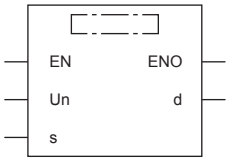
G(P).UINI

FX5S FX5UJ FX5U FX5UC

These instructions specify the station number and IP address to the own station which they are not set yet.

Ladder	ST
	ENO:=G_UINI(EN,Un,s,d); ENO:=GP_UINI(EN,Un,s,d);

FBD/LD



("G_UINI", "GP_UINI" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Number of installation position of the installed module	1H to 10H	16-bit unsigned binary	ANY16
(s)	Own station start device where control data is stored	Refer to (☞ Page 1074)	Device name	ANY16*2
(d)	Device of the own station, which turns on for one scan upon completion of the instruction. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.
*2 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

*2 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s)+0	Not used	—	—	—
(s)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System
(s)+2	Change target specification	Specify the parameters to be changed (station number and IP address). 0003H (fixed)	0003H	User
(s)+3	Station number of own station	Specifies the station number to be set. • 0: Master station • 1 to 120: Local station	0 to 120	User
(s)+4	IP address	Specify the IP address (third and fourth octets) of the own station. b15 ··· b8 b7 ··· b0 3 | 4 • b8 to b15: Third octet • b0 to b7: Fourth octet	00000001H to DFFFFFFEH	User
(s)+5		Specify the IP address (first and second octets) of the own station. b15 ··· b8 b7 ··· b0 1 | 2 • b8 to b15: First octet • b0 to b7: Second octet		User
(s)+6 to (s)+9	Not used	—	—	—

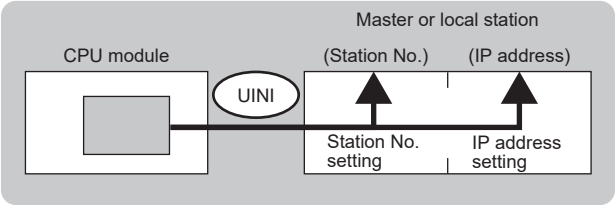
Point

- The G(P).UINI instruction can be executed only once.
- If the G(P).UINI instruction is executed after the station number and IP address have been determined, it will complete with an error.
- If the G(P).UINI instruction completes with an error before the station number and IP address are determined, take corrective action to correct the error content before retrying to execute the instruction.
- Set the subnet mask and default gateway using the engineering tool as necessary.

Processing details

- Sets the station number and IP address for the own station.

[Own station]



- The execution status and the completion status of the G(P).UINI instruction can be checked with the completion device (d) and the completion status indication device (d)+1.

Device	Operation
Completion device (d)	The device is turned on by END processing for the scan in which the G(P).UINI instruction is completed and turned off by next END processing.
Completion status indication device (d)+1	The device is turned on or off depending on the status when the G(P).UINI instruction is completed. When completed normally: The device remains off. When completed abnormally: The device is turned on by END processing for the scan in which the G(P).UINIT instruction is completed and turned off by next END processing.

Precautions

Do not set the broadcast address and the reserved address to the IP address ((s)+4, 5). If set, the instruction may not be executed properly.

Operation error

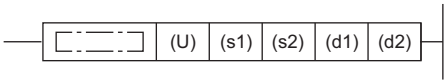
Error code ((s)+1)	Description
C000H to CFFFH D000H to DFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual.

14.2 Sending an SLMP Frame

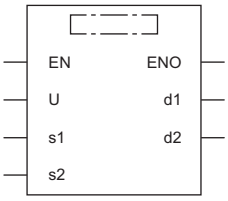
G(P).SLMPSND

FX5S FX5UJ FX5U FX5UC

These instructions send/receive an SLMP message to the SLMP-compatible device on the same network.

Ladder	ST
	ENO:=G_SLMPSND(EN,U,s1,s2,d1,d2); ENO:=GP_SLMPSND(EN,U,s1,s2,d1,d2);

FBD/LD



("GP_SLMPSND" enters ☐.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)	Position number of the module connected	00H to FEH	16-bit unsigned binary	ANY16
(s1)	Own station start device where control data is stored	☞ Page 1077 Control data	Device name	ANY16*1
(s2)	Own station start device where request data is stored	☞ Page 1078 Request data	Device name	ANY16*1
(d1)	Own station start device for storing response data	☞ Page 1078 Response data	Device name	ANY16*1
(d2)	Device of the own station, which turns on for one scan upon completion of the instruction. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

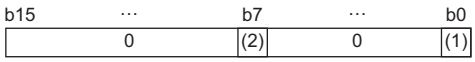
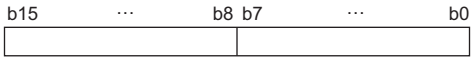
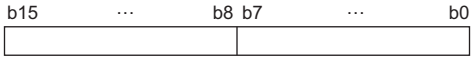
*1 When specifying data with a label, define the array so that an area required for operation can be secured, and specify the array label element.

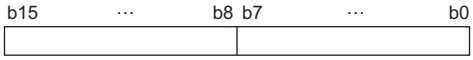
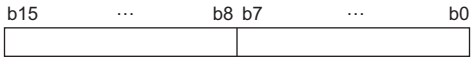
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U☐\G☐	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s1)+0	Execution/error completion type	 (1) Execution type (b0) • 0: Without arrival check (The instruction is regarded as completed when a request message is sent from own station.)*1 • 1: With arrival check (The instruction is regarded as completed when a response message is received from the target device.) (2) Error completion type (b7) Specify whether to set data at completion with an error. • 0: Do not set data in (s1)+13 and later at completion with an error. (Clear (s1)+13 and later.) • 1: Set data at completion with an error in (s1)+13 and later.	0000H 0001H 0080H 0081H	User
(s1)+1	Completion status	The instruction completion status is stored. 0000H: Normal Other than 0000H: Error (error code)	—	System
(s1)+2	Own station channel	Specify the channel to be used by the own station. Whether or not a serial number is given to the request message depends on the channel.*2 • 1: No serial number is given • 2 to 9: Serial number is given • 10 to 17: Station number extension frame	1 to 17	User
(s1)+3	IP address of target device (third and fourth octets)	Specify the IP address (third and fourth octets) of target device.  • b8 to b15: Third octet • b0 to b7: Fourth octet	00000001H to DFFFFFFEH (both (s1)+3 and (s1)+4 together) (1 to 3758096382)	User
(s1)+4	IP address of target device (first and second octets)	Specify the IP address (first and second octets) of target device.  • b8 to b15: First octet • b0 to b7: Second octet		User
(s1)+5	Destination port number	Specify the destination port number.	1 to 65534 (1 to FFFEH)	User
(s1)+6	Request destination network number	0000H (fixed)	0000H	User
(s1)+7	Request destination station number	00FFH (fixed)	00FFH	User
(s1)+8	Request destination module I/O number	Specify the request destination module I/O number. • 03E0H: Multiple CPU No. 1 • 03E1H: Multiple CPU No. 2 • 03E2H: Multiple CPU No. 3 • 03E3H: Multiple CPU No. 4 • 03FFH: Own station/control CPU	03E0H to 03E3H, 03FFH	User
(s1)+9	Request destination multidrop station number	Specify a multidrop number of the target device. If the device does not have the multidrop number, fixed to 0000H.	0000H to FFFEH	User
(s1)+10	Number of resends	Effective when the execution type specified by (s1)+0 is "1: With arrival check". ■At instruction execution Specify the number of resends to be performed if the instruction is not completed within the monitoring time specified by (s1)+11. • 0 to 15 (times) ■At completion of instruction The number of resends performed (result) is stored. • 0 to 15 (times)*3	0 to 15	User/ system
(s1)+11	Arrival monitoring time	Set the monitoring time until completion of processing. If processing is not completed within the monitoring time, the request is resent the number of times specified in (s1)+10. • 0: 10 seconds • 1 to 32767: 1 to 32767 seconds	0 to 32767	User

Device	Item	Description	Setting range	Set by
(s1)+12	Clock setting flag	The validity status (valid or invalid) of the data in (s1)+13 and later is stored. Note that the data in (s1)+13 and later is cleared when the instruction is completed successfully. • 0: Invalid • 1: Valid	—	System
(s1)+13	Clock data (Set only in an abnormal state)	Upper 8 bits: Month (01H to 12H) Lower 8 bits: Year (00H to 99H: Lower two digits of the year)	—	System
(s1)+14		Upper 8 bits: Hour (00H to 23H) Lower 8 bits: Day (01H to 31H)		
(s1)+15		Upper 8 bits: Second (00H to 59H) Lower 8 bits: Minute (00H to 59H)		
(s1)+16		Upper 8 bits: Year (00H to 99H: Upper two digits of the year) Lower 8 bits: Day of the week (00H (Sun.) to 06H (Sat.))		
(s1)+17	IP address of error detected device (third and fourth octets)	The IP address (third and fourth octets) of the station where an error was detected is stored.  • b8 to b15: Third octet • b0 to b7: Fourth octet	—	System
(s1)+18	IP address of error detected device (first and second octets)	The IP address (first and second octets) of the station where an error was detected is stored.  • b8 to b15: First octet • b0 to b7: Second octet	—	System

*1 If (s1)+0 is set to "0: Without arrival check", receive data is not set. Set 0 in (s1)+0 in the following cases:
· When a command that does not return a response message is used
· When a response message is not referred to

*2 Set the serial numbers when sending multiple request messages to the same SLMP-compatible device.
Serial numbers to be set are automatically numbered by the system.

*3 If an error was detected, the number of resends performed (result) between error detection and resend stop is stored.

■Request data

Device	Item	Description	Setting range	Set by
(s2)+0	Request data length	Specify the data length from the monitoring timer to the request data. (In units of bytes)	1 to 2000	User
(s2)+1	Monitoring timer	Set the waiting time for the external device that received a request message to wait for the response after it issued a processing request to the request destination. (Unit: Increments of 250 ms) • 0: Infinite wait • 1 to 65535: 1 to 65535 × 250 ms	0 to 65535	User
(s2)+2 to (s2)+n	Request data	The request data of the SLMP message is stored.	Refer to the specifications of the target device.	User

■Response data

Device	Item	Description	Setting range	Set by
(d1)+0	Response data length	The data length from the end code to the response data is stored. (In units of bytes)	—	System
(d1)+1	End code	The result of command processing is stored. In normal end, 0 is stored. In abnormal end, an error code set by the target device is stored.*1	—	System
(d1)+2 to (d1)+n	Response data	Execution results for the request data are set. (Some commands do not return response data.)	—*2	System

*1 For the set error code and the corresponding error description, check the specifications of the target device.

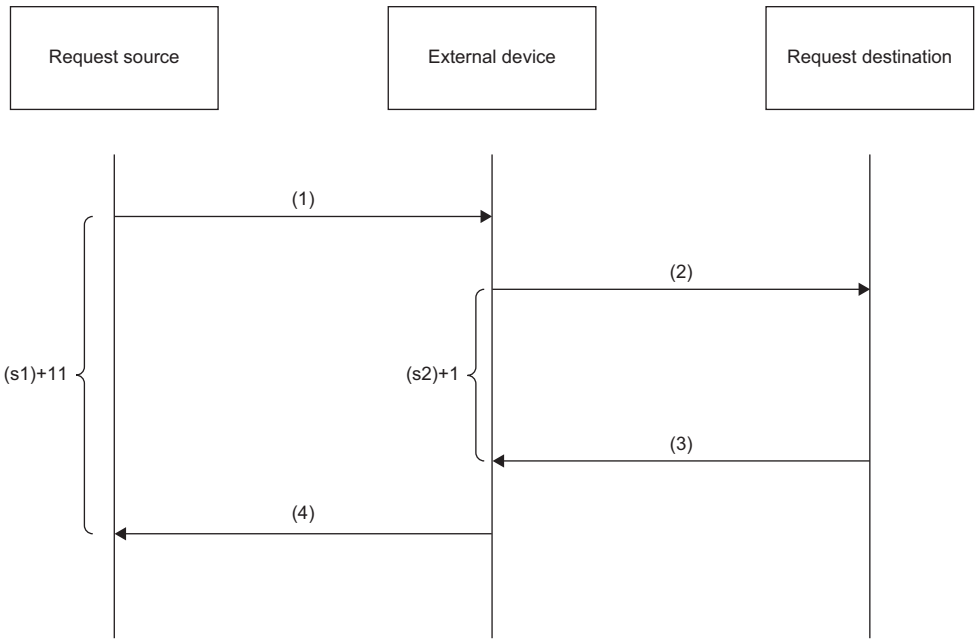
*2 For details on the response data, refer to the MELSEC iQ-F FX5 User's Manual (Communication).

Processing details

- This instruction sends the request data in the device specified by (s2) and later to the external device specified by the target device IP address in the control data. When a response message is received from the target device, it is stored in the device specified by (d1).
- The SLMPSND instruction uses the UDP/IP communication methods in binary code.
- The completion status of the SLMPSND instruction can be checked with the completion device (d2) and the completion status indication device (d2)+1.

Device	Operation
Completion device (d2)	This device turns on in the END processing of the scan where the SLMPSND instruction completes, and turns off in the next END processing.
Completion status indication device (d2)+1	This device turns on or off depending on the completion status of the SLMPSND instruction. When completed normally: The device remains off. When completed with an error: The device turns on in the END processing of the scan where the SLMPSND instruction completes, and turns off in the next END processing.

- When executing multiple SLMPSND instructions concurrently, be careful not to overlap the channels of the SLMPSND instructions. Multiple SLMPSND instructions specifying the same channels cannot be used concurrently.
- When the execution conditions of the SLMPSND instructions in the same channel are satisfied in the same sequence scan, only the SLMPSND instruction that has been executed first is enabled and the subsequent SLMPSND instructions are not executed. In addition, any subsequent SLMPSND instruction of the same channel setting as the SLMPSND instruction being executed is not executed.
- Specify the arrival monitoring time ((s1)+11) of the control data and monitoring timer ((s2)+1) of the request frame so that the arrival monitoring time ≥ monitoring timer.

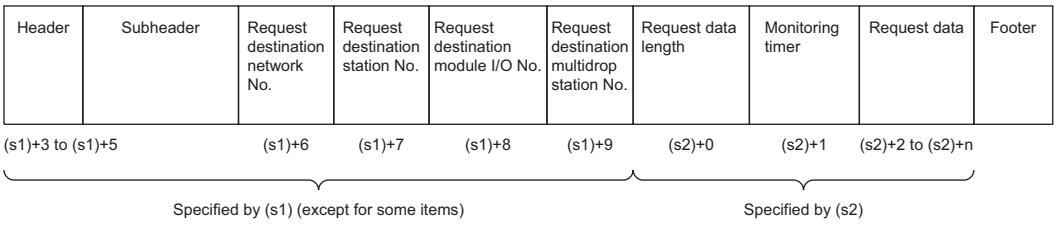


(1) Request message
(2) Processing request from target device to request destination
(3) Processing response from request destination to target device
(4) Response message

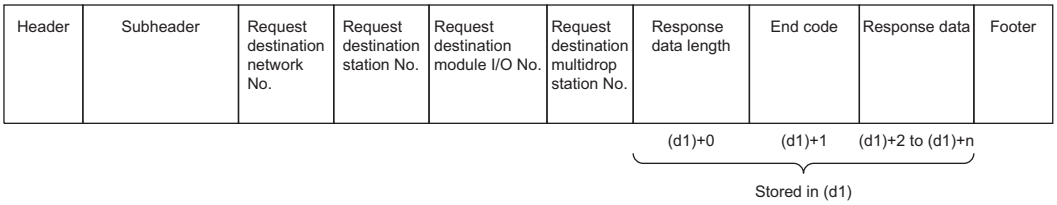
■Message format (3E/4E frame)

The following figures show the request data of 3E/4E frame and the response data in normal/abnormal end.

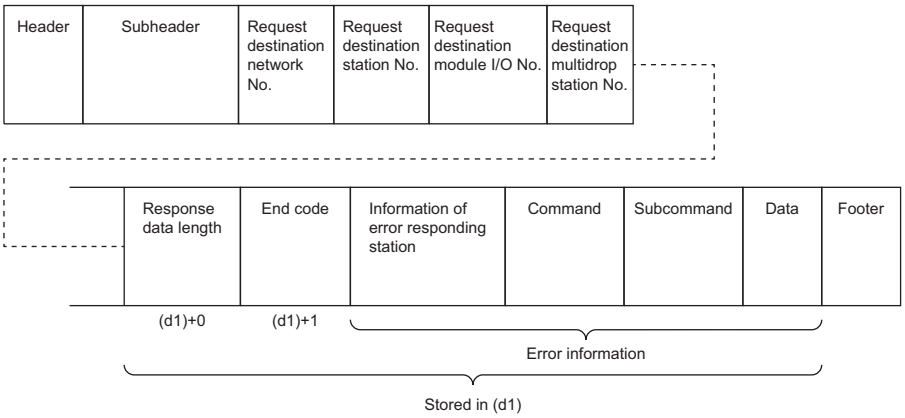
● Request message



● Response message (in normal end)



● Response message (in abnormal end)

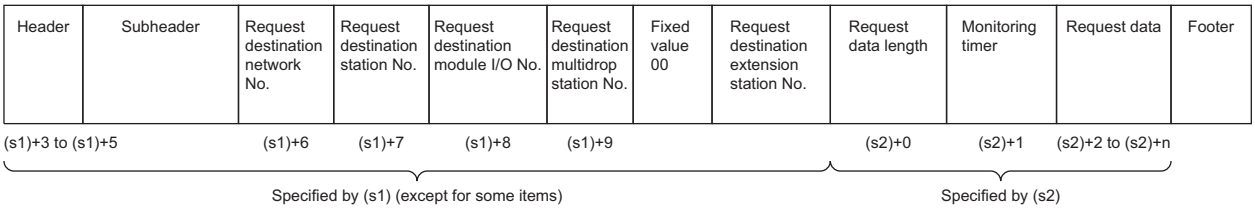


The network number, station number, request destination module I/O number, and multidrop station number are stored to the information of the error responding station.

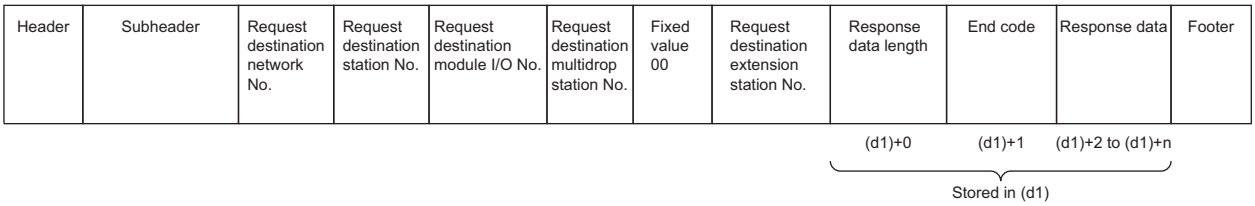
■Message format (station number extension frame)

The following figures show the request data of station number extension frame and the response data in normal/abnormal end.

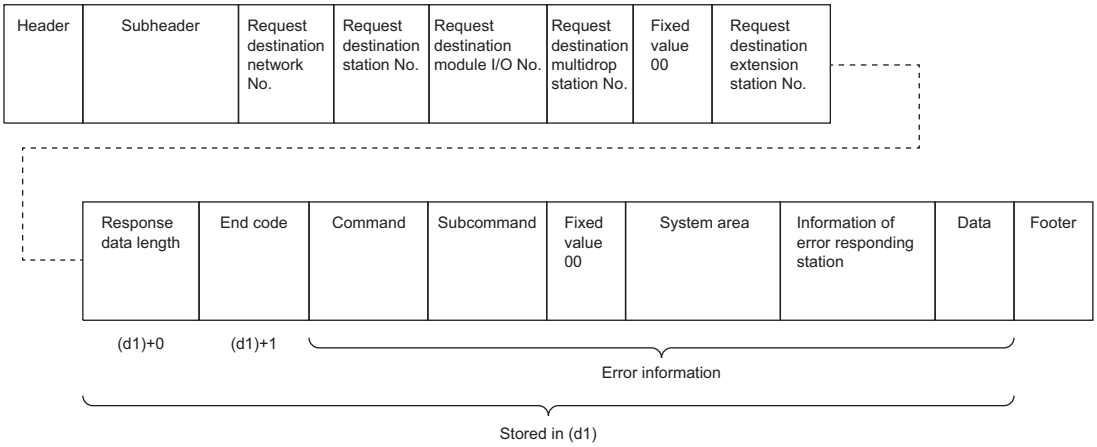
● Request message



● Response message (in normal end)



● Response message (in abnormal end)

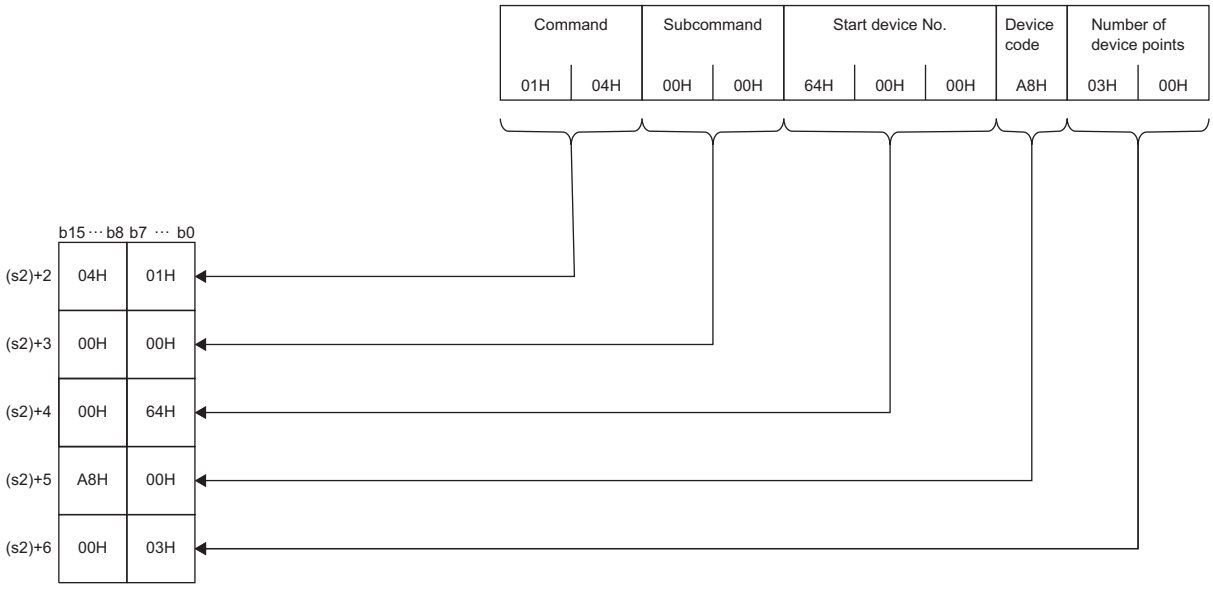


The network number, station number, request destination module I/O number, multidrop station number, 00 (fixed value), and extension station number are stored to the information of the error responding station.

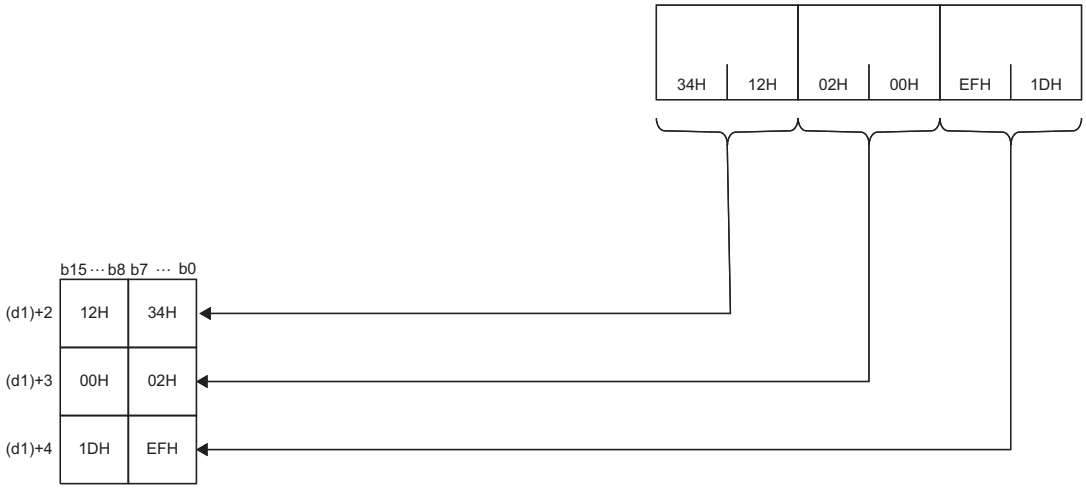
Ex.

When sending "Read (command: 0401H)" (reading in units of words) which reads the value in D100 to D102

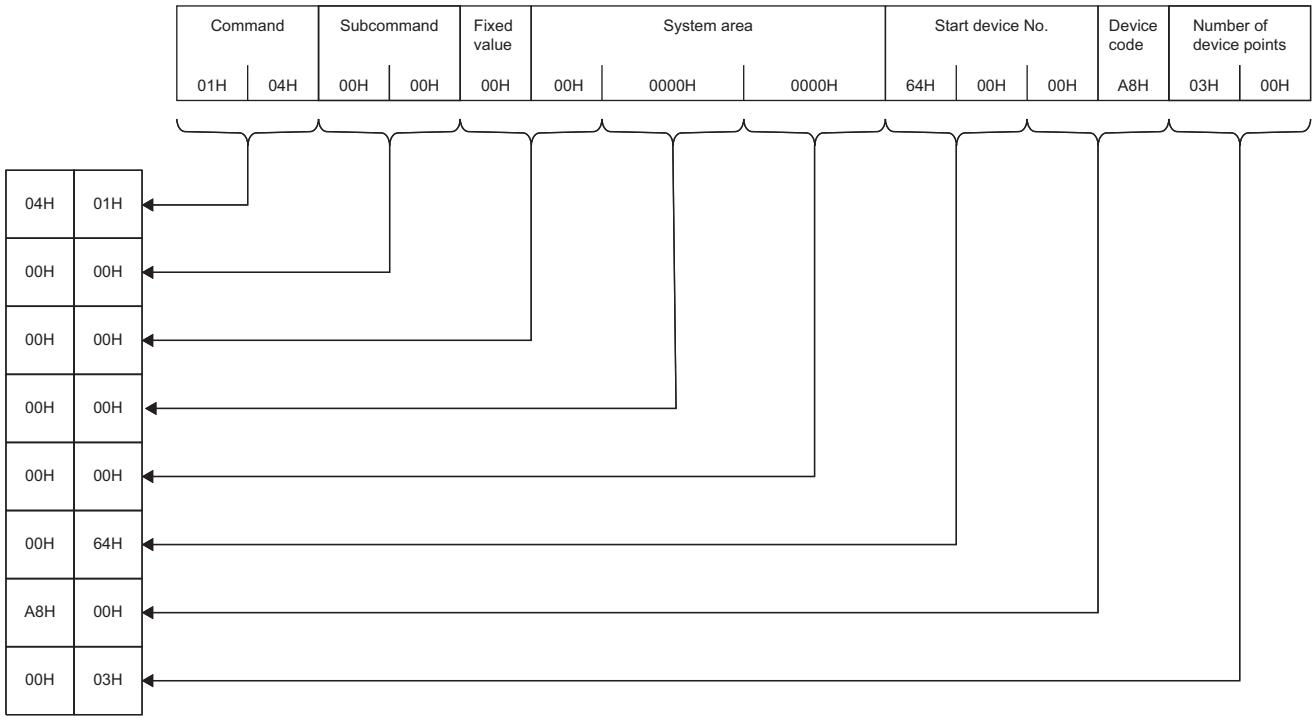
● The following example shows that request data is stored in (s2)+2 and later (3E/4E frame).



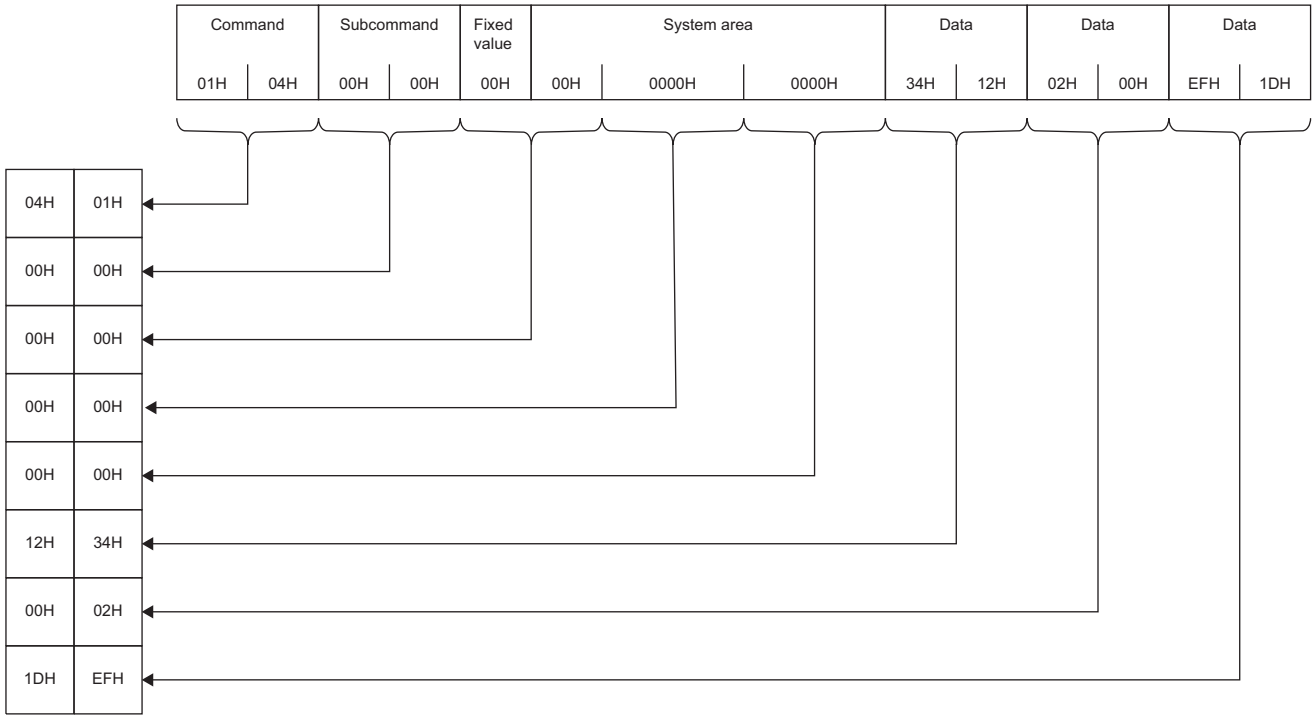
● The following example shows that response data is stored in (d1)+2 and later (3E/4E frame).



● The following example shows that request data is stored in (s2)+2 and later (station number extension frame).



● The following example shows that response data is stored in (d1)+2 and later (station number extension frame).



Operation error

Error code ((s1)+1)	Description
1000H to 3FFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual.
4000H to 4FFFH	Refer to MELSEC iQ-F FX5 User's Manual (Application).
D000H to DFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE TSN Master/Local Module User's Manual.

Point

The SLMPSND instruction is successfully completed even if the target device returns an abnormal response. When the SLMPSND instruction is completed successfully, the response is whether normal or abnormal can be identified by the end code of the response frame.
When an abnormal response is returned, check the manual of the SLMP-compatible device being used and take corrective action.

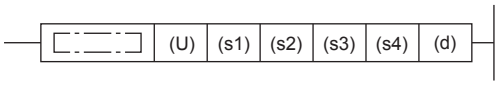
15 CC-LINK IE FIELD NETWORK INSTRUCTION

15.1 Setting parameters

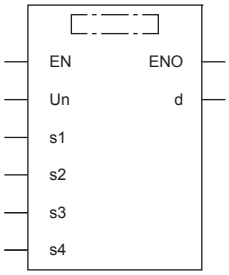
G(P).CCPASET

FX5S FX5UJ FX5U FX5UC

Set the parameters to the FX5-CCLIEF.

Ladder diagram	Structured text
	ENO:=G_CCPASET(EN,Un,s1,s2,s3,s4,d); ENO:=GP_CCPASET(EN,Un,s1,s2,s3,s4,d);

FBD/LD



("G_CCPASET", "GP_CCPASET" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s1)	Own station start device where control data is stored	☞ Page 1086 Control dataRefer to	Device name	ANY16*3
(s2)	Dummy device*2	—	Device name	ANY16*3
(s3)	Dummy device*2	—	Device name	ANY16*3
(s4)	Dummy device*2	—	Device name	ANY16*3
(d)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.
*2 Since the operands (s2) to (s4) are not used for the FX5-CCLIEF, specify dummy devices.
*3 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	○	—	—	—	—	○	—	—	—	—
(s4)	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.
*2 T, ST, and C cannot be used.

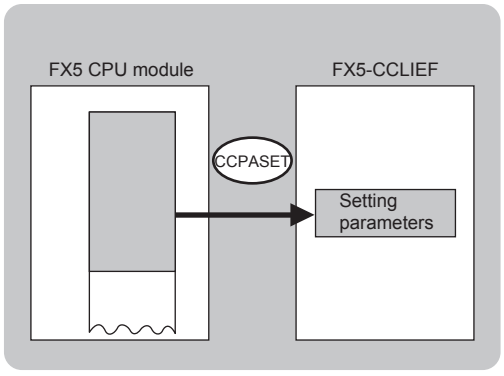
■Control data

Device	Item	Description	Setting range	Set by
(s1)+0	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System
(s1)+1	Setting flag	b15 … b13 b12 b11 b10 b9 b8 b7 … b0 0 | (3) | 0 | (2) | (1) | 0 (1) Input setting for data link error (b8) • 0: Cleared • 1: Held (2) Output setting for CPU STOP (b9) • 0: Held • 1: Cleared (3) Output setting for CPU error (b12) • 0: Cleared • 1: Held	Shown on left side	User

Processing details

- Set the parameters to the FX5-CCLIEF.

[Own station]

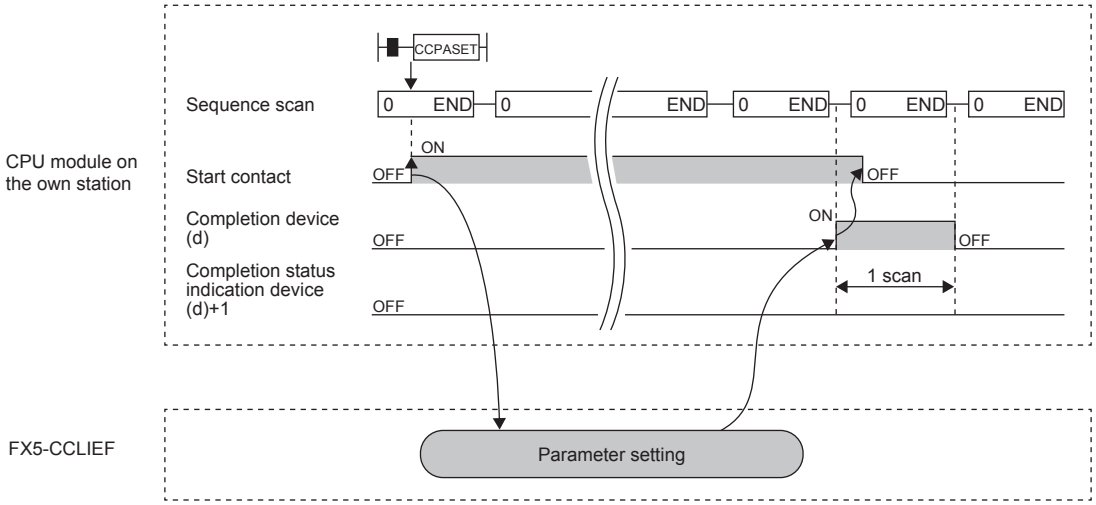


- The execution of the G(P).CCPASET instruction and whether it has been completed normally or completed with an error can be checked with the completion device (d) or completion status indication device (d)+1.

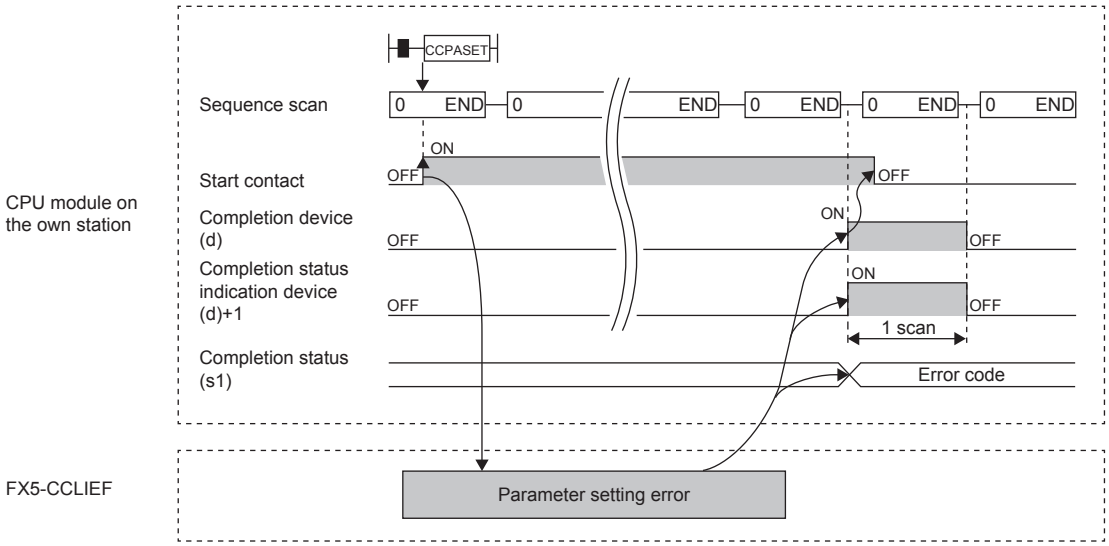
Device	Operation
Completion device (d)	The device turns on during the END processing for the scan in which the G(P).CCPASET instruction is completed, and turns off during the next END processing.
Completion status indication device (d)+1	The device turns on or off depending on the completion status of the G(P).CCPASET instruction. When completed normally: The device does not change (remains off). When completed with an error: The device turns on during the END processing for the scan in which the G(P).CCPASET instruction is completed, and turns off during the next END processing.

• The following figure shows the operation at completion of the G(P).CCPASET instruction.

When completed normally



When completed with an error



Operation error

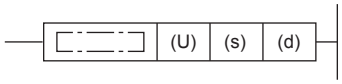
Error code ((s1)+0)	Description
D000H to DFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE Field Network Module User's Manual.

15.2 Setting the station number to own station

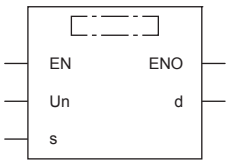
G(P).UINI

~~FX5S~~ FX5UJ FX5U FX5UC

This instruction sets the station number to the intelligent device station (own station) whose station number has not yet been set.

Ladder diagram	Structured text
	ENO:=G_UINI(EN,Un,s,d); ENO:=GP_UINI(EN,Un,s,d);

FBD/LD



("G_UINI", "GP_UINI" enters ☐.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s)	Own station start device where control data is stored	☞ Page 1089 Control dataRefer to	Device name	ANY16*2
(d)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.
*2 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U☐\G☐	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.
*2 T, ST, and C cannot be used.

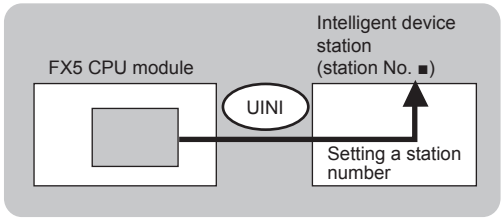
■Control data

Device	Item	Description	Setting range	Set by
(s)+0	—	Not used	—	System
(s)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System
(s)+2	Change target specification	0001H (fixed)	0001H	User
(s)+3	Station number of own station	Specifies the station number to be set.	1 to 120	User
(s)+4 to (s)+9	—	Not used	—	System

Processing details

- Set the station number to the intelligent device station.

[Own station]

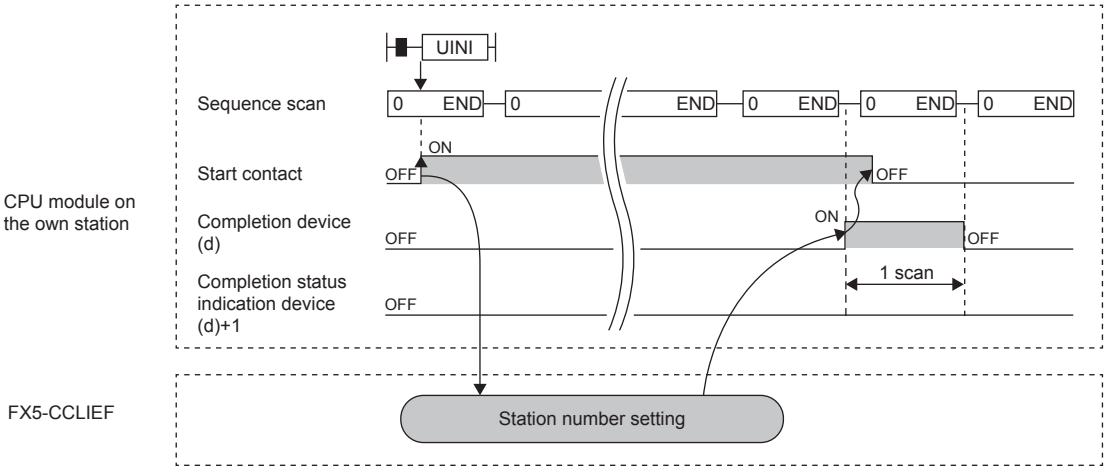


- The execution status and the completion status of the G(P).UINI instruction can be checked with the completion device (d) and the completion status indication device (d)+1.

Device	Operation
Completion device (d)	The device turns on during the END processing for the scan in which the G(P).UINI instruction is completed, and turns off during the next END processing.
Completion status indication device (d)+1	The device turns on or off depending on the completion status of the G(P).UINI instruction. When completed normally: The device does not change (remains off). When completed with an error: The device turns on during the END processing for the scan in which the G(P).UINI instruction is completed, and turns off during the next END processing.

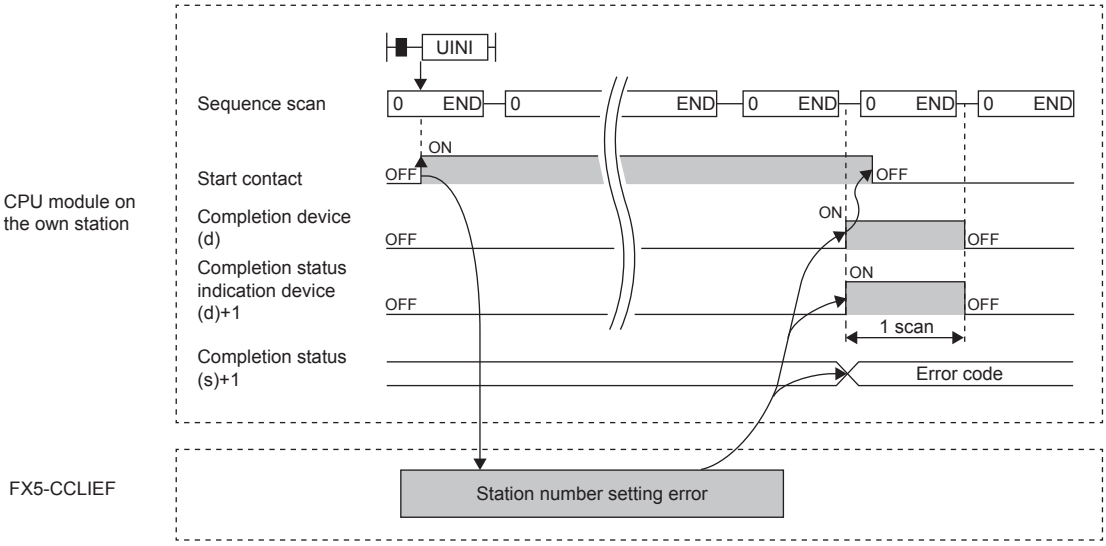
- The following figure shows the operation at completion of the G(P).UINI instruction.

When completed normally



15

When completed with an error



Precautions

- The station number set by the G(P).UINI instruction is cleared when the FX5 CPU module is powered off and on or reset.
- The G(P).UINI can be executed on the intelligent device station with no station number setting.
- If the station number set by the G(P).UINI instruction is already used for another station, the instruction is completed with an error. Set a unique station number. Note that such error cannot be detected before data link start (e.g. absence of the master station).

Operation error

Error code ((s)+1)	Description
D000H to DFFFH	Refer to MELSEC iQ-F FX5 CC-Link IE Field Network Module User's Manual.

16 HIGH-SPEED COUNTER INSTRUCTION

16.1 High-speed Processing Instruction

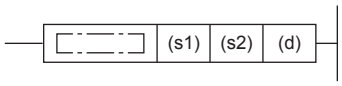
Setting 32-bit data comparison

DHSCS

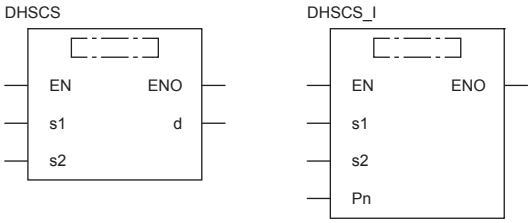
FX5S FX5UJ FX5U FX5UC

This instruction compares the value counted by a high-speed counter with a specified value, and immediately sets a bit device if the two values are equivalent to each other.

The high-speed pulse input/output module is not supported.

Ladder diagram	Structured text*1
	ENO:=DHSCS(EN,s1,s2,d); ENO:=DHSCS_I(EN,s1,s2,Pn);

FBD/LD*1



*1 When the interrupt pointer (I) is specified in operand (d) by ST language and the FBD/LD language, use the DHSCS_I instruction.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)		Data to be compared with the current value of a high-speed counter or word device number storing the data to be compared	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)		Channel number of a high-speed counter	K1 to 8	32-bit signed binary	ANY32
(d)	DHSCS	Bit device number to be set to ON when the compared two values are equivalent to each other	—	Bit	ANY_BOOL
	DHSCS_I*1			—	POINTER
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, d displays as Pn.

■Applicable devices

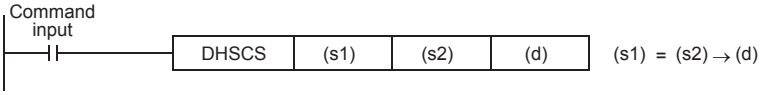
Operand	Bit	Word			Double word		Indirect specification	Constant			Others (I)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○*1	○	○	○	—	—	—
(d)	○	—	—	—	—	—	—	—	—	—	○*2

*1 Enable the FX3 compatible function, and specify a device between LC35 and 55 that is designated as an FX3 compatible high-speed counter.
For FX3 compatible function, refer to MELSEC iQ-F FX5 User's Manual (Application).

*2 I16 to I23 can be used.

Processing details

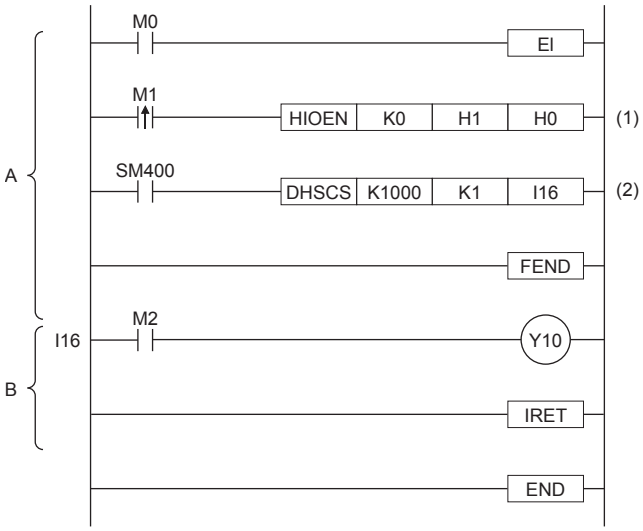
- When the current value of a high-speed counter of the channel specified in (s2) becomes the comparison value (s1) (for example, when the current value changes from "199" to "200" or from "201" to "200" if the comparison value is K200), the bit device (d) is set to ON regardless of the scan time. In this instruction, the comparison processing is executed after the count processing in the high-speed counter. For details, refer to MELSEC iQ-F FX5 User's Manual (Application).



Point

Use DHSCS if the output should be given when the counting result becomes equivalent to the comparison value regardless of the scan time of the CPU module.
When the number of instructions that can be simultaneously used is exceeded, use a general-purpose comparison instruction.

If using the high-speed comparison match interrupt, the corresponding interrupt programs can be executed by setting the interrupt pointers (I16 to I23) to (d) as shown below.



A: Main routine program
B: I16 Interrupt Program
(1): Start the high-speed counter CH1.
(2): When the current value of the high-speed counter CH1 reaches 1000, the interrupt program (I16) is executed.

Precautions

The value specified in (s2) should only be the channel of high-speed counter number (1 to 8) set by the parameter.
An operation error occurs in the following cases.

- When a channel which is not set by the parameter or a value other than K1 to 8 is specified
- When an LC device number which is not set by the parameter is specified

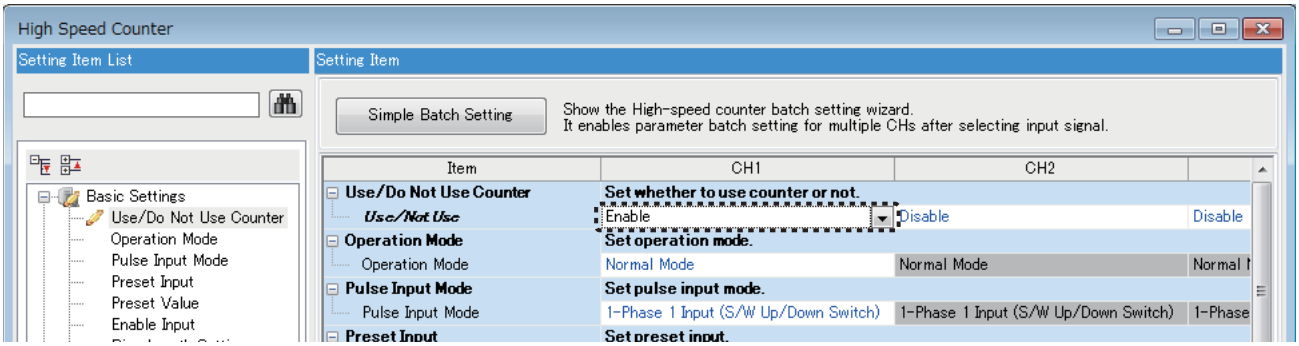
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

Program example

• Parameter

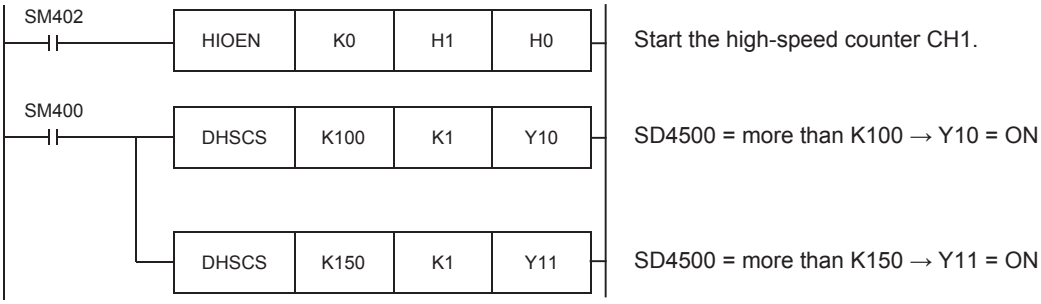
When using the data comparison by the high-speed counter, set "Use/Do Not Use Counter" of CH1 to "Enable" with the following parameter.

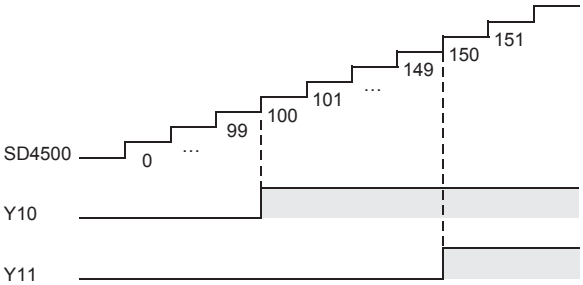
Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [High Speed I/O] ⇨ "Input Function" ⇨ "High Speed Counter" ⇨ "Detailed Setting" ⇨ "Basic Settings"



• Program

In the program example shown below, with regard to the current value of a counter, different outputs (Y10 and Y11) are arbitrarily set to ON by two values.





Operation error

Error code (SD0/SD8067)	Description
1810H	The channel specified by instructions using communication functions or built-in I/O is already used by other instructions.
2820H	A device used as an instruction operand is outside the allowable device range.
3405H	A channel number LC device, or device (I) number outside the range is specified.
3582H	The DHSCS instruction is executed in an interrupt program
3600H	A channel number for which the channel setting is not set is specified in the operand for channel number specification of the high-speed counter.
3780H	The DHSCS, DHSCR, and DHSZ instructions are used exceeding the maximum limit of the number of these instructions.

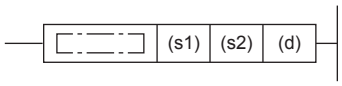
Reset 32-bit data comparison

DHSCR

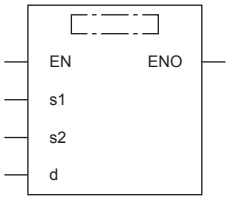
FX5S FX5UJ FX5U FX5UC

This instruction compares the value counted by a high-speed counter with a specified value, and immediately resets a bit device if the two values are equivalent to each other, or resets the high speed counter.

The high-speed pulse input/output module is not supported.

Ladder diagram	Structured text
	ENO:=DHSCR(EN,s1,s2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Data to be compared with the current value of a high-speed counter or word device number storing the data to be compared	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Channel number of a high-speed counter	K1 to 8	32-bit signed binary	ANY32
(d)	Bit device number to be reset (set to OFF) when both values become equivalent to each other, or channel number of self-reset high speed counter	—	Bit/32-bit signed binary	ANY_ELEMENTARY
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

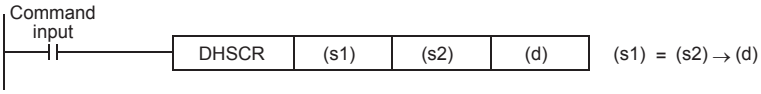
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○*1	○	○	○	—	—	—
(d)	○	○	—	—	○	—	○	○	—	—	—

*1 Enable the FX3 compatible function, and specify a device between LC35 and 55 that is designated as an FX3 compatible high-speed counter.
For FX3 compatible function, refer to MELSEC iQ-F FX5 User's Manual (Application).

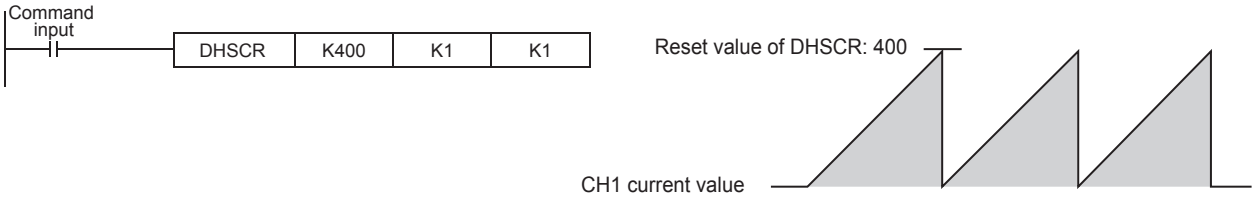
Processing details

- When the current value of a high-speed counter of the channel specified in (s2) becomes the comparison value (s1) (for example, when the current value changes from "199" to "200" or from "201" to "200" if the comparison value is K200), the bit device (d) is reset to OFF regardless of the scan time. For details, refer to MELSEC iQ-F FX5 User's Manual (Application).

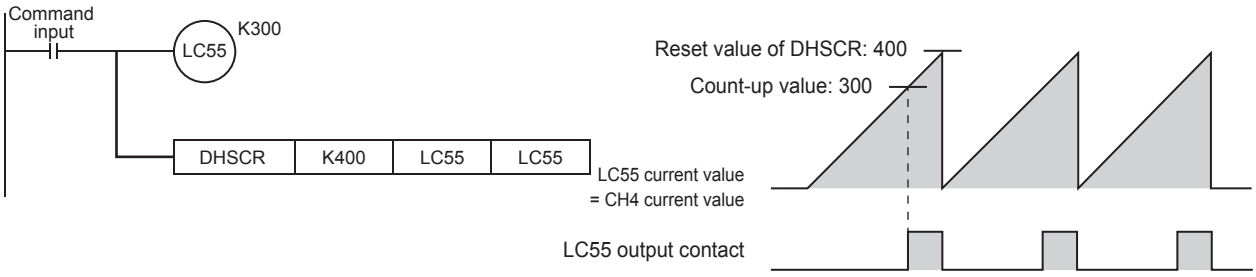


- The same channel number as (s2) or the LC device is specified in (d), (d) is reset to OFF by itself.

Example of specifying the same channel number as (s2) in (d)
When the current value of the high-speed counter of CH1 becomes 400, the current value is overwritten to the preset value.
[Program example]



Example of specifying the LC device in (d)
When the current value of LC55, the high-speed counter of CH4 with 2-phase 2-input (1-multiplication/4-multiplication) becomes 300, the output contact of LC55 turns ON. When the current value of LC55 becomes 400, the current value is cleared to 0 and the output contact turns OFF.
[Program example]



Point

Use DHSCR if the output should be given when the counting result becomes equivalent to the comparison value regardless of the scan time of the CPU module.
When the number of instructions that can be simultaneously used is exceeded, use a general-purpose comparison instruction.

Precautions

The value specified in (s2) should only be the channel of high-speed counter number (1 to 8) set by the parameter.
An operation error occurs in the following cases.

- When a channel which is not set by the parameter or a value other than K1 to 8 is specified
- When an LC device number which is not set by the parameter is specified

For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

Operation error

Error code (SD0/SD8067)	Description
1810H	The channel specified by instructions using communication functions or built-in I/O is already used by other instructions.
2820H	A device used as an instruction operand is outside the allowable device range.
3405H	A channel number outside the range or the LC device is specified.
3582H	The DHSCR instruction is executed in an interrupt program.
3600H	A channel number for which the channel setting is not set is specified in the operand in channel number specification of the high-speed counter.
3780H	The DHSCS, DHSCR, and DHSZ instructions are used exceeding the maximum limit of the in number of these instructions.

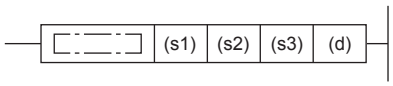
Comparison of 32-bit data band

DHSZ

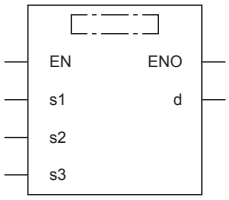
FX5S FX5UJ FX5U FX5UC

This instruction compares the current value of a high-speed counter with two values (one zone), and outputs the comparison result (refresh).

The high-speed pulse input/output module is not supported.

Ladder diagram	Structured text
	ENO:=DHSZ(EN,s1,s2,s3,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Data to be compared with the current value of a high-speed counter or word device number storing data to be compared (comparison value 1)	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Data to be compared with the current value of a high-speed counter or word device number storing data to be compared (comparison value 2)	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s3)	Channel number of a high-speed counter	K1 to 8	32-bit signed binary	ANY32
(d)	Head bit device number to which the comparison result is output based on the comparison value 1 and the comparison value 2	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

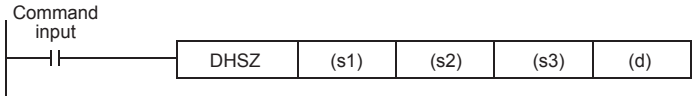
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○*1	○	○	○	—	—	—
(d)	○	—	—	—	—	—	—	—	—	—	—

*1 Enable the FX3 compatible function, and specify a device between LC35 and 55 that is designated as an FX3 compatible high-speed counter.
For FX3 compatible function, refer to MELSEC iQ-F FX5 User's Manual (Application).

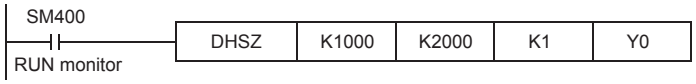
Processing details

- The current value of a high-speed counter specified in (s3) is compared with two comparison points (comparison value 1 and comparison value 2). Based on the zone comparison result, "smaller than the lower comparison value", "inside the comparison zone" or "larger than the upper comparison value", one among (d), (d)+1 and (d)+2 is set to ON regardless of the scan time. For details, refer to MELSEC iQ-F FX5 User's Manual (Application).



(s1) > (s3) → (d)
(s1) ≤ (s3) ≤ (s2) → (d) +1
(s3) > (s2) → (d) +2

- Make sure that the comparison value 1 and the comparison value 2 have the following relationship: [Comparison value 1] ≤ [Comparison value 2]. When the setting differs from the above, an operation error occurs and the DHSZ instruction will not perform any operation.
- When the current value of the high-speed counter CH1 changes (counts) as shown below, the comparison result is turn on to one of the outputs Y0, Y1 or Y2.



ON
K1000 > CH1 current value → Y0
K1000 ≤ CH1 current value ≤ K2000 → Y1
CH1 current value > K2000 → Y2

Comparison pattern	Current value of CH1 (s3)	Change of output contact (Y)		
		Y0	Y1	Y2
(s1)>(s3)	1000>(s3)	ON	OFF	OFF
	999→1000	ON→OFF	OFF→ON	OFF
	1000→999	OFF→ON	ON→OFF	OFF
(s1) ≤ (s3) ≤ (s2)	999→1000	ON→OFF	OFF→ON	OFF
	1000→999	OFF→ON	ON→OFF	OFF
	1000 ≤ (s3) ≤ 2000	OFF	ON	OFF
	2000→2001	OFF	ON→OFF	OFF→ON
	2001→2000	OFF	OFF→ON	ON→OFF
(s3)>(s2)	2000→2001	OFF	ON→OFF	OFF→ON
	2001→2000	OFF	OFF→ON	ON→OFF
	(s3)>2000	OFF	OFF	ON

Point

It is used when the output should be given when the counting result becomes equivalent to the comparison value regardless of the scan time of the CPU module.
When the number of instructions that can be simultaneously used is exceeded, use a general-purpose comparison instruction.

Precautions

An operation error occurs in the following cases.

- When a channel which is not set to (s3) by the parameter or a value other than K1 to 8 is specified
- When an LC device number which is not set by the parameter is specified

Three devices are occupied from the device specified in (d). Make sure that these devices are not used in other controls.
When designating a Y device, do not set a device that crosses over a module or a device No. that is a multiple of 16

Ex.

Example: Designating Y36 or Y37 when using FX5U-64M□.

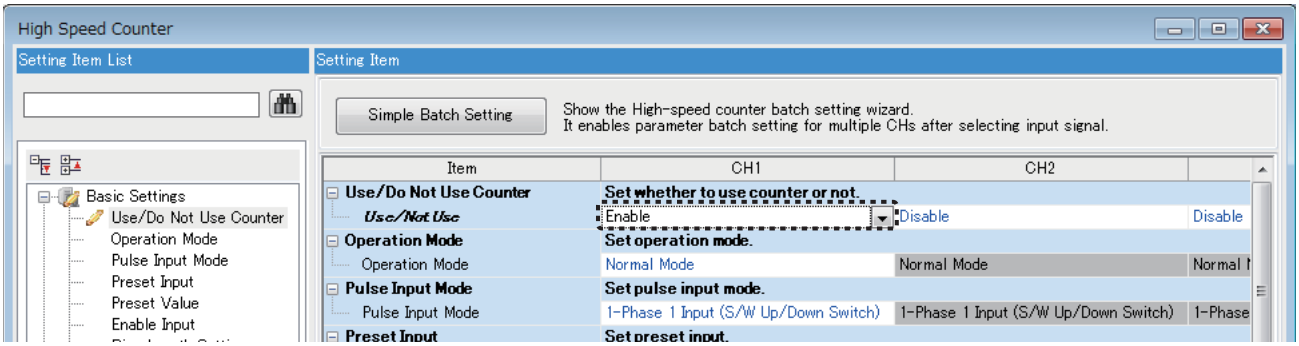
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

Program example

• Parameter

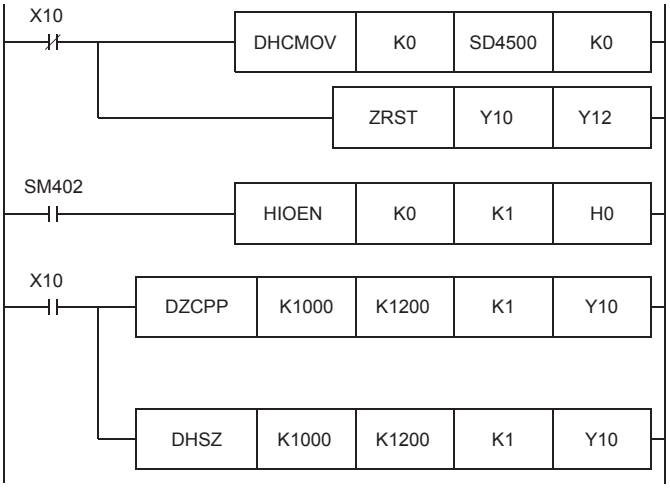
When using the data band comparison by the high-speed counter, set "Use/Do Not Use Counter" of CH1 to "Enable" with the following parameter.

Navigation window ⇨ [Parameter] ⇨ Module model name ⇨ [Module Parameter] ⇨ [High Speed I/O] ⇨ "Input Function" ⇨ "High Speed Counter" ⇨ "Detailed Setting" ⇨ "Basic Settings"



• Program

In the program shown below, the data range (K1000 to K1200) specified for the current value of the counter is compared, and the output (Y10 to Y12) is set (ON) according to the comparison results.



Y10 to Y12 are reset.

Start the high-speed counter CH1.

When X10 turns ON, comparison is executed only once.
K1000>SD4500 : Y10 ON
K1000≤SD4500≤K1200 : Y11 ON
K1200<SD4500 : Y12 ON

When X10 turns ON, comparison is executed when each pulse is input from X0.
K1000>SD4500 : Y10 ON
K1000≤SD4500≤K1200 : Y11 ON
K1200<SD4500 : Y12 ON

Operation error

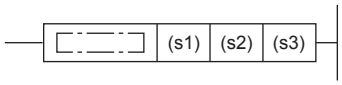
Error code (SD0/SD8067)	Description
1810H	The channel specified by instructions using communication functions or built-in I/O is already used by other instructions.
2820H	The number of devices is insufficient.
3405H	A channel number or LC device outside the range is specified.
	The comparison value 1 is greater than the comparison value 2.
3582H	The DHSZ instruction is executed in an interrupt program.
3600H	A channel number for which the channel setting is not set is specified in the operand for channel number specification of the high-speed counter.
3780H	The DHSCS, DHSCR, and DHSZ instructions are used exceeding the maximum limit of the number of these instructions.

Start/stop of the 16-bit data high-speed I/O function

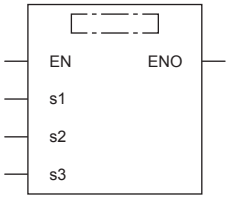
HIOEN(P)

FX5S FX5UJ FX5U FX5UC

These instructions control the start and stop operations of a high-speed I/O function.

Ladder diagram	Structured text
	ENO:=HIOEN(EN,s1,s2,s3); ENO:=HIOENP(EN,s1,s2,s3);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Function number to be started or stopped	Refer to Function number (☞ Page 1101)	16-bit signed binary	ANY16
(s2)	Set the bit of the channel number where the function is started.	-32768 to +32767	16-bit signed binary	ANY16
(s3)	Set the bit of the channel number where the function is stopped.	-32768 to +32767	16-bit signed binary	ANY16
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	○	—	—	○	○	—	—	—

Processing details

Specify the number of the function to be started or stopped in (s1), the bit of the channel to be started in (s2), and the bit of the channel to be stopped in (s3).

The following table shows the function numbers which can be specified in (s1).

■Function number

Function number	Function name
K0	High-speed counter
K10*1	Pulse density/rotation speed measurement
K20*1	High-speed comparison table (CPU module)
K21*1	High-speed comparison table (first high-speed pulse input/output module)
K22*1	High-speed comparison table (second high-speed pulse input/output module)
K23*1	High-speed comparison table (third high-speed pulse input/output module)
K24*1	High-speed comparison table (fourth high-speed pulse input/output module)
K30*1*2	Multi-output high-speed comparison table
K40	Pulse width measurement
K50	PWM

*1 When high-speed counter (function number: K0) is stopped during function operation, the function continues to operate, but nothing will be processed.

*2 When multi-output high-speed comparison table (function number: K30) is stopped, high-speed counter of the same ch is also stopped.

The following table shows the values which can be specified in (s2) and (s3) for each function number.

• Function number K0

The counting start and stop of a high-speed counter can be controlled for each channel of high-speed counter.

CH1 to CH8 are for the CPU module, and CH9 to CH16 are for the high-speed pulse input/output module.

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

Ex.

To start CH3, set 04H in (s2). To stop it, set 04H in (s3).

To start CH1, CH4, and CH5, set 19H in (s2). To stop them, set 19H in (s3).

To start CH1 and CH4 and to stop CH5, set 09H in (s2) and set 10H in (s3).

• Function number K10

The measuring start and stop of the pulse density (rotation speed measurement) can be controlled for each channel of the high-speed counter.

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—								CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

• Function number K20 to K24

Set the value to turn on the bit of the high-speed comparison table number which is to be started or stopped.

In the case of the CPU module (K20), b0 to b15 can be used.

In the case of the high-speed pulse input/output module (K21 to K24), b0 to b14 can be used.

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

• Function number K30

For the multi-output high-speed comparison table, specification of a channel is not required. To start the multi-output high-speed comparison table, set 01H in (s2). To stop it, set 01H in (s3).

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—															Valid

• Function numbers K40 and K50

The measuring start and stop of pulse width measurement and PWM can be controlled for each channel.

CH1 to CH4 are for the CPU module, and CH5 to CH12 are for the high-speed pulse input/output module.

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—				CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

Precautions

- Do not specify channel numbers of the high-speed pulse input/output module in a program with interrupt priority set to "1".
- When values that turn on the same channel are set for start and stop, the stop operation is prioritized.
- If there is a channel where an error occurs (error code: 3781H) when the high-speed counter (function number: K0) is executed by the HIOEN instruction, the counter can no longer be started/stopped for the applicable channel and subsequent channels.
- When the high-speed comparison table (function number: K20) is used with the HIOEN instruction, the total number of high-speed comparisons, including the DHSCS instruction, DHSCR instruction, DHSZ instruction, and interrupt input of built-in positioning, must be 32 or less.
- When using the high-speed comparison table (function numbers: K21 to K24) with the HIOEN instruction, set the number of high-speed comparisons used within the same high-speed pulse input/output module to 15 times or less.
- For the high-speed comparison table numbers and total number of high-speed comparisons, refer to ☞ Page 1452 Added and Changed Functions.
- To start the high-speed comparison table (function numbers: K20 to K24) or the multi-output high-speed comparison table (function number: K30), the high-speed counter must be started using the HIOEN instruction in advance.
- The high-speed input/output instructions operate according to the following parameters.

Function number	Function specified by the HIOEN instruction	Parameter setting
K0	High-speed counter	Channel setting of the high-speed counter
K10	Pulse density (rotation speed measurement)	Channel setting of the pulse density/rotation speed measurement High-speed counter
K20	High-speed comparison table (CPU module)	Output setting of the high-speed counter
K21	High-speed comparison table (first high-speed pulse input/output module)	Output setting of the high-speed counter
K22	High-speed comparison table (second high-speed pulse input/output module)	Output setting of the high-speed counter
K23	High-speed comparison table (third high-speed pulse input/output module)	Output setting of the high-speed counter
K24	High-speed comparison table (fourth high-speed pulse input/output module)	Output setting of the high-speed counter
K30	Multi-output high-speed comparison table	Output setting of the high-speed counter
K40	Pulse width measurement	Channel setting of the pulse width measurement
K50	PWM	Channel setting of PWM

Operation error

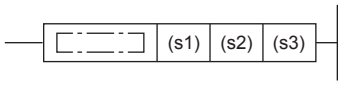
Error code (SD0/SD8067)	Description
1810H	The channel specified by instructions using communication functions or built-in I/O is already used by other instructions.
2801H	The channel number of a module that does not exist is specified.
3056H	Timeout occurred while communicating with the target modules during execution of the instruction.
3060H	Signal error was detected while accessing the target modules during execution of the instruction.
3405H	An invalid function number is specified in (s).
3580H	An instruction that cannot be used in an interrupt program is used.
3600H	A channel number which is not selected in the parameter setting is executed.
3781H	Ring length ≤ preset value is specified and executed in channel for which ring length is set and preset input is enabled.

Start/stop of the 32-bit data high-speed I/O function

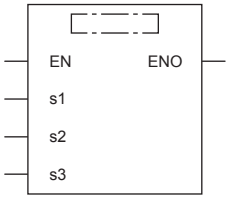
DHIOEN(P)

FX5S FX5UJ FX5U FX5UC

These instructions control the start and stop operations of a high-speed I/O function.

Ladder diagram	Structured text
	ENO:=DHIOEN(EN,s1,s2,s3); ENO:=DHIOENP(EN,s1,s2,s3);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Function number to be started or stopped	Refer to Function number (☞ Page 1105)	16-bit signed binary	ANY16
(s2)	Set the bit of the channel number where the function is started.	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s3)	Set the bit of the channel number where the function is stopped.	-2147483648 to +2147483647	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○	○	○	○	—	—	—

Processing details

Specify the number of the function to be started or stopped in (s1), the bit of the channel to be started in (s2), and the bit of the channel to be stopped in (s3).

The following table shows the function numbers which can be specified in (s1).

■Function number

Function number	Function name
K0	High-speed counter
K10*1	Pulse density/rotation speed measurement
K20*1	High-speed comparison table (CPU module)
K21	High-speed comparison table (first high-speed pulse input/output module)
K22	High-speed comparison table (second high-speed pulse input/output module)
K23	High-speed comparison table (third high-speed pulse input/output module)
K24	High-speed comparison table (fourth high-speed pulse input/output module)
K30*1*2	Multi-output high-speed comparison table
K40	Pulse width measurement
K50	PWM

*1 When high-speed counter (function number: K0) is stopped during function operation, the function continues to operate, but nothing will be processed.

*2 When multi-output high-speed comparison table (function number: K30) is stopped, high-speed counter of the same ch is also stopped.

The following table shows the values which can be specified in (s2) and (s3) for each function number.

• Function number K0

The counting start and stop of a high-speed counter can be controlled for each channel of high-speed counter.

CH1 to CH8 are for the CPU module, and CH9 to CH16 are for the high-speed pulse input/output module.

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

Ex.

To start CH3, set 04H in (s2). To stop it, set 04H in (s3).

To start CH1, CH4, and CH5, set 19H in (s2). To stop it, set 19H in (s3).

To start CH1 and CH4 and to stop CH5, set 09H in (s2) and set 10H in (s3).

• Function number K10

The measuring start and stop of the pulse density (rotation speed measurement) can be controlled for each channel of the high-speed counter.

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—								CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

• Function number K20 to K24

Set the value to turn on the bit of the high-speed comparison table number which is to be started or stopped.

In the case of the CPU module (K20), b0 to b31 can be used.

In the case of the high-speed pulse input/output module (K21 to K24), b0 to b14 can be used.

Low-order bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

High-order bit position															
b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17

• Function number K30

For the multi-output high-speed comparison table, specification of a channel is not required. To start the multi-output high-speed comparison table, set 01H in (s2). To stop it, set 01H in (s3).

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—															Valid

• Function numbers K40 and K50

The measuring start and stop of pulse width measurement and PWM can be controlled for each channel.
CH1 to CH4 are for the CPU module, and CH5 to CH12 are for the high-speed pulse input/output module.

Bit position															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—				CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

Precautions

- Do not specify channel numbers of the high-speed pulse input/output module in a program with interrupt priority set to "1".
- When values that turn on the same channel are set for start and stop, the stop operation is prioritized.
- If there is a channel where an error occurs (error code: 3781H) when the high-speed counter (function number: K0) is executed by the DHIOEN instruction, the counter can no longer be started/stopped for the applicable channel and subsequent channels.
- When the high-speed comparison table (function number: K20) is used with the DHIOEN instruction, the total number of high-speed comparisons, including the DHSCS instruction, DHSCR instruction, DHSZ instruction, and interrupt input of built-in positioning, must be 32 or less.
- When using the high-speed comparison table (function numbers: K21 to K24) with the DHIOEN instruction, set the number of high-speed comparisons used within the same high-speed pulse input/output module to 15 times or less.
- For the high-speed comparison table numbers and total number of high-speed comparisons, refer to ☞ Page 1452 Added and Changed Functions.
- To start the high-speed comparison table (function numbers: K20 to K24) or the multi-output high-speed comparison table (function number: K30), the high-speed counter must be started using the DHIOEN instruction in advance.
- The high-speed input/output instructions operate according to the following parameters.

Function number	Function specified by the DHIOEN instruction	Parameter setting
K0	High-speed counter	Channel setting of the high-speed counter
K10	Pulse density (rotation speed measurement)	Channel setting of the pulse density/rotation speed measurement High-speed counter
K20	High-speed comparison table (CPU module)	Output setting of the high-speed counter
K21	High-speed comparison table (first high-speed pulse input/output module)	Output setting of the high-speed counter
K22	High-speed comparison table (second high-speed pulse input/output module)	Output setting of the high-speed counter
K23	High-speed comparison table (third high-speed pulse input/output module)	Output setting of the high-speed counter
K24	High-speed comparison table (fourth high-speed pulse input/output module)	Output setting of the high-speed counter
K30	Multi-output high-speed comparison table	Output setting of the high-speed counter
K40	Pulse width measurement	Channel setting of the pulse width measurement
K50	PWM	Channel setting of PWM

Operation error

Error code (SD0/SD8067)	Description
1810H	The channel specified by instructions using communication functions or built-in I/O is already used by other instructions.
2801H	The channel number of a module that does not exist is specified.
3056H	Timeout occurred while communicating with the target modules during execution of the instruction.
3060H	Signal error was detected while accessing the target modules during execution of the instruction.
3405H	An invalid function number is specified in (s).
3580H	An instruction that cannot be used in an interrupt program is used.
3600H	A channel number which is not selected in the parameter setting is executed.
3781H	Ring length ≤ preset value is specified and executed in channel for which ring length is set and preset input is enabled.

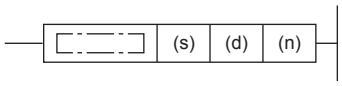
16.2 High-speed Current Value Transfer Instruction

High-speed current value transfer of 16-bit data

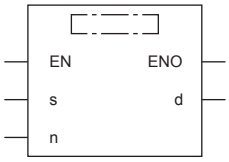
HCMOV(P)

FX5S FX5UJ FX5U FX5UC

These instructions read and write (updates) special register for high-speed counter, pulse width measurement, PWM, and positioning.

Ladder diagram	Structured text
	ENO:=HCMOV(EN,s,n,d); ENO:=HCMOVP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Transfer source device number	—	Bit/16-bit signed binary	ANY_ELEMENTARY
(d)	Transfer destination device number	—	Bit/16-bit signed binary	ANY_ELEMENTARY
(n)	Specification to clear the device value of the transfer source after the transfer	K0, K1	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

These instructions transfer the data in the device specified by (s) to the device specified by (d). At this time, if the value of (n) is K0, the value of (s) is not cleared. If the value of (n) is K1, the value of (s) is cleared to "0" after the transfer. The value is cleared only for the SD8099 special device supporting high-speed transfer.

Point

When (s) is a device supporting high-speed transfer

- When the HCMOV instruction is executed, the latest value is acquired such as the current value of a high-speed counter and transferred to (d).

When (d) is a device supporting high-speed transfer

- When the HCMOV instruction is executed, value such as the current value of a high-speed counter is changed.

■Effect of HCMOV instruction

- By using both input interrupt and HCMOV instruction, the current value of a high-speed counter can be received at the rising edge or falling edge of an external input.
- When HCMOV instruction is used just before a comparison instruction (CMP, ZCP or comparison contact instruction), the latest value of the high-speed counter is used in comparison.

Precautions

- When it is necessary to execute comparison and outputting as soon as the current value of a high-speed counter changes, use the high-speed comparison table, multi-output high-speed comparison table, or one of the DHSCS, DHSCR, and DHSZ instructions.
- If 32-bit binary data special device which supports the high-speed transfer (such as the current value of a high-speed counter) is read using the HCMOV instruction, the operation is the same as that when the MOV instruction is used.
- Do not overwrite the current value of a high-speed counter using the HCMOV instruction while executing the pulse density (rotation speed measurement) or the SPD instruction.
- If (s) is SD8099, and the (n) value is K1, SD8099 is cleared at the timing the instruction is executed (after the SD8099 current value is transferred). Do not clear the SD8099 current value with an application instruction such as a MOV instruction, which is affected by the scan.

Point

The HCMOV instruction is mainly used to read the current value of the high-speed counter/pulse width measurement and change the current address (in the user units) or the current address (in the pulse unit) of positioning.

Operation error

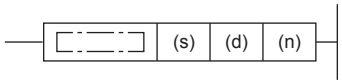
Error code (SD0/SD8067)	Description
2801H	The channel number of a module that does not exist is specified.
2821H	Operands that execute transmission between an SM supporting high-speed transfer and an SD supporting high-speed transfer is designated.
3056H	Timeout occurred while communicating with the target modules during execution of the instruction.
3060H	Signal error was detected while accessing the target modules during execution of the instruction.
3405H	A value outside the data range is set in (n).
3580H	An instruction that cannot be used in an interrupt program is used.

High-speed current value transfer of 32-bit data

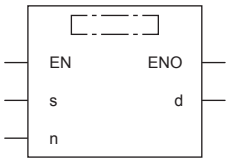
DHCMOV(P)

FX5S FX5UJ FX5U FX5UC

These instructions read and write (updates) special register for high-speed counter, pulse width measurement, PWM, and positioning.

Ladder diagram	Structured text
	ENO:=DHCMOV(EN,s,n,d); ENO:=DHCMOVP(EN,s,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Transfer source device number	—	Bit/32-bit signed binary	ANY_ELEMENTARY
(d)	Transfer destination device number	—	Bit/32-bit signed binary	ANY_ELEMENTARY
(n)	Specification to clear the device value of the transfer source after the transfer	K0, K1	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

These instructions transfer the data in the device specified by (s) to the device specified by (d). At this time, if the value of (n) is K0, the value of (s) is not cleared. If the value of (n) is K1, the value of (s) is cleared to "0" after the transfer. The value is cleared only for SD devices for the current value of high-speed counters or LC devices used as a high-speed counter when using the FX3 compatible high-speed counter.

Point

When (s) is a device supporting high-speed transfer

- When the DHCMOV instruction is executed, the latest value is acquired such as the current value of a high-speed counter and transferred to (d).

When (d) is a device supporting high-speed transfer

- When the DHCMOV instruction is executed, value such as the current value of a high-speed counter is changed.

■Effect of DHCMOV instruction

- By using both input interrupt and DHCMOV instruction, the current value of a high-speed counter can be received at the rising edge or falling edge of an external input.
- When DHCMOV instruction is used just before a comparison instruction (DCMP, DZCP or comparison contact instruction), the latest value of the high-speed counter is used in comparison.

Precautions

- When it is necessary to execute comparison and outputting as soon as the current value of a high-speed counter changes, use the high-speed comparison table, multi-output high-speed comparison table, or one of the DHSCS, DHSCR, and DHSZ instructions.
- Do not overwrite the current value of a high-speed counter using the DHCMOV instruction while executing the pulse density (rotation speed measurement) or the DSPD instruction.
- Transfer is not possible between an SM supporting high-speed transfer and an SD supporting high-speed transfer.
- When the device supporting high-speed transfer is set as the transfer source (s) by the DHCMOV instruction while the high-speed I/O function is stopped, the previous value before stop is read out. However, if the function is not executed even once, the initial value is read out.

Ex.

When SD5303, SD5302 (PWM pulse width) is set as the transfer source (s), the operation is executed as follows.

Condition	Result
When the PWM function is not executed	"0" is read out. (This is not the value of the parameter that is set by GX Works3.)
When the PWM function was executed but it is currently stopped	The value when the PWM function was stopped is read out.
When the PWM function is executed	The latest value that is currently operating is read out.

- When a high-speed counter SD device (current value, maximum value, minimum value) is read out individually, only the read SD device will be updated. Thus, there may be cases when the high-speed counter's SD device does not satisfy the relation of minimum value ≤ current value ≤ maximum value temporarily. Refer to the MELSEC iQ-F FX5 User's Manual (Application) for details on the timing that the high-speed counter's SD device is updated.

Point

The DHCMOV instruction is mainly used to read the current value of the high-speed counter/pulse width measurement and change the current address (in the user units) or the current address (in the pulse unit) of positioning.

Operation error

Error code (SD0/SD8067)	Description
2801H	The channel number of a module that does not exist is specified.
2821H	Operands that execute transmission between an SM supporting high-speed transfer and an SD supporting high-speed transfer is designated.
3056H	Timeout occurred while communicating with the target modules during execution of the instruction.
3060H	Signal error was detected while accessing the target modules during execution of the instruction.
3405H	A value outside the data range is set in (n).
3580H	An instruction that cannot be used in an interrupt program is used.

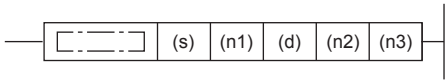
17 EXTERNAL DEVICE COMMUNICATION INSTRUCTION

17.1 Serial Communication 2

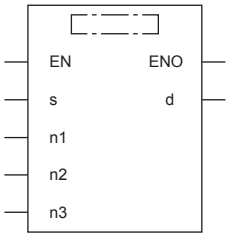
RS2

FX5S FX5UJ FX5U FX5UC

This instruction sends or receives data by non-protocol communication via serial ports of RS-232C or RS-485.

Ladder diagram	Structured text
	ENO:=RS2(EN,s,n1,n2,n3,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device storing send data	—	16-bit signed binary/ character string	ANY16
(n1)	Number of bytes of data to be sent	0 to 4096	16-bit unsigned binary	ANY16_U
(d)	Head device storing receive data	—	16-bit signed binary/ character string	ANY16
(n2)	Number of bytes to be received	0 to 4096	16-bit unsigned binary	ANY16_U
(n3)	Communication channel	■FX5S/FX5UJ CPU module K2 to 4 ■FX5U CPU module K1 to 4 ■FX5UC CPU module K1, K3 to 4	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	—	○*1	—	—	—	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○*1	—	—	—	—	○	—	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—
(n3)	—	—	—	—	—	—	—	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction sends or receives data by non-protocol communication via the built-in RS-485 port or the optional RS-232C or RS-485 serial ports installed in the CPU module. This instruction specifies the head device storing the sent data from the CPU module, the number of data bytes, the head device storing the received data, and the maximum number of bytes that can be received. For details, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Precautions

- It is not permitted to use other external device communication instructions on the same port.
- While this instruction is being driven, the communication format cannot be changed. Set this instruction to OFF before changing the communication format.
- When using the header and terminator, set them before driving this instruction. Do not change the values of the header and terminator while this instruction is being driven.

For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Operation error

Error code (SD0/SD8067)	Description
3405H	Data outside the allowable range was input.
2820H	The device specified by (s) or (d) exceeds the corresponding device range.
1810H	Channel number which is used in another instruction is specified.
3600H	Channel number specified by (n3) is not set by parameters.

For communication errors, refer to MELSEC iQ-F FX5 User's Manual (Communication).

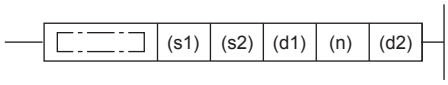
17.2 Inverter Communication Instruction

Inverter operation monitoring (Status check)

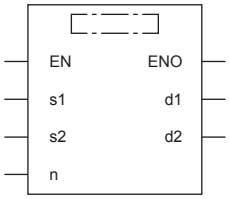
IVCK

FX5S FX5UJ FX5U FX5UC

This instruction reads the operation status of an inverter to the CPU module.

Ladder diagram	Structured text
	ENO:=IVCK(EN,s1,s2,n,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Inverter instruction codes	*1	16-bit signed binary	ANY16
(d1)	Device number storing the read value	—	16-bit signed binary	ANY16
(n)	Communication channel	■FX5S/FX5UJ CPU module K2 to 4 ■FX5U CPU module K1 to 4 ■FX5UC CPU module K1, K3 to 4	16-bit unsigned binary	ANY16_U
(d2)	Head bit device to which the execution status of the instruction is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Refer to MELSEC iQ-F FX5 User's Manual (Communication) or respective inverter manual.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	○	○	—	—	○	○	—	—	—
(s2)	—	○*1	○	○	—	—	○	○	—	—	—
(d1)	○	○	○	—	—	—	○	—	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

The operation status corresponding to the instruction code specified in (s2) of an inverter connected to the communication channel (n) whose station number is specified in (s1) is read and transferred to (d1). For details, refer to MELSEC iQ-F FX5 User's Manual (Communication). (For the instruction codes, refer to the each inverter manual.)

Precautions

Three devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

Operation error

Error code (SD0/SD8067)	Description
1810H	Channel number specified by (d) is used by another instruction.
2820H	The specified device exceeds the range of the corresponding device.
3405H	The value specified by (s1) is other than any of K0 to 31.
	When the value specified for (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to 4 • FX5U/FX5UC CPU module K1 to 4
3600H	Channel number specified by (d) is not set by parameters.

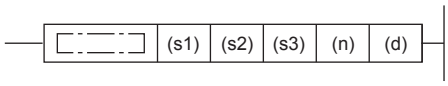
For communication errors, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Inverter operations control (Drive)

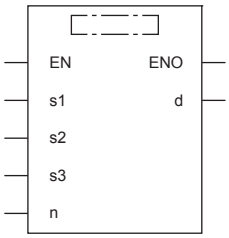
IVDR

FX5S FX5UJ FX5U FX5UC

This instruction writes a control value necessary for inverter operation to a CPU module using the computer link operation function of the inverter.

Ladder diagram	Structured text
	ENO:=IVDR(EN,s1,s2,s3,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Inverter instruction codes	*1	16-bit signed binary	ANY16
(s3)	Set value to be written to the inverter parameter or device number storing the data to be set	—	16-bit signed binary	ANY16
(n)	Communication channel	■FX5S/FX5UJ CPU module K2 to 4 ■FX5U CPU module K1 to 4 ■FX5UC CPU module K1, K3 to 4	16-bit unsigned binary	ANY16_U
(d)	Head bit device to which the execution status of the instruction is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Refer to MELSEC iQ-F FX5 User's Manual (Communication) or respective inverter manual.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	○	○	—	—	○	○	—	—	—
(s2)	—	○*1	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	—	—	—	○	○	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

The control value specified in (s3) is written to the instruction code specified in (s2) of an inverter connected to the communication channel (n) whose station number is specified in (s1). For details, refer to MELSEC iQ-F FX5 User's Manual (Communication). (For the instruction codes, refer to the each inverter manual.)

Precautions

Three devices are occupied from the device specified in (d). Make sure that these devices are not used in other controls.

Operation error

Error code (SD0/SD8067)	Description
1810H	Channel number specified by (d) is used by another instruction.
2820H	The specified device exceeds the range of the corresponding device.
3405H	The value specified by (s1) is other than any of K0 to 31.
	When the value specified for (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to 4 • FX5U/FX5UC CPU module K1 to 4
3600H	Channel number specified by (d) is not set by parameters.

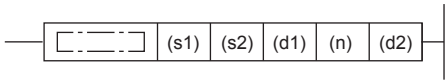
For communication errors, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Inverter parameter read

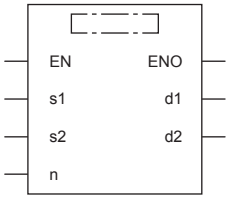
IVRD

FX5S FX5UJ FX5U FX5UC

This instruction reads a parameter of an inverter to the CPU module.

Ladder diagram	Structured text
	ENO:=IVRD(EN,s1,s2,n,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Inverter parameter number	*1	16-bit signed binary	ANY16
(d1)	Device number storing the read value	—	16-bit signed binary	ANY16
(n)	Communication channel	■FX5S/FX5UJ CPU module K2 to 4 ■FX5U CPU module K1 to 4 ■FX5UC CPU module K1, K3 to 4	16-bit unsigned binary	ANY16_U
(d2)	Head bit device to which the execution status of the instruction is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Refer to MELSEC iQ-F FX5 User's Manual (Communication) or respective inverter manual.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	○	○	—	—	○	○	—	—	—
(s2)	—	○*1	○	○	—	—	○	○	—	—	—
(d1)	○	○	○	—	—	—	○	—	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

The value of the parameter (s2) is read from an inverter connected to the communication channel (n) whose station number is (s1), and output to (d1). For details, refer to MELSEC iQ-F FX5 User's Manual (Communication). (For the parameter numbers, refer to the each inverter manual.)

Precautions

Three devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

Operation error

Error code (SD0/SD8067)	Description
1810H	Channel number specified by (d) is used by another instruction.
2820H	The specified device exceeds the range of the corresponding device.
3405H	The value specified by (s1) is other than any of K0 to 31.
	The value specified by (s2) is outside the allowable range. (Less than K0, K3000 to 9999, or K13000 to 32767)
	When the value specified for (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to 4 • FX5U/FX5UC CPU module K1 to 4
3600H	Channel number specified by (d) is not set by parameters.

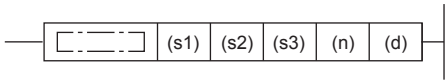
For communication errors, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Inverter parameter write

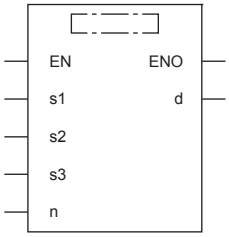
IVWR

FX5S FX5UJ FX5U FX5UC

This instruction writes a parameter of an inverter from the CPU module.

Ladder diagram	Structured text
	ENO:=IVWR(EN,s1,s2,s3,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Inverter parameter number	*1	16-bit signed binary	ANY16
(s3)	Set value to be written to the inverter parameter or device number storing the data to be set	—	16-bit signed binary	ANY16
(n)	Communication channel	■FX5S/FX5UJ CPU module K2 to 4 ■FX5U CPU module K1 to 4 ■FX5UC CPU module K1, K3 to 4	16-bit unsigned binary	ANY16_U
(d)	Head bit device to which the execution status of the instruction is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Refer to MELSEC iQ-F FX5 User's Manual (Communication) or respective inverter manual.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	○	○	—	—	○	○	—	—	—
(s2)	—	○*1	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	—	—	—	○	○	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

A value specified in (s3) is written to a parameter (s2) in an inverter connected to the communication channel (n) whose station number is (s1). For details, refer to MELSEC iQ-F FX5 User's Manual (Communication). (For the parameter numbers, refer to the each inverter manual.)

Precautions

Three devices are occupied from the device specified in (d). Make sure that these devices are not used in other controls.

Operation error

Error code (SD0/SD8067)	Description
1810H	Channel number specified by (d) is used by another instruction.
2820H	The specified device exceeds the range of the corresponding device.
3405H	The value specified by (s1) is other than any of K0 to 31.
	The value specified by (s2) is outside the allowable range. (Less than K0, K3000 to 9999, or K13000 to 32767)
	When the value specified for (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to 4 • FX5U/FX5UC CPU module K1 to 4
3600H	Channel number specified by (d) is not set by parameters.

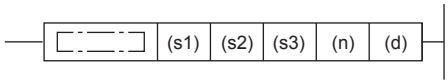
For communication errors, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Inverter parameter block write

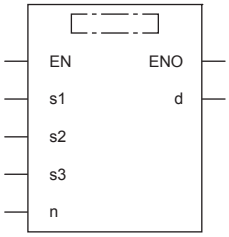
IVBWR

FX5S FX5UJ FX5U FX5UC

This instruction writes parameters of an inverter from the CPU module in a batch.

Ladder diagram	Structured text
	ENO:=IVBWR(EN,s1,s2,s3,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Number of parameters in an inverter to be written at one time	*1	16-bit signed binary	ANY16
(s3)	Head device of a parameter table to be written to an inverter	—	16-bit signed binary	ANY16
(n)	Communication channel	■FX5S/FX5UJ CPU module K2 to 4 ■FX5U CPU module K1 to 4 ■FX5UC CPU module K1, K3 to 4	16-bit unsigned binary	ANY16_U
(d)	Head bit device to which the execution status of the instruction is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Refer to MELSEC iQ-F FX5 User's Manual (Communication) or respective inverter manual.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	○	○	—	—	○	○	—	—	—
(s2)	—	○*1	○	○	—	—	○	○	—	—	—
(s3)	—	○	○	—	—	—	○	—	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

A data table specified in (s2) and (s3) is written to an inverter connected to the communication channel (n) whose station number is (s1) in batch. For details, refer to MELSEC iQ-F FX5 User's Manual (Communication). (For the parameter numbers, refer to the each inverter manual.)

Precautions

Three devices are occupied from the device specified in (d). Make sure that these devices are not used in other controls.

Operation error

Error code (SD0/SD8067)	Description
1810H	Channel number specified by (d) is used by another instruction.
2820H	The specified device exceeds the range of the corresponding device.
3405H	The value specified by (s1) is other than any of K0 to 31.
	The value specified by (s2) is K0 or less.
	The value specified by (s3) is outside the allowable range. (Less than K0, K3000 to 9999, or K13000 to 32767)
	When the value specified for (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to 4 • FX5U/FX5UC CPU module K1 to 4
3600H	Channel number specified by (d) is not set by parameters.

For communication errors, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Inverter multi command

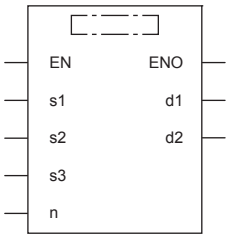
IVMC

FX5S FX5UJ FX5U FX5UC

This instruction writes 2 types of settings (operation command and set frequency) to the inverter, and reads 2 types of data (inverter status monitor, output frequency, etc.) from the inverter at the same time.

Ladder diagram	Structured text
	ENO:=IVMC(EN,s1,s2,s3,n,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Inverter station number	K0 to 31	16-bit signed binary	ANY16
(s2)	Multiple instructions for inverter: Send/receive data type specification	*1	16-bit signed binary	ANY16
(s3)	Head device which stores data to be written to the inverter	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 2)
(d1)	Head device which stores values to be read from the inverter	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 2)
(n)	Communication channel	■FX5S/FX5UJ CPU module K2 to 4 ■FX5U CPU module K1 to 4 ■FX5UC CPU module K1, K3 to 4	16-bit unsigned binary	ANY16_U
(d2)	Head bit device to which the execution status of the instruction is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Refer to MELSEC iQ-F FX5 User's Manual (Communication).

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	○	○	—	—	○	○	—	—	—
(s2)	—	○*1	○	○	—	—	○	○	—	—	—
(s3)	—	○	○	—	—	—	○	—	—	—	—
(d1)	—	○	○	—	—	—	○	—	—	—	—
(n)	—	—	—	—	—	—	—	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes multiple commands of an inverter connected to the communication channel (n) whose station number is specified in (s1). Specify the send/receive data type using (s2), the head device which stores data to be written to the inverter using (s3), and the head device which stores values to be read from the inverter using (d1). For details, refer to 📖MELSEC iQ-F FX5 User's Manual (Communication).

Precautions

- If a device number outside the range due to indexing, etc. is specified in (d1), the receive data from the inverter is not stored in (d1). However, values set in (s3) and (s3)+1 may be written to the inverter.
- If any unspecified value is set in (s2), unexpected data may be written to and read from the inverter, and values of (d1) and (d1)+1 may be updated.
- The IVMC instruction reads the inverter status at the time of communication with the inverter, and stores it in (d1). Accordingly, the inverter status written by the IVMC instruction can be read when the next reading instruction (IVCK, IVMC, etc.) is executed.
- Two devices are occupied from the device specified in (s3) and (d1). Make sure that these devices are not used in other controls.
- Three devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

Operation error

Error code (SD0/SD8067)	Description
1810H	Channel number specified by (d) is used by another instruction.
2820H	The specified device exceeds the range of the corresponding device.
3405H	The value specified by (s1) is other than any of K0 to 31.
	When the value specified for (n) is other than the following channel numbers. • FX5S/FX5UJ CPU module K2 to 4 • FX5U/FX5UC CPU module K1 to 4
3600H	Channel number specified by (d) is not set by parameters.

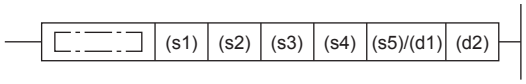
For communication errors, refer to 📖MELSEC iQ-F FX5 User's Manual (Communication).

17.3 MODBUS Communication Instruction

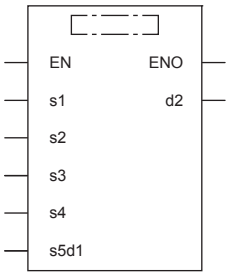
ADPRW

FX5S FX5UJ FX5U FX5UC

This instruction allows the MODBUS Master to communicate (read/write data) with the Slaves.

Ladder diagram	Structured text
	ENO:=ADPRW(EN,s1,s2,s3,s4,s5d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Slave station No.	0 to F7H*1	16-bit signed binary	ANY16
(s2)	Function code	01H to 06H, 0FH, 10H	16-bit signed binary	ANY16
(s3)	Function parameters depending on the function code	0 to FFFFH	16-bit signed binary	ANY16
(s4)	Function parameters depending on the function code	1 to 2000	16-bit signed binary	ANY16
(s5)/(d1)	Function parameters depending on the function code	—	Bit/16-bit signed binary	ANY_ELEMENTARY*2
(d2)	Head bit device number to which the execution status of the communication is output	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Slave station No. that can be used in FX5 master station differs depending on the version. (☞ Page 1452 Added and Changed Functions)

*2 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○*1	○	○	—	—	○	○	—	—	—
(s2)	—	○*1	○	○	—	—	○	○	—	—	—
(s3)	—	○*1	○	○	—	—	○	○	—	—	—
(s4)	—	○*1	○	○	—	—	○	○	—	—	—
(s5)/(d1)	○	○*1	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

- Function code (s2) is operated on Slave station No. (s1) according to Parameters (s3), (s4), and (s5)/(d1). Use 0 as the Slave station No. for Broadcast commands. For details, refer to MELSEC iQ-F FX5 User's Manual (Communication).
- The communication execution status (d2) is output according to the status of the ADPRW instruction such as communicating/completed normally/completed with an error.

Precautions

Three devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

Operation error

Error code (SD0/SD8067)	Description
1810H	Channel used by the instruction is used by other instruction.
3600H	Invalid parameter setup.
2822H	Device that cannot be used by this instruction is specified.
3405H	Data outside the allowable range was input.
2820H	The specified device exceeds the range of the corresponding device.

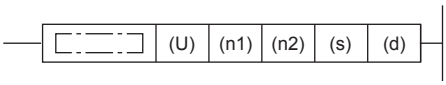
For communication errors, refer to MELSEC iQ-F FX5 User's Manual (Communication).

17.4 Predefined Protocol Support Function Instruction

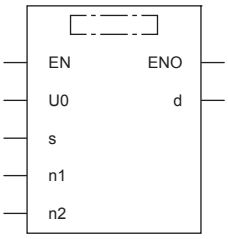
S(P).CPRTCL

FX5S FX5UJ FX5U FX5UC

This instruction executes the communication protocol registered using the engineering tool.

Ladder diagram	Structured text
	ENO:=S_CPRTCL(EN,U0,s,n1,n2,d); ENO:=SP_CPRTCL(EN,U0,s,n1,n2,d);

FBD/LD



("S_CPRTCL", "SP_CPRTCL" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Dummy (Input the character string ['U0'].)	—	Character string	ANYSTRING_SINGLE
(n1)	Communication channel	■FX5S/FX5UJ CPU module K2 to 4 ■FX5U CPU module K1 to 4 ■FX5UC CPU module K1, K3 to 4	16-bit unsigned binary	ANY16_U
(n2)	Number of protocols to be executed continuously	1 to 8	16-bit unsigned binary	ANY16_U
(s)	Head device number for storing the control data	Refer to Control data (☞ Page 1130)	Word	ANY16_ARRAY (Number of elements: 18)
(d)	Head device number which turns ON when the execution of the instruction is completed and remains on for 1 scan. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as U0.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	—	—	○	—
(n1)	—	○	—	—	—	—	○	○	—	—	—
(n2)	—	○	—	—	—	—	○	○	—	—	—
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by*1
(s)+0	Completion status	The completion status is stored upon completion of the instruction. When two or more protocols are executed, the execution result of the protocol executed last is stored. 0: Normal completion Other than 0: Error completion (error code)	—	System
(s)+1	Resulting number of executed protocols	The number of protocols executed by the S(P).CPRTCL instruction is stored. Any protocol where an error occurred is also included in the execution number. If the setting of setting data or control data contains an error, "0" is stored.	0, 1 to 8	System
(s)+2	Execution protocol number 1	Specify the number of the protocol to be executed first.	1 to 64	User
(s)+3	Execution protocol number 2	Specify the number of the protocol to be executed second.	0, 1 to 64	
(s)+4	Execution protocol number 3	Specify the number of the protocol to be executed third.	0, 1 to 64	
(s)+5	Execution protocol number 4	Specify the number of the protocol to be executed fourth.	0, 1 to 64	
(s)+6	Execution protocol number 5	Specify the number of the protocol to be executed fifth.	0, 1 to 64	
(s)+7	Execution protocol number 6	Specify the number of the protocol to be executed sixth.	0, 1 to 64	
(s)+8	Execution protocol number 7	Specify the number of the protocol to be executed seventh.	0, 1 to 64	
(s)+9	Execution protocol number 8	Specify the number of the protocol to be executed eighth.	0, 1 to 64	
(s)+10	Collation match Receive packet number 1	If receiving is included in the communication type of the protocol that has been executed first, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the first protocol, "0" is stored.	0, 1 to 16	System
(s)+11	Collation match Receive packet number 2	If receiving is included in the communication type of the protocol that has been executed second, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the second protocol, "0" is stored. If the number of protocols executed is less than 2, "0" is stored.	0, 1 to 16	
(s)+12	Collation match Receive packet number 3	If receiving is included in the communication type of the protocol that has been executed third, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the third protocol, "0" is stored. If the number of protocols executed is less than 3, "0" is stored.	0, 1 to 16	
(s)+13	Collation match Receive packet number 4	If receiving is included in the communication type of the protocol that has been executed fourth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the fourth protocol, "0" is stored. If the number of protocols executed is less than 4, "0" is stored.	0, 1 to 16	
(s)+14	Collation match Receive packet number 5	If receiving is included in the communication type of the protocol that has been executed fifth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the fifth protocol, "0" is stored. If the number of protocols executed is less than 5, "0" is stored.	0, 1 to 16	
(s)+15	Collation match Receive packet number 6	If receiving is included in the communication type of the protocol that has been executed sixth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the sixth protocol, "0" is stored. If the number of protocols executed is less than 6, "0" is stored.	0, 1 to 16	
(s)+16	Collation match Receive packet number 7	If receiving is included in the communication type of the protocol that has been executed seventh, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the seventh protocol, "0" is stored. If the number of protocols executed is less than 7, "0" is stored.	0, 1 to 16	
(s)+17	Collation match Receive packet number 8	If receiving is included in the communication type of the protocol that has been executed eighth, the receive packet number successful in collation match is stored. If the communication type is "receive only", "0" is stored. If an error occurs during execution of the eighth protocol, "0" is stored. If the number of protocols executed is less than 8, "0" is stored.	0, 1 to 16	

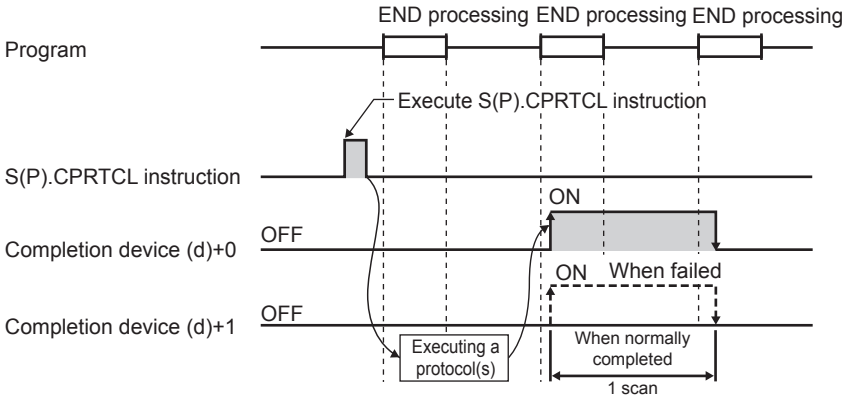
*1 User: Data to be set before the execution of the instruction. System: The CPU module stores the execution result of the instruction.

Processing details

This instruction executes the protocol registered using the engineering tool. Using CH specified by (n1), the instruction executes the protocol in accordance with the control data stored in the device specified by (s) and later. The instruction continuously executes as many protocols as specified by (n2) (a maximum of 8 protocols) at one time.
The number of executed protocols is stored in the device specified by (s)+1.
The completion of the S(P).CPRTCL instruction can be checked using the completion devices (d)+0 and (d)+1.

- Completion device (d)+0: Turns ON during the END processing for the scan in which the S(P).CPRTCL instruction is completed, and turns OFF during the next END processing.
- Completion device (d)+1: Turns ON or OFF depending on the status when the S(P).CPRTCL instruction is completed.

Status	Description
When completed normally	The device does not change (remains OFF).
When completed with an error	The device turns ON during the END processing for the scan in which the S(P).CPRTCL instruction is completed, and turns OFF during the next END processing.



For details, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Precautions

- If an error occurs in the mth protocol while multiple protocols are being executed, the instruction does not execute the "m+1"th protocol and after and is completed with an error.
- The communication CH for which the S(P).CPRTCL instruction can be executed are only those for which "Predefined protocol support function" is specified for the communication protocol.
- If a cancel request is received during execution of the mth protocol while multiple protocols are executed continuously, the following is stored in (s).

Device	Item	Description
(s)+0	Completion status	The error codes.
(s)+1	Resulting number of executed protocols	The executed protocol number.
(s)+10	Collation match Receive packet number 1	The receive packet number successful in collation match for the already executed protocol.
⋮	⋮	
(s)+m+8	Collation match Receive packet number m-1	

- If same instructions are executed for the same CH, the subsequent instruction is ignored and is not executed until the preceding instruction is completed.

For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Communication).

Operation error

Error code (SD0/SD8067)	Description
2820H	The device used exceeded the specified range.
2821H	The device used to store data are overlapping.
2822H	Device that cannot be specified is specified.
3405H	The input data was out of range.

18 POSITIONING INSTRUCTION

18.1 Dedicated Instruction (Positioning Function)

Instruction correspondence table for CPU module built-in positioning and high-speed pulse input/output module

The table for instructions corresponding to the CPU module and the high-speed pulse input/output module are shown below.

○: Supported, —: Not supported

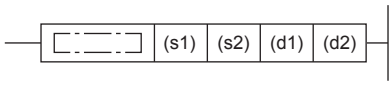
Instruction	CPU module	High-speed pulse input/ output module
DSZR [For the FX3 compatible operand specification]	○	—
DSZR [For the FX5 operand specification]	○	○
DDSZR	○	○
DVIT [For the FX3 compatible operand specification]	○	—
DVIT [For the FX5 operand specification]	○	○
DDVIT [For the FX3 compatible operand specification]	○	—
DDVIT [For the FX5 operand specification]	○	○
TBL [For the FX3 compatible operand specification]	○	—
TBL [For the FX5 operand specification]	○	—
DRVTBL	○	○
DRVMUL	○	○
DABS	○	○
PLSV [For the FX3 compatible operand specification]	○	—
PLSV [For the FX5 operand specification]	○	○
DPLSV [For the FX3 compatible operand specification]	○	—
DPLSV [For the FX5 operand specification]	○	○
DRVI [For the FX3 compatible operand specification]	○	—
DRVI [For the FX5 operand specification]	○	○
DDRVI [For the FX3 compatible operand specification]	○	—
DDRVI [For the FX5 operand specification]	○	○
DRVA [For the FX3 compatible operand specification]	○	—
DRVA [For the FX5 operand specification]	○	○
DDRVA [For the FX3 compatible operand specification]	○	—
DDRVA [For the FX5 operand specification]	○	○

Zero return(OPR) with 16-bit data DOG search

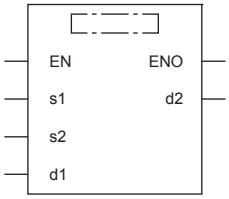
DSZR [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes mechanical zero return. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=DSZR(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Bit device number to which the near-point dog signal is input	—	Bit	ANY_ELEMENTARY (BOOL)
(s2)	Bit device number to which the zero-phase signal is input	—	Bit	ANY_ELEMENTARY (BOOL)
(d1)	Bit device number (Y) from which pulses are output	■FX5S/FX5UJ CPU module 0 to 2 ■FX5U/FX5UC CPU module 0 to 3	Bit	ANY_ELEMENTARY (BOOL)
(d2)	Bit device number from which the rotation direction is output	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○*1	○*5	—	—	—	—	—	—	—	—	—
(s2)	○*1*2	○*5	—	—	—	—	—	—	—	—	—
(d1)	○*3	—	—	—	—	—	—	—	—	—	—
(d2)	○*4	○*5	—	—	—	—	—	—	—	—	—

*1 When using X, always specify a device that has been set by parameter.
*2 Specify the device set with a parameter or same as the one set in (s1).
*3 Only Y can be used.
*4 When the output mode is CW/CCW, specify the CCW axis. When the output mode is PULSE/SIGN and using Y, only the SIGN output or general-purpose output of the self-axis can be specified.
*5 T, ST, and C cannot be used.

Processing details

This instruction executes mechanical zero return. The values of special devices are applied as the zero return speed and creep speed. With the forward limit or reverse limit, zero return with the dog search function can be executed.

- For (s1), specify the near-point dog signal input device number. When an X device is specified, the near-point dog signal functions follow the logic set by parameter. When other than X device is specified, the device functions follow the positive logic.
- For (s2), specify the zero-phase signal input device number. When an X device is specified, the zero-phase signal functions follow the logic set by parameter. When other than X device is specified, the device functions follow the positive logic.
- For (d1), specify the device from which pulses are output. Only the output devices (Y) having positioning parameters can be specified.
- For (d2), specify the bit device from which the rotation direction signal is output. Only the device specified with a parameter or general-purpose outputs can be specified. When the output devices (Y) is executed by another function (PWM, positioning PULSE axis, or CW/CCW axis etc.), the device does not function and causes an error.

For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

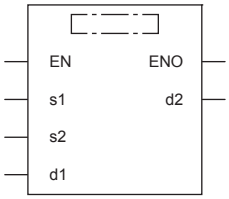
DSZR [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes mechanical zero return.

Ladder diagram	Structured text
	ENO:=DSZR(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Zero return speed	1 to 65535	16-bit unsigned binary	ANY_ELEMENTARY (WORD)
(s2)	Creep speed	1 to 65535	16-bit unsigned binary	ANY_ELEMENTARY (WORD)
(d1)	Axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
(d2)	Bit device number of the zero return complete flag or abnormal end flag	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d1)	—	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes mechanical zero return. The near-point dog signal and zero-phase signal function follow the device set with parameters. With the forward limit or reverse limit, zero return with the dog search function can be executed.

- For (s1), specify the zero return speed in the user units. (The speed must be 200 kpps or lower in frequency.)
- For (s2), specify the creep speed in the user units. Set the creep speed equal to or slower than the zero return speed set in (s1). (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the axis number for which zero return is performed.
- For (d2), specify the bit device of the zero return complete flag or abnormal end flag.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

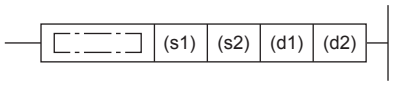
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

Zero return(OPR) with 32-bit data DOG search

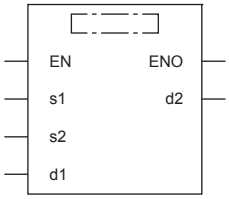
DDSZR

FX5S FX5UJ FX5U FX5UC

This instruction executes mechanical zero return.

Ladder diagram	Structured text
	ENO:=DDSZR(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Zero return speed	1 to 2147483647	32-bit signed binary	ANY32
(s2)	Creep speed	1 to 2147483647	32-bit signed binary	ANY32
(d1)	Axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY16
(d2)	Bit device number of the zero return complete flag or abnormal end flag	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d1)	—	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes mechanical zero return. The near-point dog signal and zero-phase signal function follow the device set with parameters. With the forward limit or reverse limit, zero return with the dog search function can be executed.

- For (s1), specify the zero return speed in the user units. (The speed must be 200 kpps or lower in frequency.)
- For (s2), specify the creep speed in the user units. Set the creep speed equal to or slower than the zero return speed set in (s1). (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the axis number for which zero return is performed.
- For (d2), specify the bit device of the zero return complete flag or abnormal end flag.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

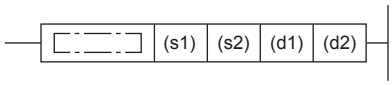
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

16-bit data interrupt positioning

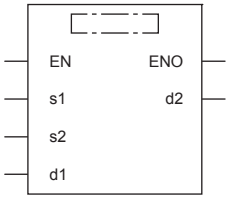
DVIT [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes interrupt 1-speed constant quantity feed. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=DVIT(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address after an interrupt input	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Command speed	1 to 65535	16-bit unsigned binary	ANY16
(d1)	Bit device number (Y) from which pulses are output	■FX5S/FX5UJ CPU module 0 to 2 ■FX5U/FX5UC CPU module 0 to 3	Bit	ANY_ELEMENTARY (BOOL)
(d2)	Bit device number from which the rotation direction is output	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d1)	○*1	—	—	—	—	—	—	—	—	—	—
(d2)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 Only Y can be used.
*2 When the output mode is CW/CCW, specify the CCW axis. When the output mode is PULSE/SIGN and using Y, only the SIGN output or general-purpose output of the self-axis can be specified.
*3 T, ST, and C cannot be used.

Processing details

This instruction executes interrupt 1-speed constant quantity feed.

- For (s1), specify the transfer distance that is output after an interrupt, in user units. (The distance must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the device from which pulses are output. Only the output devices (Y) having positioning parameters can be specified.
- For (d2), specify the bit device from which the rotation direction signal is output. Only the device specified with a parameter or general-purpose outputs can be specified. When the output devices (Y) is executed by another function (PWM, positioning PULSE axis, or CW/CCW axis etc.), the device does not function and causes an error.

For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

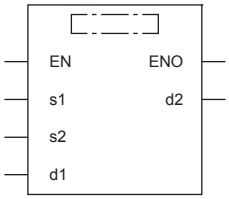
DVIT [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes interrupt 1-speed constant quantity feed.

Ladder diagram	Structured text
	ENO:=DVIT(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address after an interrupt input	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Command speed	1 to 65535	16-bit unsigned binary	ANY16
(d1)	Specify the axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
(d2)	Bit device number of the positioning complete flag or abnormal end flag	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d1)	—	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes interrupt 1-speed constant quantity feed.

- For (s1), specify the transfer distance that is output after an interrupt, in user units. (The distance must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the axis number from which pulses are output.
- For (d2), specify the bit device of the normal complete flag or abnormal end flag for the DVIT instruction.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

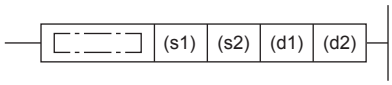
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

32-bit data interrupt positioning

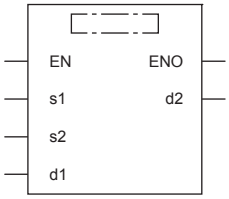
DDVIT [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes interrupt 1-speed constant quantity feed. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=DDVIT(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address after an interrupt input	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Command speed	1 to 2147483647	32-bit signed binary	ANY32
(d1)	Bit device number (Y) from which pulses are output	■FX5S/FX5UJ CPU module 0 to 2 ■FX5U/FX5UC CPU module 0 to 3	Bit	ANY_ELEMENTARY (BOOL)
(d2)	Bit device number from which the rotation direction is output	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d1)	○*1	—	—	—	—	—	—	—	—	—	—
(d2)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 Only Y can be used.
*2 When the output mode is CW/CCW, specify the CCW axis. When the output mode is PULSE/SIGN and using Y, only the SIGN output or general-purpose output of the self-axis can be specified.
*3 T, ST, and C cannot be used.

Processing details

This instruction executes interrupt 1-speed constant quantity feed.

- For (s1), specify the transfer distance that is output after an interrupt, in user units. (The distance must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the device from which pulses are output. Only the output devices (Y) having positioning parameters can be specified.
- For (d2), specify the bit device from which the rotation direction signal is output. Only the device specified with a parameter or general-purpose outputs can be specified. When the output devices (Y) is executed by another function (PWM, positioning PULSE axis, or CW/CCW axis etc.), the device does not function and causes an error.

For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

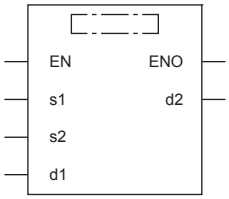
DDVIT [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes interrupt 1-speed constant quantity feed.

Ladder diagram	Structured text
	ENO:=DDVIT(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address after an interrupt input	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Command speed	1 to 2147483647	32-bit signed binary	ANY32
(d1)	Specify the axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
(d2)	Bit device number of the positioning complete flag or abnormal end flag	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d1)	—	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes interrupt 1-speed constant quantity feed.

- For (s1), specify the transfer distance that is output after an interrupt, in user units. (The distance must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the axis number from which pulses are output.
- For (d2), specify the bit device of the normal complete flag or abnormal end flag for the DDVIT instruction.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

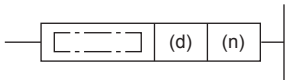
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

Positioning by one table operation

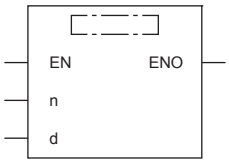
TBL [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes one specified table operation from the instructions set in the data table using the engineering tool etc. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=TBL(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Bit device number (Y) from which pulses are output	■FX5S/FX5UJ CPU module 0 to 2 ■FX5U/FX5UC CPU module 0 to 3	Bit	ANY_ELEMENTARY (BOOL)
(n)	Table number to be executed	1 to 100*1	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 If the table data is not stored to devices, the table number range is 1 to 32.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	○*1	—	—	—	—	—	—	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

*1 Only Y can be used.

Processing details

This instruction operates one table of the positioning table that is set with parameters in the engineering tool.

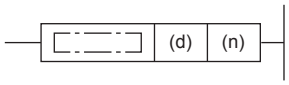
- For (d), specify the device from which pulses are output. Only the output devices (Y) having positioning parameters can be specified.
- For (n), specify the table number to be executed according to the output specified in (d).

For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

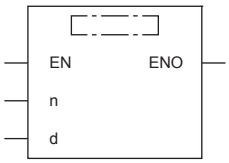
TBL [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes one specified table operation from the instructions set in the data table using the engineering tool etc. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=TBL(EN,n,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d)	Axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3 ■FX5U/FX5UC CPU module K1 to K4	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*2
(n)	Table number to be executed	1 to 100*1	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 If the table data is not stored to devices, the table number range is 1 to 32.
*2 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d)	—	○	○	○	—	—	○	○	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

Processing details

This instruction operates one table of the positioning table that is set with parameters in the engineering tool.

- For (d), specify the axis number from which pulses are output.
- For (n), specify the table number to be executed according to the output specified in (d).

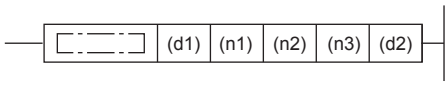
For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Positioning by multiple table operation

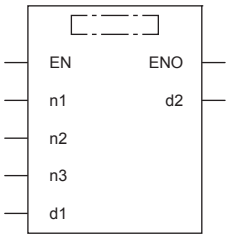
DRVTBL

FX5S FX5UJ FX5U FX5UC

This instruction executes positioning operation set in multiple data tables with the engineering tool in continuous operation or stepping operation. To execute such operation, this instruction needs to be executed only once.

Ladder diagram	Structured text
	ENO:=DRVTBL(EN,n1,n2,n3,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(d1)	Axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY16
(n1)	Start table number to be executed	1 to 100*1	16-bit unsigned binary	ANY16_U
(n2)	Last table number to be executed	1 to 100*1	16-bit unsigned binary	ANY16_U
(n3)	Table execution method	K0, K1	16-bit unsigned binary	ANY16_U
(d2)	Bit device number of the positioning complete flag or abnormal end flag	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 If the table data is not stored to devices in the CPU module, the table number range is 1 to 32.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(d1)	○	○	○	○	—	—	○	○	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—
(n3)	○	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes positioning operation set in multiple data tables with the engineering tool in the continuous operation or stepping operation. To execute such operation, this instruction needs to be executed only once.

- For (d1), specify the axis number from which pulses are output.
- For (n1), specify the start table to be executed according to the output specified in (d1).
- For (n2), specify the last table. When (n1) and (n2) are the same, only one table is executed. The table operation keeps executing until the last table or a table that is not set with parameters is executed.
- For (n3), specify the table execution method. (K0 = Stepping operation, K1 = Continuous operation)
- For (d2), specify the bit device of the normal complete flag or abnormal end flag.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

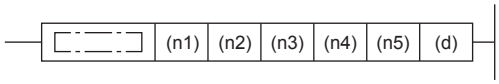
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

Multiple axes concurrent drive positioning

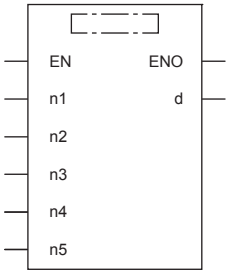
DRVMUL

FX5S FX5UJ FX5U FX5UC

This instruction executes tables of multiple axes of one module simultaneously.

Ladder diagram	Structured text
	ENO:=DRVMUL(EN,n1,n2,n3,n4,n5,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(n1)	Start axis number	K1, K5, K7, K9, K11	16-bit unsigned binary	ANY16_U
(n2)	Table number of the axis 1	K0 to 100*1	16-bit unsigned binary	ANY16_U
(n3)	Table number of the axis 2	K0 to 100*1	16-bit unsigned binary	ANY16_U
(n4)	Table number of the axis 3	K0 to 100*1	16-bit unsigned binary	ANY16_U
(n5)	Table number of the axis 4	K0 to 100*1	16-bit unsigned binary	ANY16_U
(d)	Bit device number of the positioning complete flag or abnormal end flag	—	Bit	ANYBIT_ARRAY (Number of elements: 8)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 If the table data is not stored to devices in the CPU module, the table number range is 1 to 32.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—
(n3)	○	○	○	○	—	—	○	○	—	—	—
(n4)	○	○	○	○	—	—	○	○	—	—	—
(n5)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes tables of multiple axes of one module simultaneously.

- For (n1), specify the head axis number for which pulses are output.

Designation number	Axis numbers to be used
K1	■FX5S/FX5U/FX5UC CPU module Execute axis 1 to axis 4 (CPU module) simultaneously. ■FX5UJ CPU module Execute axis 1 to axis 3 (CPU module) simultaneously.
K5	Execute axis 5, axis 6 (high-speed pulse input/output module first module) simultaneously.
K7	Execute axis 7, axis 8 (high-speed pulse input/output module second module) simultaneously.
K9	Execute axis 9, axis 10 (high-speed pulse input/output module third module) simultaneously.
K11	Execute axis 11, axis 12 (high-speed pulse input/output module fourth module) simultaneously.

- For (n2), specify the table number that is executed with the axis (n1). When not executing the axis (n1), specify K0.
- For (n3), specify the table number that is executed with the axis (n1)+1. When not executing the axis (n1)+1, specify K0.
- For (n4), specify the table number that is executed with the axis (n1)+2. If axis (n1) + 2 is not executed or the axis number of the high-speed pulse input/output module is specified to (n1), specify K0.
- For (n5), specify the table number that is executed with the axis (n1)+3. If axis (n1)+3 is not executed or the axis number of the FX5S/FX5UJ CPU module or high-speed pulse input/output module is specified to (n1), specify K0.
- For (d), specify the device*1 of the instruction execution complete flag for each axis. Each device functions as the following flag.

Device	Description
(d)	Instruction execution complete flag for the axis (n1)
(d)+1	Instruction execution abnormal end flag for the axis (n1)
(d)+2	Instruction execution complete flag for the axis (n1)+1
(d)+3	Instruction execution abnormal end flag for the axis (n1)+1
(d)+4	Instruction execution complete flag for the axis (n1)+2
(d)+5	Instruction execution abnormal end flag for the axis (n1)+2
(d)+6	Instruction execution complete flag for the axis (n1)+3*2
(d)+7	Instruction execution abnormal end flag for the axis (n1)+3*2

*1 The following number of device points are occupied starting with (d).
FX5S/FX5UJ CPU module: 6 points
FX5U/FX5UC CPU module: 8 points

*2 The complete flag is assigned only in FX5U/FX5UC CPU module.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Make sure that the instruction execution complete flag and devices used with other controls are not duplicated.

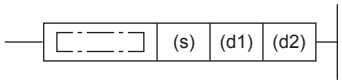
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

32-bit data ABS current value read

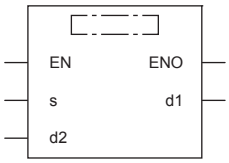
DABS

FX5S FX5UJ FX5U FX5UC

This instruction reads the absolute position (ABS) data when a servo amplifier (equipped with the absolute position detection function) is connected. The data is converted into pulse when read.

Ladder diagram	Structured text
	ENO:=DABS(EN,s,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Head device number that inputs the output signal for absolute position (ABS) data from the servo amplifier	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
(d1)	Head device number that outputs the absolute position (ABS) data control signal to the servo amplifier	—	Bit	ANYBIT_ARRAY (Number of elements: 3)
(d2)	Absolute position (ABS) data (32-bit value)	—	32-bit signed binary	ANY32
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○*1	—	—	—	—	—	—	—	—	—
(d1)	○	○*1	—	—	—	—	—	—	—	—	—
(d2)	○	○	○	○	○	○	○	○	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction reads the absolute position (ABS) data when a servo amplifier (equipped with the absolute position detection function) is connected. The data is converted into pulse when being read.

- For (s), specify the head device number that inputs the output signal for absolute position (ABS) data from the servo amplifier.
- For (d1), specify the head device number that outputs the absolute position (ABS) data control signal to the servo amplifier. Be sure to use transistor outputs for the CPU module outputs.
- For (d2), specify the device that stores the absolute position (ABS) data read from the servo amplifier.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Three devices are occupied from the device specified in (s) and (d1). Make sure that these devices are not used in other controls.

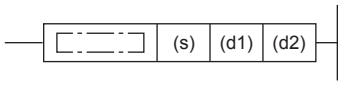
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

16-bit data variable speed pulse

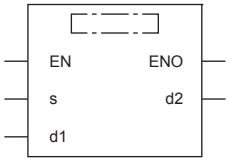
PLSV [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction outputs variable speed pulses with an assigned rotation direction output. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=PLSV(EN,s,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Command speed	-32768 to +32767	16-bit signed binary	ANY16
(d1)	Bit device number (Y) from which pulses are output	■FX5S/FX5UJ CPU module 0 to 2 ■FX5U/FX5UC CPU module 0 to 3	Bit	ANY_ELEMENTARY (BOOL)
(d2)	Bit device number from which the rotation direction is output	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d1)	○*1	—	—	—	—	—	—	—	—	—	—
(d2)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 Only Y can be used.

*2 When the output mode is CW/CCW, specify the CCW axis. When the output mode is PULSE/SIGN and using Y, only the SIGN output or general-purpose output of the self-axis can be specified.

*3 T, ST, and C cannot be used.

Processing details

This instruction outputs variable speed pulses with an assigned rotation direction output.

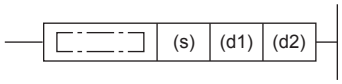
- For (s), specify the command speed to be output. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the device from which pulses are output. Only the output devices (Y) having positioning parameters can be specified.
- For (d2), specify the bit device from which the rotation direction signal is output. Only the device specified with the parameter or general-purpose outputs can be specified. When the output devices (Y) is executed by another function (PWM, positioning PULSE axis, or CW/CCW axis etc.), the device does not function and causes an error.

For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

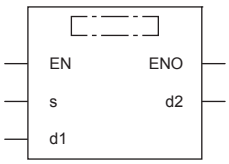
PLSV [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction outputs variable speed pulses with an assigned rotation direction output.

Ladder diagram	Structured text
	ENO:=PLSV(EN,s,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Command speed	-32768 to +32767	16-bit signed binary	ANY16
(d1)	Axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
(d2)	Bit device number of the positioning complete flag or abnormal end flag	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d1)	—	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction outputs variable speed pulses with an assigned rotation direction output.

- For (s), specify the command speed to be output. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the axis number from which pulses are output.
- For (d2), specify the bit device of the abnormal end flag for the PLSV instruction. (This device does not have the normal complete status, and only has the abnormal end status ((d2)+1).

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

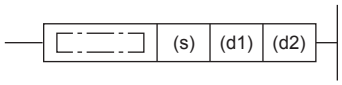
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

32-bit data variable speed pulse

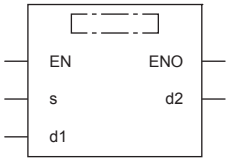
DPLSV [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction outputs variable speed pulses with an assigned rotation direction output. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=DPLSV(EN,s,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Command speed	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d1)	Bit device number (Y) from which pulses are output	■FX5S/FX5UJ CPU module 0 to 2 ■FX5U/FX5UC CPU module 0 to 3	Bit	ANY_ELEMENTARY (BOOL)
(d2)	Bit device number from which the rotation direction is output	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d1)	○*1	—	—	—	—	—	—	—	—	—	—
(d2)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 Only Y can be used.

*2 When the output mode is CW/CCW, specify the CCW axis. When the output mode is PULSE/SIGN and using Y, only the SIGN output or general-purpose output of the self-axis can be specified.

*3 T, ST, and C cannot be used.

Processing details

This instruction outputs variable speed pulses with an assigned rotation direction output.

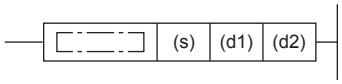
- For (s), specify the command speed to be output. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the device from which pulses are output. Only the output devices (Y) having positioning parameters can be specified.
- For (d2), specify the bit device from which the rotation direction signal is output. Only the device specified with the parameter or general-purpose outputs can be specified. When the output devices (Y) is executed by another function (PWM, positioning PULSE axis, or CW/CCW axis etc.), the device does not function and causes an error.

For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

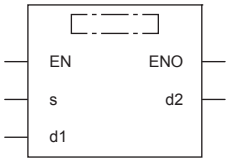
DPLSV [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction outputs variable speed pulses with an assigned rotation direction output.

Ladder diagram	Structured text
	ENO:=DPLSV(EN,s,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s)	Command speed	-2147483648 to +2147483647	32-bit signed binary	ANY32
(d1)	Axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
(d2)	Bit device number of the positioning complete flag or abnormal end flag	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d1)	—	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction outputs variable speed pulses with an assigned rotation direction output.

- For (s), specify the command speed to be output. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the axis number from which pulses are output.
- For (d2), specify the bit device of the abnormal end flag for the DPLSV instruction. (This device does not have the normal complete status, and only has the abnormal end status ((d2)+1).

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

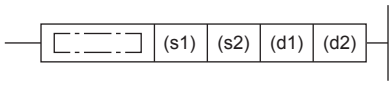
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

16-bit data relative positioning

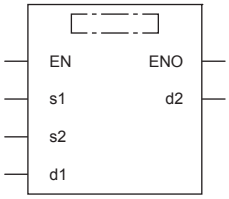
DRVI [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes one-speed positioning by incremental drive. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=DRVI(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Command speed	1 to 65535	16-bit unsigned binary	ANY16
(d1)	Output bit device number (Y) from which pulses are output	■FX5S/FX5UJ CPU module 0 to 2 ■FX5U/FX5UC CPU module 0 to 3	Bit	ANY_ELEMENTARY (BOOL)
(d2)	Bit device number from which the rotation direction is output	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d1)	○*1	—	—	—	—	—	—	—	—	—	—
(d2)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 Only Y can be used.

*2 When the output mode is CW/CCW, specify the CCW axis. When the output mode is PULSE/SIGN and using Y, only the SIGN output or general-purpose output of the self-axis can be specified.

*3 T, ST, and C cannot be used.

Processing details

This instruction executes one-speed positioning by incremental drive. Specify the positioning address in the incremental system, in which the transfer direction and transfer distance from the current position (relative address) are specified for positioning.

- For (s1), specify the relative positioning address in user units. (The address must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the command speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the device from which pulses are output. Only the Y devices having positioning parameters can be specified.
- For (d2), specify the bit device from which the rotation direction signal is output. Only the device specified with the parameter or general-purpose outputs can be specified. When the output devices (Y) is executed by another function (PWM, positioning PULSE axis, or CW/CCW axis etc.), the device does not function and causes an error.

For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

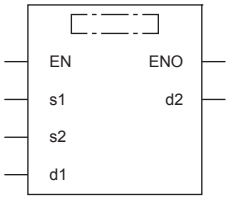
DRVI [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes one-speed positioning by incremental drive.

Ladder diagram	Structured text
	ENO:=DRVI(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Command speed	1 to 65535	16-bit unsigned binary	ANY16
(d1)	Axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
(d2)	Bit device number of the positioning complete flag or abnormal end flag	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d1)	—	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes one-speed positioning by incremental drive. Specify the positioning address in the incremental system, in which the transfer direction and transfer distance from the current position (relative address) are specified for positioning.

- For (s1), specify the relative positioning address in user units. (The address must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the command speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the axis number from which pulses are output.
- For (d2), specify the bit device of the normal complete flag or abnormal end flag for the DRVI instruction.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

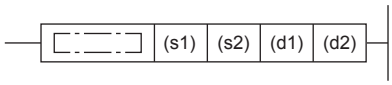
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

32-bit data relative positioning

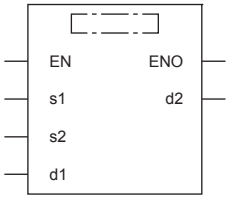
DDRVI [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes one-speed positioning by incremental drive. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=DDRVI(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Command speed	1 to 2147483647	32-bit signed binary	ANY32
(d1)	Output bit device number (Y) from which pulses are output	■FX5S/FX5UJ CPU module 0 to 2 ■FX5U/FX5UC CPU module 0 to 3	Bit	ANY_ELEMENTARY (BOOL)
(d2)	Bit device number from which the rotation direction is output	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d1)	○*1	—	—	—	—	—	—	—	—	—	—
(d2)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 Only Y can be used.
*2 When the output mode is CW/CCW, specify the CCW axis. When the output mode is PULSE/SIGN and using Y, only the SIGN output or general-purpose output of the self-axis can be specified.
*3 T, ST, and C cannot be used.

Processing details

This instruction executes one-speed positioning by incremental drive. Specify the positioning address in the incremental system, in which the transfer direction and transfer distance from the current position (relative address) are specified for positioning.

- For (s1), specify the relative positioning address in user units. (The address must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the command speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the device from which pulses are output. Only the Y devices having positioning parameters can be specified.
- For (d2), specify the bit device from which the rotation direction signal is output. Only the device specified with the parameter or general-purpose outputs can be specified. When the output devices (Y) is executed by another function (PWM, positioning PULSE axis, or CW/CCW axis etc.), the device does not function and causes an error.

For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

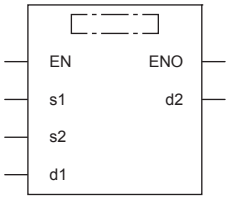
DDRVI [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes one-speed positioning by incremental drive.

Ladder diagram	Structured text
	ENO:=DDRVI(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Command speed	1 to 2147483647	32-bit signed binary	ANY32
(d1)	Axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
(d2)	Bit device number of the positioning complete flag or abnormal end flag	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d1)	—	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes one-speed positioning by incremental drive. Specify the positioning address in the incremental system, in which the transfer direction and transfer distance from the current position (relative address) are specified for positioning.

- For (s1), specify the relative positioning address in user units. (The address must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the command speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the axis number from which pulses are output.
- For (d2), specify the bit device of the normal complete flag or abnormal end flag for the DDRVI instruction.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

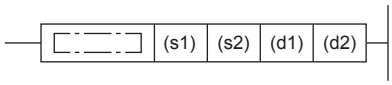
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

16-bit data absolute positioning

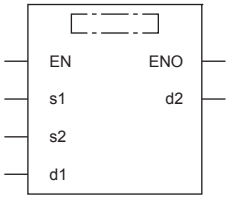
DRVA [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes one-speed positioning by absolute drive. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=DRVA(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Command speed	1 to 65535	16-bit unsigned binary	ANY16
(d1)	Output bit device number (Y) from which pulses are output	■FX5S/FX5UJ CPU module 0 to 2 ■FX5U/FX5UC CPU module 0 to 3	Bit	ANY_ELEMENTARY (BOOL)
(d2)	Bit device number from which the rotation direction is output	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d1)	○*1	—	—	—	—	—	—	—	—	—	—
(d2)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 Only Y can be used.
*2 When the output mode is CW/CCW, specify the CCW axis. When the output mode is PULSE/SIGN and using Y, only the SIGN output or general-purpose output of the self-axis can be specified.
*3 T, ST, and C cannot be used.

Processing details

This instruction executes one-speed positioning by absolute drive. Specify the positioning address in the absolute system, in which the transfer distance from the origin (absolute address) is specified for positioning.

- For (s1), specify the relative positioning address in user units. (The address must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the command speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the device from which pulses are output. Only the Y devices having positioning parameters can be specified.
- For (d2), specify the bit device from which the rotation direction signal is output. Only the device specified with the parameter or general-purpose outputs can be specified. When the output devices (Y) is executed by another function (PWM, positioning PULSE axis, or CW/CCW axis etc.), the device does not function and causes an error.

For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

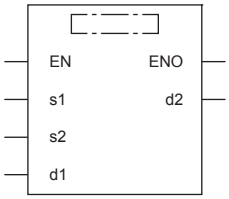
DRVA [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes one-speed positioning by absolute drive.

Ladder diagram	Structured text
	ENO:=DRVA(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address	-32768 to +32767	16-bit signed binary	ANY16
(s2)	Command speed	1 to 65535	16-bit unsigned binary	ANY16
(d1)	Axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
(d2)	Bit device number of the positioning complete flag or abnormal end flag	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d1)	—	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes one-speed positioning by absolute drive. Specify the positioning address in the absolute system, in which the transfer distance from the origin (absolute address) is specified for positioning.

- For (s1), specify the relative positioning address in user units. (The address must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the command speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the axis number from which pulses are output.
- For (d2), specify the bit device of the normal complete flag or abnormal end flag for the DRVA instruction.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

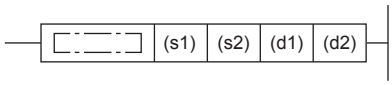
For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

32-bit data absolute positioning

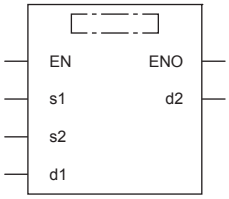
DDRVA [For the FX3 compatible operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes one-speed positioning by absolute drive. Only CPU module is supported.

Ladder diagram	Structured text
	ENO:=DDRVA(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Command speed	1 to 2147483647	32-bit signed binary	ANY32
(d1)	Output bit device number (Y) from which pulses are output	■FX5S/FX5UJ CPU module 0 to 2 ■FX5U/FX5UC CPU module 0 to 3	Bit	ANY_ELEMENTARY (BOOL)
(d2)	Bit device number from which the rotation direction is output	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d1)	○*1	—	—	—	—	—	—	—	—	—	—
(d2)	○*2	○*3	—	—	—	—	—	—	—	—	—

*1 Only Y can be used.
*2 When the output mode is CW/CCW, specify the CCW axis. When the output mode is PULSE/SIGN and using Y, only the SIGN output or general-purpose output of the self-axis can be specified.
*3 T, ST, and C cannot be used.

Processing details

This instruction executes one-speed positioning by absolute drive. Specify the positioning address in the absolute system, in which the transfer distance from the origin (absolute address) is specified for positioning.

- For (s1), specify the relative positioning address in user units. (The address must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the command speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the device from which pulses are output. Only the Y devices having positioning parameters can be specified.
- For (d2), specify the bit device from which the rotation direction signal is output. Only the device specified with the parameter or general-purpose outputs can be specified. When the output devices (Y) is executed by another function (PWM, positioning PULSE axis, or CW/CCW axis etc.), the device does not function and causes an error.

For details on the function, precautions, and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

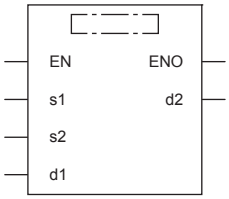
DDRVA [For the FX5 operand specification]

FX5S FX5UJ FX5U FX5UC

This instruction executes one-speed positioning by absolute drive.

Ladder diagram	Structured text
	ENO:=DDRVA(EN,s1,s2,d1,d2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Positioning address	-2147483648 to +2147483647	32-bit signed binary	ANY32
(s2)	Command speed	1 to 2147483647	32-bit signed binary	ANY32
(d1)	Axis number from which pulses are to be output	■FX5S/FX5UJ CPU module K1 to K3, K5 to K12 ■FX5U/FX5UC CPU module K1 to K12	16-bit unsigned binary	ANY_ELEMENTARY (WORD)*1
(d2)	Bit device number of the positioning complete flag or abnormal end flag	—	Bit	ANY_BOOL
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d1)	—	○	○	○	—	—	○	○	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—

*1 T, ST, and C cannot be used.

Processing details

This instruction executes one-speed positioning by absolute drive. Specify the positioning address in the absolute system, in which the transfer distance from the origin (absolute address) is specified for positioning.

- For (s1), specify the relative positioning address in user units. (The address must be within the range of -2147483648 to +2147483647 in the number of pulses.)
- For (s2), specify the command speed in user units. (The speed must be 200 kpps or lower in frequency.)
- For (d1), specify the axis number from which pulses are output.
- For (d2), specify the bit device of the normal complete flag or abnormal end flag for the DDRVA instruction.

For details on the function and error code, refer to MELSEC iQ-F FX5 User's Manual (Application).

Precautions

Two devices are occupied from the device specified in (d2). Make sure that these devices are not used in other controls.

For other precautions, refer to MELSEC iQ-F FX5 User's Manual (Application).

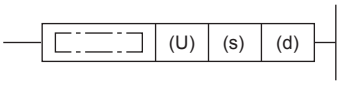
18.2 Positioning Module

Restoring the absolute position

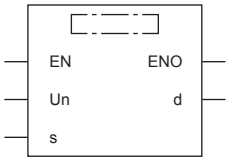
G.ABRST□

FX5S FX5UJ FX5U FX5UC

These instructions restore the absolute position of the specified axis.

Ladder diagram	Structured text
 ("G.ABRST1", "G.ABRST2" enters □.)	ENO:=G_ABRST1(EN,Un,s,d); ENO:=G_ABRST2(EN,Un,s,d);

FBD/LD



("G_ABRST1, G_ABRST2" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s)	Device where the control data is stored	☞ Page 1181 Control dataRefer to	Device name	ANY16_ARRAY*2 (Number of elements: 8)
(d)	Device which turns on when the execution of the instruction is completed and remains on for 1 scan. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.

*2 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.
Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

*2 T, ST, and C cannot be used.

■Control data

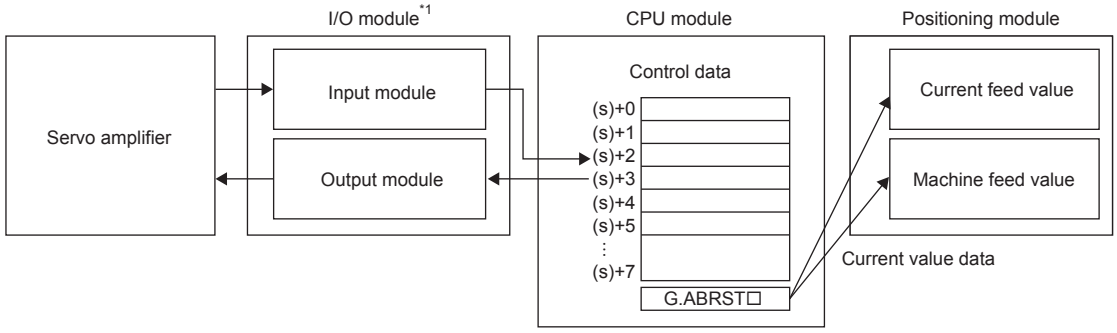
Device	Item	Description	Setting range	Set by
(s)+0	System area	—	—	—
(s)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System
(s)+2	Signal received from the servo amplifier	Write the signal status, below, input from the servo amplifier to the CPU module or I/O module. • b0: ABS data bit 0 • b1: ABS data bit 1 • b2: Send data ready flag	0, 1	User
(s)+3	Signal to be sent to the servo amplifier	Performs the operation with this instruction by using the "Signal received from the servo amplifier" in (s)+2. The ON/ OFF state of the following data, output to the servo amplifier, is stored. • b0: Servo ON • b1: ABS transfer mode • b2: ABS request flag	—	System
(s)+4	Status	Status of communication with the servo amplifier • 0: Communication complete (set by the user at start of communication) • Other than 0: Now communicating (stored by the system)	0	User/ System
(s)+5 to (s)+7	System area	—	—	—

Processing details

- The positioning data of the following axis is read from the servo amplifier supporting the absolute position and the value with the unit converted is stored in the "Current feed value" and "Machine feed value" areas in the positioning module. For the absolute position detection system, restore the absolute position once when it is powered on or the CPU module is reset.

Instruction symbol	Target axis
G.ABRST1	Axis 1
G.ABRST2	Axis 2

- The following figure shows how the G.ABRST□instruction operates.



*1 The CPU module input/output can also be used.

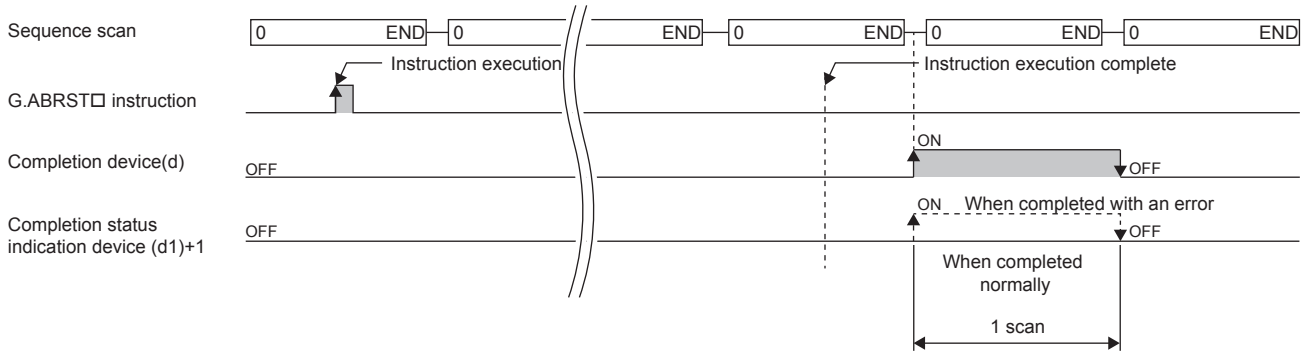
- For communication with the servo amplifier corresponding to the absolute position (data read/write), the input/output of CPU module or the I/O module is used. To use the G.ABRST□ instruction, prepare the following number of points to communicate with servo amplifier per axis.

  - Input: 3 points
  - Output: 3 points

- Whether the execution of the G.ABRST☐ instruction has been completed normally or with an error can be checked with the completion device (d) or completion status indication device (d)+1.

Device	Description
Completion device (d)	This device turns on during the END processing of the scan where the G.ABRST☐ instruction completed, and turns off during the next END processing.
Completion status indication device (d)+1	This device turns on or off depending on the completion status of the G.ABRST☐ instruction. When completed normally: Unchanged from off. When completed with an error: Turns on during the END processing of the scan where the G.ABRST☐ instruction completed, and turns off during the next END processing.

- The following figure shows the operation at completion of the G.ABRST☐ instruction.



- Completion of absolute position restoration can be checked with the (s)+4 status.
- The G.ABRST☐ instruction performs absolute position restoration by following procedure.

1. Output data from ((s)+3).

Servo on, ABS transfer mode, and ABS request flag are output by using the program.

2. Set data in ((s)+2).

ABS data bit 0/bit 1 and the send data ready flag are set using the program.

3. Execute the G.ABRST☐ instruction.

4. Check whether the value in ((s)+4) is 0 or not.

When the value is other than 0, the processing returns to step 1.

When the value is 0, the processing ends.

- For details of the function, refer to 📖MELSEC iQ-F FX5 Positioning Module User's Manual.

Precautions

- If the absolute position detection system has been constructed, absolute position restoration must be performed once after the power is turned on or reset. The servo does not turn on until the absolute position restoration completes.
- Absolute position restoration must be performed while the [Cd.190] PLC READY signal is off. While [Cd.190] PLC READY signal is ON, if the G.ABRST☐ instruction is executed, "Dedicated instruction error (Error code: 1870H)" occurs in the positioning module and absolute position restoration cannot be executed.
- The absolute position can be restored (the G.ABRST☐ instruction can be executed) while a servo amplifier is operable. Note that when the absolute position is restored, the Servo on signal may turn off (servo off) for the period of the scan time plus approximately 60 ms and the motor may move. To restore the absolute position during the servo off state, install an electromagnetic brake separately so that signals are output to the electromagnetic brake while the G.ABRST☐ instruction is being executed.
- The following instructions cannot be executed simultaneously to a single axis. For different axes, any of the following can be executed concurrently with a G.ABRSTo instruction.
  - Positioning start instruction (GP.PSTRT☐)
  - Absolute position restoration instruction (G.ABRST☐)
  - Teaching instruction (GP.TEACH☐)
- If this instruction is executed in an interrupt program with the priority 1, operation error (3580H) occurs. This instruction operates in an interrupt program with the priority 2 or 3.

Operation error

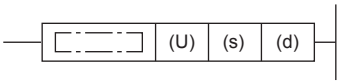
Error code ((s)+1)	Description
1860H	A value other than 0 was set in "Status" of (s)+4 (at start of communication with the servo amplifier).
1861H	"Status" of (s)+4 was changed during absolute position restoration (i.e. during communication with the servo amplifier).
1865H	An instruction was specified for an undefined axis.

Starting the positioning

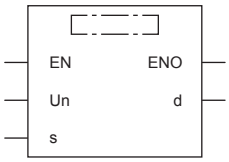
GP.PSTRT□

~~FX5S~~ FX5UJ FX5U FX5UC

This instruction starts positioning of the specified axis.

Ladder diagram	Structured text
 ("GP.PSTRT1, GP.PSTRT2" enters □.)	ENO:=GP_PSTRT1(EN,Un,s,d); ENO:=GP_PSTRT2(EN,Un,s,d);

FBD/LD



("GP_PSTRT1, GP_PSTRT2" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s)	Own station head device where control data is stored	☞ Page 1185 Control dataRefer to	Device name	ANY16_ARRAY*2 (Number of elements: 3)
(d)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.

*2 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.
Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	(U)
(U)	—	○	—	—	—	—	○	○	—	—	○
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

*2 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s)+0	System area	—	—	—
(s)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System
(s)+2	Start number	Specifies the number of the following data that is started with the GP.PSTRT☐ instruction. • Positioning data number: 1 to 600 • Block start: 7000 to 7004 • Machine OPR: 9001 • Fast OPR: 9002 • Present value change: 9003 • Multiple axes concurrent start: 9004	1 to 600 7000 to 7004 9001 to 9004	User

Processing details

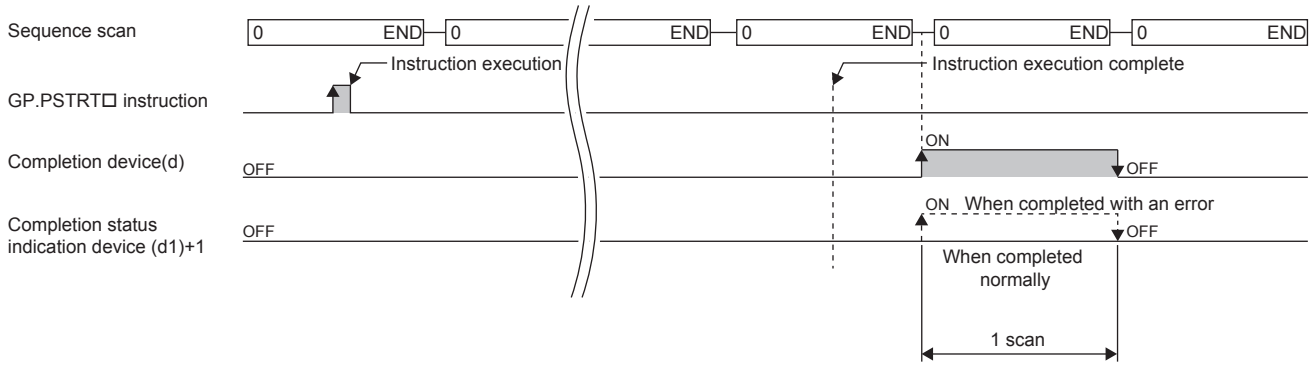
- The positioning is started for the specified axes below.

Instruction symbol	Target axis
GP.PSTRT1	Axis 1
GP.PSTRT2	Axis 2

- Block start, origin return start, present value change start, and multiple axes concurrent start are available by specifying one of 7000 to 7004 or 9001 to 9004 with "Start number" in (s)+2.
- Whether the GP.PSTRT☐ instruction has been completed normally or with an error can be checked with the completion device (d) or completion status indication device (d)+1.

Device	Description
Completion device (d)	This device turns on during the END processing of the scan where the GP.PSTRT☐ instruction completed, and turns off during the next END processing.
Completion status indication device (d)+1	This device turns on or off depending on the completion status of the GP.PSTRT☐ instruction. When completed normally: Unchanged from off. When completed with an error: Turns on during the END processing of the scan where the GP.PSTRT☐ instruction completed, and turns off during the next END processing.

- The following figure shows the operation at completion of the GP.PSTRT☐ instruction.



- For details of the function, refer to MELSEC iQ-F FX5 Positioning Module User's Manual.

Precautions

- If the positioning is started by using the GP.PSTRT☐ instruction, the [Cd.184] Positioning start signal will not turn on. In this case, although the start completion signals ([Md.31] Status: b14) turn on, the ON time is short; the program may fail to detect the ON state. For this reason, start completion cannot be checked using the start completion signals ([Md.31] Status: b14). Check the positioning control status with the start command of the GP.PSTRT☐ instruction or the [Md.141] BUSY signal.
- If the positioning is started by using the GP.PSTRT☐ instruction, and then the stop command is input before the positioning completes, the completion device (d) turns on for one scan and the execution of the GP.PSTRT☐ instruction completes.
- The following instructions cannot be executed simultaneously to a single axis. For different axes, any of the following can be executed concurrently with a GP.PSTRTo instruction.

  • Positioning start instruction (GP.PSTRT☐)
  • Absolute position restoration instruction (G.ABRST☐)
  • Teaching instruction (GP.TEACH☐)

- The GP.PSTRT☐ instruction can be executed when the READY signal ([Md.140] module status: b0) is ON. While the READY signal ([Md.140] module status: b0) is OFF, if the GP.PSTRT☐ instruction is executed, "PLC READY OFF start (Error code: 19A1H)" error occurs and positioning cannot be started. Turn the "[Cd.190] PLC READY signal" ON, and turn the READY signal ([Md.140] module status: b0) ON before executing the GP.PSTRT☐ instruction.
- If multiple axes concurrent start is executed using the GP.PSTRT☐ instruction, the completion device (d) turns on when the positioning completes for the axis where the GP.PSTRT☐ instruction has been executed (e.g. Axis 1 for GP.PSTRT1).
- If the GP.PSTRT☐ instruction is used, the starting time will delay 0 to 0.88ms relative to the [Cd.184] Positioning start signal.
- If this instruction is executed in an interrupt program with the priority 1, operation error (3580H) occurs. This instruction operates in an interrupt program with the priority 2 or 3.

Operation error

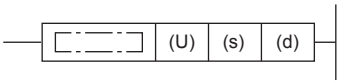
Error code ((s)+1)	Description
1862H	A value other than 1 to 600, 7000 to 7004, or 9001 to 9004 was set in "Starting number" of (s)+2.
1865H	An instruction was specified for an undefined axis.
19A1H	Start positioning when the "[Cd.190] PLC READY signal" is OFF.

Teaching

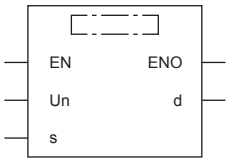
GP.TEACH□

~~FX5S~~ FX5UJ FX5U FX5UC

This instruction teaches the specified axis.

Ladder diagram	Structured text
 ("GP.TEACH1, GP.TEACH2" enters □.)	ENO:=GP_TEACH1(EN,Un,s,d); ENO:=GP_TEACH2(EN,Un,s,d);

FBD/LD



("GP_TEACH1, GP_TEACH2" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s)	Own station head device where control data is stored	☞ Page 1188 Control dataRefer to	Device name	ANY16_ARRAY*2 (Number of elements: 4)
(d)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.

*2 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.
Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	(U)
(U)	—	○	—	—	—	—	○	○	—	—	○
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

*2 T, ST, and C cannot be used.

■Control data

Device	Item	Description	Setting range	Set by
(s)+0	System area	—	—	—
(s)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System
(s)+2	Teaching data selection	Selects the address to which to write the current feed value (positioning or arc address). • 0: Writes the current feed value to the positioning address. • 1: Writes the current feed value to the arc address.	0, 1	User
(s)+3	Positioning data number	Sets the positioning data number at which to perform the teaching.	1 to 600	User

Processing details

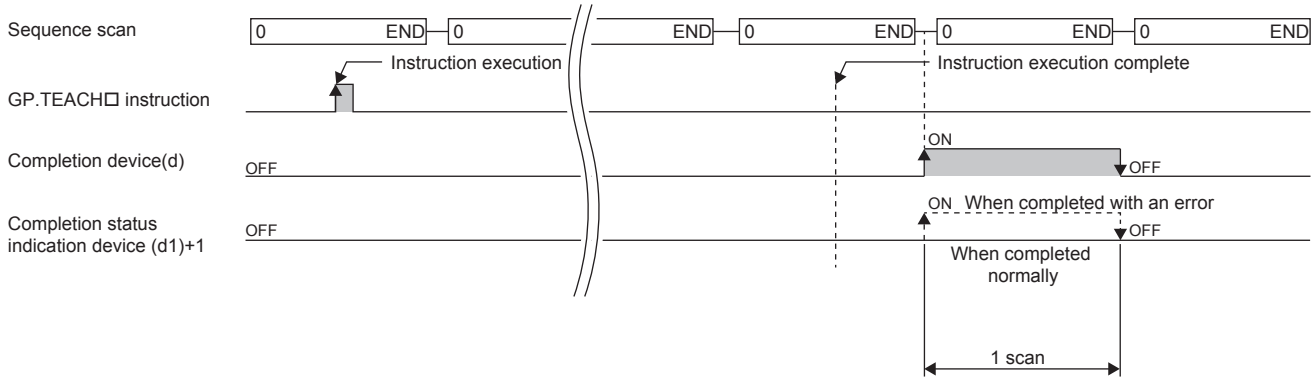
- The data of "[Md.20] Current feed value" of one of the following target axes is set in the positioning or arc address. The positioning data other than the positioning or arc address must be set using the engineering tool or program.

Instruction symbol	Target axis
GP.TEACH1	Axis 1
GP.TEACH2	Axis 2

- Positioning data No.1 to 600 can be taught.
- To move to an address (position) to be set as the positioning or arc address of the positioning data, JOG, inching, or manual pulse operation is used.
- Whether the GP.TEACH☐ instruction has been completed normally or with an error can be checked with the completion device (d) or completion status indication device (d)+1.

Device	Description
Completion device (d)	This device turns on during the END processing of the scan where the GP.TEACH☐ instruction completed, and turns off during the next END processing.
Completion status indication device (d)+1	This device turns on or off depending on the completion status of the GP.TEACH☐ instruction. When completed normally: Unchanged from off. When completed with an error: Turns on during the END processing of the scan where the GP.TEACH☐ instruction completed, and turns off during the next END processing.

- The following figure shows the operation at completion of the GP.TEACH☐ instruction.



- For details of the function, refer to MELSEC iQ-F FX5 Positioning Module User's Manual.

Precautions

- The following instructions cannot be executed simultaneously to a single axis. For different axes, any of the following can be executed concurrently with a GP.TEACHo instruction.
  - Positioning start instruction (GP.PSTRT□)
  - Absolute position restoration instruction (G.ABRST□)
  - Teaching instruction (GP.TEACH□)
- The GP.TEACH□ instruction is executed when "[Md.141] BUSY signal" is off. While "[Md.141] BUSY signal" is ON, if the GP.TEACH□ instruction is executed, "Teaching during BUSY (Warning code: 0903H)" warning occurs in the positioning module and teaching cannot be executed. Before executing the instruction, check that the "[Md.141] BUSY signal" of the corresponding axis is off.
- If this instruction is executed in an interrupt program with the priority 1, operation error (3580H) occurs. This instruction operates in an interrupt program with the priority 2 or 3.

Operation error

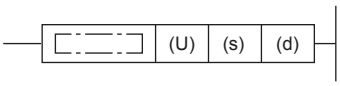
Error code ((s)+1)	Description
1863H	A value other than 0 or 1 was set in "Teaching data selection" of (s)+2.
1864H	A value other than 1 to 600 was set in "Positioning data number" of (s)+3.
1865H	An instruction was specified for an undefined axis.

Backing up module data (writing data to the flash ROM)

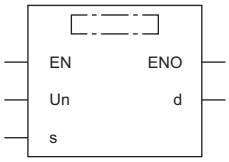
GP.PFWRT

~~FX5S~~ FX5UJ FX5U FX5UC

This instruction writes the module extension parameter (the positioning data and block start data) in the buffer memory to the module extension parameter file.

Ladder diagram	Structured text
	ENO:=GP_PFWRT(EN,Un,s,d);

FBD/LD



("GP_PFWRT" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s)	Own station head device where control data is stored	☞ Page 1191 Control dataRefer to	Device name	ANY16_ARRAY*2 (Number of elements: 2)
(d)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.

*2 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.
Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

*2 T, ST, and C cannot be used.

■Control data

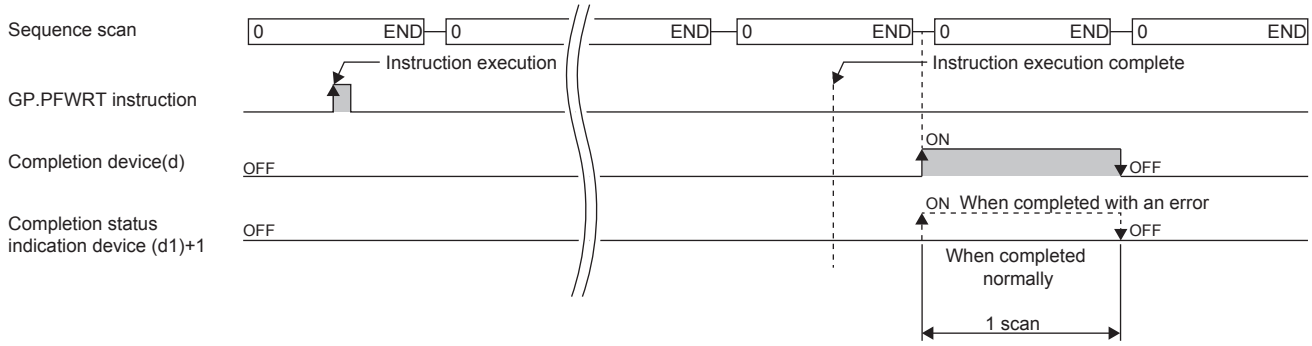
Device	Item	Description	Setting range	Set by
(s)+0	System area	—	—	—
(s)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System

Processing details

- The buffer memory module extension parameters are written to the module extension parameter file.
- Whether the GP.PFWRT instruction has been completed normally or with an error can be checked with the completion device (d) or completion status indication device (d)+1.

Device	Description
Completion device (d)	This device turns on during the END processing of the scan where the GP.PFWRT instruction completed, and turns off during the next END processing.
Completion status indication device (d)+1	This device turns on or off depending on the completion status of the GP.PFWRT instruction. When completed normally: Unchanged from off. When completed with an error: Turns on during the END processing of the scan where the GP.PFWRT instruction completed, and turns off during the next END processing.

- The following figure shows the operation at completion of the GP.PFWRT instruction.



- For details of the function, refer to MELSEC iQ-F FX5 Positioning Module User's Manual.

Precautions

- While the module extension parameters are being written by using the GP.PFWRT instruction, do not power off the system or reset the CPU module. If the power is turned off or the CPU module is reset while the module extension parameters are being written, data is not normally written and normal positioning start is disabled. If normal positioning start is disabled, restart the system as follows.

Method	Description
Restart using the engineering tool	Write the positioning and block start data to the positioning module from the "Write to programmable controller" of the engineering tool.
Restart using the program	After initializing parameters by using the GP.PINIT instruction, set the module extension parameters in the buffer memory of the positioning module. Thereafter, execute the GP.PFWRT instruction.

- Data write to the flash ROM of the positioning module can be repeated a maximum of one hundred thousand times. Any attempt to write data to the flash memory beyond this count results in failure, and the flash ROM write error (error code: 1931H) occurs.
- After the power is turned on or the CPU module is reset once, module backup (including the number of times initialization is executed) can be repeated a maximum of 25 times if the program is used. Any attempt to write data to the flash ROM memory beyond 25 times results in a flash ROM write count error at error code 1080H. If this error occurs, reset the positioning module error with "[Cd.5] axis error reset", by turning the power OFF to ON, or by resetting the CPU module.
- The GP.PFWRT instruction can be executed when the READY signal ([Md.140] module status: b0) is OFF. While the READY signal ([Md.140] module status: b0) is ON, if the GP.PFWRT instruction is executed, "Writing during PLC READY ON (Warning code: 0905H) warning occurs in the positioning module and the module backup cannot be executed. Turn the "[Cd.190] PLC READY signal" OFF, and turn the READY signal ([Md.140] module status: b0) OFF before executing the GP.PFWRT instruction.
- If this instruction is executed in an interrupt program with the priority 1, operation error (3580H) occurs. This instruction operates in an interrupt program with the priority 2 or 3.

Operation error

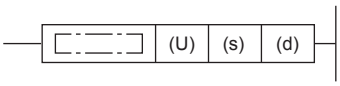
Error code ((s)+1)	Description
1080H	Flash ROM write count error
1931H	Flash ROM write error

Initializing the Module

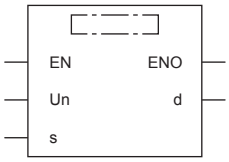
GP.PINIT

~~FX5S~~ FX5UJ FX5U FX5UC

This instruction resets the module parameter and module extension parameter (the positioning data and block start data) in the buffer memory, and the module extension parameter file setting values to the factory default settings (initial values).

Ladder diagram	Structured text
	ENO:=GP_PINIT(EN,Un,s,d);

FBD/LD



("GP_PINIT" enters □.)

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U)*1	Position number of the module connected	■FX5UJ CPU module 1H to 8H ■FX5U/FX5UC CPU module 1H to 10H	16-bit unsigned binary	ANY16
(s)	Own station head device where control data is stored	☞ Page 1194 Control dataRefer to	Device name	ANY16_ARRAY*2 (Number of elements: 2)
(d)	Own station device to be turned on for one scan when the instruction completes. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 In the case of the ST language and the FBD/LD language, U displays as Un.

*2 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.
Digit specified bit type label cannot be used.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U)	—	○	—	—	—	—	○	○	—	—	○
(s)	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	○*2	—	—	—	—	—	—	—	—	—

*1 S cannot be used.

*2 T, ST, and C cannot be used.

■Control data

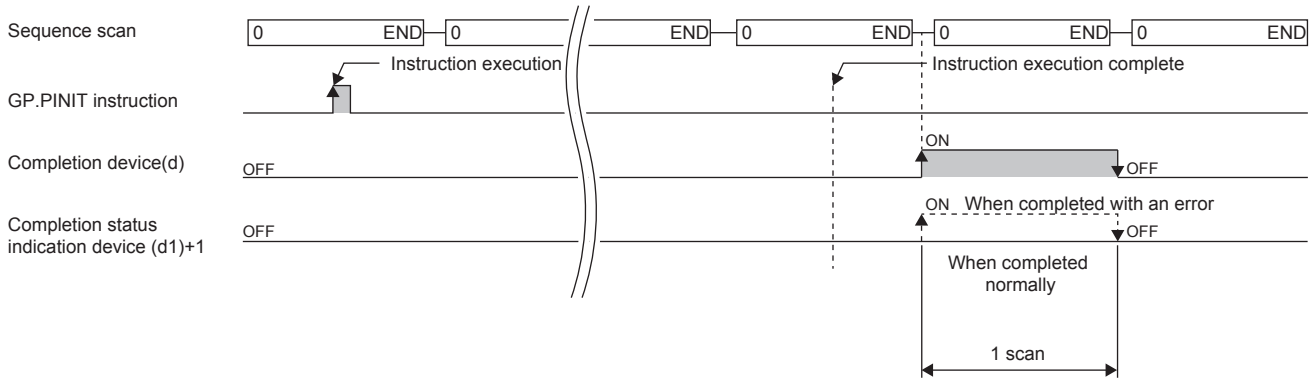
Device	Item	Description	Setting range	Set by
(s)+0	System area	—	—	—
(s)+1	Completion status	The instruction completion status is stored. • 0: Normal • Other than 0: Error (error code)	—	System

Processing details

- These instructions reset the module parameters and module extension parameters in the buffer memory of the positioning module, and the settings in the module extension parameter file to the factory default settings (initial values).
- The setting data initialized include the parameters, positioning data (No. 1 to 600), and block start data (No. 7000 to 7004).
- Whether the GP.PINIT instruction has been completed normally or with an error can be checked with the completion device (d) or completion status indication device (d)+1.

Device	Description
Completion device (d)	This device turns on during the END processing of the scan where the GP.PINIT instruction completed, and turns off during the next END processing.
Completion status indication device (d)+1	This device turns on or off depending on the completion status of the GP.PINIT instruction. When completed normally: Unchanged from off. When completed with an error: Turns on during the END processing of the scan where the GP.PINIT instruction completed, and turns off during the next END processing.

- The following figure shows the operation at completion of the GP.PINIT instruction.



- For details of the function, refer to MELSEC iQ-F FX5 Positioning Module User's Manual.

Precautions

- The GP.PINIT instruction can be executed when the READY signal ([Md.140] module status: b0) is OFF. While the READY signal ([Md.140] module status: b0) is ON, if the GP.PINIT instruction is executed, "Writing during PLC READY ON (Warning code: 0905H) warning occurs in the positioning module and the initializing the module cannot be executed. Turn the "[Cd.190] PLC READY signal" OFF, and turn the READY signal ([Md.140] module status: b0) OFF before executing the GP.PINIT instruction.
- Data write to the flash ROM of the positioning module can be repeated a maximum of one hundred thousand times. Any attempt to write data to the flash memory beyond this count results in failure, and the flash ROM write error (error code: 1931H) occurs.
- After the power is turned on or the CPU module is reset once, module backup (including the number of times initialization is executed) can be repeated a maximum of 25 times if the program is used. Any attempt to write data to the flash ROM memory beyond 25 times results in a flash ROM write count error at error code 1080H. If this error occurs, reset the positioning module error with "[Cd.5] axis error reset", by turning the power OFF to ON, or by resetting the CPU module.
- If this instruction is executed in an interrupt program with the priority 1, operation error (3580H) occurs. This instruction operates in an interrupt program with the priority 2 or 3.

Operation error

Error code ((s)+1)	Description
1080H	Flash ROM write count error
1931H	Flash ROM write error

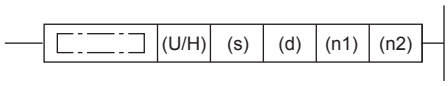
19 DIVIDED DATA READ/WRITE FROM/TO BFM INSTRUCTION

19.1 Divided BFM Read

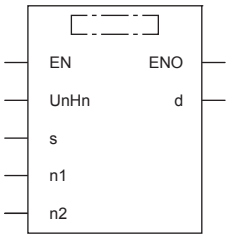
RBFM

FX5S FX5UJ FX5U FX5UC

This instruction reads data from continuous buffer memory areas in an FX3 intelligent function module

Ladder diagram	Structured text
	ENO:=RBFM(EN,UnHn,s,n1,n2,d);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U/H)*1	Module number	1H to 10H	16-bit unsigned binary	ANY16_U
(s)	Head buffer memory number	0 to 32767	16-bit unsigned binary	ANY16_U
(d)	Head device number storing data to be read from buffer memory	—	16-bit signed binary	ANY16
(n1)	Number of all buffer memory areas to be read	1 to 32768	16-bit unsigned binary	ANY16_U
(n2)	Not used	—	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

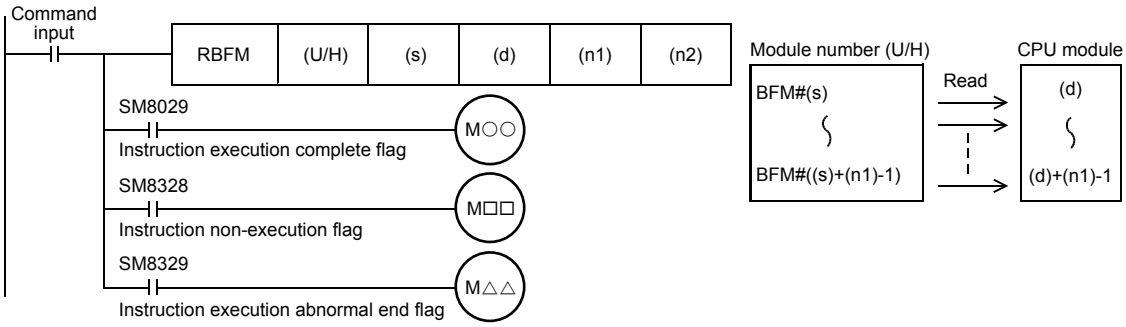
*1 In the case of the ST language and the FBD/LD language, U/H displays as UnHn.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U/H)	○	○	○	○	—	—	○	○	—	—	—
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	—	○	—	○	—	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- This instruction reads (n1) points of buffer memory starting from (s) inside the intelligent function module number (U/H) to (d) in the CPU module. When (n1) exceeds 64 points, it divides and reads by several scans. (64 points are read in one scan)



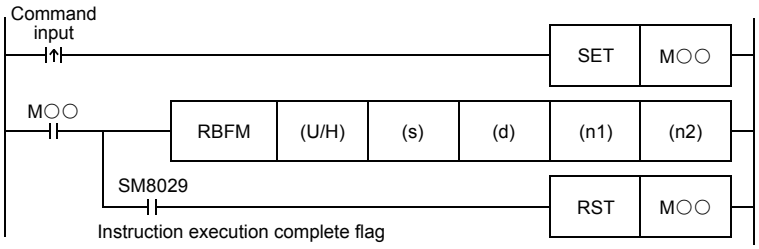
- When this instruction is finished normally, instruction execution complete flag (SM8029) turns on. When this instruction is finished abnormally, instruction execution abnormal end flag (SM8329) turns on.
- When this instruction or the WBFM instruction is executed in the same scan, instruction non-execution flag (SM8328) is set to on, and execution of such an instruction is paused. When execution of the other target instruction is complete, the paused instruction resumes.

■Related devices

Device	Name	Description
SM8029	Instruction execution complete	Turns ON when an instruction is finished normally.
SM8328	Instruction non-execution	Turns ON when the RBFM instruction or WBFM instruction in another step is executed for the same module number.
SM8329	Instruction execution abnormal end	Turns ON when an instruction is finished abnormally.

Precautions

- Do not stop the instruction while it is being executed. If driving is stopped, the buffer memory reading processing is suspended, but the data that is already read is stored in (d) onwards. Stop the instruction after execution completes as in the following program.



- When indexing is executed, the contents of index registers at the beginning of execution are used. Even if the contents of index registers are changed after the instruction, such changes do not affect the process of the instruction.
- The contents of (n1) points starting from (d) update (change) every scan while this instruction is executed. Use the data after the instruction is completed.
- Do not update (change) the contents of (n1) buffer memory areas starting from the buffer memory (s) while this instruction is executed. If the contents are updated, the intended data may not be read.
- This instruction cannot be used in FX5 intelligent function modules.
- This instruction cannot be used while a interrupt routine program is being executed.

Operation error

Error code (SD0/SD8067)	Description
2441H	Communication procedure with module is not completed correctly when this instruction is executed.
2801H	Module with the module number specified by (U/H) does not exist.
2823H	The number of transfer points specified by (n1) and the buffer memory number specified by (s) is beyond the buffer memory area range.
2820H	The number of transfer points specified by (n1) and the device number specified by (d) is beyond the specified device range.
3580H	An instruction that cannot be used in an interrupt program is used.

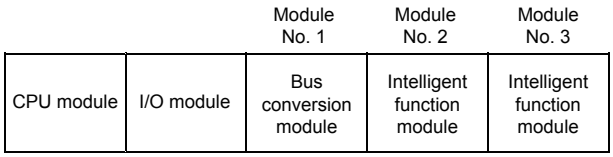
Common items between RBFM instruction and WBFM instruction

■Specification of module number of FX3 intelligent function module and buffer memory

For FX3 intelligent function module connection method, number of connectable FX3 intelligent function modules and handling of I/O numbers, refer to manuals of the CPU module and FX3 intelligent function modules.

• Module number of FX3 intelligent function module

Use the module number to specify for which equipment the RBFM/WBFM instruction is used. (Setting range: 1H to 10H (K1 to K16))



A module number is automatically assigned to each intelligent function module connected to the CPU module. The module number is assigned as No.1 → No.2 → No.3… starting from the equipment nearest the CPU module.

• Buffer memory number

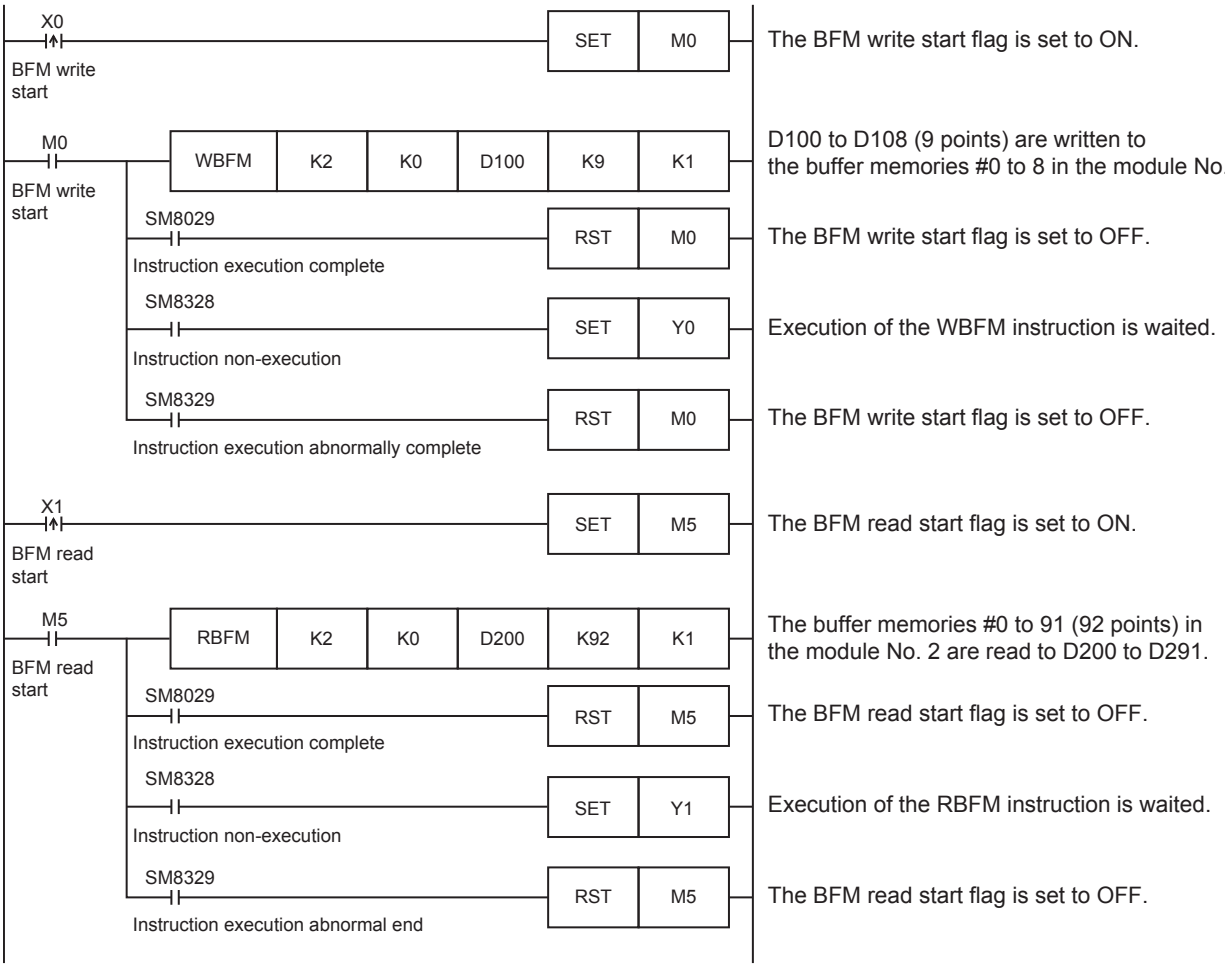
The intelligent function module incorporates a RAM memory. The RAM memory is called buffer memory. Buffer memory numbers range from #0 to #32767 and their contents vary depending on the function of the extension equipment. (Setting range: K0 to K32767)

For the contents of buffer memory areas, refer to manuals of intelligent function modules.

Program example

In the example shown below, data is read from and written to the buffer memories in the module No. 2 as follows:

- When X0 is set to ON, data stored in D100 to D108 (9 points) is written to the buffer memories #0 to #8 in the FX3 intelligent module whose module number is No. 2.
- When X1 is set to ON, the buffer memories #0 to 91 (92 points) in the FX3 intelligent module whose module number is No. 2 are written to D200 to D291.

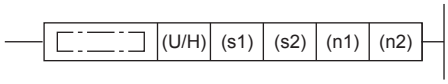


19.2 Divided BFM Write

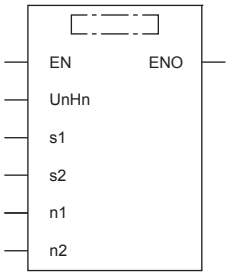
WBFM

FX5S FX5UJ FX5U FX5UC

This instruction writes data to continuous buffer memory areas in an FX3 intelligent function module.

Ladder diagram	Structured text
	ENO:=WBFM(EN,UnHn,s1,s2,n1,n2);

FBD/LD



Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(U/H)*1	Module number	1H to 10H	16-bit unsigned binary	ANY16_U
(s1)	Head buffer memory number	0 to 32767	16-bit unsigned binary	ANY16_U
(s2)	Head device number storing data to be written to buffer memory	—	16-bit signed binary	ANY16
(n1)	Number of all buffer memory areas to be written	1 to 32768	16-bit unsigned binary	ANY16_U
(n2)	Not used	—	16-bit unsigned binary	ANY16_U
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

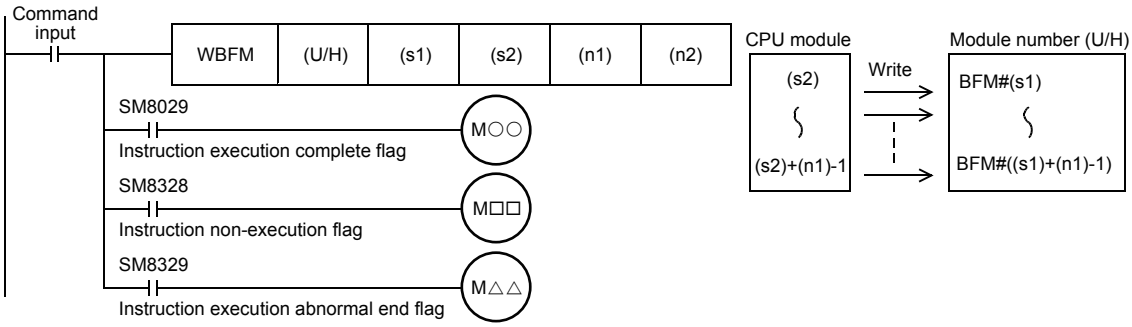
*1 In the case of the ST language and the FBD/LD language, U/H displays as UnHn.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(U/H)	○	○	○	○	—	—	○	○	—	—	—
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	—	○	—	○	—	—	○	—	—	—	—
(n1)	○	○	○	○	—	—	○	○	—	—	—
(n2)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- This instruction writes (n1) points of buffer memory starting from (s1) inside the intelligent function module number (U/H) to (s2) in the CPU module. When (n1) exceeds 64 points, it divides and writes by several scans. (64 points are read in one scan)



- When this instruction is finished normally, instruction execution complete flag (SM8029) turns on. When this instruction is finished abnormally, instruction execution abnormal end flag (SM8329) turns on.
- When this instruction or the RBFM instruction is executed in the same scan, instruction non-execution flag (SM8328) is set to on, and execution of such an instruction is paused. When execution of the other target instruction is complete, the paused instruction resumes.

■Related devices

Device	Name	Description
SM8029	Instruction execution complete	Turns ON when an instruction is finished normally.
SM8328	Instruction non-execution	Turns ON when the RBFM instruction or WBFM instruction in another step is executed for the same module number.
SM8329	Instruction execution abnormal end	Turns ON when an instruction is finished abnormally.

Precautions

- Do not stop the instruction while it is being executed. If driving is stopped, the buffer memory write processing is suspended, but the data that is already written is stored in (s1) onwards.
- When indexing is executed, the contents of index registers at the beginning of execution are used. Even if the contents of index registers are changed after the instruction, such changes do not affect the process of the instruction.
- Do not update (change) the contents of (n1) points starting from (s2) while this instruction is executed. If the contents are updated, the intended data may not be written to the buffer memory areas.
- This instruction cannot be used in FX5 intelligent function modules.
- This instruction cannot be used while a interrupt routine program is being executed.

Operation error

Error code (SD0/SD8067)	Description
2441H	Communication procedure with module is not completed correctly when this instruction is executed.
2801H	Module with the module number specified by (U/H) does not exist, or the specified module is not supported.
2823H	The number of transfer points specified by (n1) and the buffer memory number specified by (s1) is beyond the buffer memory range.
2820H	The number of transfer points specified by (n1) and the device number specified by (s2) is beyond the specified device range.
3580H	An instruction that cannot be used in an interrupt program is used.

MEMO

PART 5 STANDARD FUNCTIONS

This part consists of the following chapters.

20 TYPE CONVERSION FUNCTIONS

21 SINGLE NUMBER VARIABLE FUNCTIONS

22 ARITHMETIC OPERATION FUNCTIONS

23 BIT SHIFT FUNCTIONS

24 STANDARD BITWISE BOOLEAN FUNCTIONS

25 SELECTION FUNCTIONS

26 COMPARISON FUNCTIONS

27 CHARACTER STRING FUNCTIONS

28 TIME DATA FUNCTIONS

20 TYPE CONVERSION FUNCTIONS

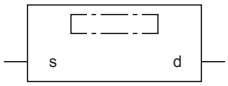
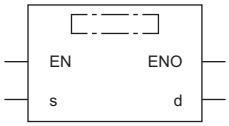
20.1 Converting BOOL to WORD

BOOL_TO_WORD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert BOOL type data to WORD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=BOOL_TO_WORD(s);
		[With EN/ENO]
		d:=BOOL_TO_WORD_E(EN,ENO,s);

Setting data

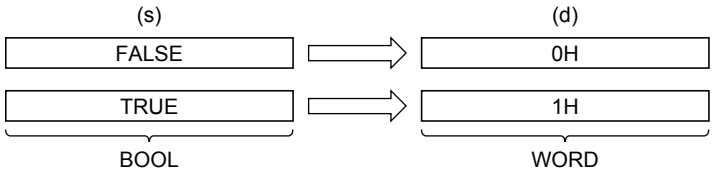
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	BOOL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(BOOL_TO_WORD(_E))	Output	Output variable	WORD

Processing details

■Operation processing

- These functions convert the BOOL type data input to (s) to WORD type data and output from (d).
- When the input value is "FALSE", these functions output 0H as the WORD type data value.
- When the input value is "TRUE", these functions output 1H as the WORD type data value.



- A value input to (s) is the BOOL type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

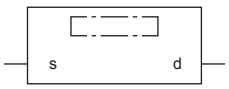
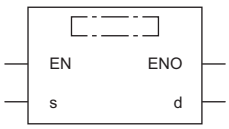
20.2 Converting BOOL to DWORD

BOOL_TO_DWORD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert BOOL type data to DWORD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=BOOL_TO_DWORD(s);
		[With EN/ENO]
		d:=BOOL_TO_DWORD_E(EN,ENO,s);

Setting data

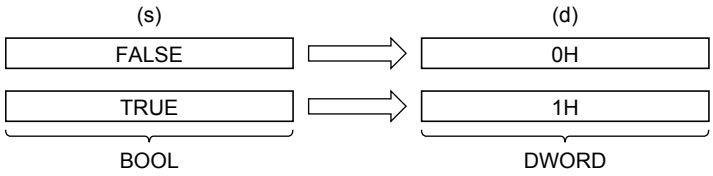
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	BOOL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(BOOL_TO_DWORD(_E))	Output	Output variable	DWORD

Processing details

■Operation processing

- These functions convert the BOOL type data input to (s) to DWORD type data and output from (d).
- When the input value is "FALSE", these functions output 0H as the DWORD type data value.
- When the input value is "TRUE", these functions output 1H as the DWORD type data value.



- A value input to (s) is the BOOL type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

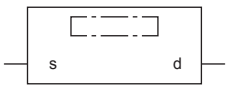
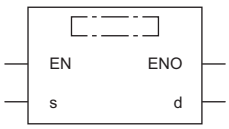
20.3 Converting BOOL to INT

BOOL_TO_INT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert BOOL type data to INT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=BOOL_TO_INT(s);
		[With EN/ENO]
		d:=BOOL_TO_INT_E(EN,ENO,s);

Setting data

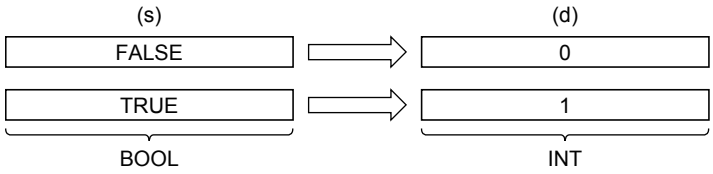
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	BOOL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(BOOL_TO_INT(_E))	Output	Output variable	INT

Processing details

■Operation processing

- These functions convert the BOOL type data input to (s) to INT type data and output from (d).
- When the input value is "FALSE", these functions output 0 as the INT type data value.
- When the input value is "TRUE", these functions output 1 as the INT type data value.



- A value input to (s) is the BOOL type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

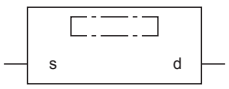
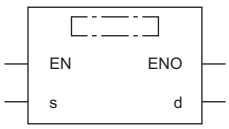
20.4 Converting BOOL to DINT

BOOL_TO_DINT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert BOOL type data to DINT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=BOOL_TO_DINT(s);
		[With EN/ENO]
		d:=BOOL_TO_DINT_E(EN,ENO,s);

Setting data

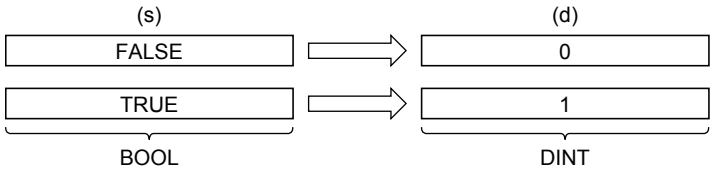
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	BOOL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(BOOL_TO_DINT(_E))	Output	Output variable	DINT

Processing details

■Operation processing

- These functions convert the BOOL type data input to (s) to DINT type data and output from (d).
- When the input value is "FALSE", these functions output 0 as the DINT type data value.
- When the input value is "TRUE", these functions output 1 as the DINT type data value.



- A value input to (s) is the BOOL type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

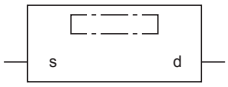
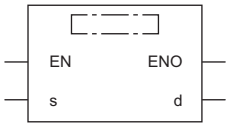
20.5 Converting BOOL to TIME

BOOL_TO_TIME(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert BOOL type data to TIME type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=BOOL_TO_TIME(s);
		[With EN/ENO]
		d:=BOOL_TO_TIME_E(EN,ENO,s);

Setting data

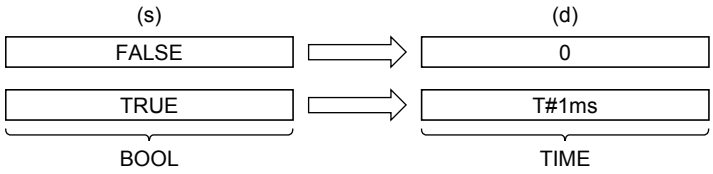
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	BOOL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(BOOL_TO_TIME(_E))	Output	Output variable	TIME

Processing details

■Operation processing

- These functions convert the BOOL type data input to (s) to TIME type data and output from (d).
- When the input value is "FALSE", these functions output 0 as the TIME type data value.
- When the input value is "TRUE", these functions output 1 as the TIME type data value.



- A value input to (s) is the BOOL type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

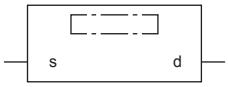
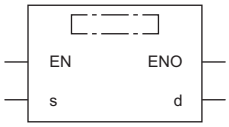
20.6 Converting BOOL to STRING

BOOL_TO_STRING(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert BOOL type data to STRING type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=BOOL_TO_STRING(s);
		[With EN/ENO]
		d:=BOOL_TO_STRING_E(EN,ENO,s);

Setting data

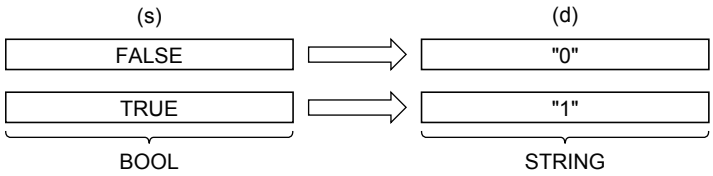
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	BOOL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(BOOL_TO_STRING(_E))	Output	Output variable	STRING

Processing details

■Operation processing

- These functions convert the BOOL type data input to (s) to STRING type data and output from (d).
- When the input value is "FALSE", these functions output 0 as the STRING type data value.
- When the input value is "TRUE", these functions output 1 as the STRING type data value.



- A value input to (s) is the BOOL type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

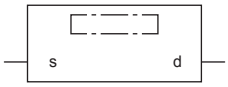
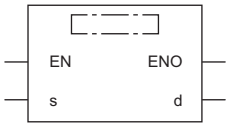
20.7 Converting WORD to BOOL

WORD_TO_BOOL(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert WORD type data to BOOL type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=WORD_TO_BOOL(s);
		[With EN/ENO]
		d:=WORD_TO_BOOL_E(EN,ENO,s);

Setting data

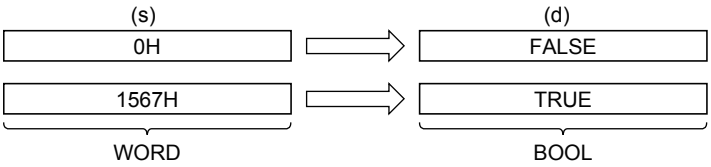
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	WORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(WORD_TO_BOOL(_E))	Output	Output variable	BOOL

Processing details

■Operation processing

- These functions convert the WORD type data input to (s) to BOOL type data and output from (d).
- When the input value is 0H, these functions output "FALSE".
- When the input value is any value other than 0H, these functions output "TRUE".



- A value input to (s) is the WORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

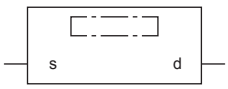
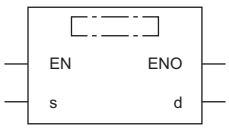
20.8 Converting WORD to DWORD

WORD_TO_DWORD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert WORD type data to DWORD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=WORD_TO_DWORD(s);
		[With EN/ENO]
		d:=WORD_TO_DWORD_E(EN,ENO,s);

Setting data

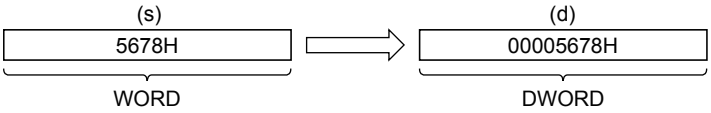
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	WORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(WORD_TO_DWORD(_E))	Output	Output variable	DWORD

Processing details

■Operation processing

- These functions convert the WORD type data input to (s) to DWORD type data and output from (d).
- Each of high-order 16 bits becomes "0" after data conversion.



- A value input to (s) is the WORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

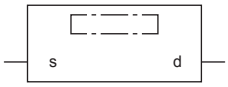
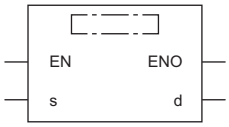
20.9 Converting WORD to INT

WORD_TO_INT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert WORD type data to INT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=WORD_TO_INT(s);
		[With EN/ENO]
		d:=WORD_TO_INT_E(EN,ENO,s);

Setting data

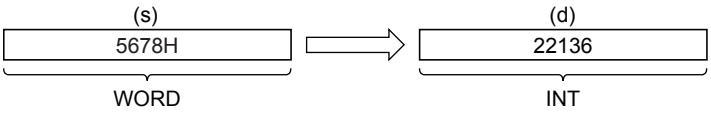
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	WORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(WORD_TO_INT(_E))	Output	Output variable	INT

Processing details

■Operation processing

- These functions convert the WORD type data input to (s) to INT type data and output from (d).



- A value input to (s) is the WORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

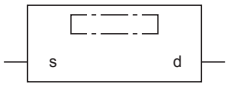
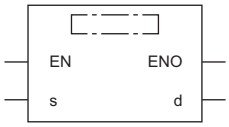
20.10 Converting WORD to DINT

WORD_TO_DINT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert WORD type data to DINT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=WORD_TO_DINT(s);
		[With EN/ENO]
		d:=WORD_TO_DINT_E(EN,ENO,s);

Setting data

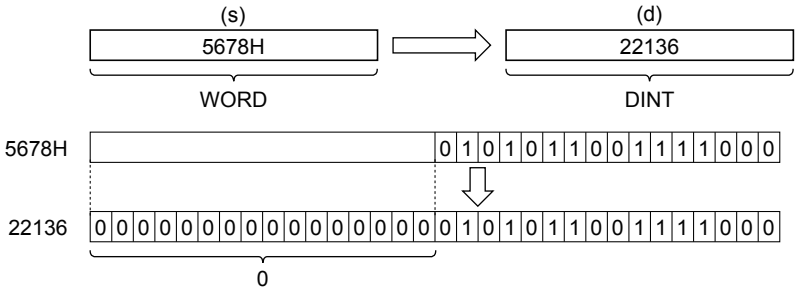
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	WORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(WORD_TO_DINT(_E))	Output	Output variable	DINT

Processing details

■Operation processing

- These functions convert the WORD type data input to (s) to DINT type data and output from (d).
- Each of high-order 16 bits becomes "0" after data conversion.



- A value input to (s) is the WORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

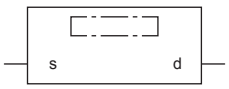
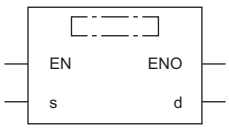
20.11 Converting WORD to TIME

WORD_TO_TIME(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert WORD type data to TIME type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=WORD_TO_TIME(s);
		[With EN/ENO]
		d:=WORD_TO_TIME_E(EN,ENO,s);

Setting data

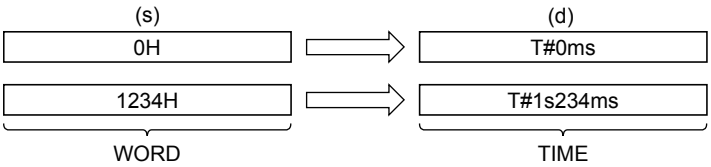
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	WORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(WORD_TO_TIME(_E))	Output	Output variable	TIME

Processing details

■Operation processing

- These functions convert the WORD type data input to (s) to TIME type data and output from (d).



- A value input to (s) is the WORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

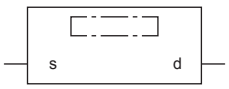
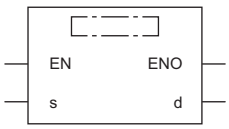
20.12 Converting DWORD to BOOL

DWORD_TO_BOOL(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DWORD type data to BOOL type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DWORD_TO_BOOL(s);
		[With EN/ENO]
		d:=DWORD_TO_BOOL_E(EN,ENO,s);

Setting data

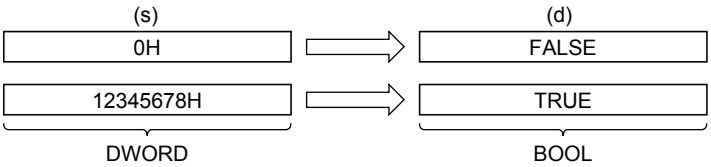
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DWORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DWORD_TO_BOOL(_E))	Output	Output variable	BOOL

Processing details

■Operation processing

- These functions convert the DWORD type data input to (s) to BOOL type data and output from (d).
- When the input value is 0H, these functions output "FALSE".
- When the input value is any value other than 0H, these functions output "TRUE".



- A value input to (s) is the DWORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

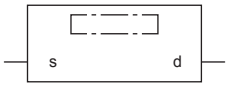
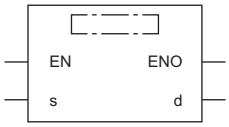
20.13 Converting DWORD to WORD

DWORD_TO_WORD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DWORD type data to WORD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DWORD_TO_WORD(s);
		[With EN/ENO]
		d:=DWORD_TO_WORD_E(EN,ENO,s);

Setting data

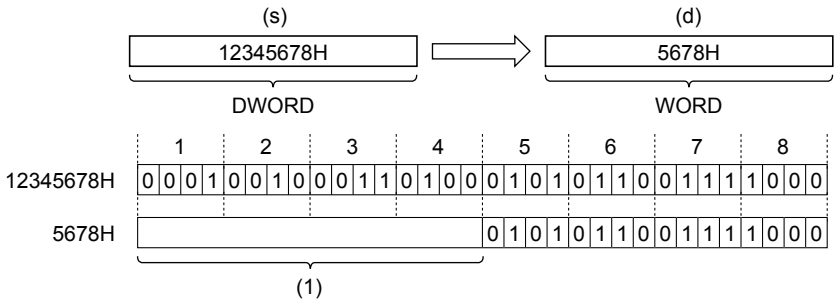
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DWORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DWORD_TO_WORD(_E))	Output	Output variable	WORD

Processing details

■Operation processing

- These functions convert the DWORD type data input to (s) to WORD type data and output from (d).
- The information stored in high-order 16 bits of an input value is discarded. (Refer to (1) in the figure below.)



- A value input to (s) is the DWORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Point

When DWORD_TO_WORD(_E) is executed, the information stored in high-order 16 bits of the DWORD type data value input from (s) is discarded.

Operation error

There is no operation error.

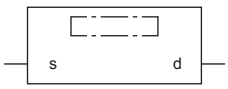
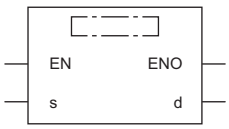
20.14 Converting DWORD to INT

DWORD_TO_INT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DWORD type data to INT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DWORD_TO_INT(s);
		[With EN/ENO]
		d:=DWORD_TO_INT_E(EN,ENO,s);

Setting data

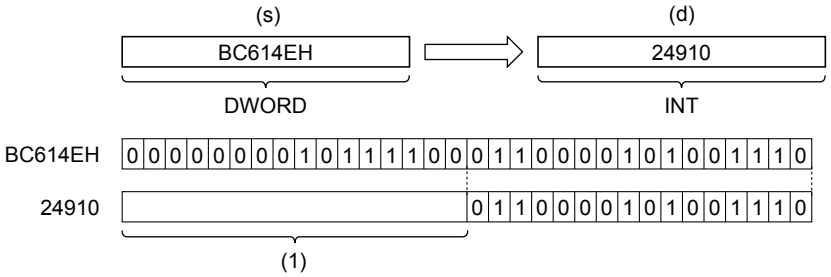
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DWORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DWORD_TO_INT(_E))	Output	Output variable	INT

Processing details

■Operation processing

- These functions convert the DWORD type data input to (s) to INT type data and output from (d).
- The information stored in high-order 16 bits of an input value is discarded. (Refer to (1) in the figure below.)



- A value input to (s) is the DWORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Point

When DWORD_TO_INT(_E) is executed, the information stored in high-order 16 bits of the DWORD type data value input from (s) is discarded.

Operation error

There is no operation error.

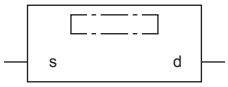
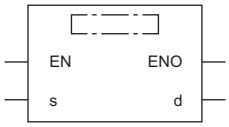
20.15 Converting DWORD to DINT

DWORD_TO_DINT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DWORD type data to DINT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DWORD_TO_DINT(s);
		[With EN/ENO]
		d:=DWORD_TO_DINT_E(EN,ENO,s);

Setting data

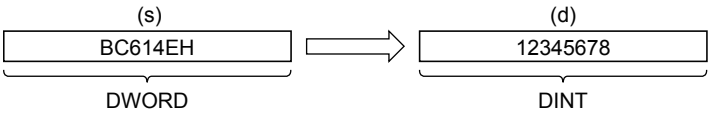
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DWORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DWORD_TO_DINT(_E))	Output	Output variable	DINT

Processing details

■Operation processing

- These functions convert the DWORD type data input to (s) to DINT type data and output from (d).



- A value input to (s) is the DWORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

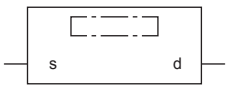
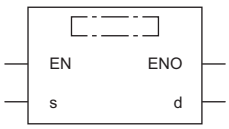
20.16 Converting DWORD to TIME

DWORD_TO_TIME(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DWORD type data to TIME type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DWORD_TO_TIME(s);
		[With EN/ENO]
		d:=DWORD_TO_TIME_E(EN,ENO,s);

Setting data

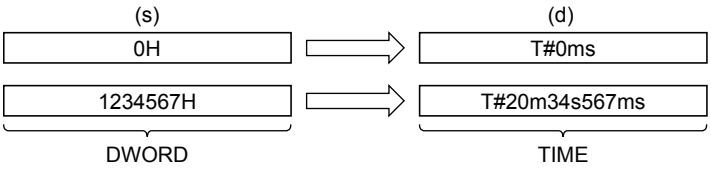
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DWORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DWORD_TO_TIME(_E))	Output	Output variable	TIME

Processing details

■Operation processing

- These functions convert the DWORD type data input to (s) to TIME type data and output from (d).



- A value input to (s) is the DWORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

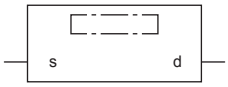
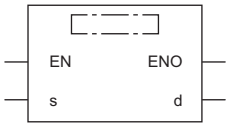
20.17 Converting INT to BOOL

INT_TO_BOOL(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert INT type data to BOOL type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=INT_TO_BOOL(s);
		[With EN/ENO]
		d:=INT_TO_BOOL_E(EN,ENO,s);

Setting data

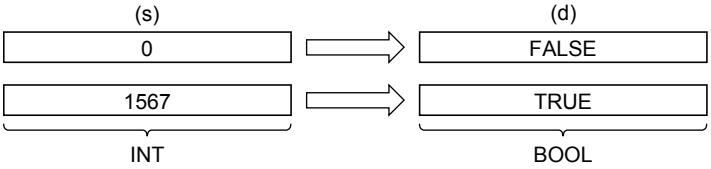
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(INT_TO_BOOL(_E))	Output	Output variable	BOOL

Processing details

■Operation processing

- These functions convert the INT type data input to (s) to BOOL type data and output from (d).
- When the input value is 0, these functions output "FALSE".
- When the input value is any value other than 0, these functions output "TRUE".



- A value input to (s) is the INT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

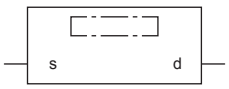
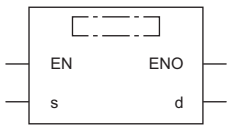
20.18 Converting INT to WORD

INT_TO_WORD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert INT type data to WORD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=INT_TO_WORD(s);
		[With EN/ENO]
		d:=INT_TO_WORD_E(EN,ENO,s);

Setting data

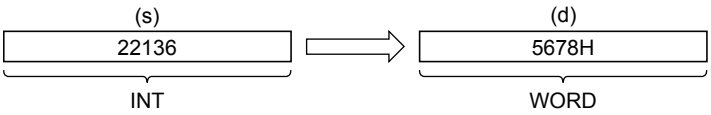
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(INT_TO_WORD(_E))	Output	Output variable	WORD

Processing details

■Operation processing

- These functions convert the INT type data input to (s) to WORD type data and output from (d).



- A value input to (s) is the INT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

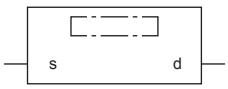
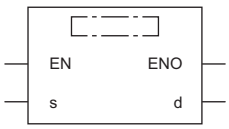
20.19 Converting INT to DWORD

INT_TO_DWORD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert INT type data to DWORD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=INT_TO_DWORD(s);
		[With EN/ENO]
		d:=INT_TO_DWORD_E(EN,ENO,s);

Setting data

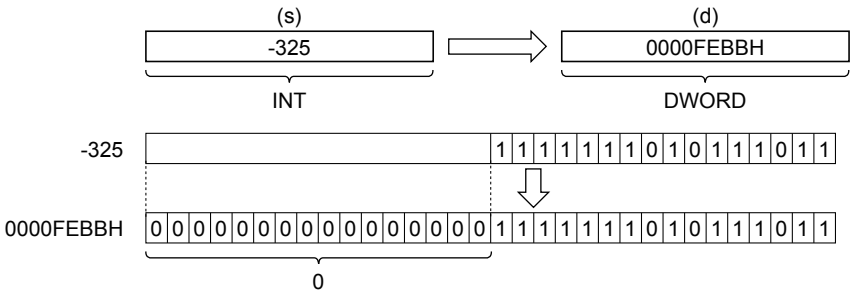
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(INT_TO_DWORD(_E))	Output	Output variable	DWORD

Processing details

■Operation processing

- These functions convert the INT type data input to (s) to DWORD type data and output from (d).
- Each of high-order 16 bits becomes "0" after data conversion.



- A value input to (s) is the INT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

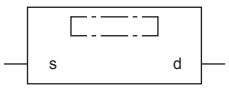
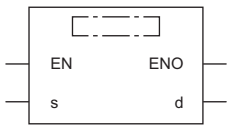
20.20 Converting INT to DINT

INT_TO_DINT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert INT type data to DINT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=INT_TO_DINT(s);
		[With EN/ENO]
		d:=INT_TO_DINT_E(EN,ENO,s);

Setting data

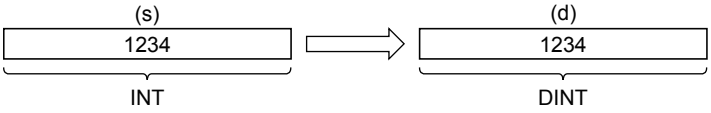
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(INT_TO_DINT(_E))	Output	Output variable	DINT

Processing details

■Operation processing

- These functions convert the INT type data input to (s) to DINT type data and output from (d).



- A value input to (s) is the INT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

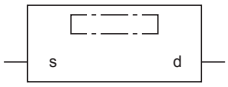
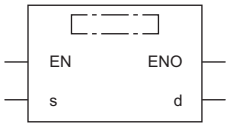
20.21 Converting INT to BCD

INT_TO_BCD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert INT type data to BCD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=INT_TO_BCD(s);
		[With EN/ENO]
		d:=INT_TO_BCD_E(EN,ENO,s);

Setting data

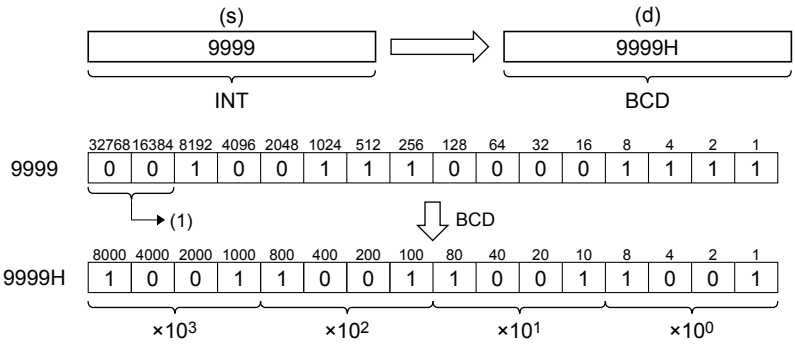
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(INT_TO_BCD(_E))	Output	Output variable	WORD

Processing details

■Operation processing

- These functions convert the INT type data input to (s) to BCD type data and output from (d).



(1): Make sure to set them to "0".

- A value input to (s) is the INT type data value and within the range from 0 to 9999.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3401H	Data in the device specified by (s) is out of the valid range (0 to 9999).

20.22 Converting INT to REAL

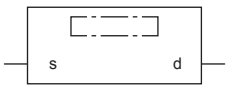
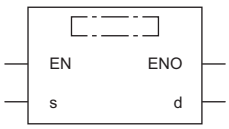
INT_TO_REAL(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert INT type data to REAL type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]	[With EN/ENO]	[Without EN/ENO]
		d:=INT_TO_REAL(s);
		[With EN/ENO]
		d:=INT_TO_REAL_E(EN,ENO,s);

Setting data

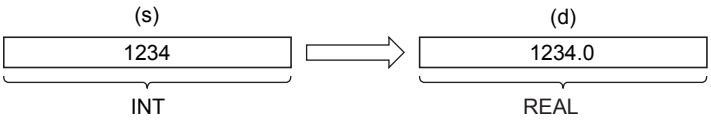
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(INT_TO_REAL(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions convert the INT type data input to (s) to REAL type data and output from (d).



- A value input to (s) is the INT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

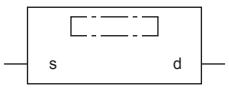
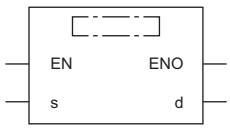
20.23 Converting INT to TIME

INT_TO_TIME(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert INT type data to TIME type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=INT_TO_TIME(s);
		[With EN/ENO]
		d:=INT_TO_TIME_E(EN,ENO,s);

Setting data

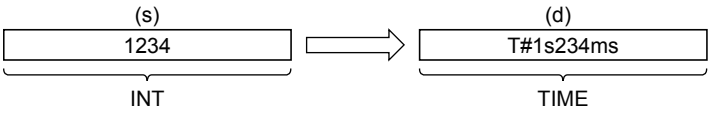
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(INT_TO_TIME(_E))	Output	Output variable	TIME

Processing details

■Operation processing

- These functions convert the INT type data input to (s) to TIME type data and output from (d).



- A value input to (s) is the INT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

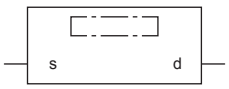
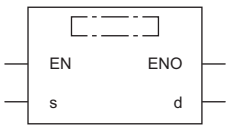
20.24 Converting INT to STRING

INT_TO_STRING(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert INT type data to STRING type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=INT_TO_STRING(s);
		[With EN/ENO]
		d:=INT_TO_STRING_E(EN,ENO,s);

Setting data

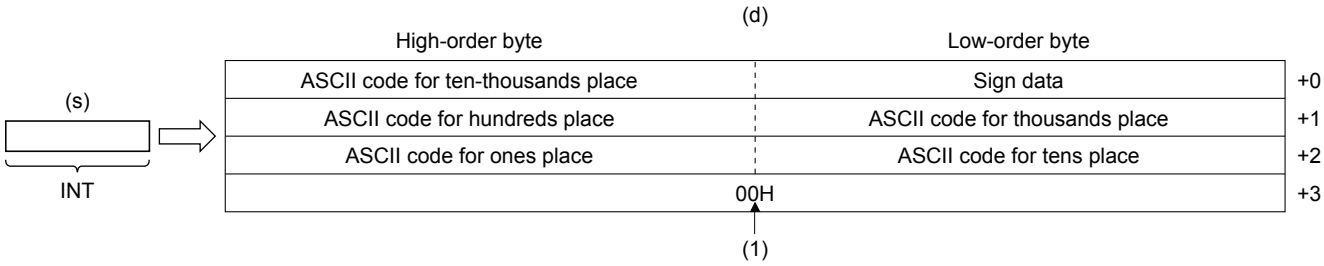
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(INT_TO_STRING(_E))	Output	Output variable	STRING(6)

Processing details

■Operation processing

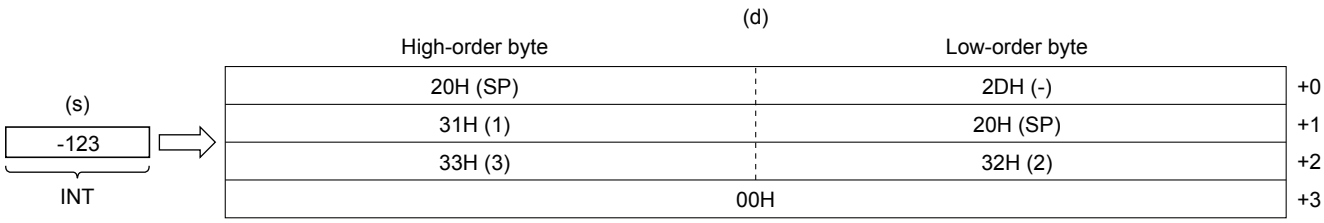
- These functions convert the INT type data input to (s) to STRING type data and output from (d).



(1): "00H" is stored when "SM701" (output character number selector) is off.

- A value input to (s) is the INT type data value.
- In "Sign data", 20H (space) is stored when the input value is positive, and 2DH (-) is stored when the input value is negative.
- 20H (space) is stored in high-order digits when the number of significant figures is small.

Ex.
When "-123" is input



- 00H is stored at the end (4th word) of the character string when SM701 (output character number selector signal) is off.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s) and later, "0000H" does not exist.
3405H	The character string specified by (s) has more than 16383 characters.
3406H	The whole specified character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.

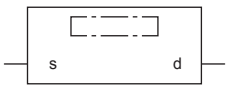
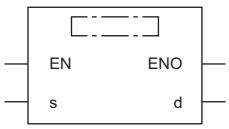
20.25 Converting DINT to BOOL

DINT_TO_BOOL(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DINT type data to BOOL type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DINT_TO_BOOL(s);
		[With EN/ENO]
		d:=DINT_TO_BOOL_E(EN,ENO,s);

Setting data

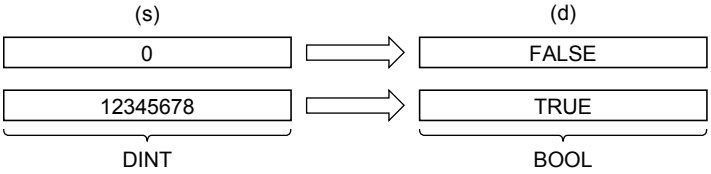
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DINT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DINT_TO_BOOL(_E))	Output	Output variable	BOOL

Processing details

■Operation processing

- These functions convert the DINT type data input to (s) to BOOL type data and output from (d).
- When the input value is 0, these functions output "FALSE".
- When the input value is any value other than 0, these functions output "TRUE".



- A value input to (s) is the DINT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

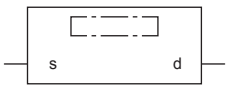
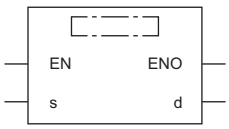
20.26 Converting DINT to WORD

DINT_TO_WORD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DINT type data to WORD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DINT_TO_WORD(s);
		[With EN/ENO]
		d:=DINT_TO_WORD_E(EN,ENO,s);

Setting data

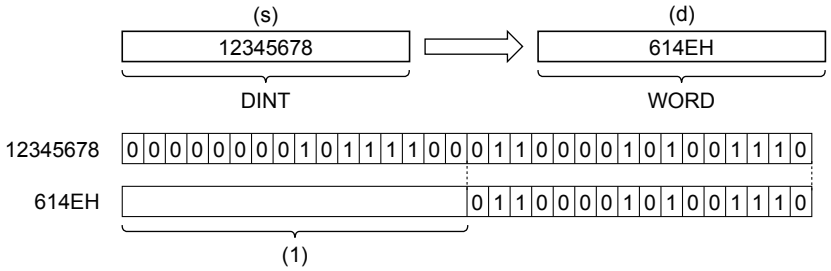
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DINT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DINT_TO_WORD(_E))	Output	Output variable	WORD

Processing details

■Operation processing

- These functions convert the DINT type data input to (s) to WORD type data and output from (d).
- The information stored in high-order 16 bits of an input value is discarded. (Refer to (1) in the figure below.)



- A value input to (s) is the DINT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Point

When DINT_TO_WORD(_E) is executed, the information stored in high-order 16 bits of the DINT type data value input from (s) is discarded.

Operation error

There is no operation error.

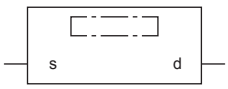
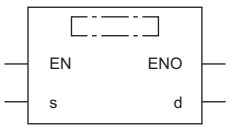
20.27 Converting DINT to DWORD

DINT_TO_DWORD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DINT type data to DWORD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DINT_TO_DWORD(s);
		[With EN/ENO]
		d:=DINT_TO_DWORD_E(EN,ENO,s);

Setting data

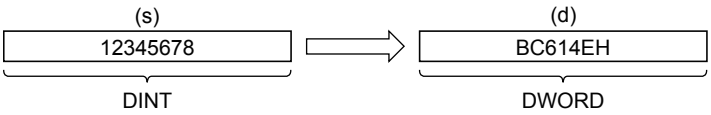
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DINT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DINT_TO_DWORD(_E))	Output	Output variable	DWORD

Processing details

■Operation processing

- These functions convert the DINT type data input to (s) to DWORD type data and output from (d).



- A value input to (s) is the DINT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

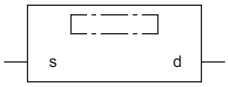
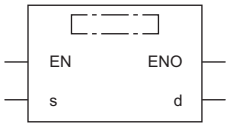
20.28 Converting DINT to INT

DINT_TO_INT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DINT type data to INT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DINT_TO_INT(s);
		[With EN/ENO]
		d:=DINT_TO_INT_E(EN,ENO,s);

Setting data

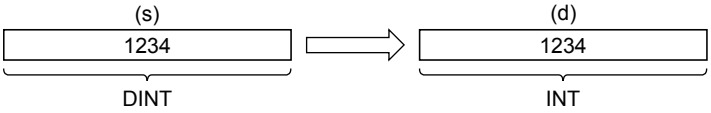
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DINT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DINT_TO_INT(_E))	Output	Output variable	INT

Processing details

■Operation processing

- These functions convert the DINT type data input to (s) to INT type data and output from (d).



- A value input to (s) is the DINT type data value.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3401H	The 32-bit signed binary data in the device specified by (s) is out of the valid range (-32768 to 32767).

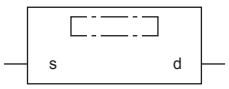
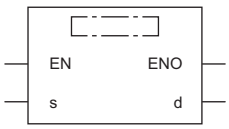
20.29 Converting DINT to BCD

DINT_TO_BCD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DINT type data to BCD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DINT_TO_BCD(s);
		[With EN/ENO]
		d:=DINT_TO_BCD_E(EN,ENO,s);

Setting data

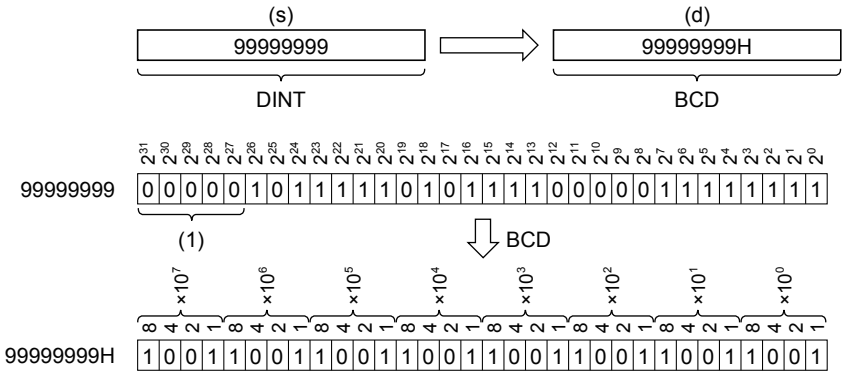
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DINT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DINT_TO_BCD(_E))	Output	Output variable	ANY_BIT

Processing details

■Operation processing

- These functions convert the DINT type data input to (s) to BCD type data and output from (d).



(1): Make sure to set them to "0".

- A value input to (s) is the DINT type data value. When (d) is WORD, the input value is within the range from 0 to 9999. When (d) is DWORD, the input value is within the range from 0 to 99999999.
- WORD or DWORD can be specified to (d). BOOL cannot be specified.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

- When (d) is WORD

Error code (SD0/SD8067)	Description
3401H	The 32-bit signed binary data in the device specified by (s) is out of the valid range (-32768 to 32767).
	Data in the device specified by (s) is out of the valid range (0 to 9999).

- When (d) is DWORD

Error code (SD0/SD8067)	Description
3401H	Data in the device specified by (s) is out of the valid range (0 to 99999999).

20.30 Converting DINT to REAL

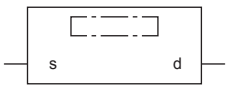
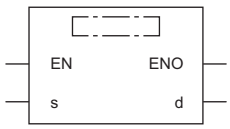
DINT_TO_REAL(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DINT type data to REAL type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]	[With EN/ENO]	[Without EN/ENO]
		d:=DINT_TO_REAL(s);
		[With EN/ENO]
		d:=DINT_TO_REAL_E(EN,ENO,s);

Setting data

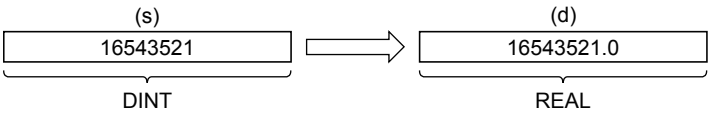
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DINT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DINT_TO_REAL(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions convert the DINT type data input to (s) to REAL type data and output from (d).



- A value input to (s) is the DINT type data value.
- The number of significant figures of the REAL type data is approximately 7 since the data is processed in 32-bit single precision.
- The converted data includes an error (rounding error) if an integer value is outside the range of -16777216 to 16777215.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

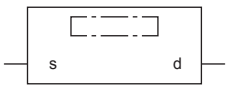
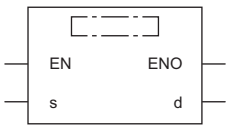
20.31 Converting DINT to TIME

DINT_TO_TIME(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DINT type data to TIME type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DINT_TO_TIME(s);
		[With EN/ENO]
		d:=DINT_TO_TIME_E(EN,ENO,s);

Setting data

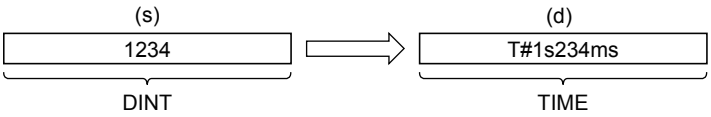
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DINT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DINT_TO_TIME(_E))	Output	Output variable	TIME

Processing details

■Operation processing

- These functions convert the DINT type data input to (s) to TIME type data and output from (d).



- A value input to (s) is the DINT type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

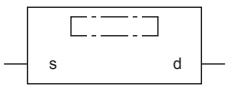
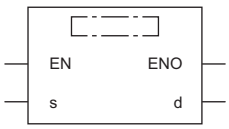
20.32 Converting DINT to STRING

DINT_TO_STRING(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert DINT type data to STRING type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DINT_TO_STRING(s);
		[With EN/ENO]
		d:=DINT_TO_STRING_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	DINT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DINT_TO_STRING(_E))	Output	Output variable	STRING(11)

Processing details

■Operation processing

- These functions convert the DINT type data input to (s) to STRING type data and output from (d).

(s)

DINT

(d)

High-order byte	Low-order byte	
ASCII code for billions place	Sign data	+0
ASCII code for ten-millions place	ASCII code for hundred-millions place	+1
ASCII code for hundred-thousands place	ASCII code for millions place	+2
ASCII code for thousands place	ASCII code for ten-thousands place	+3
ASCII code for tens place	ASCII code for hundreds place	+4
00H (1)	ASCII code for ones place	+5

(1): "00H" is stored when "SM701" (output character number selector) is off.

- A value input to (s) is the DINT type data value.
- In "Sign data", 20H (space) is stored when the input value is positive, and 2DH (-) is stored when the input value is negative.
- 20H (space) is stored in high-order digits when the number of significant figures is small.

Ex.

When "-123456" is input

(s)

-123456

DINT

(d)

High-order byte	Low-order byte	
20H (SP)	2DH (-)	+0
20H (SP)	20H (SP)	+1
31H (1)	20H (SP)	+2
33H (3)	32H (2)	+3
35H (5)	34H (4)	+4
00H	36H (6)	+5

- 00H is stored at the end (high-order byte of the 6th word) of the character string when SM701 (output character number selector signal) is off.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

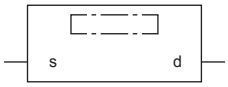
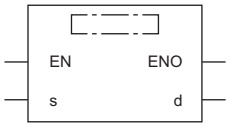
20.33 Converting BCD to INT

BCD_TO_INT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert BCD type data to INT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=BCD_TO_INT(s);
		[With EN/ENO]
		d:=BCD_TO_INT_E(EN,ENO,s);

Setting data

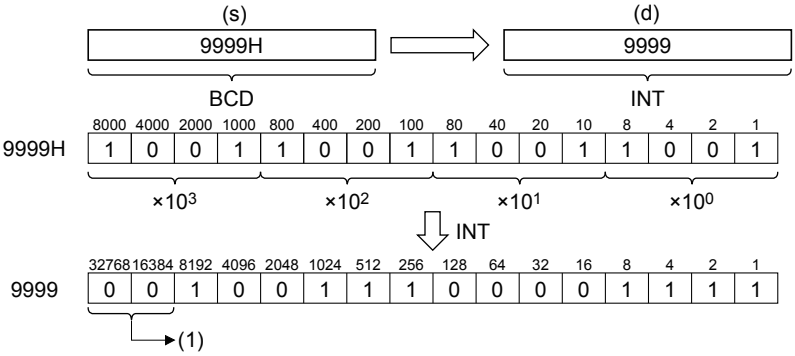
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	WORD
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(BCD_TO_INT(_E))	Output	Output variable	INT

Processing details

■Operation processing

- These functions convert the BCD type data input to (s) to INT type data and output from (d).



(1): Always becomes "0".

- A value input to (s) is the WORD type data value and within the range from 0H to 9999H (from 0 to 9 for each digit).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3401H	A value other than 0 to 9 exists in each digit of (s).

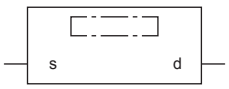
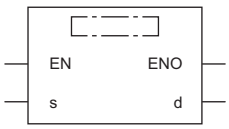
20.34 Converting BCD to DINT

BCD_TO_DINT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert BCD type data to DINT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=BCD_TO_DINT(s);
		[With EN/ENO]
		d:=BCD_TO_DINT_E(EN,ENO,s);

Setting data

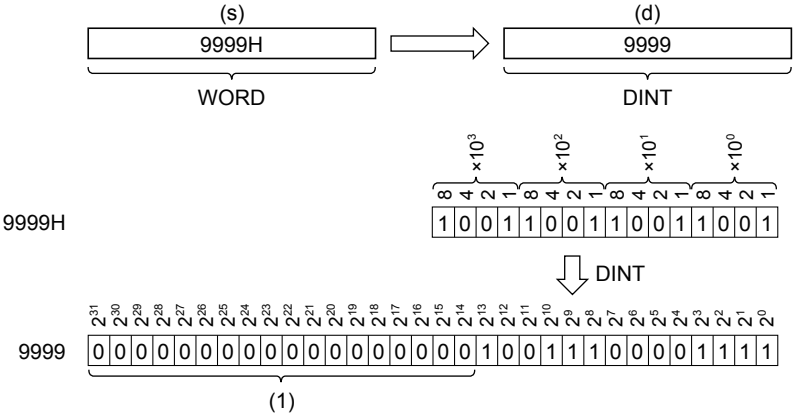
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	ANY_BIT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(BCD_TO_DINT(_E))	Output	Output variable	DINT

Processing details

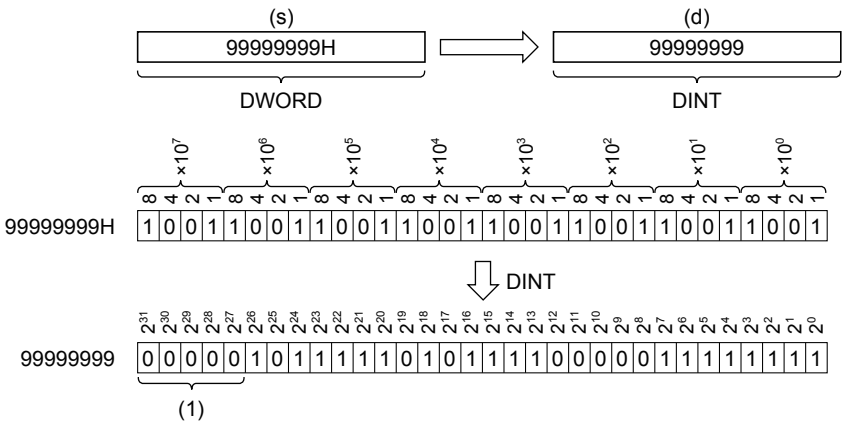
■Operation processing

- These functions convert the BCD type data input to (s) to DINT type data and output from (d).
  - When WORD is specified to (s)



(1): Always becomes "0".

• When DWORD is specified to (s)



(1): Always becomes "0".

- A value input to (s) is within the range from 0H to 9999H (from 0 to 9 for each digit) for the WORD type data value and from 0H to 99999999H (from 0 to 9 for each digit) for the DWORD type data value.
- WORD or DWORD can be specified to (s). BOOL cannot be specified.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

- When (s) is WORD

Error code (SD0/SD8067)	Description
3401H	A value other than 0 to 9 exists in each digit of (s).

- When (s) is DWORD

Error code (SD0/SD8067)	Description
3401H	A value other than 0 to 9 exists in each digit of (s).

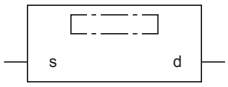
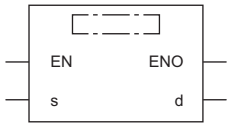
20.35 Converting REAL to INT

REAL_TO_INT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert REAL type data to INT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=REAL_TO_INT(s);
		[With EN/ENO]
		d:=REAL_TO_INT_E(EN,ENO,s);

Setting data

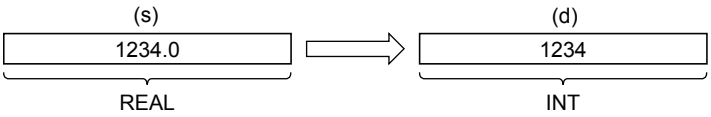
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(REAL_TO_INT(_E))	Output	Output variable	INT

Processing details

■Operation processing

- These functions convert the REAL type data input to (s) to INT type data and output from (d).



- A value input to (s) is the REAL type data value and within the range from -32768 to 32767.
- After conversion, the first digit after the decimal point of the REAL type data value is rounded down.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Precautions

If the single-precision real number set in (s) is out of the effective value range, the program will not operate correctly.

Operation error

Error code (SD0/SD8067)	Description
3402H	• A special number is set to (s). • The set single-precision real number is not located within the following range. $0, 2^{-126} \leq \lvert(s)\rvert < 2^{128}$ • The set device or label value is -0, denormalized number, NaN (not a number), or $\pm\infty$.

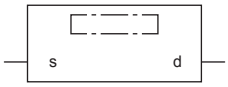
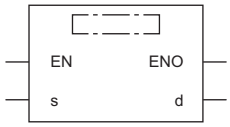
20.36 Converting REAL to DINT

REAL_TO_DINT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert REAL type data to DINT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=REAL_TO_DINT(s);
		[With EN/ENO]
		d:=REAL_TO_DINT_E(EN,ENO,s);

Setting data

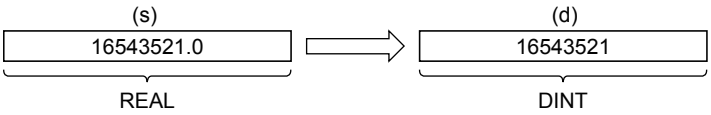
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(REAL_TO_DINT(_E))	Output	Output variable	DINT

Processing details

■Operation processing

- These functions convert the REAL type data input to (s) to DINT type data and output from (d).



- A value input to (s) is the REAL type data value and within the range from -2147483648 to 2147483647.
- After conversion, the first digit after the decimal point of the REAL type data value is rounded down.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Precautions

If the single-precision real number set in (s) is out of the effective value range, the program will not operate correctly.

Operation error

Error code (SD0/SD8067)	Description
3402H	• A special number is set to (s). • The set single-precision real number is not located within the following range. $0, 2^{-126} \leq \lvert(s)\rvert < 2^{128}$ • The set device or label value is -0, denormalized number, NaN (not a number), or $\pm\infty$.

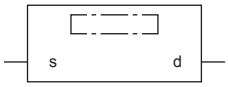
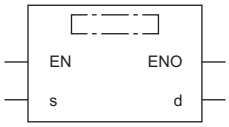
20.37 Converting REAL to STRING

REAL_TO_STRING(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert REAL type data to STRING type data (exponent format).

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=REAL_TO_STRING(s);
		[With EN/ENO]
		d:=REAL_TO_STRING_E(EN,ENO,s);

Setting data

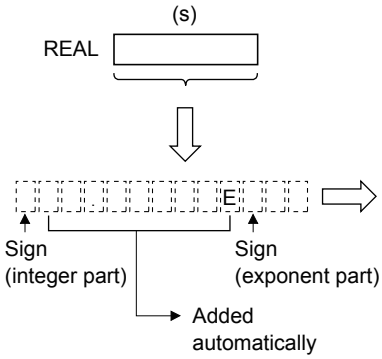
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(REAL_TO_STRING(_E))	Output	Output variable	STRING(13)

Processing details

■Operation processing

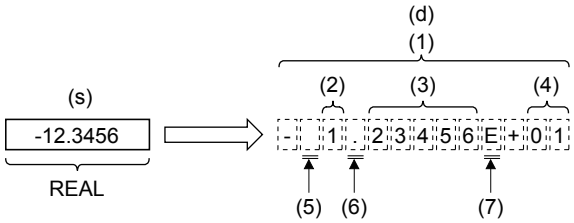
- These functions convert the REAL type data input to (s) to STRING type (exponent format) data and output from (d).



(d)

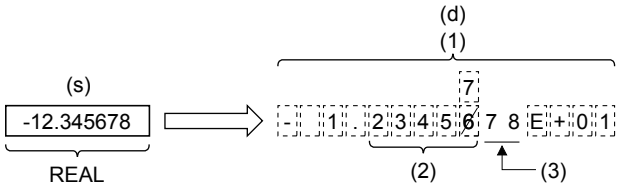
High-order byte	Low-order byte	
20H (SP)	Sign data (integer part)	+0
2EH (.)	ASCII code for integer part	+1
ASCII code for 2nd decimal place	ASCII code for 1st decimal place	+2
ASCII code for 4th decimal place	ASCII code for 3rd decimal place	+3
45H (E)	ASCII code for 5th decimal place	+4
ASCII code for tens place of exponent part	Sign data (exponent part)	+5
00H (NUL)	ASCII code for ones place of exponent part	+6

- A value input to (s) is the REAL type data value.
- The string data obtained by conversion is output from (d) as follows:
  - The number of digits is fixed respectively for the integer part, decimal part and exponent part as follows: Integer part: 1, decimal part: 5, exponent part: 2
  - "20H (space)" is stored in the 2nd byte, "2EH (.)" is stored in the 4th byte, and "45H (E)" is stored in the 10th byte automatically.



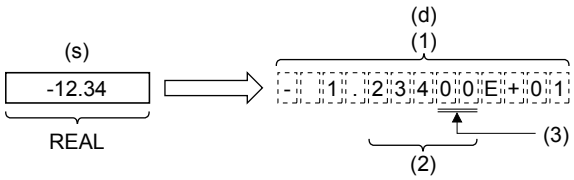
(1): Total number of digits (13 digits)
(2): Integer part (1 digit)
(3): Decimal part (5 digits)
(4): Exponent part (2 digits)
(5): "20H (SP)" is stored.
(6): "2EH (.)" is stored.
(7): "45H (E)" is stored.

- In "Sign data (integer part)", "20H (space)" is stored when the input value is positive, and "2DH (-)" is stored when the input value is negative.
- The 6th and later digits of the decimal part are rounded.



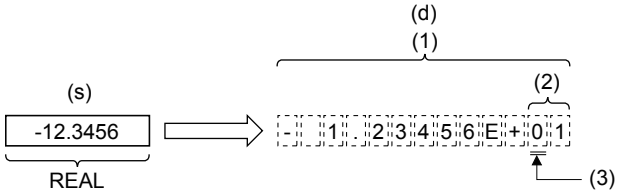
(1): Total number of digits (13 digits)
(2): Number of digits of decimal part (5)
(3): These digits are rounded off.

- "30H (0)" is stored in the decimal part when the number of significant figures is small.



(1): Total number of digits (13 digits)
(2): Number of digits of decimal part (5)
(3): "30H (0)" is stored.

- In "Sign data (exponent part)", "2BH (+)" is stored when the input value is positive, and "2DH (-)" is stored when the input value is negative.
- "30H (0)" is stored in the tens place of the exponent part when the exponent part consists of 1 digit.



(1): Total number of digits (13 digits)
(2): Number of digits of exponent part (2)
(3): "30H (0)" is stored.

- "00H" is automatically stored at the end (7th word) of the character string.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3402H	(s) is not located within the following range • 0, $2^{-126} \leq$|specified device value|$< 2^{128}$ • (s) is -0, denormalized number, NaN (not a number), or $\pm\infty$.
3406H	The whole converted character string cannot be stored in the devices from the device specified by (d) to the last device of the target device.

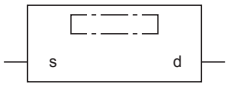
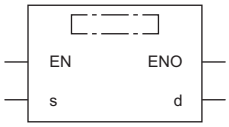
20.38 Converting TIME to BOOL

TIME_TO_BOOL(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert TIME type data to BOOL type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=TIME_TO_BOOL(s);
		[With EN/ENO]
		d:=TIME_TO_BOOL_E(EN,ENO,s);

Setting data

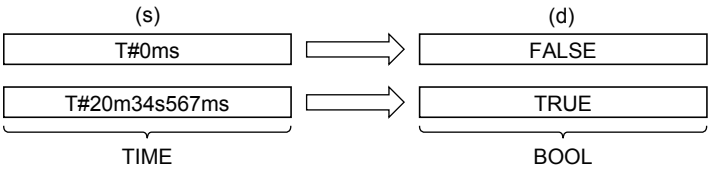
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(TIME_TO_BOOL(_E))	Output	Output variable	BOOL

Processing details

■Operation processing

- These functions convert the TIME type data input to (s) to BOOL type data and output from (d).



■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

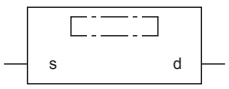
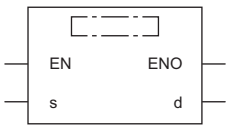
20.39 Converting TIME to WORD

TIME_TO_WORD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert TIME type data to WORD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=TIME_TO_WORD(s);
		[With EN/ENO]
		d:=TIME_TO_WORD_E(EN,ENO,s);

Setting data

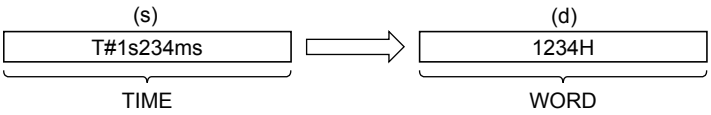
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(TIME_TO_WORD(_E))	Output	Output variable	WORD

Processing details

■Operation processing

- These functions convert the TIME type data input to (s) to WORD type data and output from (d).



- A value input to (s) is the TIME type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

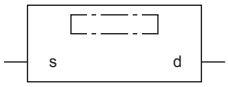
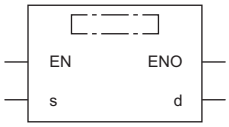
20.40 Converting TIME to DWORD

TIME_TO_DWORD(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert TIME type data to DWORD type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=TIME_TO_DWORD(s);
		[With EN/ENO]
		d:=TIME_TO_DWORD_E(EN,ENO,s);

Setting data

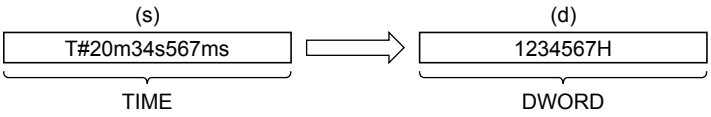
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(TIME_TO_DWORD(_E))	Output	Output variable	DWORD

Processing details

■Operation processing

- These functions convert the TIME type data input to (s) to DWORD type data and output from (d).



- A value input to (s) is the TIME type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

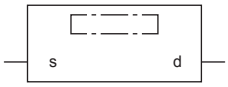
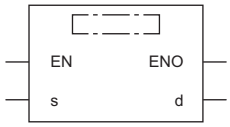
20.41 Converting TIME to INT

TIME_TO_INT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert TIME type data to INT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=TIME_TO_INT(s);
		[With EN/ENO]
		d:=TIME_TO_INT_E(EN,ENO,s);

Setting data

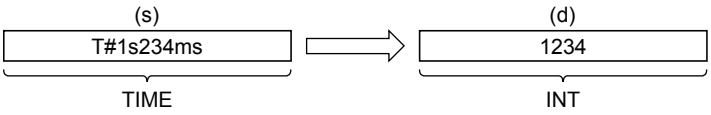
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(TIME_TO_INT(_E))	Output	Output variable	INT

Processing details

■Operation processing

- These functions convert the TIME type data input to (s) to INT type data and output from (d).



- A value input to (s) is the TIME type data value.
- When the data is converted to INT, the TIME type data stored in high-order 16 bits (1 word) are ignored.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

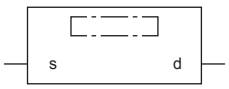
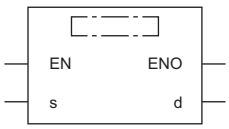
20.42 Converting TIME to DINT

TIME_TO_DINT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert TIME type data to DINT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=TIME_TO_DINT(s);
		[With EN/ENO]
		d:=TIME_TO_DINT_E(EN,ENO,s);

Setting data

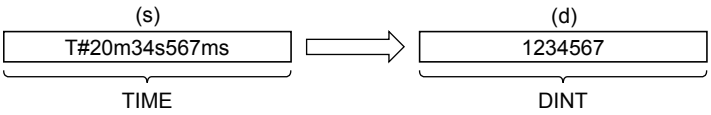
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(TIME_TO_DINT(_E))	Output	Output variable	DINT

Processing details

■Operation processing

- These functions convert the TIME type data input to (s) to DINT type data and output from (d).



- A value input to (s) is the TIME type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

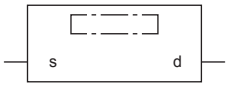
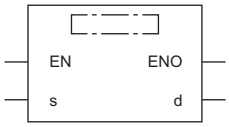
20.43 Converting TIME to STRING

TIME_TO_STRING(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert TIME type data to STRING type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=TIME_TO_STRING(s);
		[With EN/ENO]
		d:=TIME_TO_STRING_E(EN,ENO,s);

Setting data

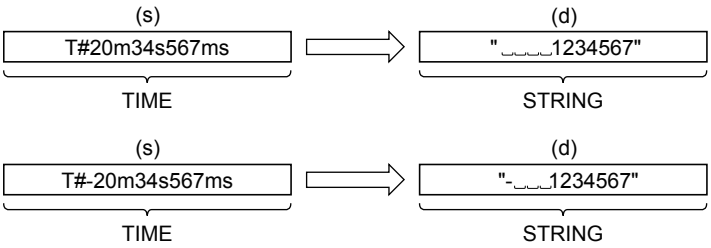
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(TIME_TO_STRING(_E))	Output	Output variable	STRING(11)

Processing details

■Operation processing

- These functions convert the TIME type data input to (s) to STRING type data and output from (d).



- A value input to (s) is the TIME type data value.
- 00H is stored at the end of the character string when SM701 (output character number selector signal) is off.
- The following shows the operation result to be stored in the output.
  - As the 1st character, "20H" (space) is stored if the binary data is positive, and "2DH" (-) is stored if the data is negative.
  - "20H" (space) is stored on the left side of the effective digits.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

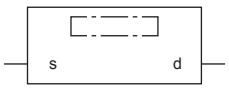
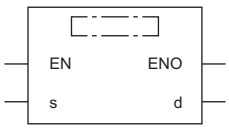
20.44 Converting STRING to BOOL

STRING_TO_BOOL(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert STRING type data to BOOL type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=STRING_TO_ BOOL(s);
		[With EN/ENO]
		d:=STRING_TO_ BOOL_E(EN,ENO,s);

Setting data

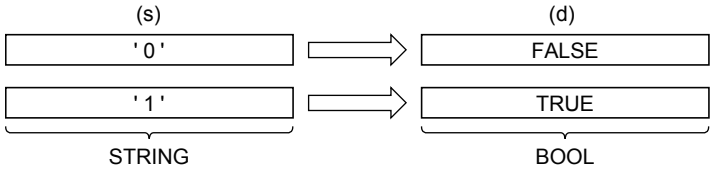
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	STRING(1)
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(STRING_TO_BOOL(_E))	Output	Output variable	BOOL

Processing details

■Operation processing

- These functions convert the STRING type (in the decimal format or exponent format) data input to (s) to BOOL type data and output from (d).



■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

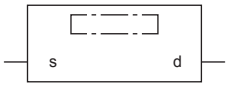
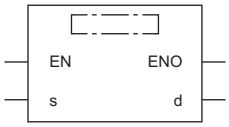
20.45 Converting STRING to INT

STRING_TO_INT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert STRING type data to INT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=STRING_TO_INT(s);
		[With EN/ENO]
		d:=STRING_TO_INT_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	STRING(6)
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(STRING_TO_INT(_E))	Output	Output variable	INT

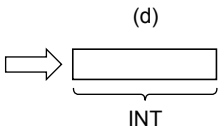
Processing details

■Operation processing

- These functions convert the STRING type data input to (s) to INT type data and output from (d).

(s)

	High-order byte	Low-order byte
+0	ASCII code for ten-thousands place	Sign data
+1	ASCII code for hundreds place	ASCII code for thousands place
+2	ASCII code for ones place	ASCII code for tens place
+3	00H (Indicates the end of the character string.)	



- A value input to (s) is the STRING type data value and within the following range.
  - Within the range of "30H" to "39H", "20H", "2DH", and "00H" in ASCII code
  - Within the range of "-32768" to "32767" as the STRING type data value

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3401H	Invalid data which cannot be converted to (s) are input. • Values for each place of the ASCII code are other than "30H" to "39H", "20H", and "00H". • Values for the ASCII data are other than "-32768" to "32767" when STRING_TO_INT(_E) is used.

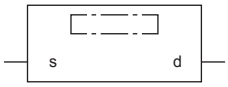
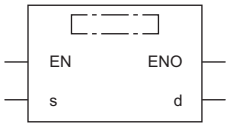
20.46 Converting STRING to DINT

STRING_TO_DINT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert STRING type data to DINT type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=STRING_TO_DINT(s);
		[With EN/ENO]
		d:=STRING_TO_DINT_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

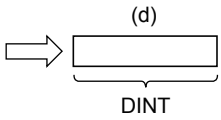
Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	STRING(11)
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(STRING_TO_DINT(_E))	Output	Output variable	DINT

Processing details

■Operation processing

- These functions convert the STRING type data input to (s) to DINT type data and output from (d).

	(s) High-order byte	Low-order byte
+0	ASCII code for billions place	Sign data
+1	ASCII code for ten-millions place	ASCII code for hundred-millions place
+2	ASCII code for hundred-thousands place	ASCII code for millions place
+3	ASCII code for thousands place	ASCII code for ten-thousands place
+4	ASCII code for tens place	ASCII code for hundreds place
+5	00H (Indicates the end of the character string.)	ASCII code for ones place



- A value input to (s) is the STRING type data value and within the following range.
  - Within the range of "30H" to "39H", "20H", "2DH", and "00H" in ASCII code
  - Within the range of "-2147483648" to "2147483647" as the STRING type data value

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3401H	Invalid data which cannot be converted to (s) are input. • Values for each place of the ASCII code are other than "30H" to "39H", "20H", and "00H". • Values for the ASCII data are other than "-2147483648" to "2147483647" when STRING_TO_DINT(_E) is used.

20.47 Converting STRING to REAL

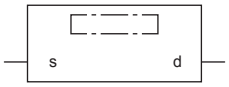
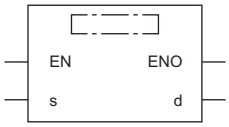
STRING_TO_REAL(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert STRING type data to REAL type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]	[With EN/ENO]	[Without EN/ENO]
		d:=STRING_TO_REAL(s);
		[With EN/ENO]
		d:=STRING_TO_REAL_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	STRING(24)
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(STRING_TO_REAL(_E))	Output	Output variable	REAL

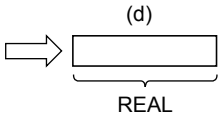
Processing details

■Operation processing

- These functions convert the STRING type (in the decimal format or exponent format) data input to (s) to REAL type data and output from (d).

(s)

	High-order byte	Low-order byte
+0	ASCII code for 1st character	Sign data
+1	ASCII code for 3rd character	ASCII code for 2nd character
+2	ASCII code for 5th character	ASCII code for 4th character
+3	ASCII code for 7th character	ASCII code for 6th character
+4	ASCII code for 9th character	ASCII code for 8th character
+5	ASCII code for 11th character	ASCII code for 10th character
+6	00H (Indicates the end of the character string.)	

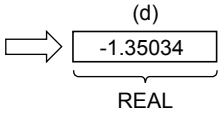


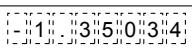
- The conversion source STRING type data can be in the decimal format or exponent format.
  - Decimal point format

(s)

	High-order byte	Low-order byte
+0	31H (1)	2DH (-)
+1	33H (3)	2EH (.)
+2	30H (0)	35H (5)
+3	34H (4)	33H (3)
+4	00H	

-1.35034





• Exponent format

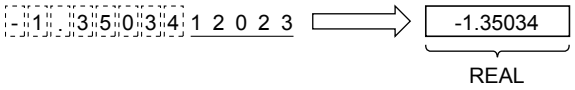
(s)

	High-order byte	Low-order byte
+0	31H (1)	2DH (-)
+1	33H (3)	2EH (.)
+2	30H (0)	35H (5)
+3	34H (4)	33H (3)
+4	2DH (-)	45H (E)
+5	30H (0)	31H (1)
+6	00H	

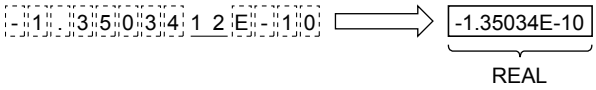
(d) → -1.35034E-10 (REAL)

-1.35034E-10

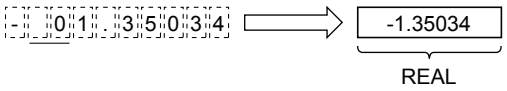
- With regard to STRING type data, six digits excluding the sign, decimal point and exponent part are valid, and the 7th and later digits are discarded during conversion.
  - Decimal point format



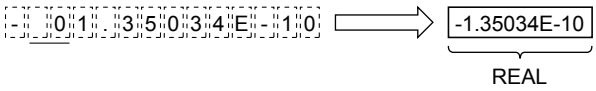
  - Exponent format



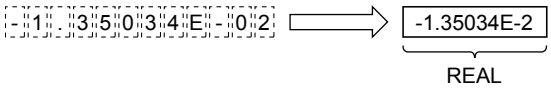
- When 2BH (+) is specified as the sign in the floating point format or when the sign is omitted, a character string is converted into a positive value. It is handled as negative value during conversion when the sign is set to 2DH (-).
- String data in the exponent format is handled as positive value during conversion when the sign of the exponent part is set to 2BH (+) or when the sign is omitted. When 2DH (-) is specified as the sign, a character string is converted into a negative value.
- When 20H (space) or 30H (0) exists between numbers except the first 0 in STRING type data, 20H or 30H is ignored during conversion.
  - Decimal point format



  - Exponent format



- When 30H (0) exists between a number and "E" in STRING type data (exponent format), 30H is ignored during conversion.



- When 20H (space) is contained in character string, 20H is ignored during conversion.
- Up to 24 characters can be input as STRING type data. 20H (space) and 30H (0) in a character string are counted as one character respectively.
- A value input to (s) is the STRING type data value and within the following range.
  - Within the range of "30H" to "39H", "45H", "2BH", "2DH", "2EH", "20H" and "00H" in ASCII code

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
2820H	00H does not exist in the corresponding device range starting from (s)
3401H	Invalid data which cannot be converted to (s) are set. • Any character other than "30(0)" to "39(9)" exists in the integer part or decimal part. • 2EH (.) exists in two or more positions in the specified character string. • Any character other than 45H (E), 65(e), 2B(+) , or 2D(-) exists in the specified exponent part. • Two or more exponent parts of 45H (E) or 65(e) exist in the specified character string. • Three or more digits of numerical values in the exponent parts are described in the specified character string. • Two or more signs of exponent parts of 2B(+) or 2D(-) exist in the specified character string. • Two or more signs of 2B(+) or 2D(-) exist in the integral part for the decimal point format and exist in the mantissa part for the exponent format in the specified character string. • The number of characters after (s) is 0 or more than 24
3403H	(d) exceeds the following range. (An overflow has occurred.) $\|(d)\|<2^{128}$

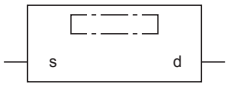
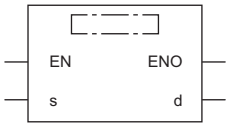
20.48 Converting STRING to TIME

STRING_TO_TIME(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert STRING type data to TIME type data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=STRING_TO_TIME(s);
		[With EN/ENO]
		d:=STRING_TO_TIME_E(EN,ENO,s);

Setting data

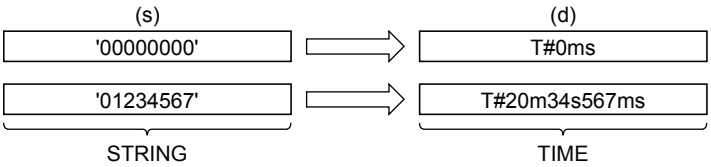
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	STRING(11)
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(STRING_TO_TIME(_E))	Output	Output variable	TIME

Processing details

■Operation processing

- These functions convert the STRING type data input to (s) to TIME type data and output from (d).



- A value input to (s) is the STRING type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3401H	Values for each place of the ASCII code for input are other than "30H" to "39H", "20H", and "00H".
	Value of the ASCII code for input are outside the following range. -2147483648 to 4147483647

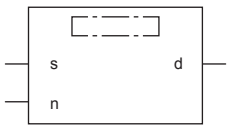
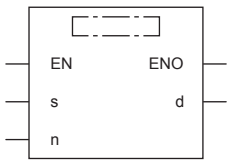
20.49 Converting Bit Array to INT

BITARR_TO_INT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert a bit array to INT type data for a specified number of bits.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
 	
	[With EN/ENO]
	 
	
		[Without EN/ENO]
		d:=BITARR_TO_INT(s,n);
		[With EN/ENO]
		d:=BITARR_TO_INT_E(EN,ENO,s,n);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(BitArr)	Input (Variables are available for element specification.)	Input variable	BOOL array element
n	Only a constant 4, 8, 12 or 16 can be specified.	Input variable	INT
ENO	Output status (TRUE: Normal , FALSE: Abnormal)	Output variable	BOOL
d(BITARR_TO_INT(_E))	Output	Output variable	ANY16

Processing details

■Operation processing

- These functions convert the data for bits specified by (n) starting from the bit array element input to (s) to ANY 16 type data and output from (d).
- "0" is set to output bits beyond the specified number of bits.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

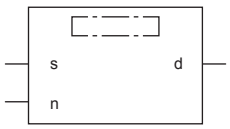
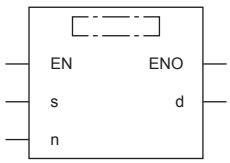
There is no operation error.

20.50 Converting Bit Array to DINT

BITARR_TO_DINT(_E)

FX5S FX5UJ FX5U FX5UC

These functions convert a bit array to DINT type data for a specified number of bits.

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO]  	[With EN/ENO]  	[Without EN/ENO] d:=BITARR_TO_DINT(s,n) [With EN/ENO] d:=BITARR_TO_DINT_E(EN,ENO,s,n);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(BitArr)	Input (Variables are available for element specification.)	Input variable	BOOL array element
n	Only a constant 4, 8, 12, 16, 20, 24, 28 or 32 can be specified.	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(BITARR_TO_DINT(_E))	Output	Output variable	ANY32

Processing details

■Operation processing

- These functions convert the data for bits specified by (n) starting from the bit array element input to (s) to ANY 32 type data and output from (d).
- "0" is set to output bits beyond the specified number of bits.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

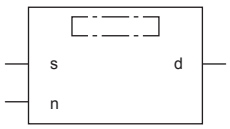
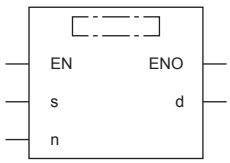
20.51 Converting INT to Bit Array

INT_TO_BITARR(_E)

FX5S FX5UJ FX5U FX5UC

These functions output low-order (n) bits of INT type data to a bit array.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=INT_TO_BITARR(s,n);
		[With EN/ENO]
		d:=INT_TO_BITARR_E(EN,ENO,s,n);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s	Input	Input variable	ANY16
n	Only a constant 4, 8, 12 or 16 can be specified.	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(INT_TO_BITARR(_E))	Output (Variables are available for element specification.)	Output variable	BOOL array element

Processing details

■Operation processing

- These functions output low-order (n) bits of ANY 16 type data specified to (s).
- Output bits beyond the specified number of bits are not changed.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

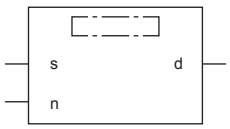
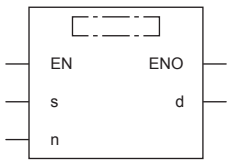
20.52 Converting DINT to Bit Array

DINT_TO_BITARR(_E)

FX5S FX5UJ FX5U FX5UC

These functions output low-order (n) bits of DINT type data to a bit array.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DINT_TO_BITARR(s,n);
		[With EN/ENO]
		d:=DINT_TO_BITARR_E(EN,ENO,s,n);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s	Input	Input variable	ANY32
n	Only a constant 4, 8, 12, 16, 20, 24, 28 or 32 can be specified.	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DINT_TO_BITARR(_E))	Output (Variables are available for element specification.)	Output variable	BOOL array element

Processing details

■Operation processing

- These functions output low-order (n) bits of ANY 32 type data specified to (s) to (d).
- Output bits beyond the specified number of bits are not changed.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

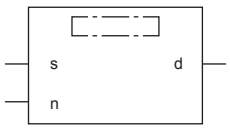
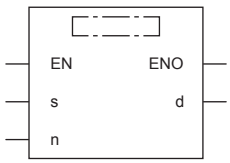
20.53 Bit Array Copy

CPY_BITARR(_E)

FX5S FX5UJ FX5U FX5UC

These functions copy specified number of bits of a bit array.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=CPY_BITARR(s,n);
		[With EN/ENO]
		d:=CPY_BITARR_E(EN,ENO,s,n);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(BitArrIn)	Input	Input variable	BOOL array element
n	Only a constant 4, 8, 12, 16, 20, 24, 28 or 32 can be specified.	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(CPY_BITARR(_E))	Output	Output variable	BOOL array element

Processing details

■Operation processing

- These functions output (n) bits of a bit array specified to (s) to (d).

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

20.54 Reading the Specified Bit of Word Label

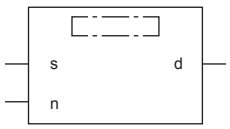
GET_BIT_OF_INT(_E)

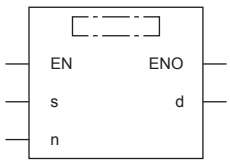
FX5S FX5UJ FX5U FX5UC

These functions reads the specified bit of the word label

Ladder diagram		Structured text	FBD/LD

[Without EN/ENO]	[With EN/ENO]	[Without EN/ENO]
		d:=GET_BIT_OF_INT(s,n);
		[With EN/ENO]
		d:=GET_BIT_OF_INT_E(EN,ENO,s,n);
			Not supported





Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s	Input	Input variable	ANY16
n	Only a constant 0 to 15 can be specified.	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(GET_BIT_OF_INT(_E))	Output	Output variable	BOOL

Processing details

■Operation processing

- These functions output (n)th bit of (s).

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

20.55 Writing the Specified Bit of Word Label

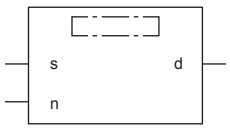
SET_BIT_OF_INT(_E)

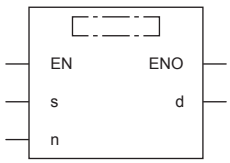
FX5S FX5UJ FX5U FX5UC

These functions writes the specified bit of the word label.

Ladder diagram		Structured text	FBD/LD

[Without EN/ENO]	[With EN/ENO]	[Without EN/ENO]
		d:=SET_BIT_OF_INT(s,n);
		[With EN/ENO]
		d:=SET_BIT_OF_INT_E(EN,ENO,s,n);
			Not supported





Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s	Input	Input variable	BOOL
n	Only a constant 0 to 15 can be specified.	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(SET_BIT_OF_INT(_E))	Output	Output variable	ANY16

Processing details

■Operation processing

- These functions write the BOOL value specified by (s) in the (n)th bit of (d).

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

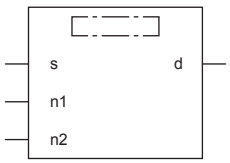
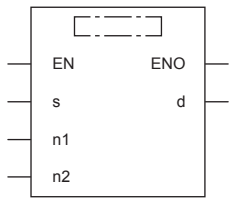
20.56 Copying the Specified Bit of Word Label

CPY_BIT_OF_INT(_E)

FX5S FX5UJ FX5U FX5UC

These functions copy the specified bit of the word label to the one of another word label.

Ladder diagram		Structured text	FBD/LD

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=CPY_BIT_OF_INT(s,n1,n2);
		[With EN/ENO]
		d:=CPY_BIT_OF_INT_E(EN,ENO,s,n1,n2);
			Not supported

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s	Input	Input variable	ANY16
n1	Bit specification of input variable (Only a constant 0 to 15 can be specified.)	Input variable	INT
n2	Bit specification of output variable (Only a constant 0 to 15 can be specified.)	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(CPY_BIT_OF_INT(_E))	Output	Output variable	ANY16

Processing details

■Operation processing

- These function copy the value of the (n1)th bit of the word specified by (s) to the (n2)th bit of (d).

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

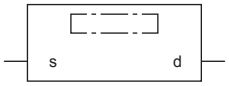
20.57 Unnecessary of Type Conversion

GET_BOOL_ADDR, GET_INT_ADDR, GET_WORD_ADDR

FX5S FX5UJ FX5U FX5UC

These functions output the input variable as the output variable type.

Ladder diagram, FBD/LD	Structured text

	d:=GET_BOOL_ADDR(s)
	d:=GET_INT_ADDR(s);
	d:=GET_WORD_ADDR(s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
s	Input	Input variable	ANY
d(GET_BOOL_ADDR / GET_INT_ADDR / GET_WORD_ADDR)	Output	Output variable	BOOL/INT/WORD

Processing details

■Operation processing

- These functions output the input data variable as the output variable type according to the following table.

General function	Input data type	Output data type
GET_BOOL_ADDR	BOOL ARRAY OF BOOL	BOOL
GET_INT_ADDR	INT DINT WORD REAL TIME STRING ARRAY OF INT ARRAY OF DINT ARRAY OF WORD ARRAY OF DWORD ARRAY OF REAL ARRAY OF TIME	INT
GET_WORD_ADDR		WORD

■Operation result

The operation processing is executed. The operation output value is output from (d).

Operation error

There is no operation error.

21 SINGLE NUMBER VARIABLE FUNCTIONS

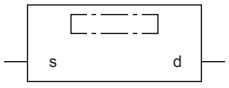
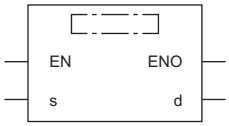
21.1 Absolute Value

ABS(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the absolute value of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=ABS(s);
		[With EN/ENO]
		d:=ABS_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	ANY_NUM
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(ABS(_E))	Output	Output variable	ANY_NUM

Processing details

■Operation processing

- These functions output the absolute value of the INT, DINT, or REAL type data input to (s) in the same data type as (s) from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".
  $B=|A|$
- A value input to (s) is the INT, DINT, or REAL type data value.
- When -32768 is input while the data type of (s) is INT, -32768 is output from (d).
- When -2147483648 is input while the data type of (s) is DINT, -2147483648 is output from (d). (An operation error does not occur. "ABS_E" outputs "TRUE" from output variable ENO.)

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

- When (s) is REAL

Error code (SD0/SD8067)	Description
3402H	The data specified by (s) is -0, denormalized number, NaN (not a number), or ±∞.
3403H	(d) exceeds the following range. (An overflow has occurred.) $\|(d)\|<2^{128}$

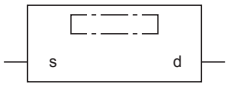
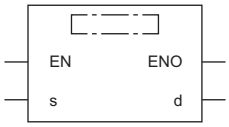
21.2 Square Root

SQRT(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the square root of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=SQRT(s);
		[With EN/ENO]
		d:=SQRT_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(SQRT(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions output the square root of the REAL type data input to (s) from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".

$B=\sqrt{A}$

- A value input to (s) is the REAL type data value and within the positive value range.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3405H	A negative value is input.

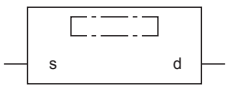
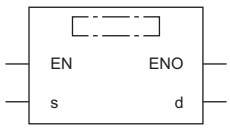
21.3 Natural Logarithm Operation

LN(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the natural logarithm operation result of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=LN(s);
		[With EN/ENO]
		d:=LN_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(LN(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions calculate the logarithm whose base is "e" of the REAL type data input to (s), and output from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".
  $B=\log_e A$
- In the natural logarithm operation, the base "e" is set to "2.71828".

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3405H	A negative value is input.

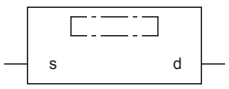
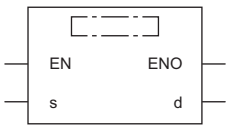
21.4 Calculating the Common Logarithm

LOG(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the operation result of the common logarithm (the logarithm whose base is 10) of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=LOG(s);
		[With EN/ENO]
		d:=LOG_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output condition (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(LOG(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions calculate the logarithm whose base is "10" of the REAL type data input to (s), and output from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".
  $B=\log_{10}A$
- A value input to (s) is the REAL type data value.
- Only a positive value can be set in (s). (The logarithm operation cannot be executed for a negative value).
- When the operation result is -0 or underflow occurs, the operation result is regarded as 0.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

• When (s) is REAL

Error code (SD0/SD8067)	Description
3402H	The value specified in (s) is -0, denormalized number, NaN (not a number), or ±∞.
3403H	The value of (d) exceeds the following range. (An overflow has occurred.) $\|(d)\|<2^{128}$
3405H	Data outside the allowable range was set to (s). • A negative value is specified. • "0" is specified.

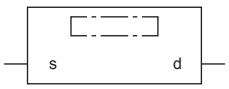
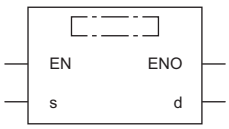
21.5 Exponential Operation

EXP(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the exponential operation result of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=EXP(s);
		[With EN/ENO]
		d:=EXP_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(EXP(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions calculate the exponent of the REAL type data input to (s), and output from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".
  $B=e^{A}$
- In the exponential operation, the base "e" is set to "2.71828".
- A value input to (s) is the REAL type data value.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3403H	The data after conversion is not -3.40282^{+38} to -1.17549^{-38}, or 1.17549^{-38} to 3.40282^{+38}.

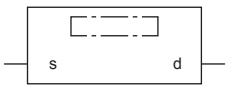
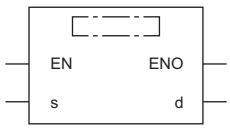
21.6 Sine Operation

SIN(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the sine of the angle of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=SIN(s);
		[With EN/ENO]
		d:=SIN_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(SIN(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions calculate the sine of the angle of the REAL type data input to (s), and output from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".
  B=SIN A
- A value (angle) input to (s) is the REAL type data value. Input a value in radians (angle×π/180).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3402H	A negative value is input.

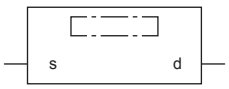
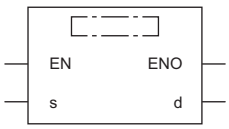
21.7 Cosine Operation

COS(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the cosine of the angle of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=COS(s);
		[With EN/ENO]
		d:=COS_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(COS(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions calculate the cosine of the angle of the REAL type data input to (s), and output from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".
  B=COS A
- A value (angle) input to (s) is the REAL type data value. Input a value in radians (angle×π/180).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3402H	A negative value is input.

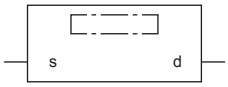
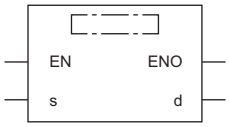
21.8 Tangent Operation

TAN(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the tangent of the angle of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=TAN(s);
		[With EN/ENO]
		d:=TAN_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(TAN(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions calculate the tangent of the angle data of the REAL type data (angle) input to (s), and output from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".
  B=TAN A
- Even when the input value is π/2 radian or (3/2) π radian, no error occurs because an operation error occurs in a radian value.
- A value (angle) input to (s) is the REAL type data value. Input a value in radians (angle×π/180).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3402H	A negative value is input.

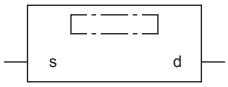
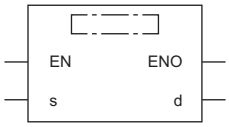
21.9 Arc Sine Operation

ASIN(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the arc sine value of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=ASIN(s);
		[With EN/ENO]
		d:=ASIN_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(ASIN(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions calculate the arc sine of the REAL type data input to (s), and output from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".
  $B=SIN^{-1} A$
- A value input to (s) is the REAL type data value and within the following range.
  ASIN(_E): -1.0 to 1.0
- A value (angle) in radians (angle$\times\pi$/180) is output from (d).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3402H	A negative value is input.
3405H	A value input by these functions is other than -1.0 to 1.0.

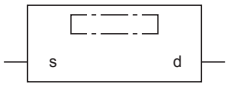
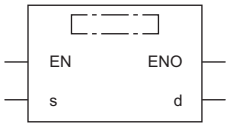
21.10 Arc Cosine Operation

ACOS(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the arc cosine value of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=ACOS(s);
		[With EN/ENO]
		d:=ACOS_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(ACOS(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions calculate the arc cosine of the REAL type data input to (s), and output from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".
  $B=COS^{-1} A$
- A value input to (s) is the REAL type data value and within the following range.
  ACOS(_E): -1.0 to 1.0
- A value (angle) in radians (angle$\times\pi$/180) is output from (d).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3402H	A negative value is input.
3405H	A value input by these functions is other than -1.0 to 1.0.

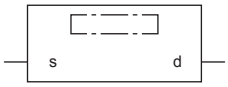
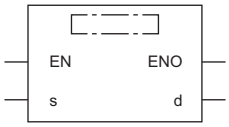
21.11 Arc Tangent Operation

ATAN(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the arc tangent value of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=ATAN(s);
		[With EN/ENO]
		d:=ATAN_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	REAL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(ATAN(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions calculate the arc tangent value of the REAL type data input to (s), and output from (d).
- These functions are expressed as follows when the input value is "A" and the output operation result is "B".
  $B=TAN^{-1} A$
- A value input to (s) is the REAL type data value and within the following range.
  ATAN(_E): $\pm 1.17549^{-38}$ to $\pm 3.40282^{+38}$
- A value (angle) in radians (angle$\times\pi$/180) is output from (d).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3402H	-0 is input.

22 ARITHMETIC OPERATION FUNCTIONS

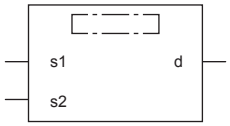
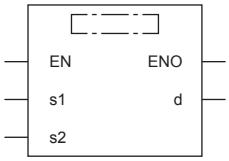
22.1 Addition

ADD(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the sum of input values ((s1) + (s2) + ... + (s28)).

Ladder diagram, FBD/LD*1		Structured text*1

[Without EN/ENO]
 
 	
	[With EN/ENO]
	 
	
		[Without EN/ENO]
		d:=ADD(s1,s2);
		[With EN/ENO]
		d:=ADD_E(EN,ENO,s1,s2);

*1 The input variable "s" can be changed in the range of 2 to 28.

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1) to s28(IN28)	Input	Input variable	ANY_NUM
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(ADD(_E))	Output	Output variable	ANY_NUM

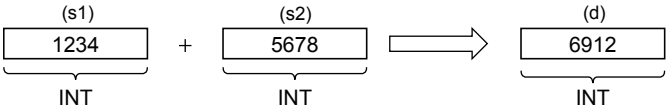
Processing details

■Operation processing

- These functions add the INT, DINT, or REAL type data ((s1) + (s2) + ... + (s28)) input to (s1) to (s28), and output from (d) in the same data type as (s).

Ex.

Data type is the INT type



- A value input to (s1) to (s28) is the INT, DINT, or REAL type data value.
- If an underflow and an overflow occur in the operation result, the result will be output as follows from (d).

Data type is INT	Data type is DINT	Data type is REAL
• Even if underflow or overflow occurs in the operation result, it is not regarded as an operation error. "ADD_E" outputs "TRUE" from ENO. [Example 1] 32767+2=-32767 (7FFFH)+(0002H)=8001H The most significant bit becomes 1, and a negative value is output. [Example 2] -32768+(-2)=32766 (8000H)+(FFFEH)=(7FFFH) The most significant bit becomes 0, and a positive value is output.	• Even if underflow or overflow occurs in the operation result, it is not regarded as an operation error. "ADD_E" outputs "TRUE" from ENO. [Example 1] 2147483647+2=-2147483647 (7FFFFFFFH)+(00000002H)=(80000001H) The most significant bit becomes 1, and a negative value is output. [Example 2] -2147483648+(-2)=2147483646 (80000000H)+(FFFFFFFEH)=(7FFFFFFEH) The most significant bit becomes 1, and a positive value is output.	An operation error occurs and an undefined value is output.

- When the operation result is 0, the zero flag (SM8020) turns on.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

- (s1) to (s28) are REAL

Error code (SD0/SD8067)	Description
3402H	The data specified by (s1) to (s28) is -0, denormalized number, NaN (not a number), or ±∞.
3403H	(d) exceeds the following range. (An overflow has occurred.) $\|(d)\|<2^{128}$

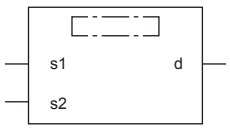
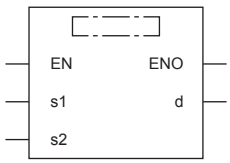
22.2 Multiplication

MUL(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the product input values ((s1)×(s2)× ... ×(s28)).

Ladder diagram, FBD/LD*1		Structured text*1

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=MUL(s1,s2);
		[With EN/ENO]
		d:=MUL_E(EN,ENO,s1,s2);

*1 The input variable "s" can be changed in the range of 2 to 28.

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1) to s28(IN28)	Input	Input variable	ANY_NUM
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(MUL(_E))	Output	Output variable	ANY_NUM

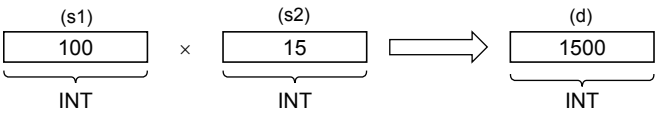
Processing details

■Operation processing

- These functions multiply the INT, DINT, or REAL type data input to (s1) to (s28) ((s1)×(s2)× ...×(s28)), and output the operation result from (d) in the same data type as (s).

Ex.

Data type is INT



- A value input to (s1) to (s28) is the INT, DINT, or REAL type data value.
- If an underflow occurs in the operation result, the result will be output as follows from (d).

Data type is INT	Data type is DINT	Data type is REAL
• Even if underflow or overflow occurs in the operation result, it is not regarded as an operation error. "MUL_E" outputs "TRUE" from ENO. • Even when the operation result exceeds the INT type data range, the INT type data is output. (The operation result is the DINT type, however, the output data is the INT type data with high-order 16 bits deleted.) • When the operation result exceeds the INT type data, convert an input value into the DINT type data by INT_TO_DINT then perform the operation.	• Even if underflow or overflow occurs in the operation result, it is not regarded as an operation error. "MUL_E" outputs "TRUE" from ENO. • Even when the operation result exceeds the DINT type data range, the DINT type data is output. (The operation result is the 64-bit data, however, the output data is the DINT type data with high-order 32 bits deleted.) • When the operation result exceeds the DINT type data, convert an input value into the REAL type data by DINT_TO_REAL then perform the operation.	An operation error occurs and an undefined value is output.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Point

When the operation result exceeds the data type range, convert the data type of an input value then perform the operation.

Operation error

• (s1) to (s28) are REAL

Error code (SD0/SD8067)	Description
3402H	The data specified by (s1) to (s28) is -0, denormalized number, NaN (not a number), or ±∞.
3403H	(d) exceeds the following range. (An overflow has occurred.) $\|(d)\|<2^{128}$

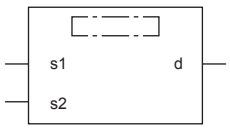
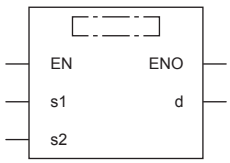
22.3 Subtraction

SUB(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the difference of input values ((s1) - (s2)).

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=SUB(s1,s2);
		[With EN/ENO]
		d:=SUB_E(EN,ENO,s1,s2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1), s2(IN2)	Input	Input variable	ANY_NUM
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(SUB(_E))	Output	Output variable	ANY_NUM

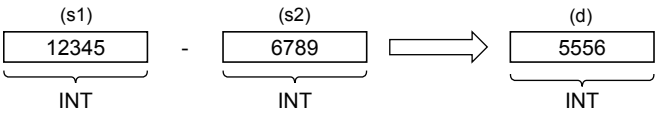
Processing details

■Operation processing

- These functions subtract the INT, DINT, or REAL type data input to (s1) and (s2) ((s1)-(s2)), and output the operation result from (d) in the same data type as (s).

Ex.

Data type is INT



- A value input to (s1) and (s2) is the INT, DINT, or REAL type data value.
- If an underflow and an overflow occur in the operation result, the result will be output as follows from (d).

Data type is INT	Data type is DINT	Data type is REAL
• Even if underflow or overflow occurs in the operation result, it is not regarded as an operation error. "SUB_E" outputs "TRUE" from ENO. [Example 1] 32767-(-2)=-32767 (7FFFH)-(FFFEH)=(8001H) The most significant bit becomes 1, and a negative value is output. [Example 2] -32768-2=32766 (8000H)-(0002H)=(7FFEH) The most significant bit becomes 0, and a positive value is output.	• Even if underflow or overflow occurs in the operation result, it is not regarded as an operation error. "SUB_E" outputs "TRUE" from ENO. [Example 1] 2147483647-(-2)=-2147483647 (7FFFFFFFH)-(FFFFFFFEH)=(80000001H) The most significant bit becomes 1, and a negative value is output. [Example 2] -2147483648-2=2147483646 (80000000H)-(00000002H)=(7FFFFFFEH) The most significant bit becomes 0, and a positive value is output.	An operation error occurs and an undefined value is output.

- When the operation result is 0, the zero flag (SM8020) turns on.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

- (s1) and (s2) are REAL

Error code (SD0/SD8067)	Description
3402H	The data specified by (s1) is -0, denormalized number, NaN (not a number), or ±∞.
	The data specified by (s2) is -0, denormalized number, NaN (not a number), or ±∞.
3403H	(d) exceeds the following range. (An overflow has occurred.) $\|(d)\|<2^{128}$

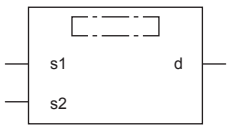
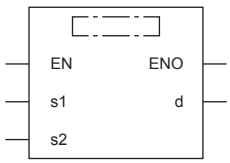
22.4 Division

DIV(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the quotient of input values ((s1) ÷ (s2)).

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DIV(s1,s2);
		[With EN/ENO]
		d:=DIV_E(EN,ENO,s1,s2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1)	Dividend	Input variable	ANY_NUM
s2(IN2)	Divisor	Input variable	ANY_NUM
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DIV(_E))	Output	Output variable	ANY_NUM

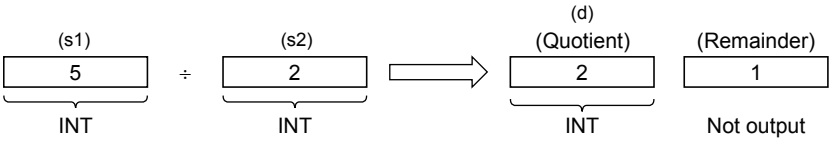
Processing details

■Operation processing

- These functions divide the INT, DINT, or REAL type data input to (s1) and (s2) ((s1) ÷ (s2)), and output the operation result from (d) in the same data type as (s).

Ex.

Data type is INT



- A value input to (s1) and (s2) is the INT, DINT, or REAL type data value. (However, input other than 0 to (s2).)

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

• (s1) and (s2) are INT

Error code (SD0/SD8067)	Description
3400H	The value (divisor) specified by (s2) is 0.

• (s1) and (s2) are DINT

Error code (SD0/SD8067)	Description
3400H	The value (divisor) specified by (s2) is 0.

• (s1) and (s2) are REAL

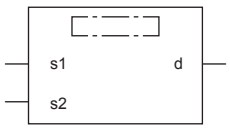
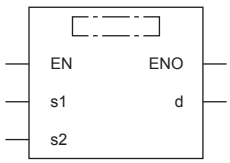
Error code (SD0/SD8067)	Description
3400H	The value (divisor) specified by (s2) is 0.
3402H	The data specified by (s1) is -0, denormalized number, NaN (not a number), or ±∞.
	The data specified by (s2) is -0, denormalized number, NaN (not a number), or ±∞.
3403H	(d) exceeds the following range. (An overflow has occurred.) $\|(d)\|<2^{128}$

22.5 Remainder

MOD(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the remainder of input values ((s1) ÷ (s2)).

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO] 	[With EN/ENO] 	[Without EN/ENO] The function is described as an operator. (MELSEC iQ-F FX5 Programming Manual (Program Design)) [With EN/ENO] d:=MOD_E(EN,ENO,s1,s2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1)	Dividend	Input variable	ANY_INT
s2(IN2)	Divisor	Input variable	ANY_INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(MOD(_E))	Output	Output variable	ANY_INT

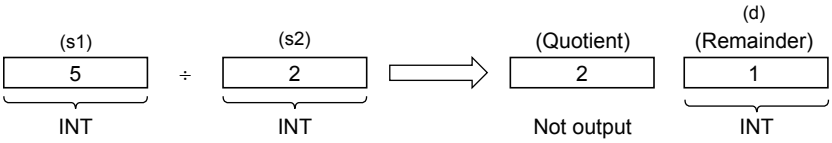
Processing details

■Operation processing

- These functions divide the INT or DINT type data input to (s1) and (s2) ((s1) ÷ (s2)), and output the remainder from (d) in the same data type as (s).

Ex.

Data type is INT



- A value input to (s1) and (s2) is the INT and DINT type data value. (However, input other than 0 to (s2).)

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

- (s1) and (s2) are INT

Error code (SD0/SD8067)	Description
3400H	The value (divisor) specified by (s2) is 0.

- (s1) and (s2) are DINT

Error code (SD0/SD8067)	Description
3400H	The value (divisor) specified by (s2) is 0.

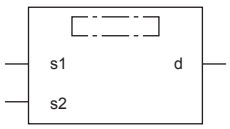
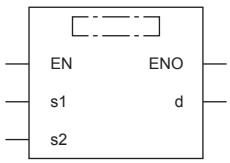
22.6 Exponentiation

EXPT(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the exponentiation of an input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=EXPT(s1,s2);
		[With EN/ENO]
		d:=EXPT_E(EN,ENO,s1,s2);

Setting data

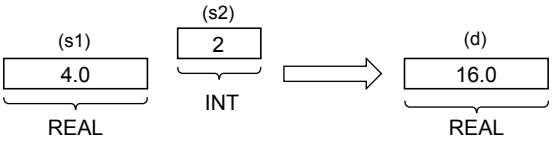
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1)	Cardinal number	Input variable	REAL
s2(IN2)	Exponent	Input variable	ANY_NUM
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(EXPT(_E))	Output	Output variable	REAL

Processing details

■Operation processing

- These functions raise the REAL type data input to (s1) by INT, DINT, or REAL specified by (s2), and output the operation result from (d).



■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

• (s1) is the REAL type and (s2) is the INT type

Error code (SD0/SD8067)	Description
3402H	The value of (s1) is outside the following range. 0, $2^{-126} \leq |(s1)| < 2^{128}$
	The data specified by (s1) is -0, denormalized number, NaN (not a number), or ±∞.
3403H	The operation result is within the following range. $2^{128} \leq$ |operation result|

• (s1) is the REAL type and (s2) is the DINT type

Error code (SD0/SD8067)	Description
3402H	The value of (s1) is outside the following range. 0, $2^{-126} \leq |(s1)| < 2^{128}$
	The data specified by (s1) is -0, denormalized number, NaN (not a number), or ±∞.
3403H	The operation result is within the following range. $2^{128} \leq$ |operation result|

• (s1) and (s2) are REAL

Error code (SD0/SD8067)	Description
3402H	The value of (s1) is outside the following range. 0, $2^{-126} \leq |(s1)| < 2^{128}$
	The data specified by (s1) is -0, denormalized number, NaN (not a number), or ±∞.
	The value of (s2) is outside the following range. 0, $2^{-126} \leq |(s2)| < 2^{128}$
	The data specified by (s2) is -0, denormalized number, NaN (not a number), or ±∞.
3403H	The operation result is within the following range. $2^{128} \leq$ |operation result|

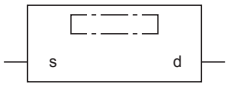
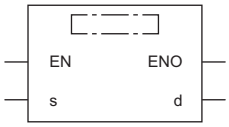
22.7 Move Operation

MOVE(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the assignment of input values.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=MOVE(s);
		[With EN/ENO]
		d:=MOVE_E(EN,ENO,s);

Setting data

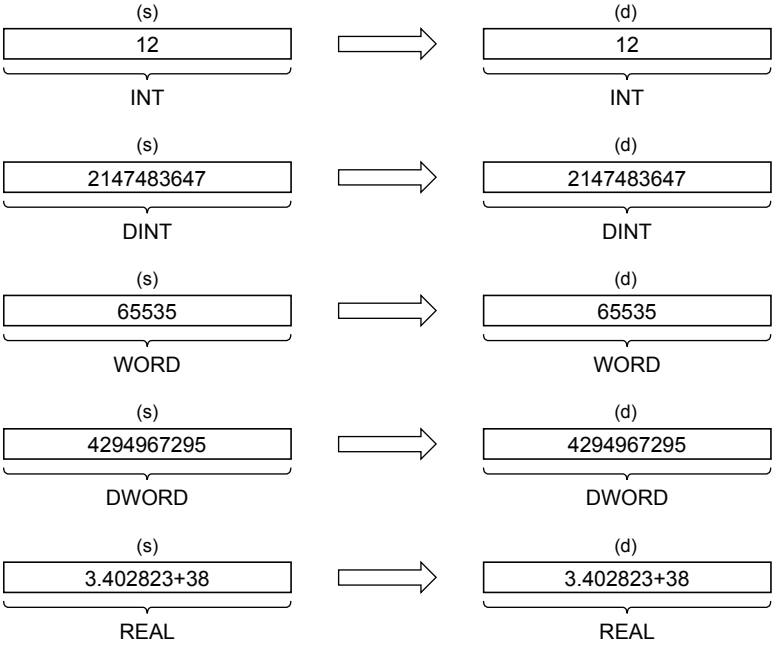
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	ANY
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(MOVE(_E))	Output	Output variable	ANY

Processing details

■Operation processing

- These functions assign the value of variable specified to (s) to the variable specified to (d).
- BOOL, INT, DINT, WORD, DWORD, REAL, STRING, TIME, structure, or array type can be specified for (s) and (d). Specify the same data type for (s) and (d).



■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s) and later, "00H" does not exist.
3405H	The character string specified by (s) has more than 16383 characters.
3406H	The whole specified character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.

23 BIT SHIFT FUNCTIONS

23.1 n-bit Left Shift

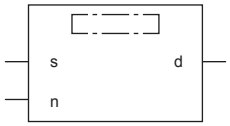
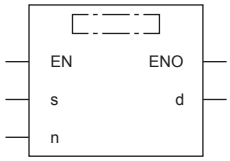
SHL(_E)

FX5S FX5UJ FX5U FX5UC

These functions shift an input value leftward by (n) bits and output the result.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]	[With EN/ENO]	[Without EN/ENO]
		d:=SHL(s,n);
		[With EN/ENO]
		d:=SHL_E(EN,ENO,s,n);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	ANY_BIT
n(N)	Number of shift bits	Input variable	ANY_BIT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(SHL(_E))	Output	Output variable	ANY_BIT

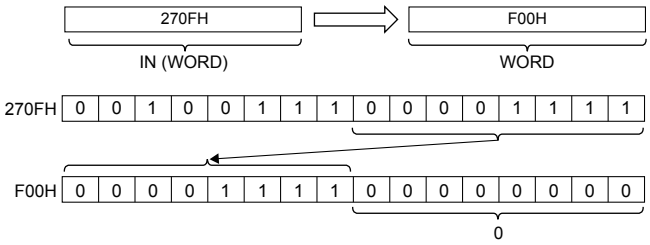
Processing details

■Operation processing

- These functions shift the 16-bit or 32-bit data (WORD or DWORD type) input to (s) left by (n) bits and output the result in the same data type as (s) from (d).
- The number input in (n) is used as the number of left-shift bits.

Ex.

When the data type of (s) is 16-bit data (WORD) and 8 is input in (n)



- "0" is set to "n" bits from the least significant bit.
- The input data to (s) is 16-bit data/32-bit data (WORD or DWORD type).
- A value input to (n) (Number of shift bits) is the INT type data value and within the following range.

When the data of (s) is 16-bit data (WORD)	When the data of (s) is 32-bit data (DWORD)
A value in (n) is within 0 to 15. The lower 4-bit data of the value in (n) is used. [Example] When the input value is 6: 6 When the input value is 22: 6	A value in (n) is within 0 to 31. The lower 5-bit data of the value in (n) is used. [Example] When the input value is 6: 6 When the input value is 22: 22

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

23.2 n-bit Right Shift

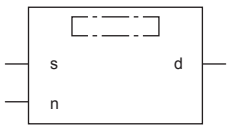
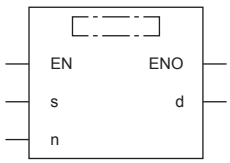
SHR(_E)

FX5S FX5UJ FX5U FX5UC

These functions shift an input value rightward by (n) bits and output the result.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]	[With EN/ENO]	[Without EN/ENO]
		d:=SHR(s,n);
		[With EN/ENO]
		d:=SHR_E(EN,ENO,s,n);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	ANY_BIT
n(N)	Number of shift bits	Input variable	ANY_BIT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(SHR(_E))	Output	Output variable	ANY_BIT

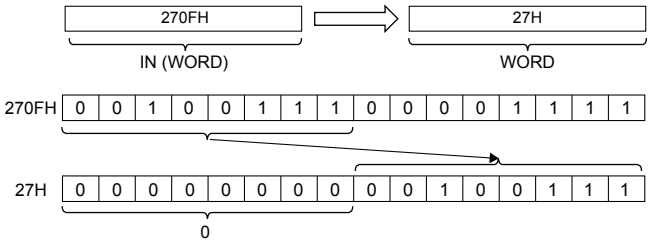
Processing details

■Operation processing

- These functions shift the 16-bit or 32-bit data (WORD or DWORD type) input to (s) right by (n) bits and output the result in the same data type as (s) from (d).
- The number input in (n) is used as the number of right-shift bits.

Ex.

When the data type of (s) is 16-bit data (WORD) and 8 is input in (n)



- "0" is set to "n" bits from the most significant bit.
- The input data to (s) is 16-bit data/32-bit data (WORD or DWORD type).
- A value input to (n) (Number of shift bits) is the INT type data value and within the following range.

When the data of (s) is 16-bit data (WORD)	When the data of (s) is 32-bit data (DWORD)
A value in (n) is within 0 to 15. The lower 4-bit data of the value in (n) is used. [Example] When the input value is 6: 6 When the input value is 22: 6	A value in (n) is within 0 to 31. The lower 5-bit data of the value in (n) is used. [Example] When the input value is 6: 6 When the input value is 22: 22

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

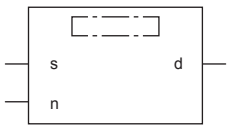
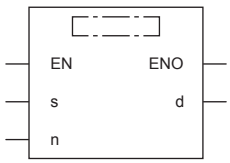
23.3 n-bit Left Rotation

ROL(_E)

FX5S FX5UJ FX5U FX5UC

These functions rotate an input value leftward by (n) bits and output the result.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=ROL(s,n);
		[With EN/ENO]
		d:=ROL_E(EN,ENO,s,n);

Setting data

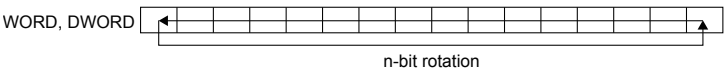
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	ANY_BIT
n(N)	Number of shift bits	Input variable	ANY_BIT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(ROL(_E))	Output	Output variable	ANY_BIT

Processing details

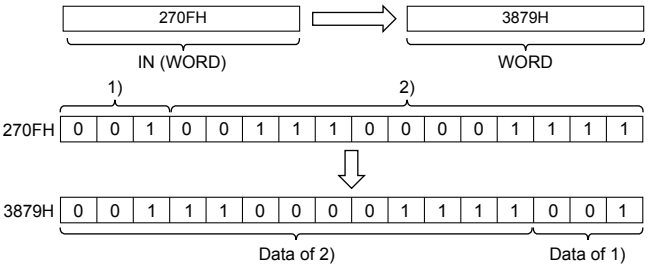
■Operation processing

- These functions rotate the 16-bit or 32-bit data (WORD or DWORD type) input to (s) left by (n) bits and output the result in the same data type as (s) from (d).
- The number input in (n) is used as the number of left-rotation bits.



Ex.

When the data type of (s) is 16-bit data (WORD) and 3 is input in (n) (The bits are rotated left by 3 bits.)



- The input data to (s) is 16-bit data/32-bit data (WORD or DWORD type).
- A value input to (n) (Number of shift bits) is the INT type data value and within the following range.

When the data of (s) is 16-bit data (WORD)	When the data of (s) is 32-bit data (DWORD)
A value in (n) is within 0 to 15. The lower 4-bit data of the value in (n) is used. [Example] When the input value is 6: 6 When the input value is 22: 6	A value in (n) is within 0 to 31. The lower 5-bit data of the value in (n) is used. [Example] When the input value is 6: 6 When the input value is 22: 22

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

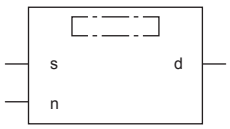
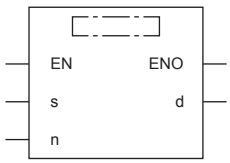
23.4 n-bit Right Rotation

ROR(_E)

FX5S FX5UJ FX5U FX5UC

These functions rotate an input value rightward by (n) bits and output the result.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=ROR(s,n);
		[With EN/ENO]
		d:=ROR_E(EN,ENO,s,n);

Setting data

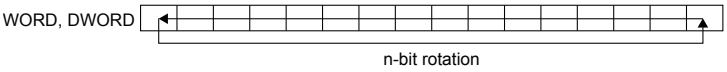
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	ANY_BIT
n(N)	Number of shift bits	Input variable	ANY_BIT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(ROR(_E))	Output	Output variable	ANY_BIT

Processing details

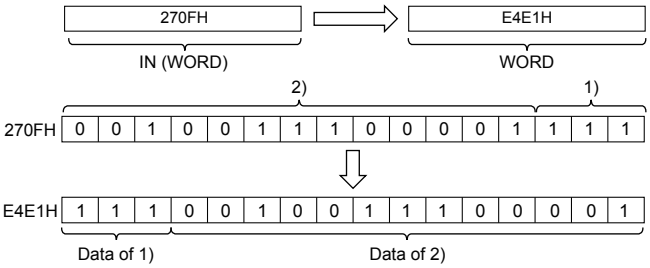
■Operation processing

- These functions rotate the 16-bit or 32-bit data (WORD or DWORD type) input to (s) right by (n) bits and output the result in the same data type as (s) from (d).
- The number input in (n) is used as the number of right-rotation bits.



Ex.

When the data type of (s) is 16-bit data (WORD) and 3 is input in (n) (The bits are rotated right by 3 bits.)



- The input data to (s) is 16-bit data/32-bit data (WORD or DWORD type).
- A value input to (n) (Number of shift bits) is the INT type data value and within the following range.

When the data of (s) is 16-bit data (WORD)	When the data of (s) is 32-bit data (DWORD)
A value in (n) is within 0 to 15. The lower 4-bit data of the value in (n) is used. [Example] When the input value is 6: 6 When the input value is 22: 6	A value in (n) is within 0 to 31. The lower 5-bit data of the value in (n) is used. [Example] When the input value is 6: 6 When the input value is 22: 22

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

24 STANDARD BITWISE BOOLEAN FUNCTIONS

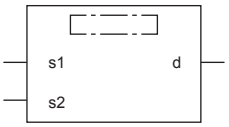
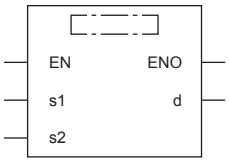
24.1 AND Operation, OR Operation, XOR Operation

AND(_E), OR(_E), XOR(_E)

FX5S FX5UJ FX5U FX5UC

- AND(_E): Outputs the logical product of input values.
- OR(_E): Outputs the logical sum of input values.
- XOR(_E): Outputs the exclusive logical sum of input values.

Ladder diagram, FBD/LD*1		Structured text*1

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=AND(s1,s2);
		d:=OR(s1,s2);
		d:=XOR(s1,s2);
		[With EN/ENO]
		d:=AND_E(EN,ENO,s1,s2);
		d:=OR_E(EN,ENO,s1,s2);
		d:=XOR_E(EN,ENO,s1,s2);

*1 The input variable "s" can be changed in the range of 2 to 28.

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1) to s28(IN28)	Input	Input variable	ANY_BIT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(AND(_E) / OR(_E) / XOR(_E))	Output	Output variable	ANY_BIT

Processing details

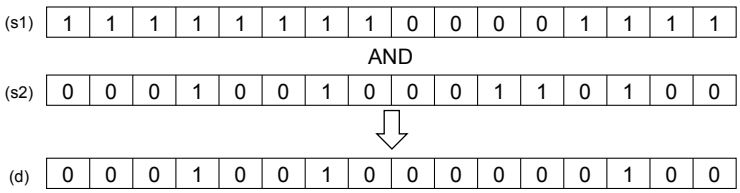
■Operation processing

1. AND(_E)

- These functions perform the logical AND on the BOOL, WORD, or DWORD type data input in (s1) to (s28) bit by bit, and output the operation result from (d) in the same data type as (s).

Ex.

When the data type is WORD

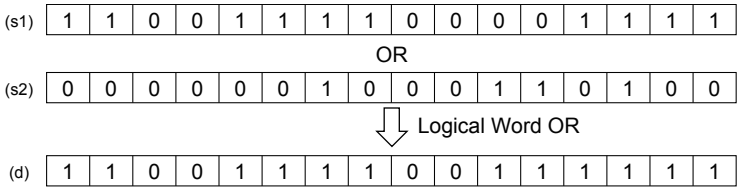


2. OR(_E)

- These functions perform the logical OR on the BOOL, WORD, or DWORD type data input in (s1) to (s28) bit by bit, and output the operation result from (d) in the same data type as (s).

Ex.

When the data type is WORD

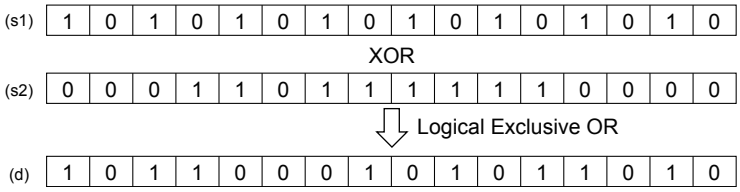


3. XOR(_E)

- These functions perform the exclusive logical OR on the BOOL, WORD, or DWORD type data input in (s1) to (s28) bit by bit, and output the operation result from (d) in the same data type as (s).

Ex.

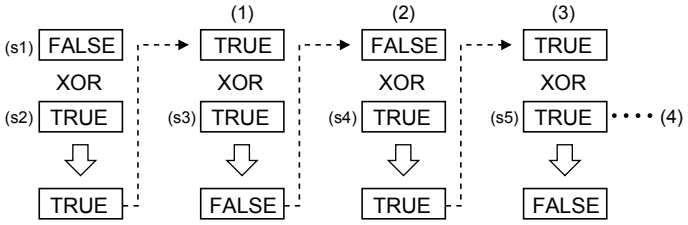
When the data type is WORD



- When three or more variables (s) exist, XOR is performed between (s1) and (s2) first, and XOR is successively performed between the result and (s3). When the expression includes (s4), XOR is performed between the result of XOR with (s3) and (s4). In this manner, XOR is repeated by the number of variables (s) in the order with (s5), (s6), and so on.

Ex.

When the data type is BOOL



(1): For 3 INs
(2): For 4 INs
(3): For 5 INs
(4): XOR is repeated by the number of variables (s).

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

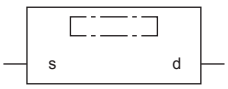
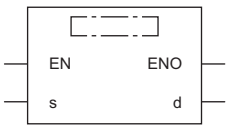
24.2 Logical Negation

NOT(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the logical negation of input values.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=NOT(s);
		[With EN/ENO]
		d:=NOT_E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	ANY_BIT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(NOT(_E))	Output	Output variable	ANY_BIT

Processing details

■Operation processing

- These functions calculate the logical negation for each bit of the BOOL, WORD, or DWORD type data input in (s), and output the operation result from (d) in the same data type as (s).

Ex.

When the data type is WORD

(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

- A value input to (s) is the BOOL, WORD, or DWORD type data value.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

25 SELECTION FUNCTIONS

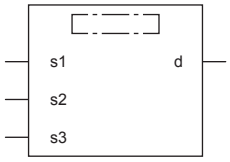
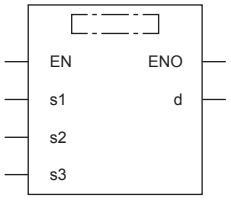
25.1 Selection

SEL(_E)

FX5S FX5UJ FX5U FX5UC

These functions output a selected input value.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=SEL(s1,s2,s3);
		[With EN/ENO]
		d:=SEL_E(EN,ENO,s1,s2,s3);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(G)	Output condition (TRUE: Output s3, FALSE: Output s2)	Input variable	BOOL
s2(IN0)	Input	Input variable	ANY
s3(IN1)			
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(SEL(_E))	Output	Output variable	ANY

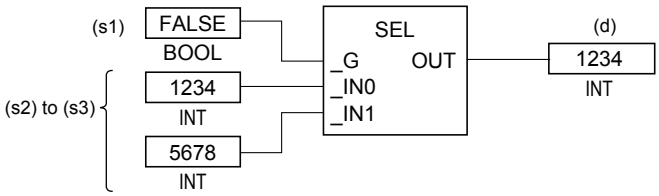
Processing details

■Operation processing

- These functions output a value input to (s2) and (s3) according to a value input to (s1) in the same data type as (s2) and (s3) from (d).
- When FALSE(=0) is input to (s1), these functions output an input value of (s2) from (d).
- When TRUE(=1) is input to (s1), these functions output an input value of (s3) from (d).

Ex.

The data type of (s2) and (s3) is the INT type ((s2) and (s3) of an argument correspond to the bit value of (s1) (0 or 1).)



- A value input to (s1) is the BOOL type data value.
- A data value of the BOOL, INT, DINT, WORD, DWORD, REAL, STRING, TIME, structure, or array type can be input to (s2) and (s3).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

• (s2) and (s3) are the STRING type

Error code (SD0/SD8067)	Description
2820H	"00H" is not set to a label specified by (s2) or devices from the device number to end device number of corresponding device.
	"00H" is not set to a label specified by (s3) or devices from the device number to the end device number of corresponding device.
3406H	The specified character string cannot be stored in a label specified by (d) or devices from the device number to the end device number of corresponding device.

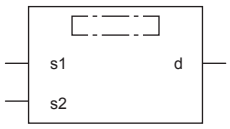
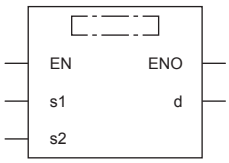
25.2 Selecting Maximum/Minimum Value

MAX(_E), MIN(_E)

FX5S FX5UJ FX5U FX5UC

- MAX(_E): These functions output the maximum value of an input value.
- MIN(_E): These functions output the minimum value of an input value.

Ladder diagram, FBD/LD*1		Structured text*1

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=MAX(s1,s2);
		d:=MIN(s1,s2);
		[With EN/ENO]
		d:=MAX_E(EN,ENO,s1,s2);
		d:=MIN_E(EN,ENO,s1,s2);

*1 The input variable "s" can be changed in the range of 2 to 28.

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1) to s28(IN28)	Input	Input variable	ANY_ELEMENTARY
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(MAX(_E) / MIN(_E))	Output	Output variable	ANY_ELEMENTARY

Processing details

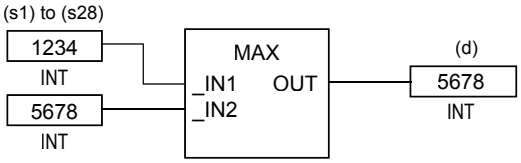
■Operation processing

- MAX(_E)

  These functions output the maximum value of the BOOL, INT, DINT, WORD, DWORD, DWORD, REAL, STRING, or TIME type data input to (s1) to (s28) in the same data type as (s) from (d).

Ex.

Data type is INT

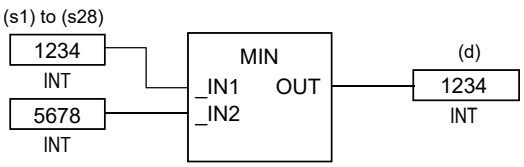


- MIN(_E)

  These functions output the minimum value of the BOOL, INT, DINT, WORD, DWORD, DWORD, REAL, STRING, or TIME type data input to (s1) to (s28) in the same data type as (s) from (d).

Ex.

Data type is INT



- A data value of the BOOL, INT, DINT, WORD, DWORD, REAL, STRING, or TIME type can be input to (s1) to (s28).
- The number of pins for (s) can be changed in the range of 2 to 28.

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

• (s1) to (s28) are STRING

Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s1) to (s28) and later, "00H" does not exist.
3405H	The character string specified by (s1) to (s28) has more than 16383 characters.
3406H	The whole specified character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.

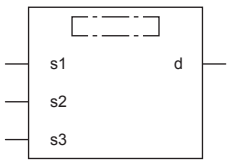
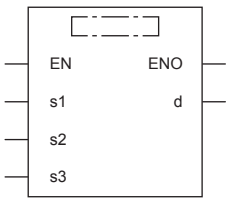
25.3 Limit Control

LIMIT(_E)

FX5S FX5UJ FX5U FX5UC

These functions output an input value controlled with the upper and lower limits.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=LIMIT(s1,s2,s3);
		[With EN/ENO]
		d:=LIMIT_E(EN,ENO,s1,s2,s3);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(MIN)	Lower limit value (minimum output threshold value)	Input variable	ANY_ELEMENTARY
s2(IN)	Input value to be controlled with the upper and lower limits	Input variable	ANY_ELEMENTARY
s3(MX)	Upper limit value (maximum output threshold value)	Input variable	ANY_ELEMENTARY
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(LIMIT(_E))	Output	Output variable	ANY_ELEMENTARY

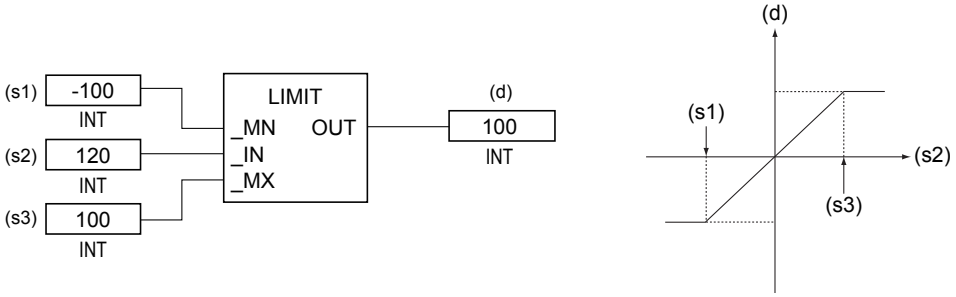
Processing details

■Operation processing

- These functions output an input value according to the BOOL, INT, DINT, WORD, DWORD, REAL, STRING, or TIME type data input to (s1), (s2), and (s3) in the same data type as (s1), (s2), and (s3) from (d).
  - When the input value of (s2) is larger than the one of (s3), these functions output the input value of (s3) from (d).
  - When the input value of (s2) is smaller than the one of (s1), these functions output the input value of (s1) from (d).
  - When the input value of (s1) ≤ the input value of (s2) ≤ the input value of (s3), these functions output the input value of (s2) from (d).

Ex.

Data type is INT



- A data value of the BOOL, INT, DINT, WORD, DWORD, REAL, STRING, or TIME type can be input to (s1), (s2), and (s3). (The input value of (s1) must be smaller than the one of (s3).)

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

• (s1), (s2), and (s3) are INT or WORD

Error code (SD0/SD8067)	Description
3405H	The lower limit value specified by (s1) is greater than the upper limit value specified by (s2).

• (s1), (s2), and (s3) are DINT, DWORD, or TIME

Error code (SD0/SD8067)	Description
3405H	The lower limit value specified by (s1) is greater than the upper limit value specified by (s2).

• (s1), (s2), and (s3) are BOOL

Error code (SD0/SD8067)	Description
3405H	The lower limit value specified by (s1) is greater than the upper limit value specified by (s3).

• (s1), (s2), and (s3) are the REAL type

Error code (SD0/SD8067)	Description
3402H	The value of (s1) is outside the following range. 0, $2^{-126} \leq \|(s1)\| < 2^{128}$
	The data specified by (s1) is -0, denormalized number, NaN (not a number), or ±∞.
	The value of (s2) is outside the following range. 0, $2^{-126} \leq \|(s2)\| < 2^{128}$
	The data specified by (s2) is -0, denormalized number, NaN (not a number), or ±∞.
	The value of (s3) is outside the following range. 0, $2^{-126} \leq \|(s3)\| < 2^{128}$
	The data specified by (s3) is -0, denormalized number, NaN (not a number), or ±∞.
3405H	The lower limit value specified by (s1) is greater than the upper limit value specified by (s3).

• (s1), (s2), and (s3) are STRING

Error code (SD0/SD8067)	Description
2820H	"00H" is not set to a label specified by (s1), (s2), and (s3) or devices from specified device number to the end device number of corresponding device.
3405H	The lower limit value specified by (s1) is greater than the upper limit value specified by (s3).
	The character strings specified by (s1), (s2), and (s3) have more than 16383 characters.
3406H	The specified character string cannot be stored in a label specified by (d) or devices from specified device number to the end device number of corresponding device.

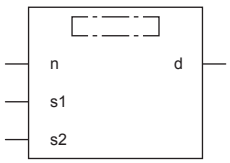
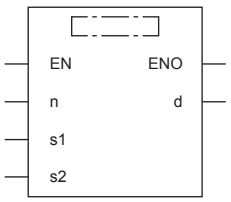
25.4 Multiplexer

MUX(_E)

FX5S FX5UJ FX5U FX5UC

These functions output one of multiple input values.

Ladder diagram, FBD/LD*1		Structured text*1

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=MUX(n,s1,s2);
		[With EN/ENO]
		d:=MUX_E(EN,ENO,n,s1,s2);

*1 The input variable "s" can be changed in the range of 2 to 28.

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
n(K)	Output value selection	Input variable	INT
s1(IN0) to s28(IN27)	Input	Input variable	ANY
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(MUX(_E))	Output	Output variable	ANY

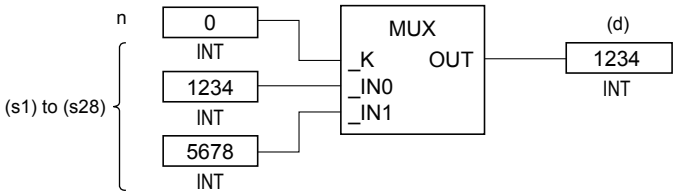
Processing details

■Operation processing

- These functions output one of values input to (s1) to (s28) according to the input value of (n) in the same data type as (s) from (d).
- When 0 is input to (n), these functions output a value input to (s1) from (d).
- When (n)-1 is input to (n), these functions output a value input to (sn) from (d).

Ex.

Data type is INT



- When a value input to (n) is outside the pin number range for (s), these functions output an indefinite value from (d). (An operation error does not occur. "MUX_E" outputs "FALSE" from ENO).
- A value input to (n) is the INT type data value and within the range from 0 to 27. (The value must be in the pin number range for (s).)
- A data value of the BOOL, INT, DINT, WORD, DWORD, REAL, STRING, TIME, structure, or array type can be input to (s).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s1) to (s28) and later, "00H" does not exist.
3405H	The character string specified by (s1) to (s28) has more than 16383 characters.
3406H	The whole specified character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.

26 COMPARISON FUNCTIONS

26.1 Compare

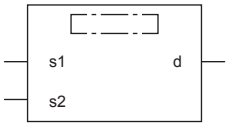
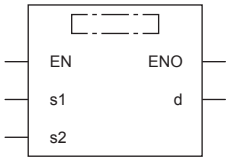
GT(_E), GE(_E), EQ(_E), LE(_E), LT(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the data comparison result of input values.

Ladder diagram, FBD/LD*1		Structured text*1

[Without EN/ENO]	[With EN/ENO]	[Without EN/ENO]*2
		d:=GT(s1,s2);
		d:=GE(s1,s2);
		d:=EQ(s1,s2);
		d:=LE(s1,s2);
		d:=LT(s1,s2);
		[With EN/ENO]
		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);

*1 The input variable "s" can be changed in the range of 2 to 28.

*2 Supported by engineering tool version "1.035M" and later. In earlier versions, write in operator form. (MELSEC iQ-F FX5 Programming Manual (Program Design))

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1) to s28(IN28)	Input	Input variable	ANY_ELEMENTARY
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(GT(_E) / GE(_E) / EQ(_E) / LE(_E) / LT(_E))	Output (TRUE: True value, FALSE: False value)	Output variable	BOOL

Processing details

■Operation processing

- These functions perform a comparison operation of input values of (s) and output operation results from (d) in the BOOL type.
  - GT(_E): These functions compare $[(s1)>(s2)]\&[(s2)>(s3)]\&\cdots\&[(s)_{(n-1)}>(s)_{(n)}]$.
    - When all the operation results are $(s)_{(n-1)}>(s)_{(n)}$, these functions output TRUE.
    - When any of the operation results is $(s)_{(n-1)}\leq(s)_{(n)}$, these functions output FALSE.
  - GE(_E): These functions compare $[(s1)\geq(s2)]\&[(s2)\geq(s3)]\&\cdots\&[(s)_{(n-1)}\geq(s)_{(n)}]$.
    - When all the operation results are $(s)_{(n-1)}\geq(s)_{(n)}$, these functions output TRUE.
    - When any of the operation result is $(s)_{(n-1)}<(s)_{(n)}$, these functions output FALSE.
  - EQ(_E): These functions compare $[(s1)=(s2)]\&[(s2)=(s3)]\&\cdots\&[(s)_{(n-1)}=(s)_{(n)}]$.
    - When all the operation results are $(s)_{(n-1)}=(s)_{(n)}$, these functions output TRUE.
    - When any of the operation results is $(s)_{(n-1)}\neq(s)_{(n)}$, these functions output FALSE.
  - LE(_E): These functions compare $[(s1)\leq(s2)]\&[(s2)\leq(s3)]\&\cdots\&[(s)_{(n-1)}\leq(s)_{(n)}]$.
    - When all the operation results are $(s)_{(n-1)}\leq(s)_{(n)}$, these functions output TRUE.
    - When any of the operation result is $(s)_{(n-1)}>(s)_{(n)}$, these functions output FALSE.
  - LT(_E): These functions compare $[(s1)<(s2)]\&[(s2)<(s3)]\&\cdots\&[(s)_{(n-1)}<(s)_{(n)}]$.
    - When all the operation results are $(s)_{(n-1)}<(s)_{(n)}$, these functions output TRUE.
    - When any of the operation results is $(s)_{(n-1)}\geq(s)_{(n)}$, these functions output FALSE.
- A data value of the INT, DINT, REAL, BOOL, WORD, DWORD, TIME, or STRING type can be input to (s).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

- (s1) to (s28) are the STRING type

Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s1) to (s28) and later, "00H" does not exist.
3405H	The character string specified by (s1) to (s28) has more than 16383 characters.
3406H	The whole specified character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.

- The above errors do not occur in LE(_E) and LT(_E)

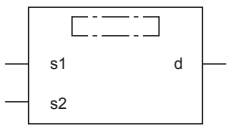
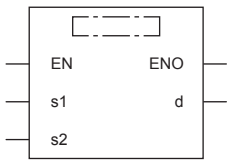
26.2 Compare

NE(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the data comparison result of input values.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]*1
		d:=NE(s1,s2);
		[With EN/ENO]
		d:=NE_E(EN,ENO,s1,s2);

*1 Supported by engineering tool version "1.035M" and later. In earlier versions, write in operator form. (MELSEC iQ-F FX5 Programming Manual (Program Design))

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1), s2(IN2)	Input	Input variable	ANY_ELEMENTARY
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(NE(_E))	Output (TRUE: True value, FALSE: False value)	Output variable	BOOL

Processing details

■Operation processing

- These functions perform a comparison operation of input values of (s) and output operation results from (d) in the BOOL type.
  - NE(_E): These functions compare [(s1)≠(s2)].
    - When (s1)≠(s2), these functions output TRUE.
    - These functions output FALSE when (s1)=(s2).
- A data value of the INT, DINT, REAL, BOOL, WORD, DWORD, TIME, or STRING type can be input to (s).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

- (s1) and (s2) are the STRING type

Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s) and later, "00H" does not exist.
3405H	The character string specified by (s) has more than 16383 characters.
3406H	The whole specified character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.

27 CHARACTER STRING FUNCTIONS

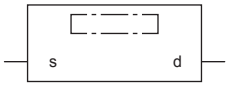
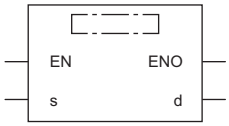
27.1 Character String Length Detection

LEN(_E)

FX5S FX5UJ FX5U FX5UC

These functions detect the length of an input character string and output the result.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=LEN(s);
		[With EN/ENO]
		d:=LEN _E(EN,ENO,s);

Setting data

■Descriptions, types, and data types

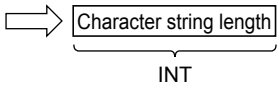
Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	STRING(255)
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(LEN(_E))	Output	Output variable	INT

Processing details

■Operation processing

- These functions detect the length of a character string input to (s) and output the result from (d).

	High-order byte	Low-order byte
+0	ASCII code for 2nd character	ASCII code for 1st character
+1	ASCII code for 4th character	ASCII code for 3rd character
+2	ASCII code for 6th character	ASCII code for 5th character
+n	00H (Indicates the end of the character string.)	ASCII code for (n)th character



- A value input to (s) is the STRING type data value and within the range from 0 to 255 byte(s).

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

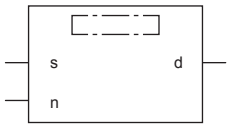
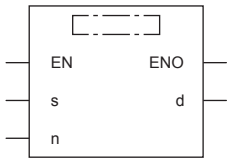
27.2 Extracting Character String Data from the Left/Right

LEFT(_E), RIGHT(_E)

FX5S FX5UJ FX5U FX5UC

- LEFT(_E): These functions output specified number of characters from the left of input character string data.
- RIGHT(_E): These functions output specified number of characters from the right of input character string data.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=LEFT(s,n);
		d:=RIGHT(s,n);
		[With EN/ENO]
		d:=LEFT_E(EN,ENO,s,n);
		d:=RIGHT_E(EN,ENO,s,n);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	STRING(255)
n(L)	Specification of number of characters to be extracted	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(LEFT(_E) / RIGHT(_E))	Output	Output variable	STRING(255)

Processing details

■Operation processing

- LEFT(_E)

These functions output the data for the specified number of characters from the left of a character string input to (s) from (d).
The value input to (n) specifies the number of characters to be extracted.

Ex.

When the value input to (n) is 7

"ABCDEF12345" ⇨ "ABCDEF1"

	High-order byte	Low-order byte
+0	42H (B)	41H (A)
+1	44H (D)	43H (C)
+2	46H (F)	45H (E)
+3	32H (2)	31H (1) ← (1)
+4	34H (4)	33H (3)
+5	00H	35H (5)

High-order byte	Low-order byte	
42H (B)	41H (A)	+0
44H (D)	43H (C)	+1
46H (F)	45H (E)	+2
00H	31H (1)	+3

(1): Number of characters to be extracted (L): 7

• RIGHT(_E)

These functions output the data for the specified number of characters from the right of a character string input to (s) from (d).
The value input to (n) specifies the number of characters to be extracted.

Ex.

When the value input to (n) is 5

"ABCDEF12345" ⇨ "12345"

	High-order byte	Low-order byte
+0	42H (B)	41H (A)
+1	44H (D)	43H (C)
+2	46H (F)	45H (E)
+3	32H (2)	31H (1)
+4	34H (4)	33H (3)
+5	00H	35H (5)

(1)

High-order byte	Low-order byte	
32H (2)	31H (1)	+0
34H (4)	33H (3)	+1
00H	35H (5)	+2

(1): Number of characters to be extracted (L): 5

• A value input to (s) is the STRING type data value and within the range from 0 to 255 byte(s).
• A value input to (n) is the INT type data value and within the range from 0 to 255. (However, the value must be within the number of characters of the character string to be input to (s).)

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

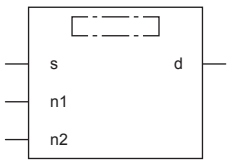
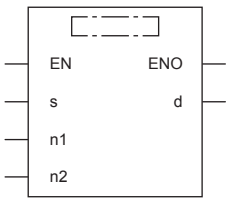
27.3 Extract Mid String

MID(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the specified number of characters from an arbitrary position of an input character string.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=MID(s,n1,n2);
		[With EN/ENO]
		d:=MID_E(EN,ENO,s,n1,n2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	STRING(255)
n1(L)	Specification of number of characters to be extracted	Input variable	INT
n2(P)	Specification of head character position of a character string to be extracted	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(MID(_E))	Output	Output variable	STRING(255)

Processing details

■Operation processing

- These functions output the data for the specified number of characters from an arbitrary position of a character string input to (s).
- The value input to (n1) specifies the number of characters to be extracted.
- The value input to (n2) specifies the number of the head character position of a character string to be extracted.

Ex.

When the value input to (n1) and (n2) is 5

(s)
"ABCDEF12345"

(d)
"EF123"

	High-order byte	Low-order byte	
+0	42H (B)	41H (A)	
+1	44H (D)	43H (C)	
+2	46H (F)	45H (E)	← (1)
+3	32H (2)	31H (1)	
+4	34H (4)	33H (3)	
+5	00H	35H (5)	← (2)

High-order byte	Low-order byte	
46H (F)	45H (E)	+0
32H (2)	31H (1)	+1
00H	33H (3)	+2

(1): Head character position of a character string to be extracted n2: 5th character
(2): Number of characters to be extracted n1: 5

- A value input to (s) is the STRING type data value and within the range from 0 to 255 byte(s).
- A value input to (n1) is the INT type data value and within the range from 0 to 255. (However, the value must be within the number of characters of the character string to be input to (s).)
- A value input to (n2) is the INT type data value and within the range from 1 to 255. (However, the value must be within the number of characters of the character string to be input to (s).)

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s) and later, "00H" does not exist.
3405H	The character string specified by (s) has more than 16383 characters.
	Data outside the allowable range was set to (n1) and (n2). • The value stored in a device specified in (n1) and (n2) is 0 or less. • The value stored in a device specified in (n2) is any value other than an effective value (-1, 0, 1, or more). • The value stored in a device specified in (n1) exceeds the number of characters of (s). • The total of the values stored in devices specified in (n1) and (n2) exceeds the number of characters of (s).

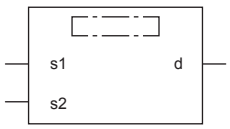
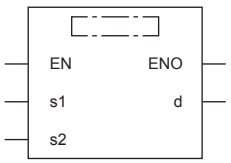
27.4 Link Character Strings

CONCAT(_E)

FX5S FX5UJ FX5U FX5UC

These functions concatenate character strings and output the result.

Ladder diagram, FBD/LD*1		Structured text*1

[Without EN/ENO]
 
 	
	[With EN/ENO]
	 
	
		[Without EN/ENO]
		d:=CONCAT(s1,s2);
		[With EN/ENO]
		d:=CONCAT_E(EN,ENO,s1,s2);

*1 The input variable "s" can be changed in the range of 2 to 28.

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1) to s28(IN28)	Input	Input variable	STRING(255)
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(CONCAT(_E))	Output	Output variable	STRING(255)

Processing details

■Operation processing

- These functions concatenate the character string input to the input variables (s2) to (s28) after the one input to (s1) and output the result from (d).
- When character strings are concatenated, 00H indicating an end of the character string specified by (s1) is ignored and the character string specified by (s2) to (s28) is concatenated.
- When the concatenated character string exceeds 255 bytes, these functions output a character string within 255 bytes.

(s1) "ABCDE" + (s2) "123456" ⇨ (d) "ABCDE123456"

	(s1) High-order byte	(s1) Low-order byte
+0	42H (B)	41H (A)
+1	44H (D)	43H (C)
+2	00H	45H (E)

	(s2) High-order byte	(s2) Low-order byte
+0	32H (2)	31H (1)
+1	34H (4)	33H (3)
+2	36H (6)	35H (5)
+3	00H	

	(d) High-order byte	(d) Low-order byte
+0	42H (B)	41H (A)
+1	44H (D)	43H (C)
+2	31H (1)	45H (E)
+3	33H (3)	32H (2)
+4	35H (5)	34H (4)
+5	00H	36H (6)

- A value input to the input variables (s1) and (s2) to (s28) is the STRING type data value and within the range from 0 to 255 byte(s).

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s1) to (s28) and later, "00H" does not exist.
3406H	The whole concatenated character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.

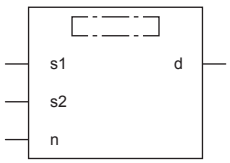
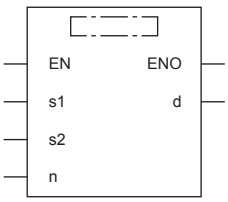
27.5 Inserting Character String

INSERT(_E)

FX5S FX5UJ FX5U FX5UC

These functions insert a character string into another character string and output the result.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
 	
	[With EN/ENO]
	 
	
		[Without EN/ENO]
		d:=INSERT(s1,s2,n);
		[With EN/ENO]
		d:=INSERT_E(EN,ENO,s1,s2,n);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1), s2(IN2)	Input	Input variable	STRING(255)
n(P)	Specification of head character position of a character string to be inserted	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(INSERT(_E))	Output	Output variable	STRING(255)

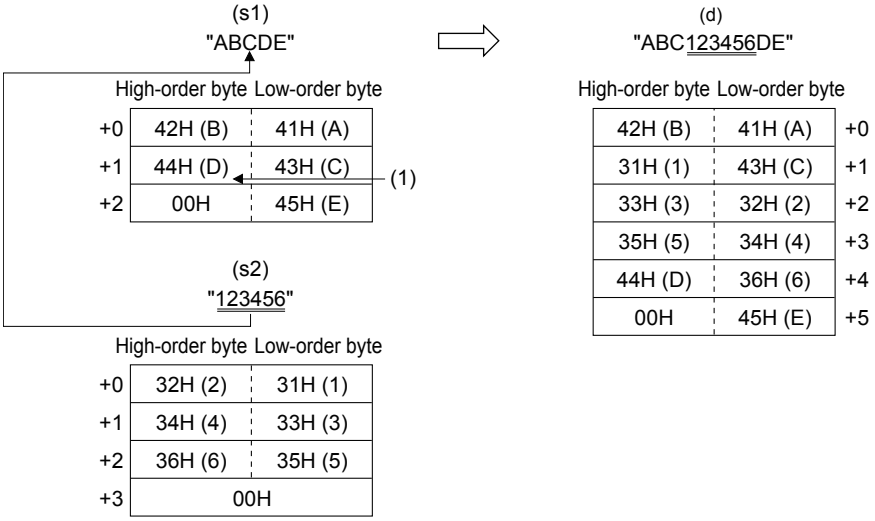
Processing details

■Operation processing

- These functions insert the character string input to (s2) at the (n)th character from the start of the character string input to (s1) (head position of the insertion) and output from (d).
- After the character string specified by (s2) is inserted to the one specified by (s1), 00H indicating an end of the character string specified by (s2) is ignored.
- When the inserted character string exceeds 255 bytes, these functions output a character string within 255 bytes.

Ex.

When the value input to (n) is 4



(1): Head character position of a character string to be inserted n: 4th character

- A value input to (s1) and (s2) is the STRING type data value and within the range from 0 to 255 byte(s).
- A value input to (n) is the INT type data value and within the range from 1 to 255. (However, the value must be within the number of characters of the character string to be input to (s1).)

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
2820H	In the corresponding device range of the device specified by (s1) to (s28) and later, "00H" does not exist.
3406H	The whole concatenated character string cannot be stored in the devices from the device specified by (d) to the last device in the corresponding device range.

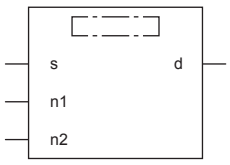
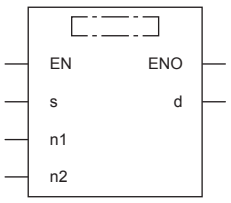
27.6 Deleting Character String

DELETE(_E)

FX5S FX5UJ FX5U FX5UC

These functions delete an arbitrary range of a character string and output the result.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DELETE(s,n1,n2);
		[With EN/ENO]
		d:=DELETE_E(EN,ENO,s,n1,n2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Input	Input variable	STRING(255)
n1(L)	Specification of number of characters to be deleted	Input variable	INT
n2(P)	Specification of head character position of a character string to be deleted	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DELETE(_E))	Output	Output variable	STRING(255)

Processing details

■Operation processing

- These functions delete the data for the specified number of characters from an arbitrary position of a character string input to (s) and output the remaining character strings from (d).
- The value input to (n1) specifies the number of characters to be deleted.
- The value input to (n2) specifies the number of the head character position of a character string to be deleted.

Ex.

When the value input to (n1) and (n2) is 5

(s) "ABCDEF12345" ⇨ (d) "ABCD45"

	High-order byte	Low-order byte	
+0	42H (B)	41H (A)	
+1	44H (D)	43H (C)	
+2	46H (F)	45H (E)	← (1)
+3	32H (2)	31H (1)	
+4	34H (4)	33H (3)	← (2)
+5	00H	35H (5)	

High-order byte	Low-order byte	
42H (B)	41H (A)	+0
44H (D)	43H (C)	+1
35H (5)	34H (4)	+2
00H		+3

(1): Head character position of a character string to be deleted n2: 5th character

(2): Number of characters to be deleted n1: 5

- A value input to (s) is the STRING type data value and within the range from 0 to 255 byte(s).
- A value input to (n1) is the INT type data value and within the range from 0 to 255. (However, the value must be within the number of characters of the character string to be input to (s).)
- A value input to (n2) is the INT type data value and within the range from 1 to 255. (However, the value must be within the number of characters of the character string to be input to (s).)

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
2820H	"00H" is not set to devices from the device number specified by (s) to the end device number of corresponding device.
3405H	The character strings specified by (s) have more than 255 characters.
	The device value specified by (n1) is out of the valid range (0 to 255).
	The device value specified by (n2) is out of the valid range (1 to 255).
	(n1) exceeds the number of characters of a character string specified by (s).
	(n2) exceeds the number of characters of a character string specified by (s).
3406H	The whole deleted character string cannot be stored in the devices from the device specified by (d) to the last device of the target device.

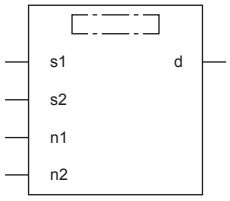
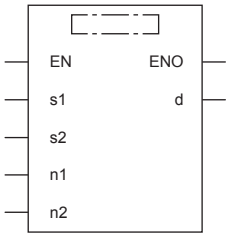
27.7 Replacing Character String

REPLACE(_E)

FX5S FX5UJ FX5U FX5UC

These functions replace an arbitrary range of a character string and output the result.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=REPLACE(s1,s2,n1,n2);
		[With EN/ENO]
		d:=REPLACE_E(EN,ENO,s1,s2,n1,n2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1), s2(IN2)	Input	Input variable	STRING(255)
n1(L)	Specification of number of characters to be replaced	Input variable	INT
n2(P)	Specification of head character position of a character string to be replaced	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(REPLACE(_E))	Output	Output variable	STRING(255)

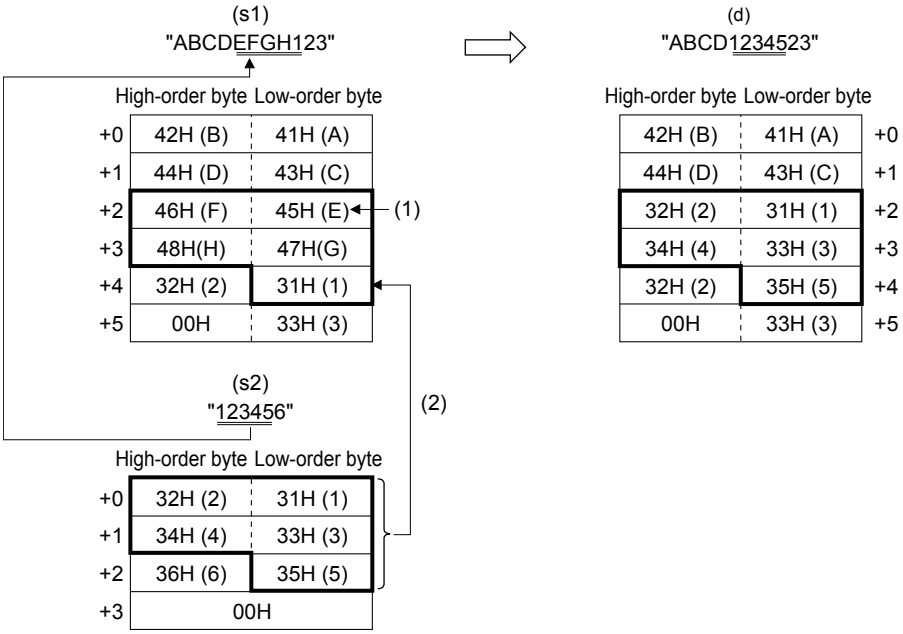
Processing details

■Operation processing

- These functions replace the data for the specified number of characters from an arbitrary position of a character string input to (s1) with a character string input to (s2) and output from (d).
- The value input to (n1) specifies the number of characters to be replaced.
- The value input to (n2) specifies the number of the head character position of a character string to be replaced.

Ex.

When the value input to (n1) and (n2) is 5



(1): Head character position of a character string to be replaced n2: 5th character

(2): Number of characters to be replaced n1: 5

- A value input to (s1) and (s2) is the STRING type data value and within the range from 0 to 255 byte(s).
- A value input to (n1) is the INT type data value and within the range from 0 to 255. (However, the value must be within the number of characters of the character string to be input to (s1).)
- A value input to (n2) is the INT type data value and within the range from 1 to 255. (However, the value must be within the number of characters of the character string to be input to (s1).)

■Operation result

1. Function without EN/ENO

The following table lists the operation results.

Operation result	(d)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	ENO	(d)
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)*1	Indefinite value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
2820H	"00H" is not set to devices from the device number specified by (s1) to the end device number of corresponding device.
	"00H" is not set to devices from the device number specified by (s2) to the end device number of corresponding device.
3405H	The character strings specified by (s1) have more than 255 characters.
	The character strings specified by (s2) have more than 255 characters.
	The device value specified by (n1) is out of the valid range (0 to 255).
	The device value specified by (n2) is out of the valid range (1 to 255).
	(n1) exceeds the number of characters of a character string specified by (s2).
	(n2) exceeds the number of characters of a character string specified by (s1).
3406H	The whole deleted character string cannot be stored in the devices from the device specified by (d) to the last device of the target device.

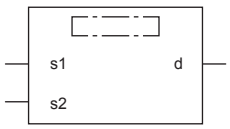
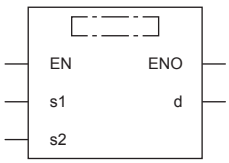
27.8 Searching Character String

FIND(_E)

FX5S FX5UJ FX5U FX5UC

These functions search for a character string and output the result.

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=FIND(s1,s2);
		[With EN/ENO]
		d:=FIND_E(EN,ENO,s1,s2);

Setting data

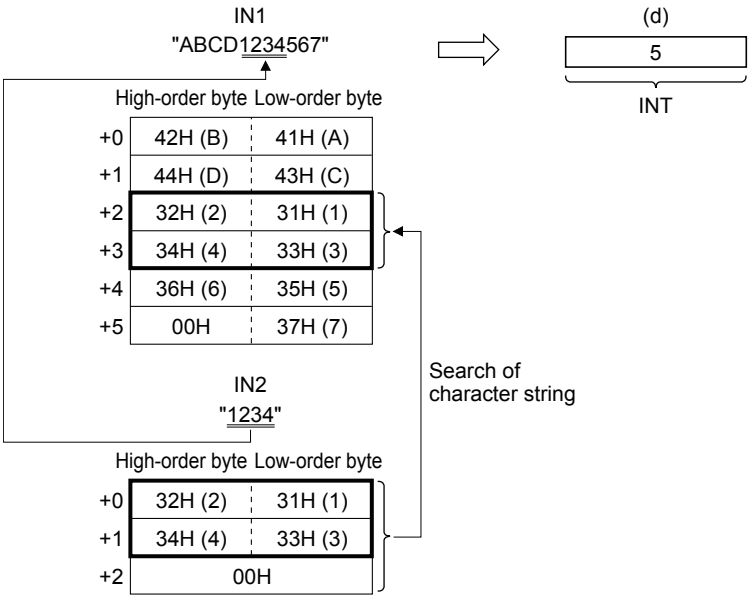
■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1), s2(IN2)	Input	Input variable	STRING(255)
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(FIND(_E))	Output	Output variable	INT

Processing details

■Operation processing

- These functions search for a character string input to (s2) from the start of the character string input to (s1) and output the result from (d).
- This function outputs the head character position of the searched character string detected first as the search result.
- If a character string specified by (s2) cannot be searched from the one specified by (s1), these functions output "0".



- A value input to (s1) and (s2) is the STRING type data value and within the range from 0 to 255 byte(s).

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

28 TIME DATA FUNCTIONS

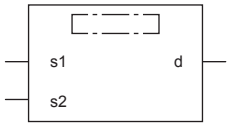
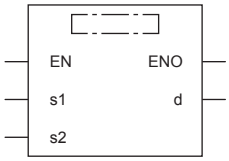
28.1 Addition

ADD_TIME(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the sum of input values (TIME data) ((s1) + (s2)).

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=ADD_TIME(s1,s2);
		[With EN/ENO]
		d:=ADD_TIME_E(EN,ENO,s1,s2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1), s2(IN2)	Input	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(ADD_TIME(_E))	Output	Output variable	TIME

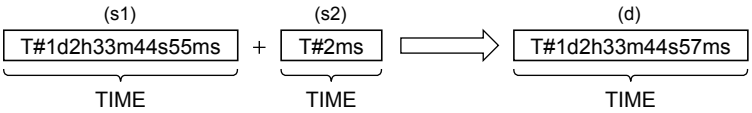
Processing details

■Operation processing

- These functions add the TIME type data input to (s1) and (s2) ((s1) + (s2)), and output the operation result from (d) as TIME type data.

Ex.

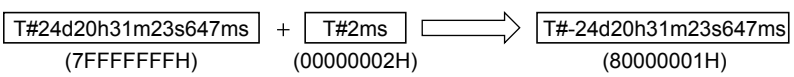
When a value input to (s1) and (s2) is T#1d2h33m44s55ms (1 day 2 hours 33 minutes 44 seconds 55 milliseconds) and T#2ms (2 milliseconds)



- A value input to (s1) and (s2) is the TIME type data value.
- Even if underflow or overflow occurs in the operation result, it is not regarded as an operation error. The data is output from (d) as follows: "ADD_TIME_E" outputs "TRUE" from the output variable ENO.

Ex.

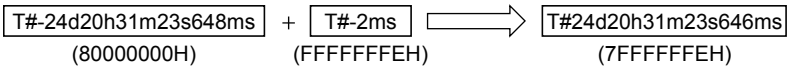
Overflow



The most significant bit becomes 1, and a negative time is output.

Ex.

Underflow



The most significant bit becomes 0, and a positive time is output.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

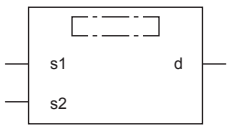
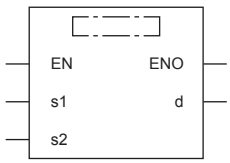
28.2 Subtraction

SUB_TIME(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the difference of input values (TIME data) ((s1) - (s2)).

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=SUB_TIME(s1,s2);
		[With EN/ENO]
		d:=SUB_TIME_E(EN,ENO,s1,s2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1), s2(IN2)	Input	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(SUB_TIME(_E))	Output	Output variable	TIME

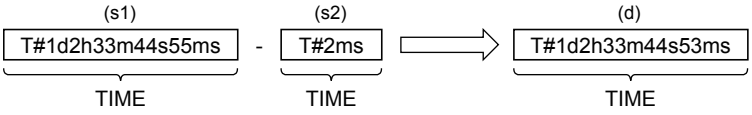
Processing details

■Operation processing

- These functions subtract the TIME type data input to (s1) and (s2) ((s1) - (s2)), and output the operation result from (d) as TIME type data.

Ex.

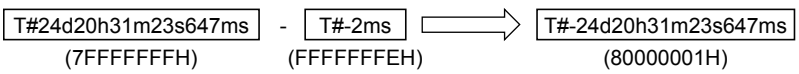
When a value input to (s1) and (s2) is T#1d2h33m44s55ms (1 day 2 hours 33 minutes 44 seconds 55 milliseconds) and T#2ms (2 milliseconds)



- A value input to (s1) and (s2) is the TIME type data value.
- Even if underflow or overflow occurs in the operation result, it is not regarded as an operation error. The data is output from (d) as follows: "SUB_TIME_E" outputs "TRUE" from the output variable ENO.

Ex.

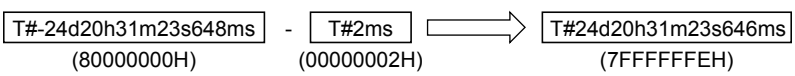
Overflow



The most significant bit becomes 1, and a negative time is output.

Ex.

Underflow



The most significant bit becomes 0, and a positive time is output.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

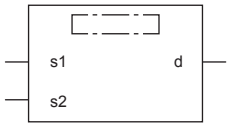
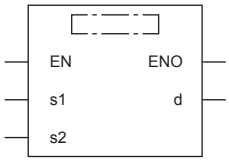
28.3 Multiplication

MUL_TIME(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the multiplication of input values (TIME) ((s1) × (s2)).

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=MUL_TIME(s1,s2);
		[With EN/ENO]
		d:=MUL_TIME_E(EN,ENO,s1,s2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1)	Input	Input variable	TIME
s2(IN2)	Input	Input variable	ANY_NUM
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(MUL_TIME(_E))	Output	Output variable	TIME

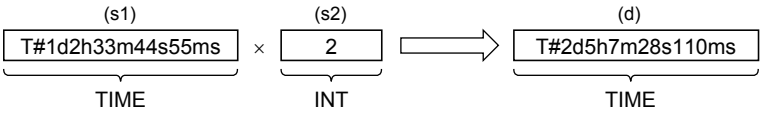
Processing details

■Operation processing

- These functions multiply the TIME type data input to (s1) and (s2) ((s1) × (s2)), and output the operation result from (d) as TIME type data.

Ex.

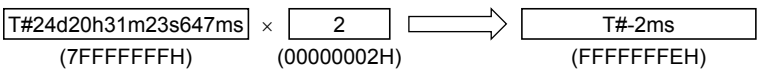
When a value input to (s1) and (s2) is T#1d2h33m44s55ms (1 day 2 hours 33 minutes 44 seconds 55 milliseconds) and 2



- A value input to (s1) is the TIME type data value.
- A value input to (s2) is the INT, DINT, or REAL type.
- Even if underflow or overflow occurs in the operation result, it is not regarded as an operation error. The data is output from (d) as follows: "MUL_TIME_E" outputs "TRUE" from the output variable ENO. (The operation result is the 64-bit data, however, the output data is the time type data with high-order 32 bits deleted.)

Ex.

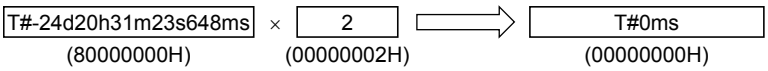
Overflow



The most significant bit becomes 1, and a negative time is output.

Ex.

Underflow



The most significant bit becomes 0, and a positive time is output.

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

There is no operation error.

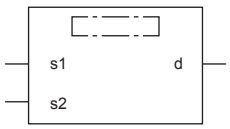
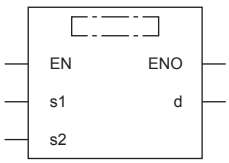
28.4 Division

DIV_TIME(_E)

FX5S FX5UJ FX5U FX5UC

These functions output the quotient of input values (TIME data) ((s1) ÷ (s2)).

Ladder diagram, FBD/LD		Structured text

[Without EN/ENO]
 
	[With EN/ENO]
	 
		[Without EN/ENO]
		d:=DIV_TIME(s1,s2);
		[With EN/ENO]
		d:=DIV_TIME_E(EN,ENO,s1,s2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(IN1)	Input	Input variable	TIME
s2(IN2)	Input	Input variable	ANY_NUM
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(DIV_TIME(_E))	Output	Output variable	TIME

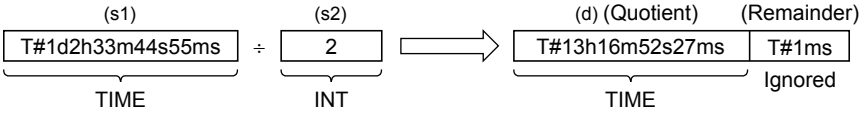
Processing details

■Operation processing

- These functions divide the TIME type data input to (s1) and (s2) ((s1) ÷ (s2)), and output the operation result from (d) as TIME type data. The remainder is ignored.

Ex.

When a value input to (s1) and (s2) is T#1d2h33m44s55ms (1 day 2 hours 33 minutes 44 seconds 55 milliseconds) and 2



- A value input to (s1) is the TIME type data value.
- A value input to (s2) is the INT, DINT, or REAL type. (However, input other than 0 to (s2).)

■Operation result

1. Function without EN/ENO

The operation processing is executed. The operation output value is output from (d).

2. Function with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE*1	Indefinite value

*1 When FALSE is output from ENO, data output from (d) is undefined. In that case, modify a program so that the data output from (d) is not used.

Operation error

Error code (SD0/SD8067)	Description
3400H	A value input to (s2) is 0. (Zero division)

MEMO

PART 6 FUNCTION BLOCKS

This part consists of the following chapters.

29 BISTABLE FUNCTION BLOCKS

30 EDGE DETECTION FUNCTION BLOCKS

31 COUNTER FUNCTION BLOCKS

32 TIMER FUNCTION BLOCKS

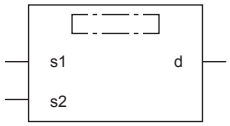
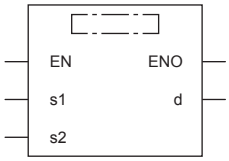
29 BISTABLE FUNCTION BLOCKS

29.1 Bistable Function Blocks (Set Priority)

SR(_E)

FX5S FX5UJ FX5U FX5UC

These function blocks judge two input values and output 1 (TRUE) or 0 (FALSE).

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO] 	[With EN/ENO] 	[Without EN/ENO] SR_1(S1:=s1,R:=s2,Q1:=d); [With EN/ENO] SR_E_1(EN:=EN,ENO:=ENO S1:=s1,R:=s2,Q1:=d);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(S1)	Set instruction	Input variable	BOOL
s2(R)	Reset instruction	Input variable	BOOL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(Q1)	Output	Output variable	BOOL

Processing details

■Operation processing

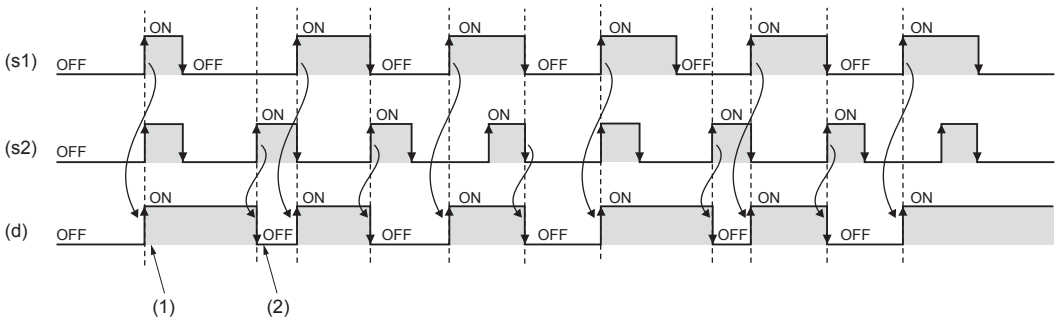
- When (s1) turns ON, (d) is set. If (s2) is turned ON when (s1) is OFF, (d) is reset.
- If (s2) is turned ON when (s1) is ON, (d) is not reset.

■Operation result

1. Function block without EN/ENO

The operation processing is executed. The operation output value is output from (d).

- Timing chart



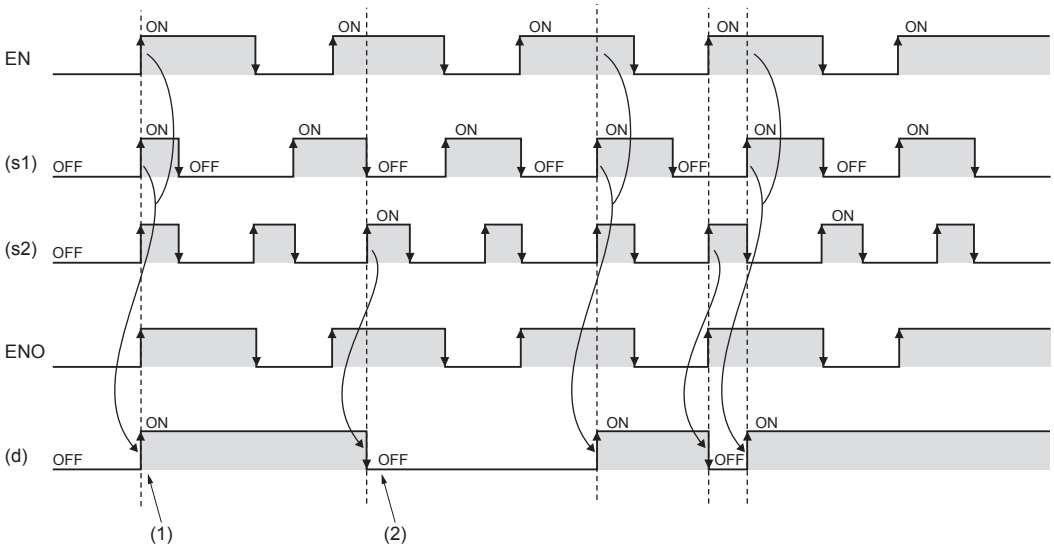
(1): When (s1) is on, (d) is on.
(2): When (s1) is off and (s2) is on, (d) is off.

2. Function block with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE	Previous output value

- Timing chart



(1): When EN and (s1) are on, (d) is on.
(2): When EN and (s2) are on and (s1) is off, (d) is off.

Operation error

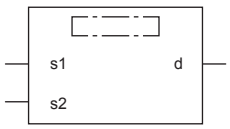
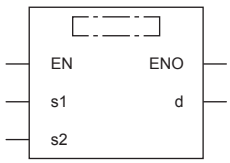
There is no operation error.

29.2 Bistable Function Blocks (Reset Priority)

RS(_E)

FX5S FX5UJ FX5U FX5UC

These function blocks judge two input values and output 1 (TRUE) or 0 (FALSE).

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO] 	[With EN/ENO] 	[Without EN/ENO] RS_1(S:=s1,R1:=s2,Q1:=d); [With EN/ENO] RS_E_1(EN:=EN, ENO:=ENO, S:=s1,R1:=s2,Q1:=d);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(S)	Set instruction	Input variable	BOOL
s2(R1)	Reset instruction	Input variable	BOOL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(Q1)	Output	Output variable	BOOL

Processing details

■Operation processing

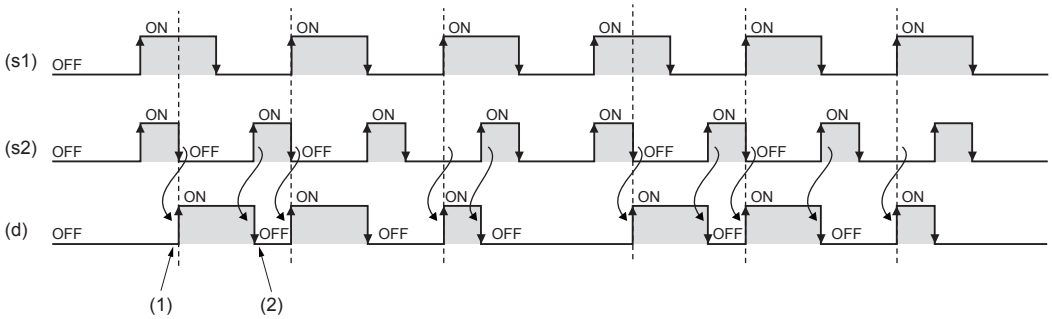
- When (s1) turns ON, (d) is set. When (s2) is turned ON, (d) is reset.
- If (s1) is turned ON when (s2) is ON, (d) is not set.

■Operation result

1. Function block without EN/ENO

The operation processing is executed. The operation output value is output from (d).

- Timing chart



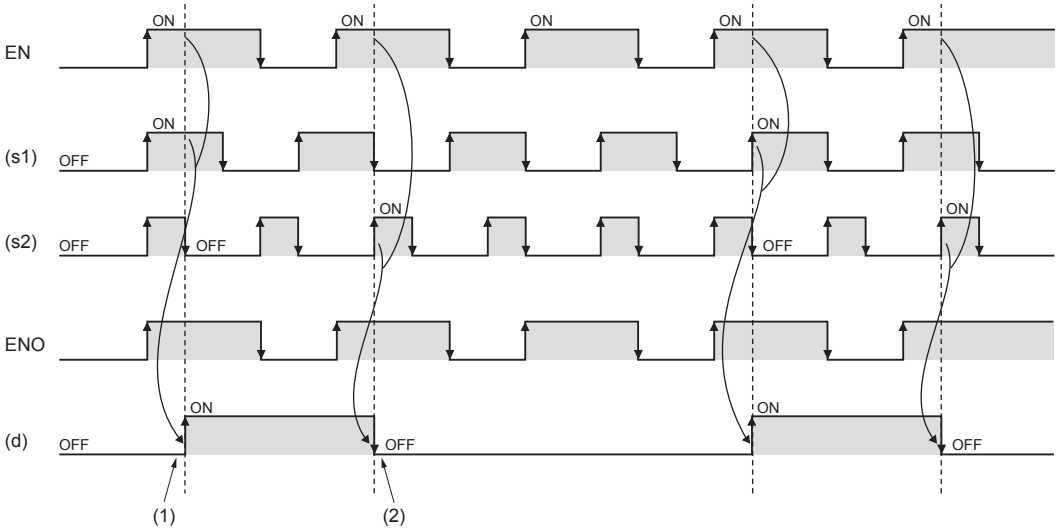
(1): When (s1) is on and (s2) is off, (d) turns on.
(2): When (s2) is on, (d) turns off.

2. Function block with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE	Previous output value

• Timing chart



(1): When EN and (s1) are on and (s2) is off, (d) turns on.
(2): When EN and (s2) are on, (d) turns off.

Operation error

There is no operation error.

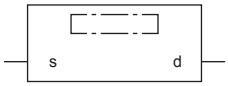
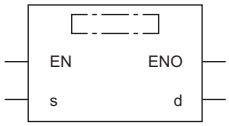
30 EDGE DETECTION FUNCTION BLOCKS

30.1 Rising Edge Detector

R_TRIG(_E)

FX5S FX5UJ FX5U FX5UC

These functions detect the rising edge of a signal, and output a pulse signal.

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO] 	[With EN/ENO] 	[Without EN/ENO] R_TRIG_1(CLK:=s,Q:=d); [With EN/ENO] R_TRIG_E_1(EN:=EN, ENO:=ENO CLK:=s,Q:=d);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(CLK)	Rising edge detector input	Input variable	BOOL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(Q)	Output	Output variable	BOOL

Processing details

■Operation processing

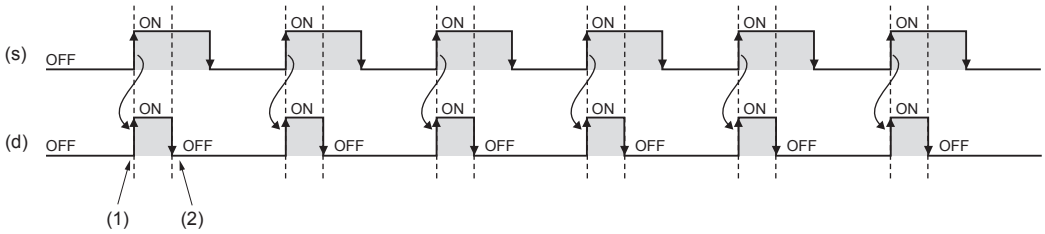
When (s) turns ON, (d) is turned ON only for one scan.

■Operation result

1. Function block without EN/ENO

The operation processing is executed. The operation output value is output from (d).

• Timing chart



(1): On the rising edge of (s), (d) turns on.

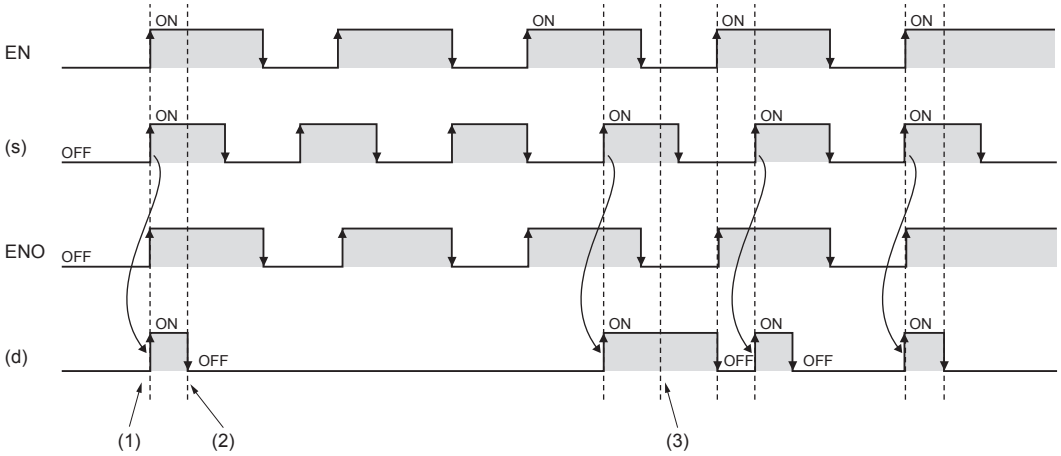
(2): At the next scan, (d) turns off.

2. Function block with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE	Previous output value

- Timing chart



(1): When EN is on and on the rising edge of (s), (d) turns on.
(2): At the next scan, (d) turns off.
(3): When EN is off, (d) holds the output of the previous scan.

Operation error

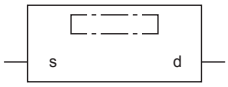
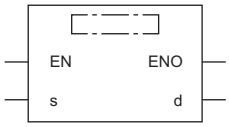
There is no operation error.

30.2 Falling Edge Detector

F_TRIG(_E)

FX5S FX5UJ FX5U FX5UC

These function blocks detect the falling edge of a signal, and output a pulse signal.

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO] 	[With EN/ENO] 	[Without EN/ENO] F_TRIG_1(CLK:=s,Q:=d); [With EN/ENO] F_TRIG_E_1(EN:=EN, ENO:=ENO CLK:=s,Q:=d);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(CLK)	Falling edge detector input	Input variable	BOOL
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d(Q)	Output	Output variable	BOOL

Processing details

■Operation processing

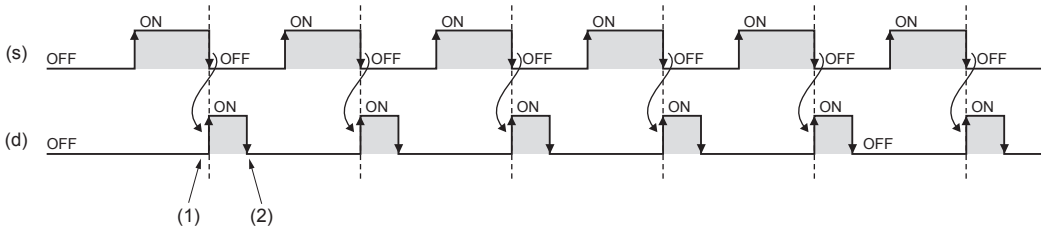
When (s) turns OFF, (d) is turned ON only for one scan.

■Operation result

1. Function block without EN/ENO

The operation processing is executed. The operation output value is output from (d).

• Timing chart



(1): On the falling edge of (s), (d) turns on.

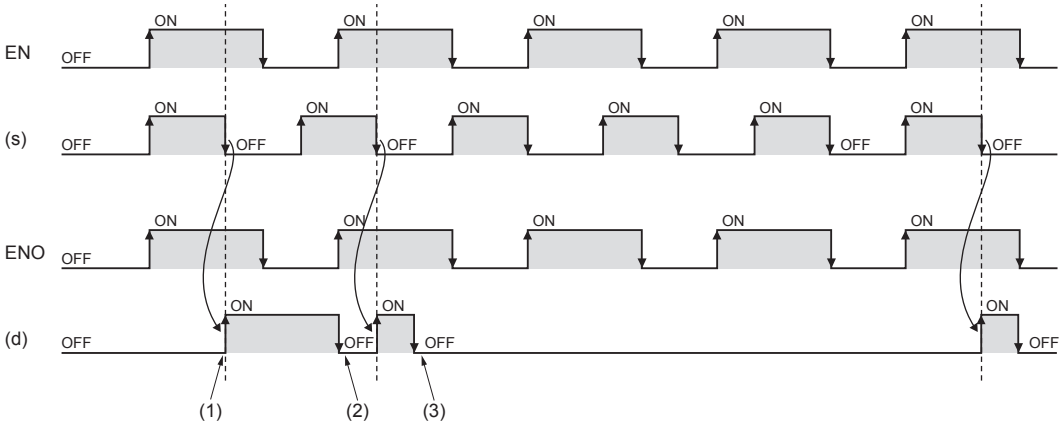
(2): At the next scan, (d) turns off.

2. Function block with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE	Previous output value

- Timing chart



(1): When EN is on and on the falling edge of (s), (d) turns on.
(2): At the next scan, (d) turns off.
(3): When EN is off, (d) holds the output of the previous scan.

Operation error

There is no operation error.

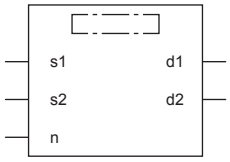
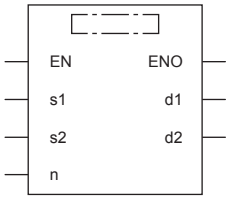
31 COUNTER FUNCTION BLOCKS

31.1 Up Counter

CTU(_E)

FX5S FX5UJ FX5U FX5UC

These function blocks count up the number of times of rising of a signal.

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO]  	[With EN/ENO]  	[Without EN/ENO] CTU_1(CU:=s1,R:=s2,PV:=n,Q:=d1,CV:=d2); [With EN/ENO] CTU_E_1(EN:=EN, ENO:=ENO CU:=s1,R:=s2,PV:=n,Q:=d1,CV:=d2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(CU)	Count signal input	Input variable	BOOL
s2(R)	Count value reset	Input variable	BOOL
n(PV)	Count maximum value	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d1(Q)	Count end	Output variable	BOOL
d2(CV)	Count value	Output variable	INT

Processing details

■Operation processing

1. Count up

- These function blocks count up (add "1" to) the value of (d2) when (s1) turns ON from OFF.
- When the value of (d2) reaches the value of (n) of the counter, (d1) turns ON and the function blocks stop counting up.
- Set the maximum value of the counter for (n). When (s2) is turned ON, (d1) turns OFF and (d2) is set to 0.

2. Count maximum value

The effective setting range of (n) is from 0 to 32767.

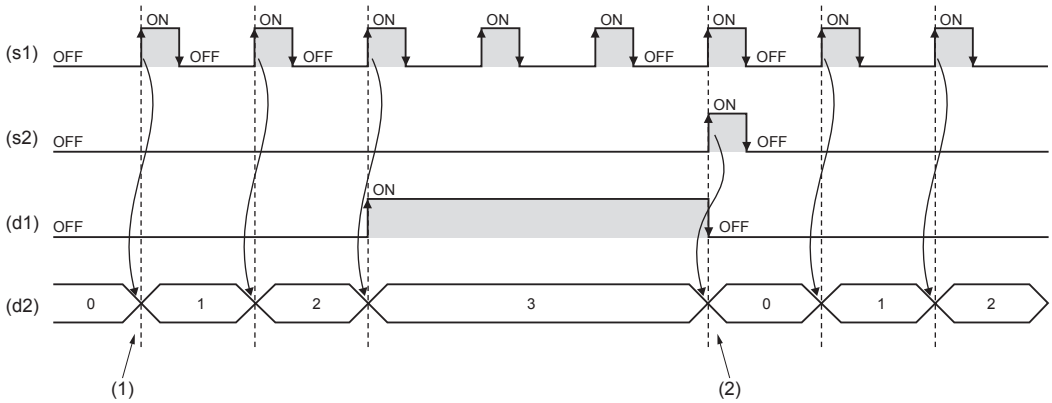
■Operation result

1. Function block without EN/ENO

The operation processing is executed. The operation output value is output from (d1) and (d2).

- Timing chart

When 3 is specified in n



(1): When (s1) is on, (d2) counts up.
(2): When (s2) is on, (d2) is cleared to 0.

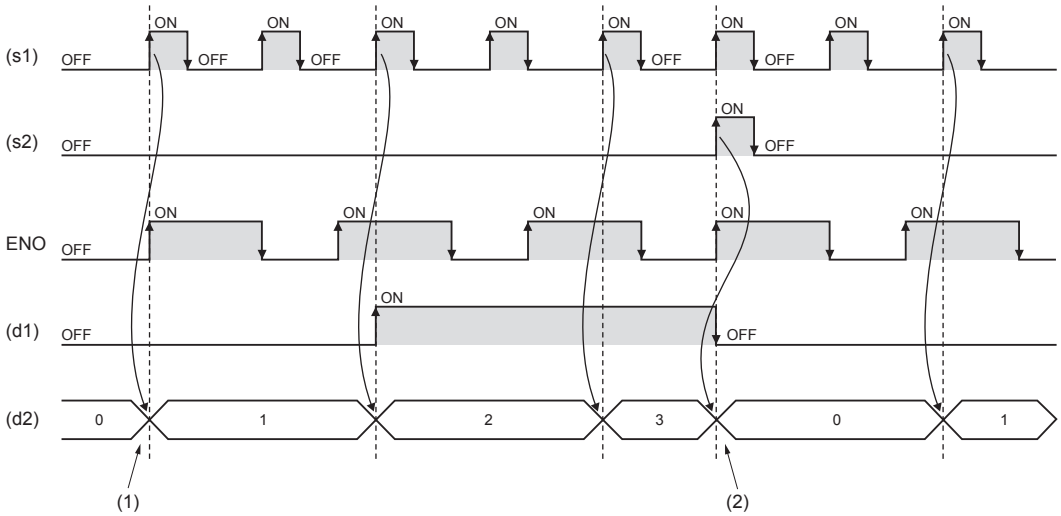
2. Function block with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d1), (d2)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE	Previous output value

- Timing chart

When 3 is specified in n



(1): When EN and (s1) are on, (d2) turns on.
(2): When (s2) is on, (d2) is cleared to 0.

Operation error

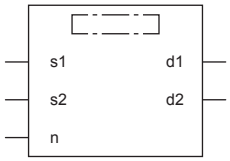
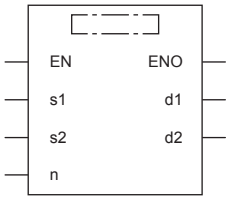
There is no operation error.

31.2 Down Counter

CTD(_E)

FX5S FX5UJ FX5U FX5UC

These function blocks count down the number of times of rising of a signal.

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO] 	[With EN/ENO] 	[Without EN/ENO] CTD_1(CD:=s1,LD:=s2,PV:=n,Q:=d1,CV:=d2); [With EN/ENO] CTD_E_1(EN:=EN, ENO:=ENO CD:=s1,LD:=s2,PV:=n,Q:=d1,CV:=d2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(CD)	Count signal input	Input variable	BOOL
s2(LD)	Count value set	Input variable	BOOL
n(PV)	Count start value	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d1(Q)	Count end	Output variable	BOOL
d2(CV)	Count value	Output variable	INT

Processing details

■Operation processing

1. Count down

- These function blocks count down (subtract "-1" from) the value of (d2) when (s1) turns ON from OFF.
- When the value of (d2) is 0, (d1) turns ON and the function blocks stop counting down.
- Set the count start value for (n). When (s2) is turned ON, (d1) turns OFF and (n) is set for (d2).

2. Count start value

The effective setting range of (n) is from 0 to 32767.

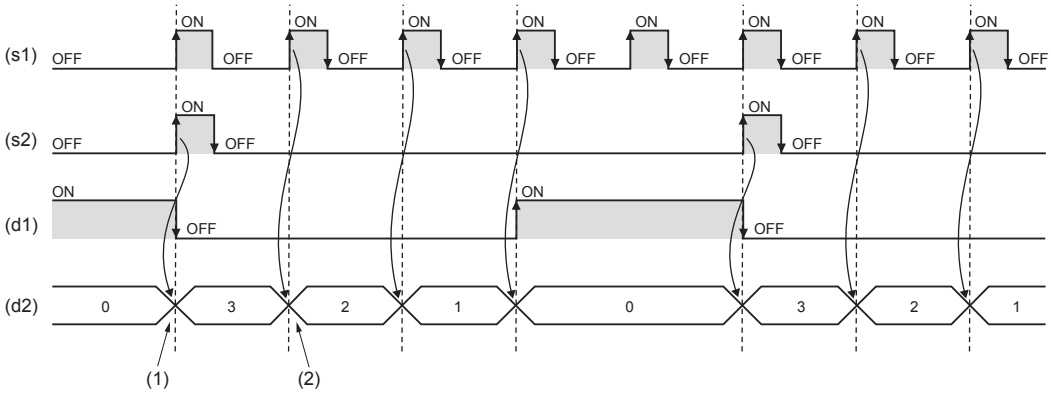
■Operation result

1. Function block without EN/ENO

The operation processing is executed. The operation output value is output from (d1) and (d2).

- Timing chart

When 3 is specified in n



(1): When (s2) is on, (d2) is initialized.
(2): On the rising edge of (s1), (d2) counts down.

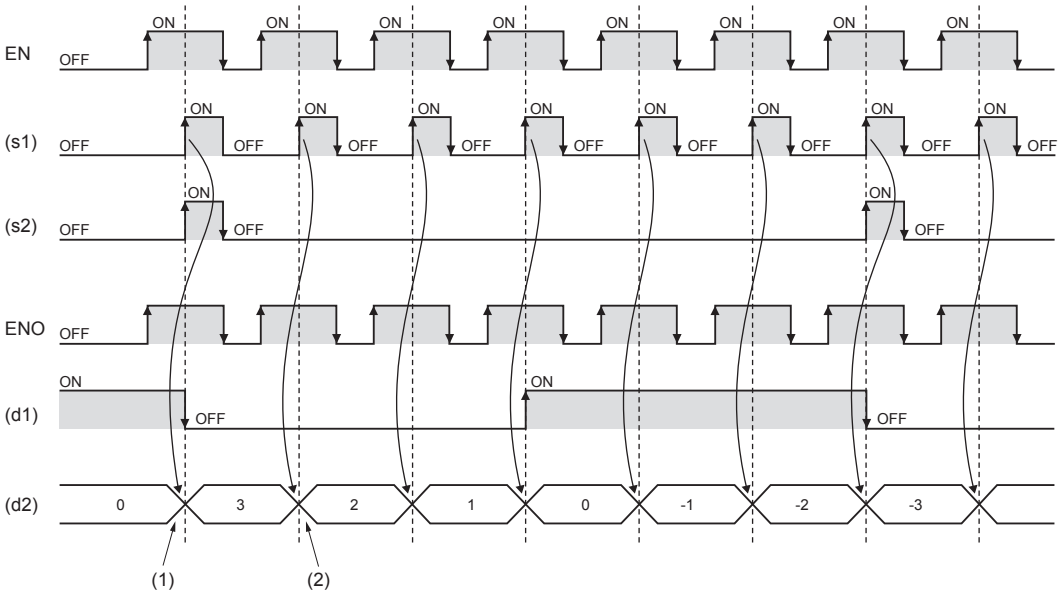
2. Function block with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d1), (d2)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE	Previous output value

- Timing chart

When 3 is specified in n



(1): When EN and (s2) are on, (d2) is initialized.
(2): When EN is on at the rising edge of (s1), (d2) counts down.

Operation error

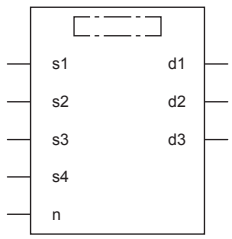
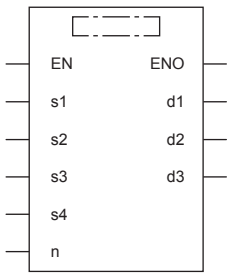
There is no operation error.

31.3 Up-down Counter

CTUD(_E)

FX5S FX5UJ FX5U FX5UC

These functions count up/down the number of times of rising of a signal.

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO] 	[With EN/ENO] 	[Without EN/ENO] CTUD_1(CU:=s1,CD:=s2,R:=s3,LD:=s4,PV:=n,QU:=d1,QD:=d2,CV:=d3); [With EN/ENO] CTUD_E_1(EN:=EN,ENO:=ENO,CU:=s1,CD:=s2,R:=s3,LD:=s4,PV:=n,QU:=d1,QD:=d2,CV:=d3);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s1(CU)	Count up signal input	Input variable	BOOL
s2(CD)	Count down signal input	Input variable	BOOL
s3(R)	Count value reset	Input variable	BOOL
s4(LD)	Count value set	Input variable	BOOL
n(PV)	Count maximum value/start value	Input variable	INT
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d1(QU)	Count up end	Output variable	BOOL
d2(QD)	Count down end	Output variable	BOOL
d3(CV)	Current count value	Output variable	INT

Processing details

■Operation processing

1. Count up

- These function blocks count up (add "1" to) the value of (d3) when (s1) turns ON from OFF.
- When the value of (d3) reaches the value of (n), (d1) turns ON and the function blocks stop counting up.
- Set the maximum value of the counter for (n). When (s3) is turned ON, (d1) turns OFF and (d3) is set to 0.

2. Count down

- These function blocks count down (subtract "-1" from) the value of (d3) when (s2) turns ON from OFF.
- When the value of (d3) is 0, (d2) turns ON and the function blocks stop counting down.
- Set the counter start value for (n). When (s4) is turned ON, (d2) turns OFF and (n) is set for (d3).

3. Count maximum value/start value

The effective setting range of (n) is from 0 to 32767.

4. Others

- When (s1) and (s2) are simultaneously turned ON from OFF, (s1) is prioritized and the function blocks count up (add "1" to) the value of (d3).
- When (s3) and (s4) are simultaneously turned ON, (s3) is prioritized and the value of (d3) is set to 0.

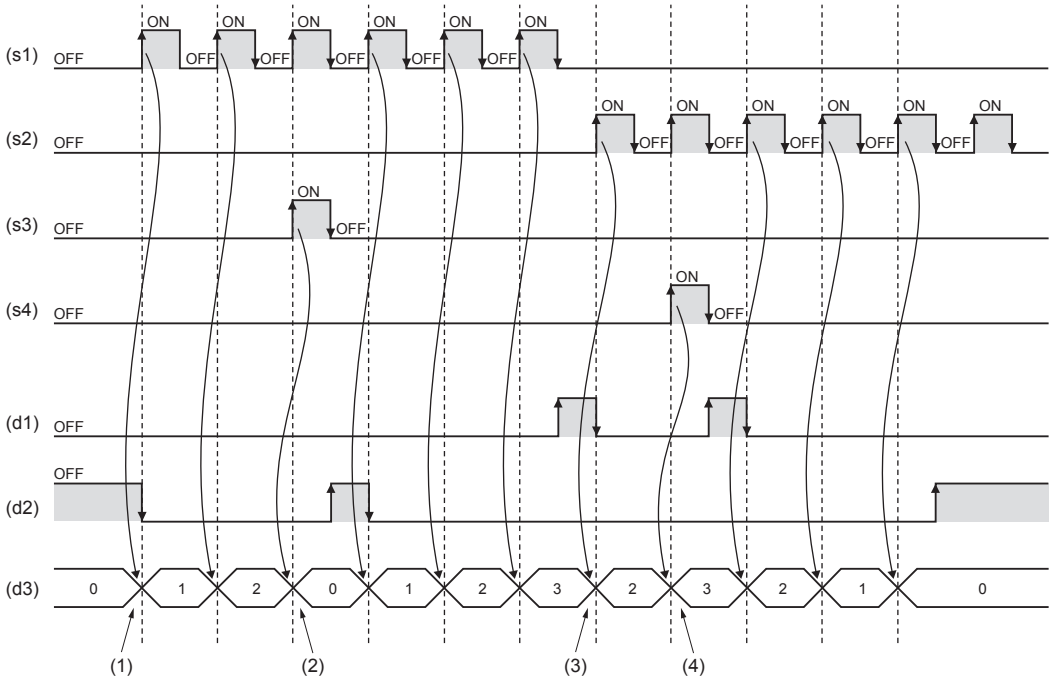
■Operation result

1. Function block without EN/ENO

The operation processing is executed. The operation output value is output from (d1), (d2), and (d3).

- Timing chart

When 3 is specified in n



(1): When (s1) turns on from off, (d3) counts up.
(2): When (s3) turns on from off, (d3) is initialized.
(3): When (s2) turns on from off, (d3) counts down.
(4): When (s4) turns on from off, (d3) is initialized.

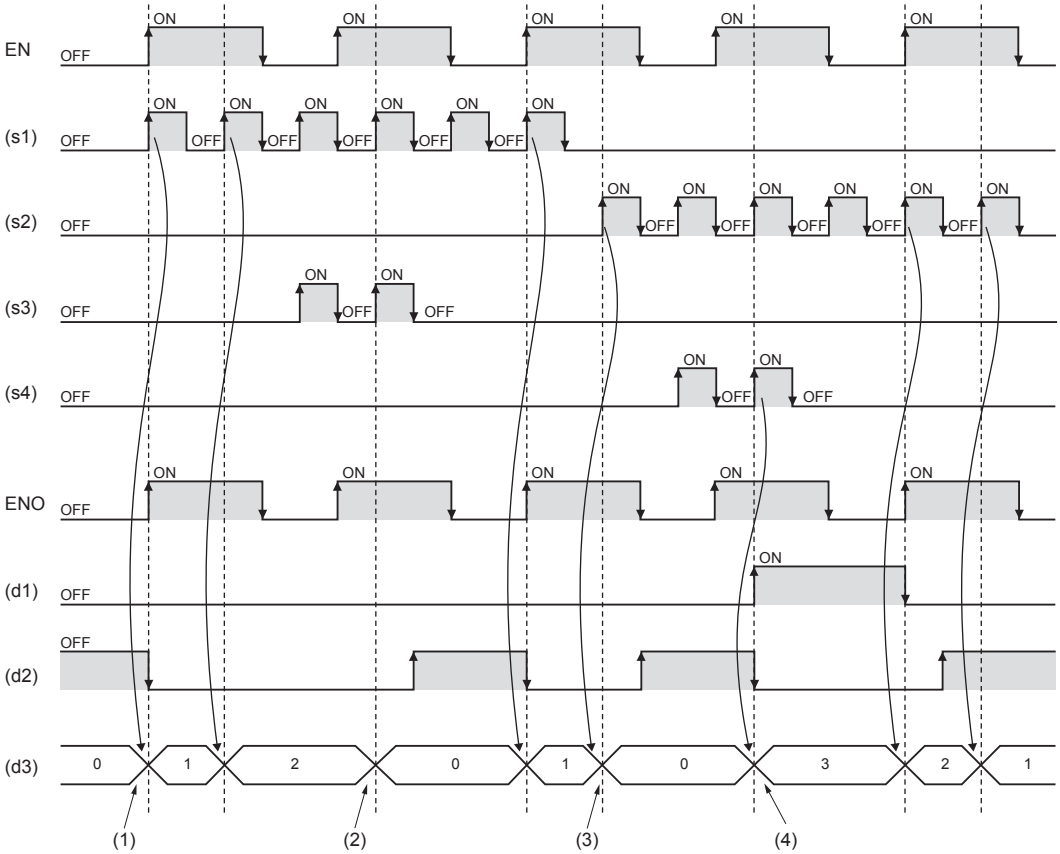
2. Function block with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d1), (d2), (d3)**
TRUE (Executes operation)	TRUE	Operation output value
FALSE (Stops operation)	FALSE	Previous output value

- Timing chart

When 3 is specified in n



(1): When EN is on and (s1) turns on from off, (d3) is counts up.
(2): When EN is on and (s3) turns on from off, (d3) is cleared to 0.
(3): When EN is on and (s2) turns on from off, (d3) is counts down.
(4): When EN is on and (s4) turns on from off, (d3) is initialized.

Operation error

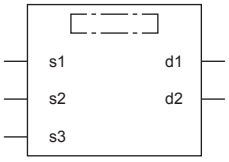
There is no operation error.

31.4 Counter Function Block

COUNTER_FB_M

FX5S FX5UJ FX5U FX5UC

When the execution condition is established, this function block starts counting up.

Ladder diagram, FBD/LD	Structured text
	COUNTER_FB_M_1(Coil:=s1,Preset:=s2,ValueIn:=s3,ValueOut:=d1,Status:=d2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
s1(Coil)	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s2(Preset)	Counter set value	Input variable	INT
s3(ValueIn)	Counter initial value	Input variable	INT
d1(ValueOut)	Counter current value	Output variable	ANY16
d2(Status)	Output	Output variable	BOOL

Processing details

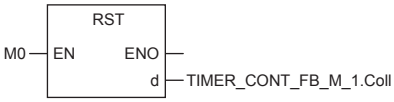
■Operation processing

- The counter starts counting when detecting the rising edge (from OFF to ON) of (s1). It does not start counting if (s1) remains ON. The counting is started from the value of (s3). When the count value reaches the value of (s2), (d2) turns ON. The current count value is stored in (d1).
- A value in the range of 0 to 32767 can be specified for (s2).
- A value in the range of -32768 to 32767 can be specified for (s3). However, when a negative value is specified, the initial value is set to 0.
- To reset the current value of the counter (d1), reset (s1) of FB directly.

Ex.

When the label name is TIMER_CONT_FB_M_1

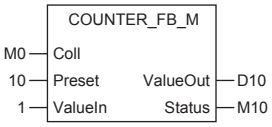
[Ladder]



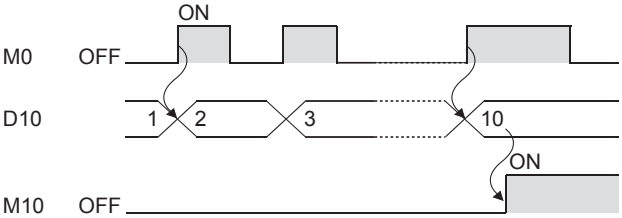
[ST]

RST(M0,TIMER_CONT_FB_M_1.Coil)

[Ladder example]



[Timing chart]



Operation error

There is no error.

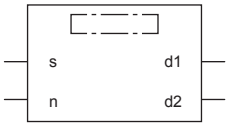
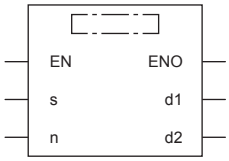
32 TIMER FUNCTION BLOCKS

32.1 Pulse Timer

TP(_E), TP_10(_E)

FX5S FX5UJ FX5U FX5UC

These function blocks keep on a signal for specified duration.

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO] 	[With EN/ENO] 	[Without EN/ENO] TP_1(IN:=s,PT:=n,Q:=d1,ET:=d2); TP_10_1(IN:=s,PT:=n,Q:=d1,ET:=d2); [With EN/ENO] TP_E_1(EN:=EN,ENO:=ENO,IN:=s,PT:=n,Q:=d1,ET:=d2); TP_10_E_1(EN:=EN,ENO:=ENO,IN:=s,PT:=n,Q:=d1,ET:=d2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Output start	Input variable	BOOL
n(PT)	Output time setting value	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d1(Q)	Output	Output variable	BOOL
d2(ET)	Elapsed time	Output variable	TIME

Processing details

■Operation processing

1. Output

- When (s) turns on, (d1) turns on for the time specified by (n). The elapsed time from when (d1) is turned on is set for (d2).
- The timer device is used for counting the elapsed time.

2. End of output

- (d1) turns off when the elapsed time reaches the set value.
- If (s) is off after (d1) is turned off, the elapsed time is reset.
- Even if (s) is turned off when (d1) is on, (d1) is not turned off.

3. Setting of output time

The effective setting range of (n) is from 0 ms to 32767 ms. For TP(_E), the output time value of (n) is set in units of 100 ms or more. For TP_10(_E), it is set in units of 10 ms or more.

The value of when (d1) is turned on from off (rising edge) is used for the setting value of (n). If the value of (n) is changed while (d1) is on, the changed value becomes valid at the next output start.

■Operation result

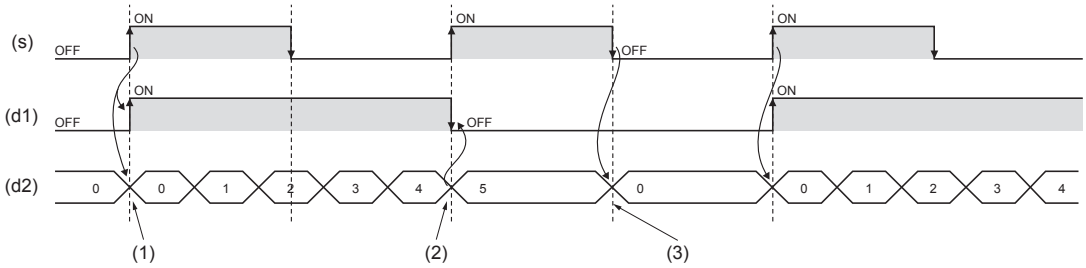
1. Function block without EN/ENO

The following table lists the operation results.

Operation result	(d1), (d2)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

- Timing chart

When T#5s (5 seconds) is specified in n



(1): When (s) is on, (d1) turns on. When (s) is on, the time measurement of (d2) starts.
(2): When (d2) reaches the time specified by n, (d1) turns off.
(3): When (s) and (d1) are off, (d2) is initialized.

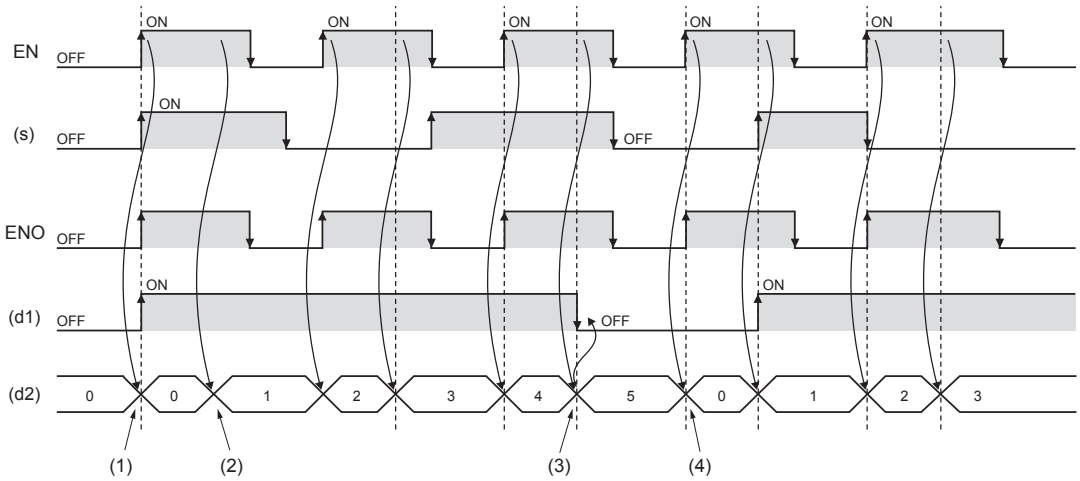
2. Function block with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d1), (d2)**
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)	Indefinite value
FALSE (Stops operation)	FALSE	Previous output value

- Timing chart

When T#5s (5 seconds) is specified in n



(1): When EN and (s) are on, (d1) turns on. When EN and (s) are on, the time measurement of (d2) starts.
(2): After the time measurement starts, the measurement time counts up when EN is on.
(3): When EN is on and (s) and (d1) are off, (d2) is initialized.
(4): When (d2) reaches the time specified by n, (d1) turns off.

Precautions

If the set value of the output time exceeds the effective range, the program will not operate correctly.

Operation error

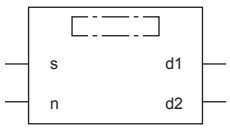
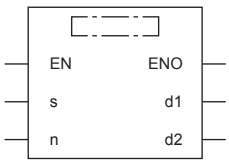
There is no operation error.

32.2 On-delay Timer

TON(_E), TON_10(_E)

FX5S FX5UJ FX5U FX5UC

These function blocks turn on a signal after a specified time.

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO] 	[With EN/ENO] 	[Without EN/ENO] TON_1(IN:=s,PT:=n,Q:=d1,ET:=d2); TON_10_1(IN:=s,PT:=n,Q:=d1,ET:=d2); [With EN/ENO] TON_E_1(EN:=EN,ENO:=ENO,IN:=s,PT:=n,Q:=d1,ET:=d2); TON_10_E_1(EN:=EN,ENO:=ENO,IN:=s,PT:=n,Q:=d1,ET:=d2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Time measurement	Input variable	BOOL
n(PT)	Delay time setting value	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d1(Q)	Output	Output variable	BOOL
d2(ET)	Elapsed time	Output variable	TIME

Processing details

■Operation processing

1. Output

- When (s) turns on, (d1) is turned on after the time specified by (n). The delay elapsed time from when (s) is turned on is set for (d2).
- When (s) is turned off, (d1) is turned off and the delay elapsed time is reset.
- The timer device is used for counting the elapsed time.

2. Setting of delay time

The effective setting range of (n) is from 0 ms to 32767 ms. For TON(_E), the output time value of (n) is set in units of 100 ms or more. For TON_10(_E), it is set in units of 10 ms or more.

The value of when (s) is turned on from off (rising edge) is used for the setting value of (n). If the value of (n) is changed while (s) is on, the changed value becomes valid on the next rising edge of (s).

■Operation result

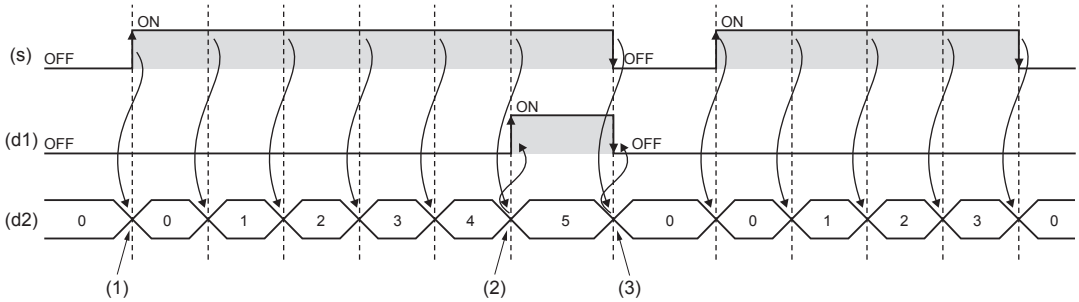
1. Function block without EN/ENO

The following table lists the operation results.

Operation result	(d1), (d2)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

- Timing chart

When T#5s (5 seconds) is specified in n



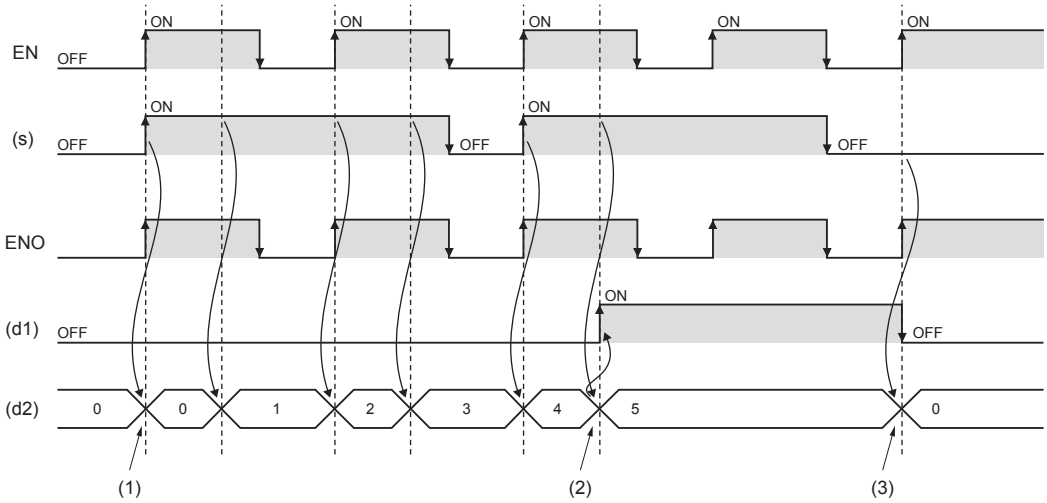
(1): When (s) is on, the time measurement of (d2) starts.
(2): When (d2) reaches the time specified by n, (d1) turns on.
(3): On the falling edge of (s), (d2) is reset.

2. Function block with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d1), (d2)**
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)	Previous output value
FALSE (Stops operation)	FALSE	Previous output value

- Timing chart



(1): When EN and (s) are on, the time measurement of (d2) starts.
(2): When (d2) reaches the time specified by n, (d1) turns on.
(3): When EN is on and (s) is off, (d1) turns off and (d2) is reset.

Precautions

If the set value of the output time exceeds the effective range, the program will not operate correctly.

Operation error

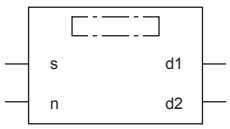
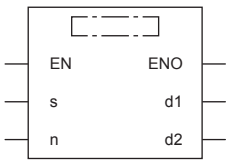
There is no operation error.

32.3 Off-delay Timer

TOF(_E), TOF_10(_E)

FX5S FX5UJ FX5U FX5UC

These function blocks turn off a signal after a specified time.

Ladder diagram, FBD/LD		Structured text
[Without EN/ENO] 	[With EN/ENO] 	[Without EN/ENO] TOF_1(IN:=s,PT:=n,Q:=d1,ET:=d2); TOF_10_1(IN:=s,PT:=n,Q:=d1,ET:=d2); [With EN/ENO] TOF_E_1(EN:=EN,ENO:=ENO,IN:=s,PT:=n,Q:=d1,ET:=d2); TOF_10_E_1(EN:=EN,ENO:=ENO,IN:=s,PT:=n,Q:=d1,ET:=d2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
EN	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s(IN)	Time measurement	Input variable	BOOL
n(PT)	Delay time setting value	Input variable	TIME
ENO	Output status (TRUE: Normal, FALSE: Abnormal)	Output variable	BOOL
d1(Q)	Output	Output variable	BOOL
d2(ET)	Elapsed time	Output variable	TIME

Processing details

■Operation processing

1. Output

- When (s) turns on, (d1) is turned on.
- When (s) turns off from on, (d1) is turned off after the time specified by (n). The elapsed time until (d1) is turned off is set for (d2).
- The timer device is used for counting the elapsed time.

2. Setting of delay time

The effective setting range of (n) is from 0 ms to 32767 ms. For TOF(_E), the output time value of (n) is set in units of 100 ms or more. For TOF_10(_E), it is set in units of 10 ms or more.

The value of when (s) is turned off from on (falling edge) is used for the setting value of (n). If the value of (n) is changed while (s) is off, the changed value becomes valid at the next falling edge of (s).

■Operation result

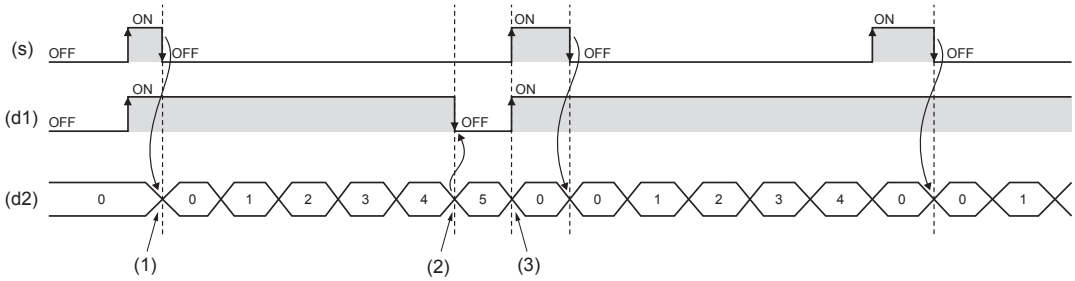
1. Function block without EN/ENO

The following table lists the operation results.

Operation result	(d1), (d2)
No operation error occurred	Operation output value
An operation error occurred	Indefinite value

- Timing chart

  When T#5s (5 seconds) is specified in n



(1): When (s) is off, the time measurement of (d2) starts.
(2): When (d2) reaches the time specified by n, (d1) turns off.
(3): When (s) is on, (d2) is reset.

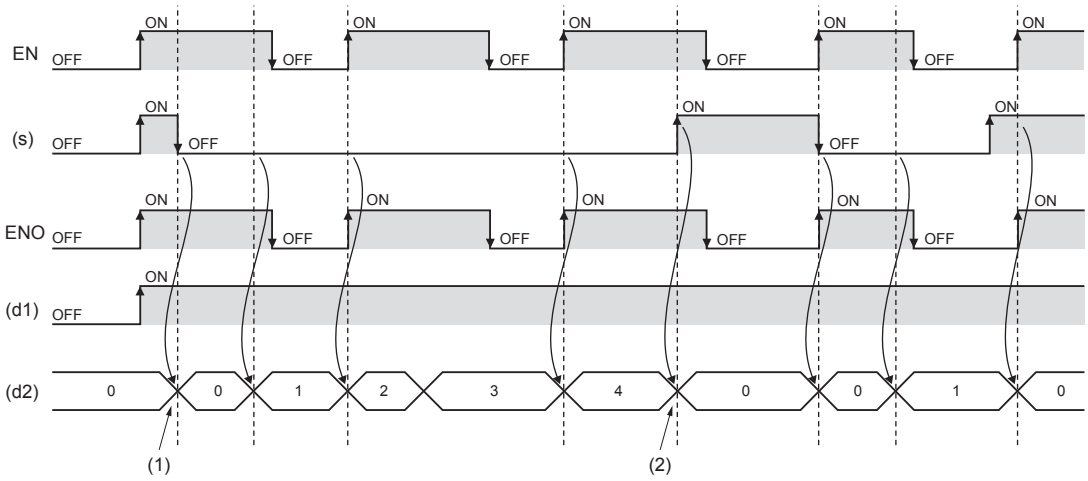
2. Function block with EN/ENO

The following table lists the execution conditions and operation results.

Execution condition	Operation result	
EN	**ENO**	**(d)**
TRUE (Executes operation)	TRUE (Operation error did not occur)	Operation output value
	FALSE (Operation error occurred)	Previous output value
FALSE (Stops operation)	FALSE	Previous output value

- Timing chart

  When T#5s (5 seconds) is specified in n



(1): When EN is on and (s) is off, the time measurement of (d2) starts.
(2): When EN and (s) are on, (d2) is reset.

Precautions

If the set value of the output time exceeds the effective range, the program will not operate correctly.

Operation error

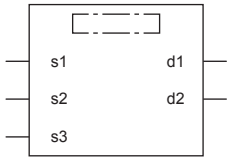
There is no operation error.

32.4 Timer Function Blocks

TIMER_□_M

FX5S FX5UJ FX5U FX5UC

When the execution condition is established, these function blocks start the timer count to the set time.

Ladder diagram, FBD/LD	Structured text
 (□ indicates TIMER_1_FB_M, TIMER_10_FB_M, TIMER_100_FB_M, TIMER_CONT_FB_M, TIMER_CONTHS_FB_M.)	TIMER_1_FB_M_1(Coil:=s1,Preset:=s2,ValueIn:=s3,ValueOut:=d1,Status:=d2); TIMER_10_FB_M_1(Coil:=s1,Preset:=s2,ValueIn:=s3,ValueOut:=d1,Status:=d2); TIMER_100_FB_M_1(Coil:=s1,Preset:=s2,ValueIn:=s3,ValueOut:=d1,Status:=d2); TIMER_CONT_FB_M_1(Coil:=s1,Preset:=s2,ValueIn:=s3,ValueOut:=d1,Status:=d2); TIMER_CONTHS_FB_M_1(Coil:=s1,Preset:=s2,ValueIn:=s3,ValueOut:=d1,Status:=d2);

Setting data

■Descriptions, types, and data types

Argument	Description	Type	Data type
s1(Coil)	Execution condition (TRUE: Execution, FALSE: Stop)	Input variable	BOOL
s2(Preset)	Timer set value	Input variable	INT
s3(ValueIn)	Timer initial value	Input variable	INT
d1(ValueOut)	Timer current value	Output variable	ANY16
d2(Status)	Output	Output variable	BOOL

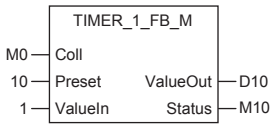
Processing details

■TIMER_1_FB_M

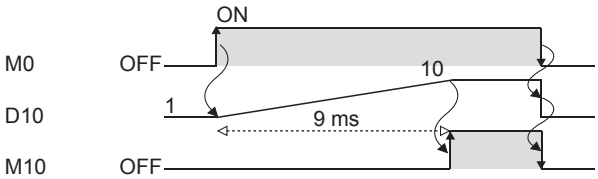
- When the execution condition of (s1) turns on, counting the current value starts. The timer starts counting from "(s3) × 1 ms". When it counts up to "(s2) × 1 ms", (d2) turns on. The current measurement value is output into (d1).
- When the execution condition of (s1) turns off, the current value is reset to (s3) and (d2) turns off.
- A value in the range of 0 to 32767 can be specified for (s2).
- A value in the range of -32768 to 32767 can be specified for (s3). However, when a negative value is specified, the initial value is set to 0.

Ex.

[Ladder example]



[Timing chart]

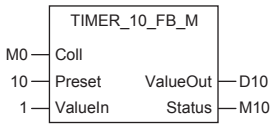


■TIMER_10_FB_M

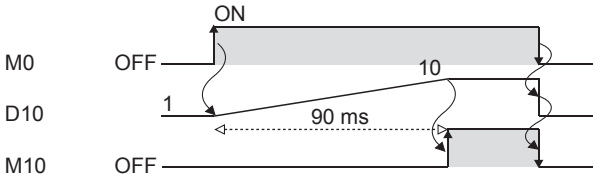
- When the execution condition of (s1) turns on, counting the current value starts. The timer starts counting from "(s3) × 10 ms". When it counts up to "(s2) × 10 ms", (d2) turns on. The current measurement value is output into (d1).
- When the execution condition of (s1) turns off, the current value is reset to (s3) and (d2) turns off.
- A value in the range of 0 to 32767 can be specified for (s2).
- A value in the range of -32768 to 32767 can be specified for (s3). However, when a negative value is specified, the initial value is set to 0.

Ex.

[Ladder example]



[Timing chart]

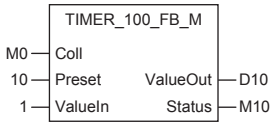


■TIMER_100_FB_M

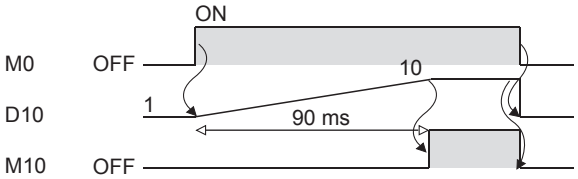
- When the execution condition of (s1) turns on, counting the current value starts. The timer starts counting from "(s3) × 100 ms". When it counts up to "(s2) × 100 ms", (d2) turns on. The current measurement value is output into (d1).
- When the execution condition of (s1) turns off, the current value is reset to (s3) and (d2) turns off.
- A value in the range of 0 to 32767 can be specified for (s2).
- A value in the range of -32768 to 32767 can be specified for (s3). However, when a negative value is specified, the initial value is set to 0.

Ex.

[Ladder example]



[Timing chart]



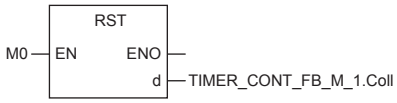
■TIMER_CONT_FB_M

- This is a retentive timer that counts the time when the variable is on. When the execution condition of (s1) turns on, counting the current value starts. There are two retentive timers: low-speed (TIMER_CONT_FB_M) and highspeed (TIMER_CONTHS_FB_M) retentive timers.
- The timer starts counting from "(s3) × 100 ms"(or 1ms if the high-speed retentive timer is used). When it counts up to "(s2) × 100 ms"(or 1ms if the high-speed retentive timer is used), (d2) turns on. The current measurement value is output into (d1).
- The on/off status of (d1) and (d2) is maintained even if the execution condition of (s1) turns off. When the execution condition of (s1) turns on, the timer resume counting from the measurement it holds.
- A value in the range of 0 to 32767 can be specified for (s2).
- A value in the range of -32768 to 32767 can be specified for (s3). However, when a negative value is specified, the initial value is set to 0.
- To reset (d1) of the retentive timer, reset (s1) of FB directly.

Ex.

For label name TIMER_CONT_FB_M_1

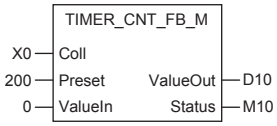
[Ladder program]



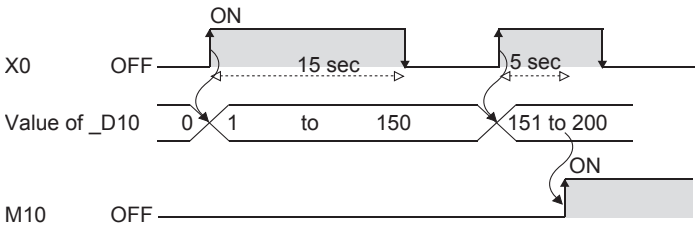
[ST]

RST(M0,TIMER_CONT_FB_M_1.Coil)

[Ladder example]



[Timing chart]



Operation error

There is no error.

APPENDIX

Appendix 1 Instruction Processing Time

Instruction processing time (High-speed instruction)

The following table shows instruction processing time of each instruction.
Because the instruction processing time slightly varies depending on the contents of the source and destination, refer to the values in the table as a reference.

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)
LD	—	0.084	—	0.034	—	0.034	—	0.350	—
LDI	—	0.084	—	0.034	—	0.034	—	0.779	—
AND	—	0.084	—	0.042	—	0.042	—	0.350	—
ANI	—	0.097	—	0.042	—	0.042	—	0.781	—
OR	—	0.091	—	0.042	—	0.042	—	0.781	—
ORI	—	0.091	—	0.042	—	0.042	—	0.781	—
LDP	—	0.117	—	0.084	—	0.084	—	0.956	—
LDF	—	0.084	—	0.084	—	0.084	—	0.956	—
ANDP	—	0.107	—	0.092	—	0.092	—	1.096	—
ANDF	—	0.094	—	0.092	—	0.092	—	1.096	—
ORP	—	0.127	—	0.092	—	0.092	—	1.096	—
ORF	—	0.127	—	0.092	—	0.092	—	1.096	—
LDPI	—	0.124	—	0.084	—	0.084	—	0.956	—
LDFI	—	0.124	—	0.084	—	0.084	—	0.956	—
ANDPI	—	0.127	—	0.092	—	0.092	—	1.096	—
ANDFI	—	0.100	—	0.100	—	0.100	—	1.169	—
ORPI	—	0.127	—	0.092	—	0.092	—	1.096	—
ORFI	—	0.100	—	0.100	—	0.100	—	1.169	—
ANB	—	0.084	—	0.025	—	0.025	—	0.341	—
ORB	—	0.084	—	0.025	—	0.025	—	0.666	—
MPS	—	0.084	—	0.017	—	0.017	—	0.333	—
MRD	—	0.084	—	0.009	—	0.009	—	0.333	—
MPP	—	0.084	—	0.009	—	0.009	—	0.333	—
INV	—	0.092	—	0.017	—	0.017	—	0.333	—
MEP	—	0.100	—	0.042	—	0.042	—	0.700	—
MEF	—	0.100	—	0.050	—	0.050	—	0.700	—
OUT	—	0.117	—	0.024	—	0.024	—	0.350	—
OUT C	—	0.384	0.354	0.341	0.275	0.341	0.275	3.653	3.397
OUT T/ST	—	0.614	0.397	0.509	0.317	0.509	0.317	4.987	3.721
SET	—	0.127	0.110	0.050	0.017	0.050	0.017	0.560	0.674
RST	—	0.127	0.110	0.050	0.017	0.050	0.017	0.560	0.674
ALT	—	0.130	0.111	0.050	0.017	0.050	0.017	0.560	0.674
SFR	(n)=K1	0.190	0.120	0.125	0.017	0.125	0.017	1.552	0.674
	(n)=K15	0.190	0.120	0.125	0.017	0.125	0.017	1.552	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)
SFL	(n)=K1	0.190	0.124	0.125	0.017	0.125	0.017	1.552	0.674
	(n)=K15	0.190	0.123	0.125	0.017	0.125	0.017	1.552	0.674
LD=	—	0.114	—	0.042	—	0.042	—	0.781	—
LD<>	—	0.114	—	0.042	—	0.042	—	0.781	—
LD>	—	0.117	—	0.042	—	0.042	—	0.781	—
LD<=	—	0.117	—	0.042	—	0.042	—	0.781	—
LD<	—	0.117	—	0.042	—	0.042	—	0.781	—
LD>=	—	0.117	—	0.042	—	0.042	—	0.781	—
AND=	—	0.111	—	0.067	—	0.067	—	1.049	—
AND<>	—	0.114	—	0.067	—	0.067	—	1.049	—
AND>	—	0.118	—	0.067	—	0.067	—	1.049	—
AND<=	—	0.117	—	0.067	—	0.067	—	1.049	—
AND<	—	0.117	—	0.067	—	0.067	—	1.049	—
AND>=	—	0.117	—	0.067	—	0.067	—	1.049	—
OR=	—	0.111	—	0.067	—	0.067	—	1.049	—
OR<>	—	0.114	—	0.067	—	0.067	—	1.049	—
OR>	—	0.117	—	0.067	—	0.067	—	1.049	—
OR<=	—	0.117	—	0.067	—	0.067	—	1.049	—
OR<	—	0.117	—	0.067	—	0.067	—	1.049	—
OR>=	—	0.117	—	0.067	—	0.067	—	1.049	—
LD=_U	—	0.114	—	0.042	—	0.042	—	0.781	—
LD<>_U	—	0.114	—	0.042	—	0.042	—	0.781	—
LD>_U	—	0.107	—	0.042	—	0.042	—	0.781	—
LD<=_U	—	0.097	—	0.042	—	0.042	—	0.781	—
LD<_U	—	0.107	—	0.042	—	0.042	—	0.781	—
LD>=_U	—	0.097	—	0.042	—	0.042	—	0.781	—
AND=_U	—	0.111	—	0.067	—	0.067	—	1.049	—
AND<>_U	—	0.114	—	0.067	—	0.067	—	1.049	—
AND>_U	—	0.107	—	0.067	—	0.067	—	1.049	—
AND<=_U	—	0.107	—	0.067	—	0.067	—	1.049	—
AND<_U	—	0.107	—	0.067	—	0.067	—	1.049	—
AND>=_U	—	0.107	—	0.067	—	0.067	—	1.049	—
OR=_U	—	0.111	—	0.067	—	0.067	—	1.049	—
OR<>_U	—	0.114	—	0.067	—	0.067	—	1.049	—
OR>_U	—	0.107	—	0.067	—	0.067	—	1.049	—
OR<=_U	—	0.107	—	0.067	—	0.067	—	1.049	—
OR<_U	—	0.107	—	0.067	—	0.067	—	1.049	—
OR>=_U	—	0.107	—	0.067	—	0.067	—	1.049	—
LDD=	—	0.134	—	0.042	—	0.042	—	0.781	—
LDD<>	—	0.134	—	0.042	—	0.042	—	0.781	—
LDD>	—	0.134	—	0.042	—	0.042	—	0.781	—
LDD<=	—	0.134	—	0.042	—	0.042	—	0.781	—
LDD<	—	0.134	—	0.042	—	0.042	—	0.781	—
LDD>=	—	0.134	—	0.042	—	0.042	—	0.781	—
ANDD=	—	0.134	—	0.067	—	0.067	—	1.049	—
ANDD<>	—	0.135	—	0.067	—	0.067	—	1.049	—
ANDD>	—	0.135	—	0.067	—	0.067	—	1.049	—

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)
ANDD<=	—	0.134	—	0.067	—	0.067	—	1.049	—
ANDD<	—	0.134	—	0.067	—	0.067	—	1.049	—
ANDD>=	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD=	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD<>	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD>	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD<=	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD<	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD>=	—	0.136	—	0.067	—	0.067	—	1.049	—
LDD=_U	—	0.134	—	0.042	—	0.042	—	0.781	—
LDD<>_U	—	0.134	—	0.042	—	0.042	—	0.781	—
LDD>_U	—	0.132	—	0.042	—	0.042	—	0.781	—
LDD<=_U	—	0.100	—	0.042	—	0.042	—	0.781	—
LDD<_U	—	0.132	—	0.042	—	0.042	—	0.781	—
LDD>=_U	—	0.100	—	0.042	—	0.042	—	0.781	—
ANDD=_U	—	0.134	—	0.067	—	0.067	—	1.049	—
ANDD<>_U	—	0.134	—	0.067	—	0.067	—	1.049	—
ANDD>_U	—	0.134	—	0.067	—	0.067	—	1.049	—
ANDD<=_U	—	0.134	—	0.067	—	0.067	—	1.049	—
ANDD<_U	—	0.134	—	0.067	—	0.067	—	1.049	—
ANDD>=_U	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD=_U	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD<>_U	—	0.135	—	0.067	—	0.067	—	1.049	—
ORD>_U	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD<=_U	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD<_U	—	0.134	—	0.067	—	0.067	—	1.049	—
ORD>=_U	—	0.134	—	0.067	—	0.067	—	1.049	—
+ (2 operands)	—	0.147	0.092	0.059	0.017	0.059	0.017	1.026	0.674
+ (3 operands)	—	0.147	0.120	0.059	0.017	0.059	0.017	1.026	0.674
+_U (2 operands)	—	0.147	0.120	0.059	0.017	0.059	0.017	1.026	0.674
+_U (3 operands)	—	0.147	0.122	0.059	0.017	0.059	0.017	1.026	0.674
- (2 operands)	—	0.147	0.092	0.059	0.017	0.059	0.017	1.026	0.674
- (3 operands)	—	0.147	0.120	0.059	0.017	0.059	0.017	1.026	0.674
-_U (2 operands)	—	0.147	0.120	0.059	0.017	0.059	0.017	1.026	0.674
-_U (3 operands)	—	0.147	0.120	0.059	0.017	0.059	0.017	1.026	0.674
D+ (2 operands)	—	0.144	0.092	0.059	0.017	0.059	0.017	1.026	0.674
D+ (3 operands)	—	0.144	0.117	0.059	0.017	0.059	0.017	1.026	0.674
D+_U (2 operands)	—	0.144	0.117	0.059	0.017	0.059	0.017	1.026	0.674
D+_U (3 operands)	—	0.144	0.117	0.059	0.017	0.059	0.017	1.026	0.674
D- (2 operands)	—	0.144	0.092	0.059	0.017	0.059	0.017	1.026	0.674
D- (3 operands)	—	0.144	0.117	0.059	0.017	0.059	0.017	1.026	0.674
D-_U (2 operands)	—	0.144	0.117	0.059	0.017	0.059	0.017	1.026	0.674
D-_U (3 operands)	—	0.144	0.117	0.059	0.017	0.059	0.017	1.026	0.674
*	—	0.147	0.084	0.059	0.017	0.059	0.017	1.026	0.674
*_U	—	0.147	0.120	0.059	0.017	0.059	0.017	1.026	0.674
INC	—	0.137	0.110	0.067	0.017	0.067	0.017	1.049	0.674
INC_U	—	0.137	0.110	0.067	0.017	0.067	0.017	1.049	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)
DEC	—	0.137	0.111	0.067	0.017	0.067	0.017	1.049	0.674
DEC_U	—	0.137	0.110	0.067	0.017	0.067	0.017	1.049	0.674
DINC	—	0.134	0.107	0.067	0.017	0.067	0.017	1.049	0.674
DINC_U	—	0.134	0.107	0.067	0.017	0.067	0.017	1.049	0.674
DDEC	—	0.134	0.107	0.067	0.017	0.067	0.017	1.049	0.674
DDEC_U	—	0.134	0.107	0.067	0.017	0.067	0.017	1.049	0.674
WAND (2 operands)	—	0.147	0.092	0.059	0.017	0.059	0.017	1.026	0.674
WAND (3 operands)	—	0.147	0.121	0.059	0.017	0.059	0.017	1.026	0.674
DAND (2 operands)	—	0.144	0.092	0.059	0.017	0.059	0.017	1.026	0.674
DAND (3 operands)	—	0.144	0.117	0.059	0.017	0.059	0.017	1.026	0.674
WOR (2 operands)	—	0.147	0.120	0.059	0.017	0.059	0.017	1.026	0.674
WOR (3 operands)	—	0.147	0.120	0.059	0.017	0.059	0.017	1.026	0.674
DOR (2 operands)	—	0.144	0.117	0.059	0.017	0.059	0.017	1.026	0.674
DOR (3 operands)	—	0.144	0.117	0.059	0.017	0.059	0.017	1.026	0.674
WXOR (2 operands)	—	0.147	0.123	0.059	0.017	0.059	0.017	1.026	0.674
WXOR (3 operands)	—	0.147	0.124	0.059	0.017	0.059	0.017	1.026	0.674
DXOR (2 operands)	—	0.144	0.121	0.059	0.017	0.059	0.017	1.026	0.674
DXOR (3 operands)	—	0.144	0.120	0.059	0.017	0.059	0.017	1.026	0.674
WXNR (2 operands)	—	0.157	0.127	0.085	0.017	0.085	0.017	1.166	0.674
WXNR (3 operands)	—	0.157	0.127	0.085	0.017	0.085	0.017	1.166	0.674
DXNR (2 operands)	—	0.150	0.117	0.085	0.017	0.085	0.017	1.166	0.674
DXNR (3 operands)	—	0.150	0.117	0.085	0.017	0.085	0.017	1.166	0.674
BSET	(n)=K1	0.144	0.124	0.075	0.017	0.075	0.017	1.061	0.674
	(n)=K15	0.144	0.123	0.075	0.017	0.075	0.017	1.061	0.674
BRST	(n)=K1	0.144	0.124	0.075	0.017	0.075	0.017	1.061	0.674
	(n)=K15	0.144	0.124	0.075	0.017	0.075	0.017	1.061	0.674
TEST	—	0.170	0.120	0.102	0.017	0.102	0.017	1.376	0.674
DTEST	—	0.171	0.124	0.102	0.017	0.102	0.017	1.376	0.674
INT2UINT	—	0.127	0.110	0.058	0.017	0.058	0.017	0.814	0.674
INT2DINT	—	0.130	0.113	0.058	0.017	0.058	0.017	0.814	0.674
INT2UDINT	—	0.130	0.114	0.058	0.017	0.058	0.017	0.814	0.674
UINT2INT	—	0.127	0.110	0.058	0.017	0.058	0.017	0.814	0.674
UINT2DINT	—	0.130	0.113	0.058	0.017	0.058	0.017	0.814	0.674
UINT2UDINT	—	0.130	0.114	0.058	0.017	0.058	0.017	0.814	0.674
DINT2INT	—	0.127	0.110	0.058	0.017	0.058	0.017	0.814	0.674
DINT2UINT	—	0.127	0.111	0.058	0.017	0.058	0.017	0.814	0.674
DINT2UDINT	—	0.127	0.111	0.058	0.017	0.058	0.017	0.814	0.674
UDINT2INT	—	0.127	0.110	0.058	0.017	0.058	0.017	0.814	0.674
UDINT2UINT	—	0.127	0.110	0.058	0.017	0.058	0.017	0.814	0.674
UDINT2DINT	—	0.127	0.111	0.058	0.017	0.058	0.017	0.814	0.674
MOV	—	0.100	0.084	0.034	0.017	0.034	0.017	0.779	0.674
DMOV	—	0.111	0.084	0.034	0.017	0.034	0.017	0.779	0.674
CML	—	0.137	0.111	0.067	0.017	0.067	0.017	1.049	0.674
DCML	—	0.134	0.107	0.067	0.017	0.067	0.017	1.049	0.674
CMLB	—	0.147	0.120	0.081	0.017	0.081	0.017	1.026	0.674
XCH	—	0.154	0.124	0.079	0.017	0.079	0.017	1.061	0.674
DXCH	—	0.145	0.120	0.079	0.017	0.079	0.017	1.061	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)
SWAP	—	0.137	0.110	0.067	0.017	0.067	0.017	1.049	0.674
DSWAP	—	0.137	0.110	0.067	0.017	0.067	0.017	1.049	0.674
MOVB	—	0.154	0.111	0.083	0.017	0.083	0.017	0.886	0.674
INT2FLT	—	0.154	0.110	0.083	0.017	0.083	0.017	0.886	0.674
UINT2FLT	—	0.154	0.111	0.083	0.017	0.083	0.017	0.886	0.674
DINT2FLT	—	0.154	0.110	0.083	0.017	0.083	0.017	0.886	0.674
UDINT2FLT	—	0.154	0.111	0.083	0.017	0.083	0.017	0.886	0.674
EMOV	—	0.127	0.110	0.059	0.017	0.059	0.017	1.026	0.674
DEMOV	—	0.134	0.110	0.059	0.017	0.059	0.017	1.026	0.674

Instruction processing time

The following table shows instruction processing time of each instruction.
Because the instruction processing time slightly varies depending on the contents of the source and destination, refer to the values in the table as a reference. Since the version supporting the SFC program, the processing time of the some instructions will become longer than the conventional processing time. (☞ Page 1452 Added and Changed Functions)

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
LD	—	1.728	—	1.333	—	1.333	—	3.041	—
LDI	—	1.763	—	1.333	—	1.333	—	3.041	—
AND	—	1.728	—	1.333	—	1.333	—	3.041	—
ANI	—	1.831	—	1.333	—	1.333	—	3.041	—
OR	—	1.764	—	1.333	—	1.333	—	3.041	—
ORI	—	1.831	—	1.333	—	1.333	—	3.041	—
LDP	—	2.062	—	1.767	—	1.759	—	3.628	—
LDF	—	2.035	—	1.767	—	1.759	—	3.628	—
ANDP	—	2.164	—	1.767	—	1.759	—	3.628	—
ANDF	—	2.096	—	1.767	—	1.759	—	3.628	—
ORP	—	2.098	—	1.767	—	1.759	—	3.628	—
ORF	—	2.130	—	1.767	—	1.759	—	3.628	—
LDPI	—	2.131	—	1.759	—	1.759	—	3.628	—
LDFI	—	2.095	—	1.759	—	1.759	—	3.628	—
ANDPI	—	2.092	—	1.759	—	1.759	—	3.628	—
ANDFI	—	2.035	—	1.759	—	1.759	—	3.628	—
ORPI	—	2.100	—	1.759	—	1.759	—	3.628	—
ORFI	—	2.066	—	1.759	—	1.759	—	3.628	—
ANB	—	0.084	—	0.025	—	0.025	—	0.341	—
ORB	—	0.084	—	0.025	—	0.025	—	0.666	—
MPS	—	0.084	—	0.017	—	0.017	—	0.333	—
MRD	—	0.084	—	0.009	—	0.009	—	0.333	—
MPP	—	0.084	—	0.009	—	0.009	—	0.333	—
INV	—	0.092	—	0.017	—	0.017	—	0.333	—
MEP	—	0.100	—	0.042	—	0.042	—	0.700	—
MEF	—	0.100	—	0.050	—	0.050	—	0.700	—
OUT	—	1.334	1.276	1.217	1.217	1.217	1.217	3.023	3.023
OUTH T/ST	—	3.934	3.577	2.973	2.395	2.853	2.361	5.013	4.126
OUT C	—	3.496	3.001	2.875	2.538	2.753	2.446	4.964	4.247
OUT LC	—	3.745	3.764	2.875	2.538	2.753	2.446	4.964	4.247
OUT F	—	25.840	1.548	24.127	0.882	16.448	0.882	19.083	2.136
OUT T/ST	—	4.427	3.852	3.267	2.705	3.160	2.664	5.307	4.426
OUTHS T/ST	—	3.654	3.277	2.960	2.278	2.760	2.211	4.920	3.904
SET	—	1.367	0.110	1.233	0.017	1.233	0.017	3.027	0.674
RST	—	1.367	0.110	1.128	0.017	1.068	0.017	2.721	0.674
SET F	—	1.002	1.006	0.967	0.996	0.959	0.987	2.484	2.198
RST F	—	1.071	1.078	0.967	0.996	0.959	0.987	2.484	2.338
ANS	(n)=K1	4.899	4.633	3.633	3.487	3.416	3.311	6.148	5.887
ANR	—	1.466	0.087	1.359	0.017	1.359	0.017	2.751	0.674
ANRP	—	2.224	0.971	2.224	0.882	2.224	0.882	4.213	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
PLS	—	1.432	1.138	1.199	1.079	1.159	1.064	2.664	2.774
PLF	—	1.383	1.155	1.199	1.079	1.159	1.064	2.664	2.774
FF	—	1.115	0.903	0.969	0.882	0.929	0.870	2.108	2.205
ALT	—	2.564	0.111	2.033	0.017	1.959	0.017	3.995	0.674
ALTP	—	2.284	0.882	2.284	0.882	2.284	0.882	5.457	2.136
SFT	—	4.231	0.087	3.267	0.017	3.241	0.017	4.351	0.674
SFTP	—	4.355	0.970	3.467	0.882	3.385	0.882	4.984	2.136
SFR	(n)=K1	5.366	0.120	4.267	0.017	4.108	0.017	6.431	0.674
	(n)=K15	5.367	0.120	4.267	0.017	4.108	0.017	6.431	0.674
SFRP	(n)=K1	5.132	0.968	5.132	0.882	4.973	0.882	7.893	2.136
	(n)=K15	5.132	0.968	5.132	0.882	4.973	0.882	7.893	2.136
SFL	(n)=K1	5.435	0.124	4.267	0.017	4.108	0.017	6.431	0.674
	(n)=K15	5.431	0.124	4.267	0.017	4.108	0.017	6.431	0.674
SFLP	(n)=K1	5.132	0.970	5.132	0.882	4.973	0.882	7.893	2.136
	(n)=K15	5.132	0.970	5.132	0.882	4.973	0.882	7.893	2.136
BSFR	(n)=K1	4.771	0.091	3.767	0.017	3.508	0.017	6.736	0.674
	(n)=K96	7.890	0.091	6.368	0.017	6.157	0.017	8.297	0.674
BSFRP	(n)=K1	4.996	0.970	4.051	0.882	3.861	0.882	8.198	2.136
	(n)=K96	8.086	0.970	6.616	0.882	6.447	0.882	9.759	2.136
BSFL	(n)=K1	4.631	0.090	3.767	0.017	3.508	0.017	6.736	0.674
	(n)=K96	7.699	0.090	6.368	0.017	6.157	0.017	8.297	0.674
BSFLP	(n)=K1	4.834	0.969	4.051	0.882	3.861	0.882	8.198	2.136
	(n)=K96	7.914	0.969	6.616	0.882	6.447	0.882	9.759	2.136
SFTR	(n1)=K16, (n2)=K1	9.279	0.090	7.324	0.017	7.141	0.017	9.885	0.674
	(n1)=K16, (n2)=K15	9.282	0.090	7.324	0.017	7.141	0.017	9.885	0.674
SFTRP	(n1)=K16, (n2)=K1	9.335	0.970	7.509	0.882	7.319	0.882	9.635	2.136
	(n1)=K16, (n2)=K15	9.383	0.970	7.509	0.882	7.319	0.882	9.635	2.136
SFTL	(n1)=K16, (n2)=K1	9.023	0.090	7.324	0.017	7.141	0.017	9.885	0.674
	(n1)=K16, (n2)=K15	9.022	0.090	7.324	0.017	7.141	0.017	9.885	0.674
SFTLP	(n1)=K16, (n2)=K1	8.975	0.971	7.509	0.882	7.319	0.882	9.635	2.136
	(n1)=K16, (n2)=K15	8.970	0.971	7.509	0.882	7.319	0.882	9.635	2.136
DSFR	(n)=K1	3.766	0.090	3.067	0.017	2.925	0.017	4.967	0.674
	(n)=K96	8.022	0.090	6.901	0.017	6.743	0.017	8.881	0.674
DSFRP	(n)=K1	3.904	0.968	3.287	0.882	3.143	0.882	5.753	2.136
	(n)=K96	8.075	0.968	7.163	0.882	6.959	0.882	9.500	2.136
DSFL	(n)=K1	3.099	0.091	3.067	0.017	2.925	0.017	4.967	0.674
	(n)=K96	7.224	0.091	6.901	0.017	6.743	0.017	8.881	0.674
DSFLP	(n)=K1	3.304	0.970	3.287	0.882	3.143	0.882	5.753	2.136
	(n)=K96	7.375	0.970	7.163	0.882	6.959	0.882	9.500	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
WSFR	(n1)=K16, (n2)=K1	9.176	0.087	7.696	0.017	7.417	0.017	10.208	0.674
	(n1)=K16, (n2)=K15	9.154	0.087	7.696	0.017	7.417	0.017	10.208	0.674
WSFRP	(n1)=K16, (n2)=K1	9.218	0.970	7.860	0.882	7.617	0.882	10.097	2.136
	(n1)=K16, (n2)=K15	9.227	0.970	7.860	0.882	7.617	0.882	10.097	2.136
WSFL	(n1)=K16, (n2)=K1	8.822	0.087	7.696	0.017	7.417	0.017	10.208	0.674
	(n1)=K16, (n2)=K15	8.820	0.087	7.696	0.017	7.417	0.017	10.208	0.674
WSFLP	(n1)=K16, (n2)=K1	8.867	0.970	7.860	0.882	7.617	0.882	10.097	2.136
	(n1)=K16, (n2)=K15	8.875	0.970	7.860	0.882	7.617	0.882	10.097	2.136
MC	—	3.641	3.600	2.800	2.716	2.800	2.716	4.483	4.566
MCR	—	1.950	1.958	1.591	1.600	1.591	1.600	2.941	2.950
FEND	—	0.250	—	0.250	—	0.250	—	0.250	—
END	—	0.250	—	0.250	—	0.250	—	0.250	—
STOP	—	0.991	0.091	0.925	0.017	0.925	0.017	1.566	0.674
NOP	—	0.000	—	0.000	—	0.000	—	0.000	—
LD=	—	3.796	—	2.867	—	2.751	—	4.943	—
LD<>	—	3.802	—	2.867	—	2.751	—	4.943	—
LD>	—	3.763	—	2.867	—	2.751	—	4.943	—
LD<=	—	3.766	—	2.867	—	2.751	—	4.943	—
LD<	—	3.763	—	2.867	—	2.751	—	4.943	—
LD>=	—	3.764	—	2.867	—	2.751	—	4.943	—
AND=	—	3.864	—	2.867	—	2.751	—	4.943	—
AND<>	—	3.830	—	2.867	—	2.751	—	4.943	—
AND>	—	3.831	—	2.867	—	2.751	—	4.943	—
AND<=	—	3.798	—	2.867	—	2.751	—	4.943	—
AND<	—	3.800	—	2.867	—	2.751	—	4.943	—
AND>=	—	3.830	—	2.867	—	2.751	—	4.943	—
OR=	—	3.863	—	2.867	—	2.751	—	4.943	—
OR<>	—	3.831	—	2.867	—	2.751	—	4.943	—
OR>	—	3.835	—	2.867	—	2.751	—	4.943	—
OR<=	—	3.796	—	2.867	—	2.751	—	4.943	—
OR<	—	3.835	—	2.867	—	2.751	—	4.943	—
OR>=	—	3.834	—	2.867	—	2.751	—	4.943	—
CMP	—	3.931	0.090	3.300	0.017	3.108	0.017	5.405	0.674
CMPP	—	4.107	0.969	3.469	0.882	3.304	0.882	6.091	2.136
LD=_U	—	3.796	—	2.867	—	2.751	—	4.943	—
LD<>_U	—	3.796	—	2.867	—	2.751	—	4.943	—
LD>_U	—	3.830	—	2.867	—	2.751	—	4.943	—
LD<=_U	—	3.796	—	2.867	—	2.751	—	4.943	—
LD<_U	—	3.830	—	2.867	—	2.751	—	4.943	—
LD>=_U	—	3.831	—	2.867	—	2.751	—	4.943	—
AND=_U	—	3.896	—	2.867	—	2.751	—	4.943	—

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
AND<>_U	—	3.831	—	2.867	—	2.751	—	4.943	—
AND>_U	—	3.896	—	2.867	—	2.751	—	4.943	—
AND<=_U	—	3.838	—	2.867	—	2.751	—	4.943	—
AND<_U	—	3.864	—	2.867	—	2.751	—	4.943	—
AND>=_U	—	3.863	—	2.867	—	2.751	—	4.943	—
OR=_U	—	3.896	—	2.867	—	2.751	—	4.943	—
OR<>_U	—	3.870	—	2.867	—	2.751	—	4.943	—
OR>_U	—	3.864	—	2.867	—	2.751	—	4.943	—
OR<=_U	—	3.831	—	2.867	—	2.751	—	4.943	—
OR<_U	—	3.899	—	2.867	—	2.751	—	4.943	—
OR>=_U	—	3.896	—	2.867	—	2.751	—	4.943	—
CMP_U	—	3.898	0.087	3.300	0.017	3.108	0.017	5.405	0.674
CMPP_U	—	4.076	0.969	3.469	0.882	3.304	0.882	6.091	2.136
LDD=	—	4.198	—	3.244	—	3.084	—	4.929	—
LDD<>	—	4.230	—	3.244	—	3.084	—	4.929	—
LDD>	—	4.203	—	3.244	—	3.084	—	4.929	—
LDD<=	—	4.196	—	3.244	—	3.084	—	4.929	—
LDD<	—	4.203	—	3.244	—	3.084	—	4.929	—
LDD>=	—	4.199	—	3.244	—	3.084	—	4.929	—
ANDD=	—	4.296	—	3.244	—	3.084	—	4.929	—
ANDD<>	—	4.264	—	3.244	—	3.084	—	4.929	—
ANDD>	—	4.264	—	3.244	—	3.084	—	4.929	—
ANDD<=	—	4.235	—	3.244	—	3.084	—	4.929	—
ANDD<	—	4.230	—	3.244	—	3.084	—	4.929	—
ANDD>=	—	4.264	—	3.244	—	3.084	—	4.929	—
ORD=	—	4.298	—	3.244	—	3.084	—	4.929	—
ORD<>	—	4.263	—	3.244	—	3.084	—	4.929	—
ORD>	—	4.264	—	3.244	—	3.084	—	4.929	—
ORD<=	—	4.235	—	3.244	—	3.084	—	4.929	—
ORD<	—	4.264	—	3.244	—	3.084	—	4.929	—
ORD>=	—	4.264	—	3.244	—	3.084	—	4.929	—
DCMP	—	3.967	0.087	3.333	0.017	3.133	0.017	5.420	0.674
DCMPP	—	4.100	0.970	3.547	0.882	3.341	0.882	6.109	2.136
LDD=_U	—	4.230	—	3.244	—	3.084	—	4.929	—
LDD<>_U	—	4.230	—	3.244	—	3.084	—	4.929	—
LDD>_U	—	4.263	—	3.244	—	3.084	—	4.929	—
LDD<=_U	—	4.232	—	3.244	—	3.084	—	4.929	—
LDD<_U	—	4.263	—	3.244	—	3.084	—	4.929	—
LDD>=_U	—	4.270	—	3.244	—	3.084	—	4.929	—
ANDD=_U	—	4.330	—	3.244	—	3.084	—	4.929	—
ANDD<>_U	—	4.264	—	3.244	—	3.084	—	4.929	—
ANDD>_U	—	4.330	—	3.244	—	3.084	—	4.929	—
ANDD<=_U	—	4.296	—	3.244	—	3.084	—	4.929	—
ANDD<_U	—	4.298	—	3.244	—	3.084	—	4.929	—
ANDD>=_U	—	4.296	—	3.244	—	3.084	—	4.929	—
ORD=_U	—	4.330	—	3.244	—	3.084	—	4.929	—
ORD<>_U	—	4.298	—	3.244	—	3.084	—	4.929	—

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
ORD>_U	—	4.300	—	3.244	—	3.084	—	4.929	—
ORD<=_U	—	4.298	—	3.244	—	3.084	—	4.929	—
ORD<_U	—	4.335	—	3.244	—	3.084	—	4.929	—
ORD>=_U	—	4.330	—	3.244	—	3.084	—	4.929	—
DCMP_U	—	3.931	0.087	3.333	0.017	3.133	0.017	5.420	0.674
DCMPP_U	—	4.112	0.972	3.547	0.882	3.341	0.882	6.109	2.136
ZCP	—	4.199	0.087	3.633	0.017	3.367	0.017	5.916	0.674
ZCPP	—	4.356	0.968	3.805	0.882	3.555	0.882	7.378	2.136
ZCP_U	—	4.202	0.087	3.633	0.017	3.367	0.017	5.916	0.674
ZCPP_U	—	4.366	0.970	3.805	0.882	3.555	0.882	7.378	2.136
DZCP	—	4.266	0.087	3.633	0.017	3.367	0.017	5.916	0.674
DZCPP	—	4.460	0.972	3.805	0.882	3.555	0.882	7.378	2.136
DZCP_U	—	4.231	0.087	3.633	0.017	3.367	0.017	5.916	0.674
DZCPP_U	—	4.403	0.971	3.805	0.882	3.555	0.882	7.378	2.136
BKCMP=	(n)=K1	8.419	0.087	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.559	0.087	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP<>	(n)=K1	8.450	0.087	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.566	0.087	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP>	(n)=K1	8.418	0.089	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.536	0.089	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP<=	(n)=K1	8.451	0.088	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.558	0.088	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP<	(n)=K1	8.423	0.087	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.526	0.087	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP>=	(n)=K1	8.418	0.089	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.567	0.089	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP=P	(n)=K1	8.440	0.971	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	38.963	0.971	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP<>P	(n)=K1	8.454	0.971	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	38.908	0.971	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP>P	(n)=K1	8.503	0.971	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	39.271	0.971	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP<=P	(n)=K1	8.455	0.970	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	38.908	0.970	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP<P	(n)=K1	8.439	0.970	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	38.992	0.970	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP>=P	(n)=K1	8.504	0.972	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	38.915	0.972	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP=_U	(n)=K1	8.416	0.087	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.498	0.087	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP<>_U	(n)=K1	8.418	0.087	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.500	0.087	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP>_U	(n)=K1	8.452	0.087	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.532	0.087	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP<=_U	(n)=K1	8.451	0.087	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.528	0.087	24.760	0.017	24.332	0.017	27.296	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
BKCMP<_U	(n)=K1	8.464	0.089	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.531	0.089	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP>=_U	(n)=K1	8.450	0.087	6.928	0.017	6.501	0.017	9.473	0.674
	(n)=K96	39.526	0.087	24.760	0.017	24.332	0.017	27.296	0.674
BKCMP=P_U	(n)=K1	8.428	0.970	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	38.747	0.970	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP<>P_U	(n)=K1	8.431	0.971	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	38.851	0.971	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP>P_U	(n)=K1	8.483	0.970	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	38.739	0.970	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP<=P_U	(n)=K1	8.447	0.973	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	39.140	0.973	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP<P_U	(n)=K1	8.456	0.972	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	38.830	0.972	24.900	0.882	24.456	0.882	28.758	2.136
BKCMP>=P_U	(n)=K1	8.450	0.971	7.112	0.882	6.687	0.882	10.935	2.136
	(n)=K96	39.032	0.971	24.900	0.882	24.456	0.882	28.758	2.136
DBKCMP=	(n)=K1	8.418	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	45.634	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP<>	(n)=K1	8.418	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	45.631	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP>	(n)=K1	8.416	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	45.596	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP<=	(n)=K1	8.418	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	45.592	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP<	(n)=K1	8.419	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	45.567	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP>=	(n)=K1	8.420	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	45.592	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP=P	(n)=K1	8.424	0.971	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	44.783	0.971	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP<>P	(n)=K1	8.508	0.972	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	44.823	0.972	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP>P	(n)=K1	8.424	0.971	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	44.788	0.971	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP<=P	(n)=K1	8.495	0.970	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	44.722	0.970	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP<P	(n)=K1	8.454	0.973	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	44.755	0.973	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP>=P	(n)=K1	8.439	0.972	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	30.077	0.972	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP=_U	(n)=K1	8.487	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	29.212	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP<>_U	(n)=K1	8.487	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	29.212	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP>_U	(n)=K1	8.518	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	29.212	0.087	29.212	0.017	28.435	0.017	31.341	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
DBKCMP<=_U	(n)=K1	8.519	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	29.212	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP<_U	(n)=K1	8.519	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	29.212	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP>=_U	(n)=K1	8.518	0.087	6.935	0.017	6.492	0.017	9.464	0.674
	(n)=K96	29.212	0.087	29.212	0.017	28.435	0.017	31.341	0.674
DBKCMP=P_U	(n)=K1	8.492	0.970	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	30.077	0.970	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP<>P_U	(n)=K1	8.560	0.970	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	30.077	0.970	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP>P_U	(n)=K1	8.538	0.971	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	30.077	0.971	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP<=P_U	(n)=K1	8.535	0.972	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	30.077	0.972	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP<P_U	(n)=K1	8.543	0.971	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	30.077	0.971	30.077	0.882	29.897	0.882	32.803	2.136
DBKCMP>=P_U	(n)=K1	8.535	0.971	7.119	0.882	6.684	0.882	10.926	2.136
	(n)=K96	30.077	0.971	30.077	0.882	29.897	0.882	32.803	2.136
+ (2 operands)	—	4.831	0.092	3.700	0.017	3.492	0.017	5.363	0.674
+P (2 operands)	—	4.565	0.969	4.565	0.882	3.683	0.882	6.067	2.136
+ (3 operands)	—	4.832	0.120	3.700	0.017	3.492	0.017	5.363	0.674
+P (3 operands)	—	4.565	0.969	4.565	0.882	3.683	0.882	6.067	2.136
+_U (2 operands)	—	4.831	0.120	3.700	0.017	3.492	0.017	5.363	0.674
+P_U (2 operands)	—	4.565	0.972	4.565	0.882	3.683	0.882	6.067	2.136
+_U (3 operands)	—	4.836	0.122	3.700	0.017	3.492	0.017	5.363	0.674
+P_U (3 operands)	—	4.565	0.972	4.565	0.882	3.683	0.882	6.067	2.136
ADD	—	3.298	0.087	2.900	0.017	2.733	0.017	4.157	0.674
ADDP	—	3.468	0.970	3.065	0.882	2.920	0.882	4.883	2.136
ADD_U	—	3.331	0.087	2.900	0.017	2.733	0.017	4.157	0.674
ADDP_U	—	3.467	0.969	3.065	0.882	2.920	0.882	4.883	2.136
- (2 operands)	—	4.768	0.092	3.700	0.017	3.492	0.017	5.363	0.674
-P (2 operands)	—	4.565	0.970	4.565	0.882	3.683	0.882	6.067	2.136
- (3 operands)	—	4.764	0.120	3.700	0.017	3.492	0.017	5.363	0.674
-P (3 operands)	—	4.565	0.970	4.565	0.882	3.683	0.882	6.067	2.136
-_U (2 operands)	—	4.764	0.120	3.700	0.017	3.492	0.017	5.363	0.674
-P_U (2 operands)	—	4.565	0.971	4.565	0.882	3.683	0.882	6.067	2.136
-_U (3 operands)	—	4.764	0.120	3.700	0.017	3.492	0.017	5.363	0.674
-P_U (3 operands)	—	4.565	0.971	4.565	0.882	3.683	0.882	6.067	2.136
SUB	—	3.398	0.087	2.900	0.017	2.733	0.017	4.157	0.674
SUBP	—	3.600	0.970	3.065	0.882	2.920	0.882	4.883	2.136
SUB_U	—	3.364	0.087	2.900	0.017	2.733	0.017	4.157	0.674
SUBP_U	—	3.559	0.970	3.065	0.882	2.920	0.882	4.883	2.136
D+ (2 operands)	—	5.511	0.092	4.233	0.017	3.967	0.017	6.553	0.674
D+P (2 operands)	—	5.098	0.966	5.098	0.882	4.152	0.882	8.015	2.136
D+ (3 operands)	—	5.503	0.117	4.233	0.017	3.967	0.017	6.553	0.674
D+P (3 operands)	—	5.098	0.970	5.098	0.882	4.152	0.882	8.015	2.136
D+_U (2 operands)	—	5.498	0.117	4.233	0.017	3.967	0.017	6.553	0.674

A

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
D+P_U (2 operands)	—	5.098	0.971	5.098	0.882	4.152	0.882	8.015	2.136
D+_U (3 operands)	—	5.498	0.117	4.233	0.017	3.967	0.017	6.553	0.674
D+P_U (3 operands)	—	5.098	0.970	5.098	0.882	4.152	0.882	8.015	2.136
DADD	—	3.331	0.087	2.933	0.017	2.749	0.017	4.875	0.674
DADDP	—	3.488	0.970	3.112	0.882	2.931	0.882	6.337	2.136
DADD_U	—	3.364	0.087	2.933	0.017	2.749	0.017	4.875	0.674
DADDP_U	—	3.495	0.970	3.112	0.882	2.931	0.882	6.337	2.136
D- (2 operands)	—	5.464	0.092	4.233	0.017	3.967	0.017	6.553	0.674
D-P (2 operands)	—	5.098	0.973	5.098	0.882	4.152	0.882	8.015	2.136
D- (3 operands)	—	5.467	0.117	4.233	0.017	3.967	0.017	6.553	0.674
D-P (3 operands)	—	5.098	0.970	5.098	0.882	4.152	0.882	8.015	2.136
D-_U (2 operands)	—	5.464	0.117	4.233	0.017	3.967	0.017	6.553	0.674
D-P_U (2 operands)	—	5.098	0.972	5.098	0.882	4.152	0.882	8.015	2.136
D-_U (3 operands)	—	5.464	0.117	4.233	0.017	3.967	0.017	6.553	0.674
D-P_U (3 operands)	—	5.098	0.968	5.098	0.882	4.152	0.882	8.015	2.136
DSUB	—	3.431	0.087	2.933	0.017	2.749	0.017	4.875	0.674
DSUBP	—	3.632	0.971	3.112	0.882	2.931	0.882	6.337	2.136
DSUB_U	—	3.364	0.087	2.933	0.017	2.749	0.017	4.875	0.674
DSUBP_U	—	3.510	0.972	3.112	0.882	2.931	0.882	6.337	2.136
*	—	5.344	0.084	3.967	0.017	3.749	0.017	5.616	0.674
*P	—	4.832	0.973	4.832	0.882	3.939	0.882	6.307	2.136
*_U	—	3.967	0.084	3.967	0.017	3.749	0.017	5.616	0.674
*P_U	—	4.832	0.882	4.832	0.882	3.939	0.882	6.307	2.136
MUL	—	4.399	0.087	3.300	0.017	3.067	0.017	4.933	0.674
MULP	—	4.496	0.970	3.511	0.882	3.263	0.882	5.649	2.136
MUL_U	—	4.398	0.087	3.300	0.017	3.067	0.017	4.933	0.674
MULP_U	—	4.482	0.970	3.511	0.882	3.263	0.882	5.649	2.136
/	—	4.431	0.092	3.827	0.017	3.827	0.017	5.735	0.674
/P	—	4.692	0.972	4.692	0.882	4.692	0.882	7.197	2.136
/_U	—	4.331	0.087	3.827	0.017	3.827	0.017	5.735	0.674
/P_U	—	4.692	0.971	4.692	0.882	4.692	0.882	7.197	2.136
DIV	—	4.598	0.087	3.503	0.017	3.275	0.017	5.841	0.674
DIVP	—	4.692	0.972	3.660	0.882	3.456	0.882	7.303	2.136
DIV_U	—	4.531	0.087	3.503	0.017	3.275	0.017	5.841	0.674
DIVP_U	—	4.660	0.970	3.660	0.882	3.456	0.882	7.303	2.136
D*	—	5.098	0.092	4.333	0.017	4.333	0.017	6.236	0.674
D*P	—	5.228	0.973	5.198	0.882	5.198	0.882	7.698	2.136
D*_U	—	5.100	0.087	4.333	0.017	4.333	0.017	6.236	0.674
D*P_U	—	5.224	0.972	5.198	0.882	5.198	0.882	7.698	2.136
DMUL	—	5.198	0.087	4.067	0.017	3.741	0.017	6.320	0.674
DMULP	—	5.368	0.970	4.267	0.882	3.928	0.882	7.782	2.136
DMUL_U	—	5.198	0.087	4.067	0.017	3.741	0.017	6.320	0.674
DMULP_U	—	5.368	0.973	4.267	0.882	3.928	0.882	7.782	2.136
D/	—	4.998	0.092	4.413	0.017	4.413	0.017	6.432	0.674
D/P	—	5.278	0.969	5.278	0.882	5.278	0.882	7.894	2.136
D/_U	—	4.999	0.084	4.413	0.017	4.413	0.017	6.432	0.674
D/P_U	—	5.278	0.972	5.278	0.882	5.278	0.882	7.894	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
DDIV	—	5.266	0.084	4.133	0.017	3.916	0.017	6.565	0.674
DDIVP	—	5.435	0.972	4.335	0.882	4.108	0.882	8.027	2.136
DDIV_U	—	5.131	0.084	4.133	0.017	3.916	0.017	6.565	0.674
DDIVP_U	—	5.295	0.970	4.335	0.882	4.108	0.882	8.027	2.136
B+ (2 operands)	—	5.064	0.088	4.567	0.017	4.300	0.017	6.012	0.674
B+P (2 operands)	—	5.202	0.972	4.737	0.882	4.513	0.882	7.474	2.136
B+ (3 operands)	—	4.567	0.087	4.567	0.017	4.300	0.017	6.012	0.674
B+P (3 operands)	—	4.737	0.968	4.737	0.882	4.513	0.882	7.474	2.136
B- (2 operands)	—	5.066	0.087	4.567	0.017	4.300	0.017	6.012	0.674
B-P (2 operands)	—	5.228	0.971	4.737	0.882	4.513	0.882	7.474	2.136
B- (3 operands)	—	4.567	0.087	4.567	0.017	4.300	0.017	6.012	0.674
B-P (3 operands)	—	4.737	0.969	4.737	0.882	4.513	0.882	7.474	2.136
DB+ (2 operands)	—	5.899	0.087	5.400	0.017	5.092	0.017	7.565	0.674
DB+P (2 operands)	—	6.067	0.971	5.541	0.882	5.301	0.882	9.027	2.136
DB+ (3 operands)	—	5.400	0.087	5.400	0.017	5.092	0.017	7.565	0.674
DB+P (3 operands)	—	5.541	0.970	5.541	0.882	5.301	0.882	9.027	2.136
DB- (2 operands)	—	5.903	0.087	5.400	0.017	5.092	0.017	7.565	0.674
DB-P (2 operands)	—	6.087	0.973	5.541	0.882	5.301	0.882	9.027	2.136
DB- (3 operands)	—	5.400	0.087	5.400	0.017	5.092	0.017	7.565	0.674
DB-P (3 operands)	—	5.541	0.971	5.541	0.882	5.301	0.882	9.027	2.136
B*	—	5.998	0.087	4.733	0.017	4.533	0.017	6.481	0.674
B*P	—	6.135	0.970	4.909	0.882	4.731	0.882	7.112	2.136
B/	—	6.300	0.087	4.879	0.017	4.684	0.017	7.332	0.674
B/P	—	6.479	0.968	5.071	0.882	4.889	0.882	8.794	2.136
DB*	—	6.664	0.087	5.300	0.017	4.992	0.017	7.641	0.674
DB*P	—	6.802	0.969	5.427	0.882	5.180	0.882	9.103	2.136
DB/	—	8.032	0.084	6.568	0.017	6.351	0.017	9.015	0.674
DB/P	—	8.159	0.971	6.753	0.882	6.523	0.882	10.477	2.136
BK+	(n)=K1	8.126	0.087	6.800	0.017	6.367	0.017	8.639	0.674
	(n)=K96	20.734	0.087	17.131	0.017	16.641	0.017	18.900	0.674
BK+P	(n)=K1	8.230	0.971	6.959	0.882	6.569	0.882	9.261	2.136
	(n)=K96	20.584	0.971	17.201	0.882	16.851	0.882	19.629	2.136
BK+_U	(n)=K1	8.032	0.088	6.800	0.017	6.367	0.017	8.639	0.674
	(n)=K96	20.626	0.088	17.131	0.017	16.641	0.017	18.900	0.674
BK+P_U	(n)=K1	8.135	0.970	6.959	0.882	6.569	0.882	9.261	2.136
	(n)=K96	20.440	0.970	17.201	0.882	16.851	0.882	19.629	2.136
BK-	(n)=K1	8.094	0.087	6.800	0.017	6.367	0.017	8.639	0.674
	(n)=K96	19.903	0.087	17.131	0.017	16.641	0.017	18.900	0.674
BK-P	(n)=K1	8.184	0.969	6.959	0.882	6.569	0.882	9.261	2.136
	(n)=K96	19.812	0.969	17.201	0.882	16.851	0.882	19.629	2.136
BK-_U	(n)=K1	8.055	0.088	6.800	0.017	6.367	0.017	8.639	0.674
	(n)=K96	19.830	0.088	17.131	0.017	16.641	0.017	18.900	0.674
BK-P_U	(n)=K1	8.186	0.971	6.959	0.882	6.569	0.882	9.261	2.136
	(n)=K96	19.820	0.971	17.201	0.882	16.851	0.882	19.629	2.136
DBK+	(n)=K1	8.258	0.087	6.801	0.017	6.367	0.017	9.339	0.674
	(n)=K96	25.666	0.087	21.976	0.017	21.223	0.017	24.096	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
DBK+P	(n)=K1	8.360	0.972	6.985	0.882	6.571	0.882	10.801	2.136
	(n)=K96	25.555	0.972	21.996	0.882	21.348	0.882	25.558	2.136
DBK+_U	(n)=K1	8.251	0.087	6.801	0.017	6.367	0.017	9.339	0.674
	(n)=K96	25.631	0.087	21.976	0.017	21.223	0.017	24.096	0.674
DBK+P_U	(n)=K1	8.275	0.972	6.985	0.882	6.571	0.882	10.801	2.136
	(n)=K96	25.539	0.972	21.996	0.882	21.348	0.882	25.558	2.136
DBK-	(n)=K1	8.251	0.087	6.801	0.017	6.367	0.017	9.339	0.674
	(n)=K96	24.795	0.087	21.976	0.017	21.223	0.017	24.096	0.674
DBK-P	(n)=K1	8.338	0.971	6.985	0.882	6.571	0.882	10.801	2.136
	(n)=K96	24.595	0.971	21.996	0.882	21.348	0.882	25.558	2.136
DBK-_U	(n)=K1	8.124	0.087	6.801	0.017	6.367	0.017	9.339	0.674
	(n)=K96	24.732	0.087	21.976	0.017	21.223	0.017	24.096	0.674
DBK-P_U	(n)=K1	8.224	0.972	6.985	0.882	6.571	0.882	10.801	2.136
	(n)=K96	24.560	0.972	21.996	0.882	21.348	0.882	25.558	2.136
INC	—	4.631	0.110	3.533	0.017	3.400	0.017	4.875	0.674
INCP	—	4.398	0.970	4.398	0.882	3.560	0.882	5.547	2.136
INC_U	—	4.632	0.110	3.533	0.017	3.400	0.017	4.875	0.674
INCP_U	—	4.398	0.971	4.398	0.882	3.560	0.882	5.547	2.136
DEC	—	4.634	0.111	3.533	0.017	3.400	0.017	4.875	0.674
DECP	—	4.398	0.968	4.398	0.882	3.560	0.882	5.547	2.136
DEC_U	—	4.631	0.110	3.533	0.017	3.400	0.017	4.875	0.674
DECP_U	—	4.398	0.973	4.398	0.882	3.560	0.882	5.547	2.136
DINC	—	5.098	0.107	3.900	0.017	3.716	0.017	5.901	0.674
DINCP	—	4.765	0.967	4.765	0.882	3.900	0.882	7.363	2.136
DINC_U	—	5.099	0.107	3.900	0.017	3.716	0.017	5.901	0.674
DINCP_U	—	4.765	0.969	4.765	0.882	3.900	0.882	7.363	2.136
DDEC	—	5.131	0.107	3.900	0.017	3.716	0.017	5.901	0.674
DDECP	—	4.765	0.970	4.765	0.882	3.900	0.882	7.363	2.136
DDEC_U	—	5.131	0.107	3.900	0.017	3.716	0.017	5.901	0.674
DDECP_U	—	4.765	0.970	4.765	0.882	3.900	0.882	7.363	2.136
WAND (2 operands)	—	4.731	0.092	3.633	0.017	3.467	0.017	5.341	0.674
WANDP (2 operands)	—	4.498	0.970	4.498	0.882	3.659	0.882	6.075	2.136
WAND (3 operands)	—	4.731	0.121	3.633	0.017	3.467	0.017	5.341	0.674
WANDP (3 operands)	—	4.498	0.970	4.498	0.882	3.659	0.882	6.075	2.136
DAND (2 operands)	—	5.398	0.092	4.167	0.017	3.941	0.017	6.541	0.674
DANDP (2 operands)	—	5.032	0.972	5.032	0.882	4.135	0.882	8.003	2.136
DAND (3 operands)	—	5.403	0.117	4.167	0.017	3.941	0.017	6.541	0.674
DANDP (3 operands)	—	5.032	0.969	5.032	0.882	4.135	0.882	8.003	2.136
BKAND	(n)=K1	7.527	0.089	6.433	0.017	6.033	0.017	8.305	0.674
	(n)=K96	20.104	0.089	16.809	0.017	16.403	0.017	18.655	0.674
BKANDP	(n)=K1	7.618	0.973	6.612	0.882	6.228	0.882	8.932	2.136
	(n)=K96	20.135	0.973	16.897	0.882	16.617	0.882	19.172	2.136
WOR (2 operands)	—	4.731	0.120	3.633	0.017	3.467	0.017	5.341	0.674
WORP (2 operands)	—	4.498	0.971	4.498	0.882	3.659	0.882	6.075	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
WOR (3 operands)	—	4.731	0.120	3.633	0.017	3.467	0.017	5.341	0.674
WORP (3 operands)	—	4.498	0.971	4.498	0.882	3.659	0.882	6.075	2.136
DOR (2 operands)	—	5.434	0.117	4.167	0.017	3.941	0.017	6.541	0.674
DORP (2 operands)	—	5.032	0.969	5.032	0.882	4.135	0.882	8.003	2.136
DOR (3 operands)	—	5.434	0.117	4.167	0.017	3.941	0.017	6.541	0.674
DORP (3 operands)	—	5.032	0.972	5.032	0.882	4.135	0.882	8.003	2.136
BKOR	(n)=K1	7.506	0.087	6.433	0.017	5.983	0.017	8.240	0.674
	(n)=K96	20.099	0.087	16.833	0.017	16.293	0.017	18.553	0.674
BKORP	(n)=K1	7.632	0.971	6.656	0.882	6.175	0.882	8.913	2.136
	(n)=K96	19.932	0.971	16.932	0.882	16.387	0.882	19.168	2.136
WXOR (2 operands)	—	4.734	0.123	3.633	0.017	3.467	0.017	5.341	0.674
WXORP (2 operands)	—	4.498	0.970	4.498	0.882	3.659	0.882	6.075	2.136
WXOR (3 operands)	—	4.731	0.124	3.633	0.017	3.467	0.017	5.341	0.674
WXORP (3 operands)	—	4.498	0.973	4.498	0.882	3.659	0.882	6.075	2.136
DXOR (2 operands)	—	5.398	0.121	4.167	0.017	3.941	0.017	6.541	0.674
DXORP (2 operands)	—	5.032	0.972	5.032	0.882	4.135	0.882	8.003	2.136
DXOR (3 operands)	—	5.398	0.120	4.167	0.017	3.941	0.017	6.541	0.674
DXORP (3 operands)	—	5.032	0.973	5.032	0.882	4.135	0.882	8.003	2.136
BKXOR	(n)=K1	7.523	0.088	6.400	0.017	6.016	0.017	8.289	0.674
	(n)=K96	20.099	0.088	16.783	0.017	16.375	0.017	18.645	0.674
BKXORP	(n)=K1	7.627	0.970	6.616	0.882	6.213	0.882	8.936	2.136
	(n)=K96	20.135	0.970	16.915	0.882	16.467	0.882	19.352	2.136
WXNR (2 operands)	—	4.764	0.127	3.667	0.017	3.467	0.017	5.341	0.674
WXNRP (2 operands)	—	4.532	0.973	4.532	0.882	3.659	0.882	6.075	2.136
WXNR (3 operands)	—	4.764	0.127	3.667	0.017	3.467	0.017	5.341	0.674
WXNRP (3 operands)	—	4.532	0.971	4.532	0.882	3.659	0.882	6.075	2.136
DXNR (2 operands)	—	5.431	0.117	4.200	0.017	3.941	0.017	6.541	0.674
DXNRP (2 operands)	—	5.065	0.969	5.065	0.882	4.135	0.882	8.003	2.136
DXNR (3 operands)	—	5.431	0.117	4.200	0.017	3.941	0.017	6.541	0.674
DXNRP (3 operands)	—	5.065	0.971	5.065	0.882	4.135	0.882	8.003	2.136
BKXNR	(n)=K1	7.498	0.087	6.433	0.017	6.000	0.017	8.272	0.674
	(n)=K96	20.900	0.087	16.860	0.017	16.340	0.017	18.609	0.674
BKXNRP	(n)=K1	7.704	0.969	6.645	0.882	6.197	0.882	8.897	2.136
	(n)=K96	20.747	0.969	16.928	0.882	16.495	0.882	19.297	2.136
BSET	(n)=K1	5.167	0.124	4.033	0.017	3.867	0.017	6.135	0.674
	(n)=K15	5.167	0.124	4.033	0.017	3.867	0.017	6.135	0.674
BSETP	(n)=K1	4.898	0.972	4.898	0.882	4.732	0.882	7.597	2.136
	(n)=K15	4.898	0.972	4.898	0.882	4.732	0.882	7.597	2.136
BRST	(n)=K1	5.200	0.124	4.033	0.017	3.867	0.017	6.135	0.674
	(n)=K15	5.200	0.124	4.033	0.017	3.867	0.017	6.135	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
BRSTP	(n)=K1	4.898	0.972	4.898	0.882	4.732	0.882	7.597	2.136
	(n)=K15	4.898	0.972	4.898	0.882	4.732	0.882	7.597	2.136
TEST	—	4.664	0.120	3.767	0.017	3.608	0.017	5.467	0.674
TESTP	—	4.632	0.970	4.632	0.882	3.792	0.882	6.177	2.136
DTEST	—	4.898	0.124	3.767	0.017	3.608	0.017	5.467	0.674
DTESTP	—	4.632	0.970	4.632	0.882	3.792	0.882	6.177	2.136
BKRST	(n)=K1	3.230	0.087	2.567	0.017	2.408	0.017	3.749	0.674
	(n)=K96	3.330	0.087	2.700	0.017	2.525	0.017	3.867	0.674
BKRSTP	(n)=K1	3.358	0.972	2.745	0.882	2.611	0.882	4.515	2.136
	(n)=K96	3.495	0.972	2.868	0.882	2.724	0.882	4.571	2.136
BCD	—	3.231	0.092	2.700	0.017	2.556	0.017	3.799	0.674
BCDP	—	3.565	0.969	3.565	0.882	3.421	0.882	4.512	2.136
DBCD	—	3.598	0.092	3.067	0.017	2.931	0.017	4.849	0.674
DBCDP	—	3.932	0.967	3.932	0.882	3.161	0.882	6.311	2.136
BIN	—	2.998	0.092	2.567	0.017	2.381	0.017	3.641	0.674
BINP	—	3.432	0.970	3.432	0.882	2.653	0.882	4.363	2.136
DBIN	—	3.264	0.092	2.833	0.017	2.660	0.017	4.591	0.674
DBINP	—	3.698	0.971	3.698	0.882	2.907	0.882	6.053	2.136
FLT2INT	—	3.198	0.087	3.183	0.017	3.183	0.017	4.659	0.674
FLT2INTP	—	3.404	0.968	3.368	0.882	3.368	0.882	5.360	2.136
FLT2UINT	—	3.231	0.088	3.183	0.017	3.183	0.017	4.659	0.674
FLT2UINTP	—	3.395	0.969	3.368	0.882	3.368	0.882	5.360	2.136
FLT2DINT	—	3.203	0.087	3.183	0.017	3.183	0.017	4.659	0.674
FLT2DINTP	—	3.414	0.970	3.368	0.882	3.368	0.882	5.360	2.136
FLT2UDINT	—	3.264	0.087	3.183	0.017	3.183	0.017	4.659	0.674
FLT2UDINTP	—	3.452	0.970	3.368	0.882	3.368	0.882	5.360	2.136
INT2UINT	—	3.931	0.110	3.183	0.017	3.183	0.017	4.659	0.674
INT2UINTP	—	3.368	0.971	3.368	0.882	3.368	0.882	5.360	2.136
INT2DINT	—	4.264	0.113	3.183	0.017	3.183	0.017	4.659	0.674
INT2DINTP	—	3.368	0.970	3.368	0.882	3.368	0.882	5.360	2.136
INT2UDINT	—	4.232	0.114	3.183	0.017	3.183	0.017	4.659	0.674
INT2UDINTP	—	3.368	0.972	3.368	0.882	3.368	0.882	5.360	2.136
UINT2INT	—	3.998	0.110	3.183	0.017	3.183	0.017	4.659	0.674
UINT2INTP	—	3.368	0.968	3.368	0.882	3.368	0.882	5.360	2.136
UINT2DINT	—	4.264	0.113	3.183	0.017	3.183	0.017	4.659	0.674
UINT2DINTP	—	3.368	0.970	3.368	0.882	3.368	0.882	5.360	2.136
UINT2UDINT	—	4.268	0.114	3.183	0.017	3.183	0.017	4.659	0.674
UINT2UDINTP	—	3.368	0.971	3.368	0.882	3.368	0.882	5.360	2.136
DINT2INT	—	4.298	0.110	3.333	0.017	3.183	0.017	4.659	0.674
DINT2INTP	—	4.198	0.971	4.198	0.882	3.368	0.882	5.360	2.136
DINT2UINT	—	4.298	0.111	3.333	0.017	3.183	0.017	4.659	0.674
DINT2UINTP	—	4.198	0.969	4.198	0.882	3.368	0.882	5.360	2.136
DINT2UDINT	—	4.464	0.111	3.333	0.017	3.183	0.017	4.659	0.674
DINT2UDINTP	—	4.198	0.970	4.198	0.882	3.368	0.882	5.360	2.136
UDINT2INT	—	4.298	0.110	3.333	0.017	3.183	0.017	4.659	0.674
UDINT2INTP	—	4.198	0.972	4.198	0.882	3.368	0.882	5.360	2.136
UDINT2UINT	—	4.298	0.110	3.333	0.017	3.183	0.017	4.659	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
UDINT2UINTP	—	4.198	0.972	4.198	0.882	3.368	0.882	5.360	2.136
UDINT2DINT	—	4.466	0.111	3.333	0.017	3.183	0.017	4.659	0.674
UDINT2DINTP	—	4.198	0.972	4.198	0.882	3.368	0.882	5.360	2.136
GRY	—	2.702	0.087	2.300	0.017	2.167	0.017	3.357	0.674
GRYP	—	2.894	0.969	2.515	0.882	2.369	0.882	4.060	2.137
GRY_U	—	2.698	0.087	2.300	0.017	2.167	0.017	3.357	0.674
GRYP_U	—	2.902	0.970	2.515	0.882	2.369	0.882	4.060	2.137
DGRY	—	2.732	0.087	2.300	0.017	2.184	0.017	4.075	0.674
DGRYP	—	2.854	0.970	2.515	0.882	2.388	0.882	5.537	2.137
DGRY_U	—	2.731	0.087	2.300	0.017	2.184	0.017	4.075	0.674
DGRYP_U	—	2.848	0.973	2.515	0.882	2.388	0.882	5.537	2.137
GBIN	—	4.098	0.087	3.100	0.017	2.959	0.017	4.132	0.674
GBINP	—	4.268	0.969	3.276	0.882	3.163	0.882	4.867	2.136
GBIN_U	—	4.102	0.087	3.100	0.017	2.959	0.017	4.132	0.674
GBINP_U	—	4.270	0.968	3.276	0.882	3.163	0.882	4.867	2.136
DGBIN	—	4.103	0.087	3.100	0.017	2.959	0.017	4.849	0.674
DGBINP	—	4.296	0.973	3.276	0.882	3.163	0.882	6.311	2.136
DGBIN_U	—	4.098	0.087	3.100	0.017	2.959	0.017	4.849	0.674
DGBINP_U	—	4.288	0.970	3.276	0.882	3.163	0.882	6.311	2.136
DABIN	—	4.531	0.088	3.867	0.017	3.716	0.017	5.075	0.674
DABINP	—	4.690	0.970	4.040	0.882	3.907	0.882	5.812	2.136
DABIN_U	—	4.564	0.087	3.867	0.017	3.716	0.017	5.075	0.674
DABINP_U	—	4.707	0.972	4.040	0.882	3.907	0.882	5.812	2.136
DDABIN	—	5.268	0.087	4.633	0.017	4.475	0.017	6.575	0.674
DDABINP	—	5.420	0.969	4.767	0.882	4.651	0.882	8.037	2.136
DDABIN_U	—	5.364	0.087	4.633	0.017	4.475	0.017	6.575	0.674
DDABINP_U	—	5.518	0.969	4.767	0.882	4.651	0.882	8.037	2.136
HEXA	(n)=K1	19.367	0.091	19.367	0.017	19.367	0.017	21.997	0.674
	(n)=K96	34.244	0.091	25.769	0.017	20.085	0.017	22.713	0.674
HEXAP	(n)=K1	20.232	0.970	20.232	0.882	20.232	0.882	22.651	2.136
	(n)=K96	33.674	0.970	26.634	0.882	20.950	0.882	23.407	2.136
VAL	—	6.366	0.091	5.433	0.017	5.208	0.017	7.656	0.674
VALP	—	6.534	0.968	5.640	0.882	5.420	0.882	8.276	2.136
VAL_U	—	6.299	0.090	5.433	0.017	5.208	0.017	7.656	0.674
VALP_U	—	6.466	0.968	5.640	0.882	5.420	0.882	8.276	2.136
DVAL	—	6.536	0.087	5.433	0.017	5.208	0.017	7.743	0.674
DVALP	—	6.710	0.969	5.640	0.882	5.420	0.882	8.361	2.136
DVAL_U	—	6.499	0.088	5.433	0.017	5.208	0.017	7.743	0.674
DVALP_U	—	6.651	0.968	5.640	0.882	5.420	0.882	8.361	2.136
NEG	—	3.064	0.087	2.500	0.017	2.400	0.017	3.591	0.674
NEGP	—	3.231	0.965	2.675	0.882	2.629	0.882	4.320	2.136
DNEG	—	3.131	0.087	2.517	0.017	2.483	0.017	4.375	0.674
DNEGP	—	3.259	0.969	2.729	0.882	2.668	0.882	5.837	2.136
DECO	(n)=K2	4.664	0.087	4.200	0.017	3.975	0.017	6.697	0.674
	(n)=K8	4.664	0.087	4.408	0.017	4.408	0.017	7.831	0.674
DECOP	(n)=K2	4.828	0.971	4.371	0.882	4.164	0.882	7.329	2.136
	(n)=K8	4.835	0.971	4.593	0.882	4.593	0.882	9.293	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
ENCO	(n)=K2	3.931	0.091	3.367	0.017	3.241	0.017	5.531	0.674
	(n)=K8	4.636	0.091	4.167	0.017	3.625	0.017	5.904	0.674
ENCOP	(n)=K2	4.232	0.972	4.232	0.882	4.106	0.882	6.993	2.136
	(n)=K8	5.032	0.972	5.032	0.882	3.828	0.882	7.366	2.136
DIS	(n)=K1	3.331	0.087	2.867	0.017	2.749	0.017	4.332	0.674
	(n)=K4	3.668	0.087	3.167	0.017	3.067	0.017	4.649	0.674
DISP	(n)=K1	3.539	0.971	3.028	0.882	2.939	0.882	5.037	2.136
	(n)=K4	3.867	0.971	3.340	0.882	3.256	0.882	5.336	2.136
UNI	(n)=K1	3.531	0.087	3.067	0.017	2.949	0.017	4.532	0.674
	(n)=K4	3.900	0.087	3.400	0.017	3.292	0.017	4.875	0.674
UNIP	(n)=K1	3.691	0.973	3.248	0.882	3.147	0.882	5.216	2.136
	(n)=K4	3.691	0.973	3.605	0.882	3.473	0.882	5.567	2.136
NDIS	—	10.643	0.087	8.828	0.017	8.583	0.017	11.224	0.674
NDISP	—	10.640	0.973	8.897	0.882	8.703	0.882	12.686	2.136
NUNI	—	10.646	0.087	8.781	0.017	8.591	0.017	11.233	0.674
NUNIP	—	10.656	0.973	8.921	0.882	8.692	0.882	12.695	2.136
WTOB	(n)=K1	6.064	0.090	5.033	0.017	4.759	0.017	7.289	0.674
	(n)=K96	17.180	0.090	11.699	0.017	11.423	0.017	13.873	0.674
WTOBP	(n)=K1	6.168	0.972	5.271	0.882	4.983	0.882	7.891	2.136
	(n)=K96	17.215	0.972	11.855	0.882	11.548	0.882	14.617	2.136
BTOW	(n)=K1	6.266	0.091	5.267	0.017	5.008	0.017	7.539	0.674
	(n)=K96	20.171	0.091	13.335	0.017	13.083	0.017	15.535	0.674
BTOWP	(n)=K1	6.398	0.971	5.473	0.882	5.225	0.882	8.133	2.136
	(n)=K96	20.080	0.971	13.439	0.882	13.196	0.882	16.319	2.136
MOV	—	3.468	0.084	2.633	0.017	2.549	0.017	4.016	0.674
MOVP	—	3.498	0.970	3.498	0.882	2.744	0.882	4.703	2.136
DMOV	—	3.734	0.084	2.800	0.017	2.683	0.017	4.833	0.674
DMOVP	—	3.665	0.969	3.665	0.882	2.877	0.882	6.295	2.136
CML	—	3.899	0.111	2.967	0.017	2.867	0.017	4.341	0.674
CMLP	—	3.832	0.973	3.832	0.882	3.057	0.882	5.012	2.136
DCML	—	4.399	0.107	3.333	0.017	3.175	0.017	5.359	0.674
DCMLP	—	4.198	0.971	4.198	0.882	3.353	0.882	6.821	2.136
SMOV	(n1)=K4, (n2)=K1, (n3)=K4	6.531	0.087	5.800	0.017	5.549	0.017	8.723	0.674
	(n1)=K4, (n2)=K4, (n3)=K4	6.531	0.087	5.800	0.017	5.549	0.017	8.723	0.674
SMOVP	(n1)=K4, (n2)=K1, (n3)=K4	6.698	0.970	5.967	0.882	5.747	0.882	9.329	2.136
	(n1)=K4, (n2)=K4, (n3)=K4	6.694	0.970	5.967	0.882	5.747	0.882	9.329	2.136
CMLB	—	4.264	0.120	3.267	0.017	3.125	0.017	4.599	0.674
CMLBP	—	4.132	0.969	4.132	0.882	3.300	0.882	5.329	2.136
BMOV	(n)=K1	5.198	0.087	4.333	0.017	4.208	0.017	5.975	0.674
	(n)=K96	7.827	0.087	6.867	0.017	6.767	0.017	8.599	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
BMOVP	(n)=K1	5.335	0.973	4.504	0.882	4.385	0.882	6.673	2.136
	(n)=K96	7.970	0.973	7.068	0.882	6.959	0.882	9.193	2.136
FMOV	(n)=K1	3.898	0.087	3.267	0.017	3.175	0.017	4.757	0.674
	(n)=K96	8.484	0.087	5.667	0.017	5.575	0.017	7.239	0.674
FMOVP	(n)=K1	4.079	0.972	3.436	0.882	3.357	0.882	5.440	2.136
	(n)=K96	8.491	0.972	5.829	0.882	5.756	0.882	7.900	2.136
DFMOV	(n)=K1	3.998	0.087	3.300	0.017	3.149	0.017	5.432	0.674
	(n)=K96	9.388	0.087	7.033	0.017	6.956	0.017	9.256	0.674
DFMOVP	(n)=K1	4.135	0.971	3.461	0.882	3.339	0.882	6.894	2.136
	(n)=K96	9.390	0.971	7.240	0.882	7.079	0.882	10.718	2.136
XCH	—	6.366	0.124	4.967	0.017	4.708	0.017	7.047	0.674
XCHP	—	5.832	0.970	5.832	0.882	4.899	0.882	7.757	2.136
DXCH	—	7.224	0.120	5.633	0.017	5.300	0.017	8.340	0.674
DXCHP	—	6.498	0.967	6.498	0.882	5.491	0.882	9.802	2.136
SWAP	—	4.631	0.110	3.533	0.017	3.383	0.017	4.857	0.674
SWAPP	—	4.398	0.969	4.398	0.882	3.547	0.882	5.539	2.136
DSWAP	—	5.098	0.110	3.867	0.017	3.692	0.017	5.867	0.674
DSWAPP	—	4.732	0.970	4.732	0.882	3.864	0.882	7.329	2.136
MOVB	—	4.264	0.111	3.233	0.017	3.092	0.017	4.567	0.674
MOVBP	—	4.098	0.970	4.098	0.882	3.263	0.882	5.256	2.136
BLKMOVB	(n)=K1	5.331	0.087	4.400	0.017	4.108	0.017	5.859	0.674
	(n)=K96	5.664	0.087	4.833	0.017	4.549	0.017	6.331	0.674
BLKMOVBP	(n)=K1	5.471	0.969	4.580	0.882	4.291	0.882	6.539	2.136
	(n)=K96	5.786	0.969	4.976	0.882	4.744	0.882	7.096	2.136
PRUN	—	7.224	0.087	5.667	0.017	5.375	0.017	6.931	0.674
PRUNP	—	7.338	0.970	5.823	0.882	5.557	0.882	7.608	2.136
DPRUN	—	8.351	0.087	6.467	0.017	6.167	0.017	8.431	0.674
DPRUNP	—	8.366	0.968	6.632	0.882	6.357	0.882	8.348	2.136
ROR	(n)=K1	3.452	0.087	3.069	0.017	2.917	0.017	4.920	0.674
	(n)=K15	3.452	0.087	3.069	0.017	2.917	0.017	4.920	0.674
RORP	(n)=K1	3.735	0.970	3.240	0.882	3.137	0.882	6.382	2.136
	(n)=K15	3.735	0.970	3.240	0.882	3.137	0.882	6.382	2.136
RCR	(n)=K1	3.502	0.087	3.125	0.017	2.975	0.017	4.988	0.674
	(n)=K15	3.508	0.087	3.125	0.017	2.975	0.017	4.988	0.674
RCRP	(n)=K1	3.766	0.971	3.303	0.882	3.204	0.882	6.450	2.136
	(n)=K15	3.830	0.971	3.303	0.882	3.204	0.882	6.450	2.136
DROR	(n)=K1	3.509	0.087	3.116	0.017	2.941	0.017	4.953	0.674
	(n)=K31	3.515	0.087	3.116	0.017	2.941	0.017	4.953	0.674
DRORP	(n)=K1	3.774	0.968	3.304	0.882	3.177	0.882	6.415	2.136
	(n)=K31	3.759	0.968	3.304	0.882	3.177	0.882	6.415	2.136
DRCR	(n)=K1	3.558	0.087	3.175	0.017	3.000	0.017	5.022	0.674
	(n)=K31	3.574	0.087	3.175	0.017	3.000	0.017	5.022	0.674
DRCRP	(n)=K1	3.796	0.972	3.379	0.882	3.221	0.882	6.484	2.136
	(n)=K31	3.823	0.972	3.379	0.882	3.221	0.882	6.484	2.136
ROL	(n)=K1	3.434	0.087	3.023	0.017	2.917	0.017	4.920	0.674
	(n)=K15	3.435	0.087	3.023	0.017	2.917	0.017	4.920	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
ROLP	(n)=K1	3.775	0.973	3.247	0.882	3.136	0.882	6.382	2.136
	(n)=K15	3.770	0.973	3.247	0.882	3.136	0.882	6.382	2.136
RCL	(n)=K1	3.502	0.088	3.118	0.017	2.975	0.017	4.987	0.674
	(n)=K15	3.518	0.088	3.118	0.017	2.975	0.017	4.987	0.674
RCLP	(n)=K1	3.780	0.970	3.304	0.882	3.212	0.882	6.449	2.136
	(n)=K15	3.784	0.970	3.304	0.882	3.212	0.882	6.449	2.136
DROL	(n)=K1	3.492	0.087	3.102	0.017	2.941	0.017	4.961	0.674
	(n)=K31	3.498	0.087	3.102	0.017	2.941	0.017	4.961	0.674
DROLP	(n)=K1	3.818	0.972	3.276	0.882	3.172	0.882	6.423	2.136
	(n)=K31	3.767	0.972	3.276	0.882	3.172	0.882	6.423	2.136
DRCL	(n)=K1	3.562	0.087	3.165	0.017	3.000	0.017	5.023	0.674
	(n)=K31	3.587	0.087	3.165	0.017	3.000	0.017	5.023	0.674
DRCLP	(n)=K1	3.819	0.971	3.343	0.882	3.215	0.882	6.485	2.136
	(n)=K31	3.896	0.971	3.343	0.882	3.215	0.882	6.485	2.136
CJ	—	3.300	0.087	2.504	0.017	2.317	0.017	3.883	0.674
CJP	—	3.467	0.982	2.761	0.882	2.493	0.882	4.560	2.136
GOEND	—	0.108	0.083	0.091	0.017	0.091	0.017	0.408	0.674
DI (0 operands)	—	1.828	0.088	1.168	0.017	1.168	0.017	1.792	0.674
DI (1 operand)	—	2.840	0.090	2.107	0.017	2.107	0.017	3.747	0.674
EI	—	1.828	0.087	1.263	0.017	1.235	0.017	1.859	0.674
IMASK	—	7.570	—	6.441	—	5.183	—	6.879	—
SIMASK	—	4.099	0.088	3.000	0.017	2.833	0.017	4.308	0.674
IRET	—	0.100	—	0.050	—	0.050	—	0.358	—
WDT	—	1.498	0.087	1.141	0.017	1.141	0.017	1.832	0.674
WDTP	—	1.774	0.972	1.540	0.882	1.452	0.882	2.657	2.136
FOR	(n)=K1	2.444	—	2.023	—	2.023	—	4.109	—
NEXT	—	1.386	—	1.227	—	1.156	—	2.325	—
BREAK	—	4.172	0.104	3.203	0.017	2.971	0.017	5.249	0.674
BREAKP	—	4.450	1.037	3.515	0.882	3.215	0.882	5.661	2.136
CALL	—	4.508	0.100	3.851	0.017	3.335	0.017	5.816	0.674
CALLP	—	4.800	1.020	4.111	0.882	3.489	0.882	6.224	2.136
RET	—	2.474	—	2.279	—	1.939	—	3.448	—
SRET	—	2.470	—	2.279	—	1.939	—	3.448	—
XCALL	—	4.771	0.894	4.077	0.882	3.544	0.882	6.335	2.136
SFRD	(n)=K2	5.899	0.087	5.069	0.017	4.864	0.017	7.123	0.674
	(n)=K97	10.134	0.087	9.039	0.017	8.905	0.017	11.132	0.674
SFRDP	(n)=K2	6.215	0.977	5.344	0.882	5.097	0.882	8.585	2.136
	(n)=K97	10.486	0.977	9.299	0.882	9.111	0.882	11.568	2.136
POP	(n)=K2	5.072	0.088	4.335	0.017	4.119	0.017	6.392	0.674
	(n)=K97	5.022	0.088	4.312	0.017	4.104	0.017	6.383	0.674
POPP	(n)=K2	5.334	0.972	4.563	0.882	4.359	0.882	7.854	2.136
	(n)=K97	5.372	0.972	4.583	0.882	4.364	0.882	6.776	2.136
SFWR	(n)=K2	3.802	0.088	3.067	0.017	2.941	0.017	4.507	0.674
	(n)=K97	3.798	0.088	3.067	0.017	2.941	0.017	4.507	0.674
SFWRP	(n)=K2	3.924	0.970	3.204	0.882	3.127	0.882	5.219	2.136
	(n)=K97	3.916	0.970	3.203	0.882	3.135	0.882	5.200	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
FINS	(n)=K1	5.768	0.087	4.877	0.017	4.623	0.017	6.879	0.674
	(n)=K96	5.868	0.087	4.892	0.017	4.724	0.017	6.975	0.674
FINSP	(n)=K1	6.080	0.971	5.111	0.882	4.859	0.882	8.341	2.136
	(n)=K96	6.163	0.971	5.180	0.882	4.967	0.882	7.404	2.136
FDEL	(n)=K1	6.372	0.087	5.373	0.017	5.124	0.017	7.387	0.674
	(n)=K96	6.627	0.087	5.561	0.017	5.364	0.017	7.576	0.674
FDELP	(n)=K1	6.692	0.971	5.635	0.882	5.355	0.882	8.849	2.136
	(n)=K96	6.974	0.971	5.841	0.882	5.579	0.882	8.001	2.136
LD$=	—	6.458	—	6.384	—	6.384	—	8.319	—
LD$<>	—	6.384	—	6.384	—	6.384	—	8.319	—
LD$>	—	6.384	—	6.384	—	6.384	—	8.319	—
LD$<=	—	6.384	—	6.384	—	6.384	—	8.319	—
LD$<	—	6.384	—	6.384	—	6.384	—	8.319	—
LD$>=	—	6.384	—	6.384	—	6.384	—	8.319	—
AND$=	—	6.396	—	6.384	—	6.384	—	8.319	—
AND$<>	—	6.396	—	6.384	—	6.384	—	8.319	—
AND$>	—	6.384	—	6.384	—	6.384	—	8.319	—
AND$<=	—	6.396	—	6.384	—	6.384	—	8.319	—
AND$<	—	6.398	—	6.384	—	6.384	—	8.319	—
AND$>=	—	6.384	—	6.384	—	6.384	—	8.319	—
OR$=	—	6.396	—	6.384	—	6.384	—	8.319	—
OR$<>	—	6.384	—	6.384	—	6.384	—	8.319	—
OR$>	—	6.384	—	6.384	—	6.384	—	8.319	—
OR$<=	—	6.384	—	6.384	—	6.384	—	8.319	—
OR$<	—	6.398	—	6.384	—	6.384	—	8.319	—
OR$>=	—	6.384	—	6.384	—	6.384	—	8.319	—
$+ (2 operands)	—	12.344	0.087	10.188	0.017	10.039	0.017	12.591	0.674
$+P (2 operands)	—	12.415	0.969	10.308	0.882	10.139	0.882	14.053	2.136
$+ (3 operands)	—	12.343	0.084	10.188	0.017	10.039	0.017	12.591	0.674
$+P (3 operands)	—	12.376	0.972	10.308	0.882	10.139	0.882	14.053	2.136
$MOV	—	6.632	0.091	5.433	0.017	5.300	0.017	7.605	0.674
$MOVP	—	6.806	0.968	5.673	0.882	5.415	0.882	8.145	2.136
$MOV_WS	0 characters	6.299	0.087	4.965	0.018	4.891	0.016	7.505	0.700
	32 characters	9.788	0.087	7.800	0.018	7.733	0.016	10.363	0.700
$MOVP_WS	0 characters	6.139	0.973	5.225	0.882	5.121	0.016	8.177	0.700
	32 characters	9.055	0.973	8.128	0.882	7.951	0.016	10.939	0.700
BINDA	—	4.403	0.090	3.500	0.017	3.349	0.017	5.407	0.674
BINDAP	—	4.572	0.968	3.703	0.882	3.541	0.882	6.092	2.136
BINDA_U	—	4.331	0.090	3.500	0.017	3.349	0.017	5.407	0.674
BINDAP_U	—	4.508	0.970	3.703	0.882	3.541	0.882	6.092	2.136
DBINDA	—	5.002	0.087	4.867	0.017	4.692	0.017	6.827	0.674
DBINDAP	—	5.175	0.969	5.075	0.882	4.885	0.882	6.760	2.136
DBINDA_U	—	4.967	0.087	4.867	0.017	4.692	0.017	6.827	0.674
DBINDAP_U	—	5.151	0.970	5.075	0.882	4.885	0.882	6.760	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
ASCI	(n)=K1 SM8161 =OFF	6.432	0.087	5.300	0.017	5.075	0.017	7.605	0.674
	(n)=K96 SM8161 =OFF	20.598	0.087	16.951	0.017	16.715	0.017	19.229	0.674
	(n)=K1 SM8161=ON	6.531	0.087	5.433	0.017	5.192	0.017	9.067	0.674
	(n)=K96 SM8161=ON	21.660	0.087	14.633	0.017	14.399	0.017	16.929	0.674
ASCIP	(n)=K1 SM8161 =OFF	6.610	0.973	5.517	0.882	5.267	0.882	7.856	2.136
	(n)=K96 SM8161 =OFF	20.527	0.973	17.051	0.882	16.832	0.882	19.427	2.136
	(n)=K1 SM8161=ON	6.636	0.973	5.568	0.882	5.391	0.882	9.318	2.136
	(n)=K96 SM8161=ON	21.474	0.973	14.728	0.882	14.528	0.882	16.929	2.136
STR	—	6.431	0.087	5.377	0.017	5.175	0.017	7.620	0.674
STRP	—	6.600	0.968	5.568	0.882	5.352	0.882	9.082	2.136
STR_U	—	6.431	0.088	5.377	0.017	5.175	0.017	7.620	0.674
STRP_U	—	6.608	0.970	5.568	0.882	5.352	0.882	9.082	2.136
DSTR	—	6.364	0.084	5.377	0.017	5.175	0.017	7.620	0.674
DSTRP	—	6.486	0.970	5.568	0.882	5.352	0.882	9.082	2.136
DSTR_U	—	6.366	0.084	5.377	0.017	5.175	0.017	7.620	0.674
DSTRP_U	—	6.495	0.970	5.568	0.882	5.352	0.882	9.082	2.136
ESTR	—	19.296	0.090	16.267	0.017	16.044	0.017	18.660	0.674
ESTRP	—	19.159	0.969	16.355	0.882	16.125	0.882	20.122	2.136
DESTR	—	19.306	0.090	16.267	0.017	16.044	0.017	18.660	0.674
DESTRP	—	19.264	0.967	16.355	0.882	16.125	0.882	20.122	2.136
WS2SJIS	The number of characters = 1	11.044	0.088	8.452	0.018	8.379	0.016	11.376	0.700
	The number of characters = 96	44.459	0.088	43.433	0.018	43.025	0.016	45.961	0.700
WS2SJISP	The number of characters = 1	11.143	0.972	8.759	0.882	8.544	0.016	11.653	0.700
	The number of characters = 96	46.500	0.972	43.696	0.882	43.167	0.016	46.636	0.700
SJIS2WS	The number of characters = 1	10.042	0.087	7.799	0.018	7.717	0.016	10.355	0.700
	The number of characters = 96	52.647	0.087	48.516	0.018	48.311	0.016	52.480	0.700

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
SJIS2WSP	The number of characters = 1	10.481	0.976	8.132	0.882	7.963	0.016	11.299	0.700
	The number of characters = 96	54.096	0.976	48.749	0.882	48.531	0.016	52.840	0.700
SJIS2WSB	The number of characters = 1	10.222	0.087	7.905	0.018	7.849	0.016	10.487	0.700
	The number of characters = 96	52.714	0.087	48.515	0.018	48.317	0.016	52.521	0.700
SJIS2WSBP	The number of characters = 1	10.576	0.976	8.232	0.882	8.064	0.016	11.416	0.700
	The number of characters = 96	54.039	0.976	48.763	0.882	48.539	0.016	52.871	0.700
LEN	—	4.198	0.091	3.400	0.017	3.308	0.017	5.367	0.674
LENP	—	4.379	0.969	3.643	0.882	3.527	0.882	6.103	2.136
RIGHT	(n)=K1 The number of character strings to be extracted: 100	16.008	0.084	14.632	0.017	14.524	0.017	17.051	0.674
	(n)=K96 The number of character strings to be extracted: 100	16.707	0.084	15.332	0.017	15.216	0.017	17.749	0.674
RIGHTP	(n)=K1 The number of character strings to be extracted: 100	16.063	0.973	14.792	0.882	14.627	0.882	18.513	2.136
	(n)=K96 The number of character strings to be extracted: 100	16.695	0.973	15.471	0.882	15.332	0.882	17.733	2.136
LEFT	(n)=K1 The number of character strings to be extracted: 100	15.943	0.084	14.665	0.017	14.524	0.017	17.051	0.674
	(n)=K96 The number of character strings to be extracted: 100	16.680	0.084	15.365	0.017	15.216	0.017	17.749	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
LEFTP	(n)=K1 The number of character strings to be extracted: 100	16.076	0.973	14.768	0.882	14.627	0.882	18.513	2.136
	(n)=K96 The number of character strings to be extracted: 100	16.668	0.973	15.501	0.882	15.332	0.882	17.733	2.136
MIDR	—	7.098	0.090	5.800	0.017	5.608	0.017	8.332	0.674
MIDRP	—	7.252	0.969	6.017	0.882	5.799	0.882	9.794	2.136
MIDW	—	7.531	0.087	6.200	0.017	6.075	0.017	8.081	0.674
MIDWP	—	7.676	0.971	6.423	0.882	6.247	0.882	8.736	2.136
INSTR	—	6.936	0.090	5.867	0.017	5.659	0.017	8.481	0.674
INSTRP	—	7.103	0.968	6.075	0.882	5.851	0.882	9.943	2.136
STRINS	(s1)=1 character, (d)=128 characters, (s2)=K40	26.976	0.087	24.384	0.017	24.241	0.017	26.772	0.674
	(s1)=48 characters, (d)=128 characters, (s2)=K40	33.352	0.087	30.327	0.017	30.131	0.017	31.904	0.674
STRINSP	(s1)=1 character, (d)=128 characters, (s2)=K40	26.987	0.973	24.516	0.882	24.427	0.882	28.234	2.136
	(s1)=48 characters, (d)=128 characters, (s2)=K40	32.776	0.973	31.593	0.882	31.593	0.882	32.579	2.136
STRDEL	(d)=128 characters, (s)=K40, (n)=K1	21.535	0.084	19.905	0.017	19.823	0.017	22.069	0.674
	(d)=128 characters, (s)=K40, (n)=K48	20.858	0.084	19.328	0.017	19.224	0.017	21.449	0.674
STRDELP	(d)=128 characters, (s)=K40, (n)=K1	21.648	0.972	20.044	0.882	20.044	0.882	23.531	2.136
	(d)=128 characters, (s)=K40, (n)=K48	21.183	0.972	19.416	0.882	19.416	0.882	22.911	2.136
LDE=	—	2.566	—	2.247	—	2.100	—	3.980	—
LDE<>	—	2.563	—	2.247	—	2.100	—	3.980	—
LDE>	—	2.564	—	2.247	—	2.100	—	3.980	—
LDE<=	—	2.570	—	2.247	—	2.100	—	3.980	—

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
LDE<	—	2.536	—	2.247	—	2.100	—	3.980	—
LDE>=	—	2.530	—	2.247	—	2.100	—	3.980	—
ANDE=	—	2.763	—	2.247	—	2.100	—	3.980	—
ANDE<>	—	2.635	—	2.247	—	2.100	—	3.980	—
ANDE>	—	2.666	—	2.247	—	2.100	—	3.980	—
ANDE<=	—	2.602	—	2.247	—	2.100	—	3.980	—
ANDE<	—	2.596	—	2.247	—	2.100	—	3.980	—
ANDE>=	—	2.598	—	2.247	—	2.100	—	3.980	—
ORE=	—	2.663	—	2.247	—	2.100	—	3.980	—
ORE<>	—	2.630	—	2.247	—	2.100	—	3.980	—
ORE>	—	2.634	—	2.247	—	2.100	—	3.980	—
ORE<=	—	2.602	—	2.247	—	2.100	—	3.980	—
ORE<	—	2.635	—	2.247	—	2.100	—	3.980	—
ORE>=	—	2.630	—	2.247	—	2.100	—	3.980	—
DECMP	—	4.231	0.088	3.600	0.017	3.400	0.017	4.991	0.674
DECMPP	—	4.438	0.968	3.780	0.882	3.581	0.882	5.665	2.136
DEZCP	—	4.698	0.087	4.100	0.017	3.859	0.017	6.499	0.674
DEZCPP	—	4.950	0.968	4.337	0.882	4.055	0.882	7.961	2.136
E+ (2 operands)	—	4.498	0.087	3.967	0.017	3.692	0.017	5.389	0.674
E+P (2 operands)	—	4.656	0.971	4.180	0.882	3.888	0.882	6.101	2.136
E+ (3 operands)	—	3.998	0.087	3.967	0.017	3.692	0.017	5.389	0.674
E+P (3 operands)	—	4.180	0.971	4.180	0.882	3.888	0.882	6.101	2.136
DEADD	—	3.998	0.087	3.967	0.017	3.692	0.017	5.389	0.674
DEADDP	—	4.180	0.973	4.180	0.882	3.888	0.882	6.101	2.136
E- (2 operands)	—	4.532	0.087	4.000	0.017	3.700	0.017	5.391	0.674
E-P (2 operands)	—	4.668	0.972	4.172	0.882	3.901	0.882	6.076	2.136
E- (3 operands)	—	4.098	0.087	4.000	0.017	3.700	0.017	5.391	0.674
E-P (3 operands)	—	4.224	0.968	4.172	0.882	3.901	0.882	6.076	2.136
DESUB	—	4.098	0.087	4.000	0.017	3.700	0.017	5.391	0.674
DESUBP	—	4.228	0.970	4.172	0.882	3.901	0.882	6.076	2.136
E*	—	4.134	0.087	3.400	0.017	3.208	0.017	4.633	0.674
E*P	—	4.268	0.971	3.575	0.882	3.395	0.882	5.344	2.136
DEMUL	—	4.131	0.087	3.400	0.017	3.208	0.017	4.633	0.674
DEMULP	—	4.268	0.971	3.575	0.882	3.395	0.882	5.344	2.136
E/	—	3.998	0.087	3.433	0.017	3.259	0.017	4.683	0.674
E/P	—	4.135	0.971	3.603	0.882	3.439	0.882	5.395	2.136
DEDIV	—	3.998	0.087	3.433	0.017	3.259	0.017	4.683	0.674
DEDIVP	—	4.136	0.971	3.603	0.882	3.439	0.882	5.395	2.136
INT2FLT	—	4.303	0.110	3.333	0.017	3.183	0.017	4.657	0.674
INT2FLTP	—	4.645	0.970	4.645	0.882	4.645	0.882	6.119	2.136
UINT2FLT	—	4.264	0.111	3.333	0.017	3.183	0.017	4.657	0.674
UINT2FLTP	—	4.048	0.969	4.048	0.882	4.048	0.882	6.119	2.136
DINT2FLT	—	4.468	0.110	3.333	0.017	3.183	0.017	4.657	0.674
DINT2FLTP	—	4.645	0.970	4.645	0.882	4.645	0.882	6.119	2.136
UDINT2FLT	—	4.466	0.111	3.333	0.017	3.183	0.017	4.657	0.674
UDINT2FLTP	—	4.048	0.971	4.048	0.882	4.048	0.882	6.119	2.136
EVAL	—	5.298	0.090	4.533	0.017	4.433	0.017	6.572	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
EVALP	—	5.467	0.970	4.669	0.882	4.620	0.882	7.268	2.136
DEVAL	—	5.300	0.091	4.533	0.017	4.433	0.017	6.572	0.674
DEVALP	—	5.468	0.971	4.735	0.882	4.620	0.882	7.268	2.136
DEBCD	—	3.698	0.087	3.033	0.017	2.925	0.017	4.816	0.674
DEBCDP	—	3.868	0.971	3.248	0.882	3.124	0.882	6.278	2.136
DEBIN	—	3.698	0.088	3.033	0.017	2.916	0.017	4.107	0.674
DEBINP	—	3.899	0.971	3.241	0.882	3.112	0.882	4.784	2.136
ENEG	—	3.064	0.087	2.491	0.017	2.449	0.017	3.641	0.674
ENEGP	—	3.234	0.969	2.745	0.882	2.640	0.882	4.337	2.136
DENEG	—	3.067	0.087	2.491	0.017	2.449	0.017	3.641	0.674
DENEGP	—	3.236	0.972	2.745	0.882	2.640	0.882	4.337	2.136
EMOV	—	4.402	0.110	3.333	0.017	3.167	0.017	4.641	0.674
EMOVP	—	4.032	0.969	4.032	0.882	4.032	0.882	6.103	2.136
DEMOV	—	4.399	0.110	3.333	0.017	3.167	0.017	4.641	0.674
DEMOVP	—	4.032	0.973	4.032	0.882	4.032	0.882	6.103	2.136
SIN	—	4.302	0.087	3.584	0.017	3.584	0.017	5.475	0.674
SINP	—	4.510	0.968	3.788	0.882	3.788	0.882	6.937	2.136
DSIN	—	4.298	0.089	3.584	0.017	3.584	0.017	5.475	0.674
DSINP	—	4.504	0.971	3.788	0.882	3.788	0.882	6.937	2.136
COS	—	4.298	0.087	3.584	0.017	3.584	0.017	5.475	0.674
COSP	—	4.504	0.969	3.788	0.882	3.788	0.882	6.937	2.136
DCOS	—	4.298	0.087	3.584	0.017	3.584	0.017	5.475	0.674
DCOSP	—	4.499	0.973	3.788	0.882	3.788	0.882	6.937	2.136
TAN	—	4.367	0.087	3.675	0.017	3.584	0.017	5.475	0.674
TANP	—	4.534	0.968	3.879	0.882	3.788	0.882	6.937	2.136
DTAN	—	4.364	0.087	3.675	0.017	3.584	0.017	5.475	0.674
DTANP	—	4.534	0.969	3.879	0.882	3.788	0.882	6.937	2.136
ASIN	—	3.631	0.087	2.967	0.017	2.967	0.017	4.859	0.674
ASINP	—	3.772	0.970	3.179	0.882	3.179	0.882	6.321	2.136
DASIN	—	3.631	0.088	2.967	0.017	2.967	0.017	4.859	0.674
DASINP	—	3.771	0.969	3.179	0.882	3.179	0.882	6.321	2.136
ACOS	—	3.831	0.087	3.045	0.017	2.967	0.017	4.859	0.674
ACOSP	—	4.055	0.969	3.279	0.882	3.179	0.882	6.321	2.136
DACOS	—	3.831	0.088	3.045	0.017	2.967	0.017	4.859	0.674
DACOSP	—	4.034	0.970	3.279	0.882	3.179	0.882	6.321	2.136
ATAN	—	3.631	0.087	3.013	0.017	2.967	0.017	4.859	0.674
ATANP	—	3.834	0.969	3.251	0.882	3.179	0.882	6.321	2.136
DATAN	—	3.636	0.087	3.013	0.017	2.967	0.017	4.859	0.674
DATANP	—	3.816	0.971	3.251	0.882	3.179	0.882	6.321	2.136
RAD	—	3.331	0.088	2.772	0.017	2.667	0.017	4.559	0.674
RADP	—	3.470	0.965	2.980	0.882	2.873	0.882	6.021	2.136
DRAD	—	3.331	0.087	2.772	0.017	2.667	0.017	4.559	0.674
DRADP	—	3.467	0.971	2.980	0.882	2.873	0.882	6.021	2.136
DEG	—	3.300	0.087	2.780	0.017	2.675	0.017	4.567	0.674
DEGP	—	3.508	0.970	2.980	0.882	2.873	0.882	6.029	2.136
DDEG	—	3.298	0.088	2.780	0.017	2.675	0.017	4.567	0.674
DDEGP	—	3.532	0.971	2.980	0.882	2.873	0.882	6.029	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
ESQRT	—	3.231	0.087	2.733	0.017	2.608	0.017	3.799	0.674
ESQRTP	—	3.430	0.972	2.920	0.882	2.787	0.882	4.500	2.136
DESQR	—	3.234	0.087	2.733	0.017	2.608	0.017	3.799	0.674
DESQRP	—	3.426	0.965	2.920	0.882	2.787	0.882	4.500	2.136
EXP	—	3.966	0.087	3.333	0.017	3.225	0.017	4.416	0.674
EXPP	—	4.091	0.970	3.509	0.882	3.408	0.882	5.132	2.136
DEXP	—	3.964	0.087	3.333	0.017	3.225	0.017	4.416	0.674
DEXPP	—	4.095	0.968	3.509	0.882	3.408	0.882	5.132	2.136
LOG	—	4.698	0.087	3.633	0.017	3.508	0.017	4.699	0.674
LOGP	—	4.824	0.968	3.837	0.882	3.693	0.882	5.381	2.136
DLOGE	—	4.698	0.088	3.633	0.017	3.508	0.017	4.699	0.674
DLOGEP	—	4.818	0.969	3.837	0.882	3.693	0.882	5.381	2.136
POW	—	4.431	0.087	3.800	0.017	3.641	0.017	5.065	0.674
POWP	—	4.603	0.973	3.975	0.882	3.827	0.882	5.771	2.136
LOG10	—	4.764	0.087	3.600	0.017	3.516	0.017	4.708	0.674
LOG10P	—	4.922	0.969	3.805	0.882	3.711	0.882	5.411	2.136
DLOG10	—	4.764	0.087	3.600	0.017	3.516	0.017	4.708	0.674
DLOG10P	—	4.928	0.973	3.805	0.882	3.711	0.882	5.411	2.136
EMAX	(n)=K1	5.466	0.087	4.500	0.017	4.200	0.017	6.731	0.674
	(n)=K96	33.187	0.087	26.552	0.017	26.219	0.017	28.735	0.674
EMAXP	(n)=K1	5.604	0.970	4.741	0.882	4.396	0.882	8.193	2.136
	(n)=K96	32.520	0.970	27.084	0.882	27.084	0.882	30.197	2.136
EMIN	(n)=K1	5.498	0.087	4.533	0.017	4.225	0.017	6.699	0.674
	(n)=K96	33.232	0.087	26.112	0.017	25.616	0.017	28.156	0.674
EMINP	(n)=K1	5.655	0.973	4.691	0.882	4.404	0.882	8.161	2.136
	(n)=K96	32.540	0.973	26.481	0.882	26.481	0.882	29.618	2.136
RND	—	2.532	0.087	2.033	0.017	1.941	0.017	2.883	0.674
RNDP	—	2.830	0.971	2.201	0.882	2.116	0.882	3.565	2.136
ZPUSH (1 operand)	—	6.843	0.087	6.843	0.017	5.735	0.017	7.648	0.674
ZPUSHP (1 operand)	—	7.103	0.972	7.103	0.882	5.953	0.882	9.110	2.136
ZPUSH (2 operands)	K2 R0	6.546	0.087	6.200	0.017	5.735	0.017	7.648	0.674
ZPUSHP (2 operands)	K2 R0	6.816	0.973	6.433	0.882	5.953	0.882	9.110	2.136
ZPOP (1 operand)	—	7.587	0.084	6.843	0.017	5.735	0.017	7.648	0.674
ZPOPP (1 operand)	—	7.880	0.973	7.103	0.882	5.953	0.882	9.110	2.136
ZPOP (2 operands)	K2 R0	7.046	0.084	6.200	0.017	5.735	0.017	7.648	0.674
ZPOPP (2 operands)	K2 R0	7.312	0.970	6.433	0.882	5.953	0.882	9.110	2.136
LIMIT	—	3.196	0.087	2.667	0.017	2.567	0.017	4.883	0.674
LIMITP	—	3.335	0.969	2.843	0.882	2.771	0.882	6.345	2.136
LIMIT_U	—	3.198	0.087	2.667	0.017	2.567	0.017	4.883	0.674
LIMITP_U	—	3.362	0.971	2.843	0.882	2.771	0.882	6.345	2.136
DLIMIT	—	3.198	0.087	2.667	0.017	2.567	0.017	4.883	0.674
DLIMITP	—	3.366	0.973	2.843	0.882	2.771	0.882	6.345	2.136
DLIMIT_U	—	3.231	0.087	2.667	0.017	2.567	0.017	4.883	0.674
DLIMITP_U	—	3.391	0.969	2.843	0.882	2.771	0.882	6.345	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
BAND	—	3.134	0.088	2.667	0.017	2.567	0.017	4.883	0.674
BANDP	—	3.292	0.972	2.845	0.882	2.772	0.882	6.345	2.136
BAND_U	—	3.164	0.087	2.667	0.017	2.567	0.017	4.883	0.674
BANDP_U	—	3.344	0.972	2.845	0.882	2.772	0.882	6.345	2.136
DBAND	—	3.198	0.087	2.667	0.017	2.567	0.017	4.883	0.674
DBANDP	—	3.408	0.972	2.845	0.882	2.772	0.882	6.345	2.136
DBAND_U	—	3.198	0.087	2.667	0.017	2.567	0.017	4.883	0.674
DBANDP_U	—	3.360	0.970	2.845	0.882	2.772	0.882	6.345	2.136
ZONE	—	3.131	0.087	2.633	0.017	2.559	0.017	4.875	0.674
ZONEP	—	3.258	0.971	2.848	0.882	2.765	0.882	6.337	2.136
ZONE_U	—	3.131	0.087	2.633	0.017	2.559	0.017	4.875	0.674
ZONEP_U	—	3.287	0.970	2.848	0.882	2.765	0.882	6.337	2.136
DZONE	—	3.198	0.087	2.633	0.017	2.559	0.017	4.875	0.674
DZONEP	—	3.359	0.971	2.848	0.882	2.765	0.882	6.337	2.136
DZONE_U	—	3.170	0.087	2.633	0.017	2.559	0.017	4.875	0.674
DZONEP_U	—	3.291	0.971	2.848	0.882	2.765	0.882	6.337	2.136
SCL	—	5.998	0.091	4.733	0.017	4.541	0.017	6.733	0.674
SCLP	—	6.172	0.971	4.939	0.882	4.752	0.882	7.517	2.136
SCL_U	—	6.031	0.090	4.733	0.017	4.541	0.017	6.733	0.674
SCLP_U	—	6.198	0.970	4.939	0.882	4.752	0.882	7.517	2.136
DSCL	—	5.931	0.087	4.733	0.017	4.541	0.017	6.733	0.674
DSCLP	—	6.026	0.971	4.939	0.882	4.752	0.882	7.517	2.136
DSCL_U	—	5.868	0.087	4.733	0.017	4.541	0.017	6.733	0.674
DSCLP_U	—	6.042	0.969	4.939	0.882	4.752	0.882	7.517	2.136
SCL2	—	6.499	0.091	5.233	0.017	5.033	0.017	7.377	0.674
SCL2P	—	6.667	0.973	5.451	0.882	5.240	0.882	7.991	2.136
SCL2_U	—	6.535	0.090	5.233	0.017	5.033	0.017	7.377	0.674
SCL2P_U	—	6.710	0.970	5.451	0.882	5.240	0.882	7.991	2.136
DSCL2	—	6.498	0.087	5.233	0.017	5.033	0.017	7.377	0.674
DSCL2P	—	6.643	0.971	5.451	0.882	5.240	0.882	7.991	2.136
DSCL2_U	—	6.498	0.087	5.233	0.017	5.033	0.017	7.377	0.674
DSCL2P_U	—	6.608	0.973	5.451	0.882	5.240	0.882	7.991	2.136
UDCNTF	—	4.171	3.578	3.267	2.568	3.067	2.568	4.716	4.334
TTMR	—	4.232	3.661	3.633	2.917	3.400	2.917	4.932	4.985
STMR	—	5.666	6.460	4.433	4.624	4.183	4.624	7.003	7.199
ROTC	—	8.134	8.033	6.800	6.431	6.667	6.431	9.600	9.274
RAMPF	(n)=K1	5.966	3.818	5.467	3.263	5.067	3.028	8.067	5.013
SPD	—	13.867	3.081	12.800	2.426	10.533	2.426	13.200	4.654
DSPD	—	13.867	3.108	12.800	2.426	10.533	2.426	13.200	4.654
PLSY	—	30.400	6.573	30.400	6.573	25.217 (24.267)	6.573	27.617 (26.667)	8.096
DPLSY	—	30.400	6.573	30.400	6.573	24.267	6.573	26.667	8.096
PWM	—	20.000	9.443	18.667	6.456	15.750 (14.800)	6.456	18.417 (17.467)	7.979
DPWM	—	20.000	9.456	18.667	6.456	14.800	6.456	17.467	7.979
ZRST	—	2.800	0.084	2.533	0.017	2.425	0.017	4.703	0.674
ZRSTP	—	3.139	0.970	2.771	0.882	2.687	0.882	6.165	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
SERMM	(n)=K1 No same data	5.799	0.087	4.933	0.017	4.541	0.017	7.331	0.674
	(n)=K1 The number of same data=1	5.834	0.087	4.967	0.017	4.567	0.017	7.357	0.674
	(n)=K96 No same data	20.102	0.087	16.283	0.017	15.984	0.017	18.785	0.674
	(n)=K96 The number of same data=96	23.327	0.087	19.577	0.017	19.173	0.017	21.887	0.674
SERMMP	(n)=K1 No same data	5.924	0.974	5.105	0.882	4.727	0.882	8.793	2.136
	(n)=K1 The number of same data=1	5.963	0.974	5.137	0.882	4.753	0.882	8.819	2.136
	(n)=K96 No same data	20.000	0.974	16.389	0.882	16.084	0.882	18.981	2.136
	(n)=K96 The number of same data=96	23.272	0.974	20.442	0.882	19.219	0.882	22.020	2.136
DSERMM	(n)=K1 No same data	5.831	0.088	4.933	0.017	4.541	0.017	7.331	0.674
	(n)=K1 The number of same data=1	5.868	0.088	4.967	0.017	4.567	0.017	7.357	0.674
	(n)=K96 No same data	20.207	0.088	16.816	0.017	16.383	0.017	19.177	0.674
	(n)=K96 The number of same data=96	23.430	0.088	20.112	0.017	19.657	0.017	22.381	0.674
DSERMMP	(n)=K1 No same data	5.995	0.975	5.105	0.882	4.727	0.882	8.793	2.136
	(n)=K1 The number of same data=1	6.040	0.975	5.137	0.882	4.753	0.882	8.819	2.136
	(n)=K96 No same data	20.063	0.975	16.869	0.882	16.489	0.882	20.639	2.136
	(n)=K96 The number of same data=96	23.422	0.975	20.105	0.882	19.691	0.882	23.843	2.136
MAX	(n)=K1	5.564	0.090	4.467	0.017	4.175	0.017	6.705	0.674
	(n)=K96	18.270	0.090	14.665	0.017	14.383	0.017	16.913	0.674
MAXP	(n)=K1	5.751	0.968	4.668	0.882	4.388	0.882	8.167	2.136
	(n)=K96	18.156	0.968	14.801	0.882	14.496	0.882	18.375	2.136
MAX_U	(n)=K1	5.464	0.091	4.467	0.017	4.175	0.017	6.705	0.674
	(n)=K96	17.247	0.091	14.665	0.017	14.383	0.017	16.913	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
MAXP_U	(n)=K1	5.603	0.971	4.668	0.882	4.388	0.882	8.167	2.136
	(n)=K96	17.280	0.971	14.801	0.882	14.496	0.882	18.375	2.136
DMAX	(n)=K1	5.568	0.090	4.467	0.017	4.175	0.017	6.705	0.674
	(n)=K96	19.932	0.090	15.549	0.017	15.193	0.017	17.717	0.674
DMAXP	(n)=K1	5.771	0.970	4.668	0.882	4.388	0.882	8.167	2.136
	(n)=K96	19.854	0.970	15.643	0.882	15.307	0.882	18.035	2.136
DMAX_U	(n)=K1	5.432	0.091	4.467	0.017	4.175	0.017	6.705	0.674
	(n)=K96	19.828	0.091	15.549	0.017	15.193	0.017	17.717	0.674
DMAXP_U	(n)=K1	5.602	0.971	4.668	0.882	4.388	0.882	8.167	2.136
	(n)=K96	19.684	0.971	15.643	0.882	15.307	0.882	18.035	2.136
MIN	(n)=K1	5.598	0.090	4.467	0.017	4.167	0.017	6.697	0.674
	(n)=K96	18.171	0.090	14.699	0.017	14.408	0.017	16.947	0.674
MINP	(n)=K1	5.778	0.971	4.639	0.882	4.371	0.882	8.159	2.136
	(n)=K96	18.163	0.971	14.803	0.882	14.613	0.882	18.409	2.136
MIN_U	(n)=K1	5.543	0.090	4.467	0.017	4.167	0.017	6.697	0.674
	(n)=K96	17.319	0.090	14.699	0.017	14.408	0.017	16.947	0.674
MINP_U	(n)=K1	5.714	0.971	4.639	0.882	4.371	0.882	8.159	2.136
	(n)=K96	17.324	0.971	14.803	0.882	14.613	0.882	18.409	2.136
DMIN	(n)=K1	5.498	0.091	4.467	0.017	4.167	0.017	6.697	0.674
	(n)=K96	19.900	0.091	15.517	0.017	15.192	0.017	17.721	0.674
DMINP	(n)=K1	5.703	0.972	4.639	0.882	4.371	0.882	8.159	2.136
	(n)=K96	19.740	0.972	15.636	0.882	15.359	0.882	18.119	2.136
DMIN_U	(n)=K1	5.435	0.091	4.467	0.017	4.167	0.017	6.697	0.674
	(n)=K96	19.800	0.091	15.517	0.017	15.192	0.017	17.721	0.674
DMINP_U	(n)=K1	5.648	0.972	4.639	0.882	4.371	0.882	8.159	2.136
	(n)=K96	19.703	0.972	15.636	0.882	15.359	0.882	18.119	2.136
SORTTBL	(n1)=K1, (n2)=K1, (n3)=K1	6.000	1.112	5.467	1.083	5.200	1.083	8.400	2.417
	(n1)=K32, (n2)=K1, (n3)=K1	6.934	1.112	6.400	1.083	6.133	1.083	9.333	2.413
	(n1)=K32, (n2)=K6, (n3)=K1	12.667	1.112	11.733	1.082	11.467	1.082	14.667	2.415
SORTTBL_U	(n1)=K1, (n2)=K1, (n3)=K1	6.000	1.111	5.467	1.083	5.200	1.083	8.400	2.417
	(n1)=K32, (n2)=K1, (n3)=K1	7.067	1.111	6.400	1.083	6.133	1.083	9.333	2.413
	(n1)=K32, (n2)=K6, (n3)=K1	12.534	1.111	11.733	1.082	11.467	1.082	14.667	2.415

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
SORTTBL2	(n1)=K1, (n2)=K1, (n3)=K1	6.534	1.135	5.733	1.083	5.467	1.083	8.667	2.418
	(n1)=K32, (n2)=K1, (n3)=K1	7.600	1.135	6.667	1.083	6.267	1.083	9.600	2.414
	(n1)=K32, (n2)=K6, (n3)=K1	13.067	1.135	12.000	1.083	11.733	1.083	14.933	2.413
SORTTBL2_U	(n1)=K1, (n2)=K1, (n3)=K1	6.667	1.136	5.733	1.083	5.467	1.083	8.667	2.418
	(n1)=K32, (n2)=K1, (n3)=K1	7.600	1.136	6.667	1.083	6.267	1.083	9.600	2.414
	(n1)=K32, (n2)=K6, (n3)=K1	13.067	1.136	12.000	1.083	11.733	1.083	14.933	2.413
DSORTTBL2	(n1)=K1, (n2)=K1, (n3)=K1	6.800	1.136	5.733	1.083	5.467	1.083	8.667	2.418
	(n1)=K32, (n2)=K1, (n3)=K1	8.267	1.136	7.333	1.083	6.933	1.083	9.867	2.414
	(n1)=K32, (n2)=K1, (n3)=K1	18.800	1.136	17.467	1.083	17.067	1.083	20.000	2.416
DSORTTBL2_U	(n1)=K1, (n2)=K1, (n3)=K1	6.800	1.136	5.733	1.083	5.467	1.083	8.667	2.418
	(n1)=K32, (n2)=K1, (n3)=K1	8.267	1.136	7.333	1.083	6.933	1.083	9.867	2.414
	(n1)=K32, (n2)=K1, (n3)=K1	18.800	1.136	17.467	1.083	17.067	1.083	20.000	2.416
SUM	—	3.599	0.090	2.800	0.017	2.683	0.017	4.575	0.674
SUMP	—	3.735	0.970	2.975	0.882	2.893	0.882	4.947	2.136
BON	(n)=K1	3.098	0.090	2.600	0.017	2.475	0.017	4.591	0.674
BONP	—	3.235	0.972	2.805	0.882	2.664	0.882	6.053	2.136
DSUM	—	4.435	0.090	3.133	0.017	3.000	0.017	4.896	0.674
DSUMP	—	4.564	0.969	3.384	0.882	3.219	0.882	5.240	2.136
DBON	—	3.131	0.090	2.600	0.017	2.475	0.017	4.591	0.674
DBONP	—	3.259	0.969	2.805	0.882	2.664	0.882	6.053	2.136
WSUM	(n)=K1	5.998	0.087	4.433	0.017	4.192	0.017	6.831	0.674
	(n)=K96	13.776	0.087	7.645	0.017	7.367	0.017	10.023	0.674
WSUMP	(n)=K1	6.136	0.977	4.639	0.882	4.387	0.882	8.293	2.136
	(n)=K96	13.870	0.977	7.845	0.882	7.564	0.882	11.485	2.136
WSUM_U	(n)=K1	5.998	0.087	4.433	0.017	4.192	0.017	6.831	0.674
	(n)=K96	13.808	0.087	7.645	0.017	7.367	0.017	10.023	0.674
WSUMP_U	(n)=K1	6.211	0.977	4.639	0.882	4.387	0.882	8.293	2.136
	(n)=K96	13.854	0.977	7.845	0.882	7.564	0.882	11.485	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
DWSUM	(n)=K1	6.799	0.087	5.000	0.017	4.616	0.017	7.256	0.674
	(n)=K96	18.900	0.087	10.851	0.017	10.473	0.017	13.048	0.674
DWSUMP	(n)=K1	6.938	0.975	5.203	0.882	4.813	0.882	8.718	2.136
	(n)=K96	18.787	0.975	10.939	0.882	10.592	0.882	14.510	2.136
DWSUM_U	(n)=K1	6.799	0.087	5.000	0.017	4.616	0.017	7.256	0.674
	(n)=K96	18.067	0.087	10.851	0.017	10.473	0.017	13.048	0.674
DWSUMP_U	(n)=K1	7.024	0.974	5.203	0.882	4.813	0.882	8.718	2.136
	(n)=K96	18.044	0.974	10.939	0.882	10.592	0.882	14.510	2.136
MEAN	(n)=K1	4.731	0.088	3.815	0.017	3.675	0.017	5.975	0.674
	(n)=K96	12.544	0.088	7.015	0.017	6.892	0.017	9.273	0.674
MEANP	(n)=K1	4.896	0.971	4.012	0.882	3.871	0.882	7.437	2.136
	(n)=K96	12.659	0.971	7.217	0.882	7.104	0.882	10.735	2.136
MEAN_U	(n)=K1	4.764	0.087	3.815	0.017	3.675	0.017	5.975	0.674
	(n)=K96	12.579	0.087	7.015	0.017	6.892	0.017	9.273	0.674
MEANP_U	(n)=K1	4.912	0.971	4.012	0.882	3.871	0.882	7.437	2.136
	(n)=K96	12.642	0.971	7.217	0.882	7.104	0.882	10.735	2.136
DMEAN	(n)=K1	5.264	0.084	4.167	0.017	4.049	0.017	6.415	0.674
	(n)=K96	13.976	0.084	9.999	0.017	9.865	0.017	12.164	0.674
DMEANP	(n)=K1	5.448	0.973	4.392	0.882	4.252	0.882	7.877	2.136
	(n)=K96	14.039	0.973	10.139	0.882	10.009	0.882	13.626	2.136
DMEAN_U	(n)=K1	5.164	0.084	4.167	0.017	4.049	0.017	6.415	0.674
	(n)=K96	13.914	0.084	9.999	0.017	9.865	0.017	12.164	0.674
DMEANP_U	(n)=K1	5.339	0.971	4.392	0.882	4.252	0.882	7.877	2.136
	(n)=K96	14.000	0.971	10.139	0.882	10.009	0.882	13.626	2.136
SQRT	—	2.964	0.090	2.533	0.017	2.425	0.017	4.316	0.674
SQRTP	—	3.114	0.971	2.741	0.882	2.648	0.882	5.027	2.136
DSQRT	—	3.467	0.087	3.067	0.017	2.933	0.017	4.824	0.674
DSQRTP	—	3.679	0.966	3.237	0.882	3.157	0.882	6.286	2.136
CCD	(n)=K1 SM8161 =OFF	5.631	0.087	4.633	0.017	4.392	0.017	6.923	0.674
	(n)=K96 SM8161 =OFF	14.346	0.087	9.311	0.017	9.024	0.017	11.491	0.674
	(n)=K1 SM8161=ON	5.542	0.087	4.633	0.017	4.392	0.017	6.923	0.674
	(n)=K96 SM8161=ON	13.275	0.087	8.073	0.017	7.817	0.017	10.365	0.674
CCDP	(n)=K1 SM8161 =OFF	5.764	0.972	5.257	0.882	5.257	0.882	8.385	2.136
	(n)=K96 SM8161 =OFF	14.580	0.972	9.375	0.882	9.169	0.882	11.921	2.136
	(n)=K1 SM8161=ON	5.656	0.972	5.257	0.882	5.257	0.882	8.385	2.136
	(n)=K96 SM8161=ON	13.366	0.972	8.281	0.882	7.984	0.882	10.648	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
CRC	(n)=K1 SM8161 =OFF	5.903	0.087	4.967	0.017	4.659	0.017	7.189	0.674
	(n)=K96 SM8161 =OFF	28.151	0.087	15.028	0.017	14.724	0.017	17.260	0.674
	(n)=K1 SM8161=ON	5.864	0.087	4.967	0.017	4.659	0.017	7.189	0.674
	(n)=K96 SM8161=ON	19.299	0.087	14.399	0.017	14.107	0.017	16.651	0.674
CRCP	(n)=K1 SM8161 =OFF	6.070	0.975	5.177	0.882	4.844	0.882	8.651	2.136
	(n)=K96 SM8161 =OFF	28.178	0.975	15.216	0.882	14.829	0.882	17.440	2.136
	(n)=K1 SM8161=ON	6.024	0.975	5.177	0.882	4.844	0.882	8.651	2.136
	(n)=K96 SM8161=ON	19.236	0.975	14.493	0.882	14.195	0.882	18.113	2.136
ADRSET	—	2.598	0.090	2.300	0.017	2.208	0.017	4.099	0.674
ADRSETP	—	2.860	0.973	2.509	0.882	2.436	0.882	4.801	2.136
TRD	—	3.164	0.088	2.600	0.017	2.549	0.017	3.657	0.674
TRDP	—	3.404	0.971	2.756	0.882	2.741	0.882	4.351	2.136
TWR	—	7.270	0.087	6.143	0.017	5.149	0.017	7.037	0.674
TWRP	—	7.394	0.972	6.272	0.882	5.321	0.882	8.499	2.136
TADD	—	5.534	0.090	4.900	0.017	4.616	0.017	7.329	0.674
TADDP	—	5.667	0.972	5.061	0.882	4.819	0.882	8.791	2.136
TSUB	—	5.431	0.090	4.867	0.017	4.608	0.017	7.316	0.674
TSUBP	—	5.592	0.972	5.083	0.882	4.799	0.882	8.778	2.136
HTOS	—	3.498	0.084	3.012	0.017	2.925	0.017	4.984	0.674
HTOSP	—	3.682	0.974	3.245	0.882	3.111	0.882	6.446	2.136
DHTOS	—	3.531	0.087	3.012	0.017	2.925	0.017	4.984	0.674
DHTOSP	—	3.752	0.976	3.245	0.882	3.111	0.882	6.446	2.136
STOH	—	3.331	0.090	2.867	0.017	2.741	0.017	4.800	0.674
STOHP	—	3.551	0.972	3.064	0.882	2.945	0.882	5.497	2.136
DSTOH	—	3.368	0.088	2.867	0.017	2.741	0.017	4.800	0.674
DSTOHP	—	3.519	0.971	3.064	0.882	2.945	0.882	5.497	2.136
LDDT=	—	5.631	—	4.867	—	4.549	—	7.107	—
LDDT<>	—	4.867	—	4.867	—	4.549	—	7.107	—
LDDT>	—	4.867	—	4.867	—	4.549	—	7.107	—
LDDT<=	—	4.867	—	4.867	—	4.549	—	7.107	—
LDDT<	—	4.867	—	4.867	—	4.549	—	7.107	—
LDDT>=	—	4.867	—	4.867	—	4.549	—	7.107	—
ANDDT=	—	5.731	—	4.867	—	4.549	—	7.107	—
ANDDT<>	—	4.867	—	4.867	—	4.549	—	7.107	—
ANDDT>	—	4.867	—	4.867	—	4.549	—	7.107	—
ANDDT<=	—	4.867	—	4.867	—	4.549	—	7.107	—
ANDDT<	—	4.867	—	4.867	—	4.549	—	7.107	—
ANDDT>=	—	4.867	—	4.867	—	4.549	—	7.107	—

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
ORDT=	—	5.731	—	4.867	—	4.549	—	7.107	—
ORDT<>	—	4.867	—	4.867	—	4.549	—	7.107	—
ORDT>	—	4.867	—	4.867	—	4.549	—	7.107	—
ORDT<=	—	4.867	—	4.867	—	4.549	—	7.107	—
ORDT<	—	4.867	—	4.867	—	4.549	—	7.107	—
ORDT>=	—	4.867	—	4.867	—	4.549	—	7.107	—
LDTM=	—	5.235	—	4.615	—	4.317	—	6.843	—
LDTM<>	—	5.263	—	4.615	—	4.317	—	6.843	—
LDTM>	—	5.267	—	4.615	—	4.317	—	6.843	—
LDTM<=	—	5.363	—	4.615	—	4.317	—	6.843	—
LDTM<	—	5.298	—	4.615	—	4.317	—	6.843	—
LDTM>=	—	5.296	—	4.615	—	4.317	—	6.843	—
ANDTM=	—	5.296	—	4.615	—	4.317	—	6.843	—
ANDTM<>	—	5.368	—	4.615	—	4.317	—	6.843	—
ANDTM>	—	5.430	—	4.615	—	4.317	—	6.843	—
ANDTM<=	—	5.464	—	4.615	—	4.317	—	6.843	—
ANDTM<	—	5.430	—	4.615	—	4.317	—	6.843	—
ANDTM>=	—	5.435	—	4.615	—	4.317	—	6.843	—
ORTM=	—	5.330	—	4.615	—	4.317	—	6.843	—
ORTM<>	—	5.396	—	4.615	—	4.317	—	6.843	—
ORTM>	—	5.399	—	4.615	—	4.317	—	6.843	—
ORTM<=	—	5.463	—	4.615	—	4.317	—	6.843	—
ORTM<	—	5.463	—	4.615	—	4.317	—	6.843	—
ORTM>=	—	5.432	—	4.615	—	4.317	—	6.843	—
TCMP	—	6.031	0.084	5.200	0.017	4.792	0.017	7.833	0.674
TCMPP	—	6.136	0.973	5.393	0.882	4.975	0.882	9.295	2.136
TZCP	—	7.631	0.087	6.767	0.017	6.225	0.017	9.381	0.674
TZCPP	—	7.736	0.975	6.939	0.882	6.409	0.882	10.843	2.136
DUTY	(n1)=K1, (n2)=K1	2.700	2.683	2.300	2.235	2.225	2.235	4.488	4.292
HOURM	—	4.998	0.087	4.000	0.017	3.841	0.017	6.000	0.674
DHOURM	—	4.799	0.087	4.000	0.017	3.841	0.017	6.000	0.674
RFS	(s)=Y, (n)=K1	4.332	0.087	3.500	0.017	3.400	0.017	4.741	0.674
	(s)=Y, (n)=K64	6.732	0.087	5.633	0.017	5.525	0.017	6.947	0.674
	(s)=X, (n)=K1	4.434	0.087	3.567	0.017	3.425	0.017	4.765	0.674
	(s)=X, (n)=K64	7.856	0.087	6.400	0.017	6.192	0.017	7.619	0.674
RFSP	(s)=Y, (n)=K1	4.522	0.972	3.707	0.882	3.611	0.882	5.460	2.136
	(s)=Y, (n)=K64	6.860	0.972	5.813	0.882	5.777	0.882	7.739	2.136
	(s)=X, (n)=K1	4.558	0.972	3.743	0.882	3.628	0.882	5.521	2.136
	(s)=X, (n)=K64	7.958	0.972	6.595	0.882	6.375	0.882	8.259	2.136
REF	(s)=Y, (n)=K1	4.336	0.088	3.500	0.017	3.400	0.017	4.741	0.674
	(s)=Y, (n)=K64	6.731	0.088	5.633	0.017	5.525	0.017	6.947	0.674
	(s)=X, (n)=K1	4.432	0.088	3.567	0.017	3.425	0.017	4.765	0.674
	(s)=X, (n)=K64	7.831	0.088	6.400	0.017	6.192	0.017	7.619	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
REFP	(s)=Y, (n)=K1	4.531	0.972	3.707	0.882	3.611	0.882	5.460	2.136
	(s)=Y, (n)=K64	6.862	0.972	5.813	0.882	5.777	0.882	7.739	2.136
	(s)=X, (n)=K1	4.564	0.972	3.743	0.882	3.628	0.882	5.521	2.136
	(s)=X, (n)=K64	7.959	0.972	6.595	0.882	6.375	0.882	8.259	2.136
FROM	(n)=K1	—	—	12.020	0.017	11.600	0.017	13.401	0.674
	(n)=K1000	—	—	662.387	0.017	617.560	0.017	619.787	0.674
FROMP	(n)=K1	—	—	12.591	0.882	11.697	0.882	14.055	2.136
	(n)=K1000	—	—	663.252	0.882	619.022	0.882	621.249	2.136
DFROM	(n)=K1	—	—	13.023	0.017	11.792	0.017	14.335	0.674
	(n)=K500	—	—	662.387	0.017	617.560	0.017	619.787	0.674
DFROMP	(n)=K1	—	—	13.888	0.882	11.865	0.882	15.797	2.136
	(n)=K500	—	—	663.252	0.882	619.022	0.882	621.249	2.136
TO	(n)=K1	—	—	11.801	0.017	11.476	0.017	13.395	0.674
	(n)=K1000	—	—	716.827	0.017	632.973	0.017	634.893	0.674
TOP	(n)=K1	—	—	11.928	0.882	11.673	0.882	14.012	2.136
	(n)=K1000	—	—	717.692	0.882	633.838	0.882	636.355	2.136
DTO	(n)=K1	—	—	12.715	0.017	11.637	0.017	13.484	0.674
	(n)=K500	—	—	716.827	0.017	632.973	0.017	634.893	0.674
DTOP	(n)=K1	—	—	12.872	0.882	11.717	0.882	14.055	2.136
	(n)=K500	—	—	717.692	0.882	633.838	0.882	636.355	2.136
FROMD	(n)=K1	—	—	12.240	0.017	11.564	0.017	13.372	0.674
	(n)=K1000	—	—	662.387	0.017	618.080	0.017	619.893	0.674
FROMDP	(n)=K1	—	—	12.559	0.882	11.653	0.882	13.948	2.136
	(n)=K1000	—	—	663.252	0.882	618.945	0.882	621.355	2.136
DFROMD	(n)=K1	—	—	13.081	0.017	11.748	0.017	13.512	0.674
	(n)=K500	—	—	662.387	0.017	618.080	0.017	619.893	0.674
DFROMDP	(n)=K1	—	—	13.946	0.882	11.833	0.882	14.139	2.136
	(n)=K500	—	—	663.252	0.882	618.945	0.882	621.355	2.136
TOD	(n)=K1	—	—	11.753	0.017	11.548	0.017	13.368	0.674
	(n)=K1000	—	—	716.307	0.017	633.040	0.017	634.893	0.674
TODP	(n)=K1	—	—	12.687	0.882	11.631	0.882	13.953	2.136
	(n)=K1000	—	—	717.172	0.882	633.905	0.882	636.355	2.136
DTOD	(n)=K1	—	—	12.687	0.017	11.631	0.017	13.425	0.674
	(n)=K500	—	—	716.307	0.017	633.040	0.017	634.893	0.674
DTODP	(n)=K1	—	—	12.845	0.882	11.696	0.882	14.011	2.136
	(n)=K500	—	—	717.172	0.882	633.905	0.882	636.355	2.136
RBFM	(s)=K0, (n1)=K1	—	—	—	—	11.496	1.359	13.421	2.853
	(s)=K0, (n1)=K1000	—	—	—	—	34.631	1.360	35.936	2.851
WBFM	(s)=K0, (n1)=K1	—	—	—	—	10.583	1.341	12.337	2.835
	(s)=K0, (n1)=K1000	—	—	—	—	9.599	1.342	11.319	2.836
LOGTRG	—	27.467	0.971	27.467	0.882	23.867	0.882	25.333	2.136
LOGTRGR	—	6.400	0.976	6.400	0.882	6.400	0.882	8.000	2.136

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
SP.SOCOPEN	—	11.867	1.083	10.400	0.882	8.817 (7.867)	0.882	11.750 (10.800)	2.136
SP.SOCCLOSE	—	9.067	1.082	7.867	0.882	7.083 (6.133)	0.882	9.617 (8.667)	2.526
SP.SOCRCV	TCP Receive data length = 1 byte	9.734	1.082	8.400	0.882	7.350 (6.400)	0.882	10.283 (9.333)	2.136
	TCP Receive data length = 2046 bytes	9.734	1.082	8.400	0.882	7.483 (6.533)	0.882	10.683 (9.733)	2.136
	UDP Receive data length = 1 byte	9.734	1.082	8.400	0.882	7.483 (6.533)	0.882	10.683 (9.733)	2.136
	UDP Receive data length = 2046 bytes	9.734	1.082	8.400	0.882	7.483 (6.533)	0.882	10.683 (9.733)	2.136
SP.FTPPUT	File name + folder path of the FTP server = 32 characters	60.800	1.067	59.867	0.882	44.267	0.882	53.733	2.136
	File name + folder path of the FTP server = 64 characters	114.400	1.067	102.533	0.882	74.000	0.882	90.267	2.136
SP.FTPGET	File name + folder path of the FTP server = 32 characters	62.534	1.067	61.333	0.882	46.133	0.882	55.333	2.136
	File name + folder path of the FTP server = 64 characters	110.534	1.067	110.800	0.882	80.800	0.882	97.333	2.136
SP.SOCSND	TCP Receive data length = 1 byte	14.267	1.082	11.733	0.882	10.817 (9.867)	0.882	13.617 (12.667)	2.136
	TCP Receive data length = 2046 bytes	133.734	1.082	131.600	0.882	70.683 (69.733)	0.882	74.017 (73.067)	2.136
	UDP Receive data length = 1 byte	13.334	1.082	11.333	0.882	10.950 (10.000)	0.882	14.283 (13.333)	2.136
	UDP Receive data length = 2046 bytes	133.600	1.082	131.600	0.882	70.683 (69.733)	0.882	74.017 (73.067)	2.136
SP.SOCCINF	—	7.467	0.972	6.267	0.882	5.467	0.882	7.867	2.205

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
S.SOCRDATA	(n)=K1	5.440	0.087	4.456	0.017	4.009	0.017	6.761	0.674
	(n)=K1024	146.036	0.087	144.305	0.017	104.037	0.017	106.584	0.674
SP.SOCRDATA	(n)=K1	5.782	0.978	4.721	0.882	4.281	0.882	7.192	2.136
	(n)=K1024	146.328	0.978	144.337	0.882	104.147	0.882	107.397	2.136
SP.ECPRTCL	—	16.000	0.971	14.667	0.882	10.400	0.882	13.333	2.205
PID	—	6.066	6.197	5.115	4.996	4.925	4.996	6.785	7.016
LD [S□]	Continuity/ Non-continuity	—	—	—	—	1.400	1.400	1.700	1.700
LD [BL□\S□]	Continuity/ Non-continuity	—	—	—	—	2.300	2.300	2.700	2.700
LD [BL□]	Continuity/ Non-continuity	—	—	—	—	1.100	1.100	1.500	1.500
LDI [S□]	Continuity/ Non-continuity	—	—	—	—	1.100	1.100	1.700	1.700
LDI [BL□\S□]	Continuity/ Non-continuity	—	—	—	—	2.300	2.300	2.700	2.700
LDI [BL□]	Continuity/ Non-continuity	—	—	—	—	1.100	1.100	1.500	1.500
AND [S□]	Continuity/ Non-continuity	—	—	—	—	1.400	1.400	1.800	1.800
AND [BL□\S□]	Continuity/ Non-continuity	—	—	—	—	2.300	2.300	2.700	2.700
AND [BL□]	Continuity/ Non-continuity	—	—	—	—	1.200	1.200	1.500	1.500
ANI [S□]	Continuity/ Non-continuity	—	—	—	—	1.400	1.400	1.700	1.700
ANI [BL□\S□]	Continuity/ Non-continuity	—	—	—	—	2.300	2.300	2.600	2.600
ANI [BL□]	Continuity/ Non-continuity	—	—	—	—	1.200	1.200	1.500	1.500
OR [S□]	Continuity/ Non-continuity	—	—	—	—	1.400	1.400	1.700	1.700
OR [BL□\S□]	Continuity/ Non-continuity	—	—	—	—	2.300	2.300	2.700	2.700
OR [BL□]	Continuity/ Non-continuity	—	—	—	—	1.100	1.100	1.500	1.500
ORI [S□]	Continuity/ Non-continuity	—	—	—	—	1.400	1.400	1.700	1.700
ORI [BL□\S□]	Continuity/ Non-continuity	—	—	—	—	2.300	2.300	2.700	2.700
ORI [BL□]	Continuity/ Non-continuity	—	—	—	—	1.200	1.200	1.500	1.500
MOV [KnS□]	—	—	—	—	—	3.900	3.900	5.000	5.000
MOV [BL□\KnS□]	—	—	—	—	—	4.200	4.200	5.300	5.300
MOVP [KnS□]	—	—	—	—	—	3.900	3.900	5.000	5.000
MOVP [BL□\KnS□]	—	—	—	—	—	4.200	4.200	5.300	5.300
DMOV [KnS□]	—	—	—	—	—	4.100	4.100	5.100	5.100
DMOV [BL□\KnS□]	—	—	—	—	—	4.400	4.400	5.400	5.400
DMOVP [KnS□]	—	—	—	—	—	4.100	4.100	5.100	5.100
DMOVP [BL□\KnS□]	—	—	—	—	—	4.400	4.400	5.400	5.400

A

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
BMOV [KnS□]	(n)=K1	—	—	—	—	5.400	5.400	6.900	6.900
	(n)=K96	—	—	—	—	13.500	13.500	15.000	15.000
BMOV [BL□\KnS□]	(n)=K1	—	—	—	—	5.500	5.500	7.000	7.000
	(n)=K96	—	—	—	—	13.500	13.500	15.000	15.000
BMOVP [KnS□]	(n)=K1	—	—	—	—	5.400	5.400	6.900	6.900
	(n)=K96	—	—	—	—	13.500	13.500	15.000	15.000
BMOVP [BL□\KnS□]	(n)=K1	—	—	—	—	5.500	5.500	7.000	7.000
	(n)=K96	—	—	—	—	13.500	13.500	15.000	15.000
SET [S□]	No change time	—	—	—	—	4.300	4.300	4.800	4.800
	Change time	—	—	—	—	7.500	7.500	8.000	8.000
SET [BL□\S□]	No change time	—	—	—	—	4.100	4.100	4.700	4.700
	Change time	—	—	—	—	7.300	7.300	7.800	7.800
SET [BL□]	No change time	—	—	—	—	1.300	1.300	1.800	1.800
	Change time	—	—	—	—	1.300	1.300	1.800	1.800
RST [S□]	No change time	—	—	—	—	2.900	2.900	3.500	3.500
	Change time	—	—	—	—	3.900	3.900	4.400	4.400
RST [BL□\S□]	No change time	—	—	—	—	2.800	2.800	3.300	3.300
	Change time	—	—	—	—	3.800	3.800	4.300	4.300
RST [BL□]	No change time	—	—	—	—	1.700	1.700	2.200	2.200
	Change time	—	—	—	—	1.700	1.700	2.200	2.200
OUT [S□]	No change time	—	—	—	—	4.200	4.200	4.900	4.900
	Change time	—	—	—	—	7.400	7.400	8.100	8.100
OUT [BL□\S□]	No change time	—	—	—	—	4.400	4.400	4.700	4.700
	Change time	—	—	—	—	7.500	7.500	7.900	7.900
ZRST [S□]	1 point	—	—	—	—	7.100	7.100	8.200	8.200
	128 points	—	—	—	—	2943.400	2955.500	2954.700	2959.400
	4096 points	—	—	—	—	2946.400	2959.400	2946.100	2961.100
ZRST [BL□\S□]	1 point	—	—	—	—	6.600	6.600	7.700	7.700
	128 points	—	—	—	—	461.300	462.900	452.100	464.600
	4096 points	—	—	—	—	9600.100	9608.000	9598.300	9609.600
ZRSTP [S□]	1 point	—	—	—	—	7.100	7.100	8.200	8.200
	128 points	—	—	—	—	2943.400	2955.500	2954.700	2959.400
	4096 points	—	—	—	—	2946.400	2959.400	2946.100	2961.100
ZRSTP [BL□\S□]	1 point	—	—	—	—	6.600	6.600	7.700	7.700
	128 points	—	—	—	—	461.300	462.900	452.100	464.600
	4096 points	—	—	—	—	9600.100	9608.000	9598.300	9609.600
ABSD	(n)=K1	6.966	0.084	5.551	0.017	5.551	0.017	8.104	0.674
DABSD	(n)=K1	7.894	0.087	6.000	0.017	5.551	0.017	8.104	0.674
INCD	(n)=K1	8.886	1.187	7.216	1.149	6.725	1.149	9.947	2.475
IST	—	5.936	1.082	5.076	1.056	4.167	1.056	5.897	2.380
DHSCS	—	8.547	0.088	7.400	0.017	5.667	0.017	8.613	0.674

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
DHSCR	—	9.814	0.088	8.640	0.017	6.813	0.017	10.480	0.674
DHSZ	—	9.840	0.088	8.600	0.017	6.853	0.017	10.280	0.674
HCMOV	(n)=K0	5.398	0.087	4.567	0.017	4.375	0.017	6.737	0.674
HCMOVP	—	5.563	0.969	4.771	0.882	4.573	0.882	7.271	2.136
DHCMOV	(n)=K0	5.464	0.087	4.567	0.017	4.375	0.017	6.737	0.674
DHCMOVP	—	5.630	0.968	4.771	0.882	4.573	0.882	7.271	2.136
RS2	—	17.200	1.161	14.533	1.091	13.083 (12.133)	1.091	16.017 (15.067)	2.350
IVCK	—	13.867	1.469	12.933	1.371	12.283 (11.333)	1.371	14.950 (14.000)	2.706
IVDR	—	14.934	1.498	13.867	1.412	13.083 (12.133)	1.412	15.750 (14.800)	2.746
IVRD	—	17.467	1.507	16.667	1.395	15.750 (14.800)	1.395	18.417 (17.467)	2.732
IVWR	—	18.267	1.512	17.467	1.396	16.550 (15.600)	1.396	19.217 (18.267)	2.733
IVBWR	—	20.400	1.462	19.467	1.362	18.283 (17.333)	1.362	21.350 (20.400)	2.696
IVMC	—	18.400	1.505	16.667	1.412	15.883 (14.933)	1.412	19.750 (18.800)	2.748
ADPRW	—	25.600	1.210	23.467	1.149	18.150 (17.200)	1.149	23.083 (22.133)	2.402
S.CPRTCL	—	12.000	0.084	10.533	0.017	7.867	0.017	10.667	0.674
SP.CPRTCL	—	12.267	0.974	10.800	0.882	8.133	0.882	11.067	2.136
MTR	—	2.404	1.147	2.303	1.098	1.935	1.098	3.240	2.422
DSW	—	7.132	1.177	6.473	1.148	5.267	1.148	8.220	2.474
SEGD	—	3.899	0.087	3.400	0.017	3.292	0.017	4.816	0.674
SEGDP	—	4.006	0.972	3.531	0.882	3.468	0.882	5.515	2.136
SEGL	(n)=K1	7.102	1.184	5.940	1.148	5.251	1.148	7.929	2.471
DSZR	(s1)=K1, (s2)=K1	145.067	6.609	145.067	6.609	134.683 (133.733)	6.609	138.283 (137.333)	8.166
DDSZR	(s1)=K1, (s2)=K1	145.067	6.609	145.067	6.609	134.683 (133.733)	6.609	138.283 (137.333)	8.166
DVIT	(s1)=K100, (s2)=K1	142.533	6.585	142.533	6.585	133.217 (132.267)	6.585	136.017 (135.067)	8.141
DDVIT	(s1)=K100, (s2)=K1	142.533	6.585	142.533	6.585	132.267	6.585	135.067	8.141
TBL	(d)=K1, (n)=K1	142.667	6.575	142.667	6.575	133.883 (132.933)	6.575	136.417 (135.467)	8.097
DRVTBL	—	145.733	6.581	145.733	6.581	136.417 (135.467)	6.581	139.350 (138.400)	8.104
DABS	—	3.200	1.292	2.933	1.216	2.533	1.216	4.000	2.718
PLSV	(s)=K1	135.200	6.590	135.200	6.590	127.483 (126.533)	6.590	130.017 (129.067)	8.113
DPLSV	(s)=K1	135.200	6.590	135.200	6.590	126.533	6.590	129.067	8.113
DRVI	(s1)=K1, (s2)=K1	138.133	6.585	138.133	6.585	130.417 (129.467)	6.585	133.217 (132.267)	8.141
DDRVI	(s1)=K1, (s2)=K1	138.133	6.585	138.133	6.585	129.467	6.585	132.267	8.141
DRVA	(s1)=K500, (s2)=K1	138.267	6.584	138.267	6.584	130.417 (129.467)	6.584	133.217 (132.267)	8.141

Instruction name	Condition	FX5S CPU module		FX5UJ CPU module		FX5U/FX5UC CPU module			
		Program capacity setting: 48000 steps		Program capacity setting: 48000 steps		Program capacity setting: 64000 steps		Program capacity setting: 128000 steps	
		Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)	Execution time in ON status (μs)*1	Execution time in OFF status (μs)
DDRVA	(s1)=K500, (s2)=K1	138.267	6.584	138.267	6.584	129.467	6.584	132.267	8.141
DRVMUL	—	578.134	6.842	528.133	6.842	529.083 (528.133)	6.842	532.683 (531.733)	8.287
HIOEN	—	15.754	0.090	14.897	0.017	12.416	0.017	14.631	0.674
HIOENP	—	16.060	0.975	15.085	0.882	12.620	0.882	14.836	2.136
DHIOEN	—	15.762	0.087	14.881	0.017	12.385	0.017	14.428	0.674
DHIOENP	—	16.019	0.976	15.079	0.882	12.627	0.882	14.801	2.136
STL	—	4.667	—	4.000	—	3.333	—	4.800	—
RETSTL	—	2.000	—	1.867	—	1.467	—	2.667	—
SP.SLMPSND	—	13.867	0.971	12.533	0.882	10.017 (9.067)	0.882	12.950 (12.000)	2.136
ERREAD	(n)=K1	11.600	2.483	10.133	2.168	9.217 (8.267)	2.168	12.150 (11.200)	3.834
	(n)=K0 (all points)	11.600	2.483	10.133	2.168	9.350 (8.400)	2.168	12.150 (11.200)	3.832
ERWRITE	(n)=K1	11.734	2.480	10.267	2.140	9.483 (8.533)	2.140	12.150 (11.200)	3.806
	(n)=K0 (all points)	11.734	2.479	10.133	2.168	9.350 (8.400)	2.168	12.150 (11.200)	3.832
ERINIT	—	10.534	2.392	9.067	2.036	8.283 (7.333)	2.036	10.417 (9.467)	3.691
RTM	—	1.334	0.091	0.933	0.017	0.800	0.017	1.867	0.674
S.DEVLD	(n)=K1	4.500	0.094	3.467	0.017	3.259	0.017	5.833	0.674
	(n)=K64	13.170	0.094	12.063	0.017	9.491	0.017	12.051	0.674
SP.DEVLD	(n)=K1	4.667	1.086	3.600	0.882	3.333	0.882	7.295	2.136
	(n)=K64	13.200	1.086	12.133	0.882	10.356	0.882	13.513	2.136
SP.DEVST	(n)=K1	20.400	1.092	16.133	0.882	12.667	0.882	15.600	2.136
	(n)=K64	25.067	1.092	23.467	0.882	16.400	0.882	19.333	2.136
SP.FREAD	—	21.867	1.067	20.133	0.882	17.733	0.882	21.333	2.136
SP.FWRITE	—	22.134	1.067	20.267	0.882	17.733	0.882	21.333	2.136
SP.FDELETE	—	47.867	1.067	46.000	0.882	28.933	0.882	34.267	2.136
SP.FCOPY	—	85.334	1.067	83.067	0.882	50.133	0.882	57.067	2.136
SP.FMOVE	—	85.467	1.067	83.067	0.882	50.000	0.882	56.800	2.136
SP.FRENAME	—	86.934	1.200	84.533	0.882	51.467	0.882	58.933	2.136
SP.FSTATUS	—	50.667	1.200	48.133	0.882	30.667	0.882	36.533	2.136

*1 The value in the parentheses is the value before supporting the SFC program.

Appendix 2 Number of Instruction Steps

The number of instruction steps are shown below.

The number of steps may increase depending on the contents of the source and destination (such as specification of BFM or character string), bit specification of word device, digit specification of bit devices, indexing etc. Note that when the number of steps increases, the instruction processing time increases.

Instruction name	Number of minimum steps
LD	2
LDI	2
AND	2
ANI	2
OR	2
ORI	2
LDP	4
LDF	4
ANDP	4
ANDF	4
ORP	4
ORF	4
LDPI	4
LDFI	4
ANDPI	4
ANDFI	4
ORPI	4
ORFI	4
ANB	1
ORB	1
MPS	1
MRD	1
MPP	1
INV	1
MEP	3
MEF	3
OUT	2
OUT T/ST	5
OUTH T/ ST	5
OUTHS T/ST	5
OUT C	5
OUT LC	5
OUT F	3
SET	2
RST	2
SET F	3
RST F	3
ANS	7
ANR	1
ANRP	1
PLS	3
PLF	3
FF	3
ALT	3
ALTP	3
SFT	3

Instruction name	Number of minimum steps
SFTP	3
SFR	4
SFRP	4
SFL	4
SFLP	4
BSFR	4
BSFRP	4
BSFL	4
BSFLP	4
DSFR	4
DSFRP	4
DSFL	4
DSFLP	4
SFTR	6
SFTRP	6
SFTL	6
SFTLP	6
WSFR	6
WSFRP	6
WSFL	6
WSFLP	6
MC	5
MCR	3
FEND	1
END	1
STOP	1
LD=	4
LD<>	4
LD>	4
LD<=	4
LD<	4
LD>=	4
AND=	4
AND<>	4
AND>	4
AND<=	4
AND<	4
AND>=	4
OR=	4
OR<>	4
OR>	4
OR<=	4
OR<	4
OR>=	4
LD=_U	4
LD<>_U	4
LD>_U	4
LD<=_U	4
LD<_U	4
LD>=_U	4
AND=_U	4
AND<>_U	4
AND>_U	4

Instruction name	Number of minimum steps
AND<=_U	4
AND<_U	4
AND>=_U	4
OR=_U	4
OR<>_U	4
OR>_U	4
OR<=_U	4
OR<_U	4
OR>=_U	4
LDD=	4
LDD<>	4
LDD>	4
LDD<=	4
LDD<	4
LDD>=	4
ANDD=	4
ANDD<>	4
ANDD>	4
ANDD<=	4
ANDD<	4
ANDD>=	4
ORD=	4
ORD<>	4
ORD>	4
ORD<=	4
ORD<	4
ORD>=	4
LDD=_U	4
LDD<>_U	4
LDD>_U	4
LDD<=_U	4
LDD<_U	4
LDD>=_U	4
ANDD=_U	4
ANDD<>_U	4
ANDD>_U	4
ANDD<=_U	4
ANDD<_U	4
ANDD>=_U	4
ORD=_U	4
ORD<>_U	4
ORD>_U	4
ORD<=_U	4
ORD<_U	4
ORD>=_U	4
CMP	5
CMPP	5
CMP_U	5
CMPP_U	5
DCMP	5
DCMPP	5
DCMP_U	5
DCMPP_U	5

Instruction name	Number of minimum steps
ZCP	6
ZCPP	6
ZCP_U	6
ZCPP_U	6
DZCP	6
DZCPP	6
DZCP_U	6
DZCPP_U	6
BKCMP=	6
BKCMP<>	6
BKCMP>	6
BKCMP<=	6
BKCMP<	6
BKCMP>=	6
BKCMP=P	6
BKCMP<>P	6
BKCMP>P	6
BKCMP<=P	6
BKCMP<P	6
BKCMP>=P	6
BKCMP=_U	6
BKCMP<>_U	6
BKCMP>_U	6
BKCMP<=_U	6
BKCMP<_U	6
BKCMP>=_U	6
BKCMP=P_U	6
BKCMP<>P_U	6
BKCMP>P_U	6
BKCMP<=P_U	6
BKCMP<P_U	6
BKCMP>=P_U	6
DBKCMP=	6
DBKCMP<>	6
DBKCMP>	6
DBKCMP<=	6
DBKCMP<	6
DBKCMP>=	6
DBKCMP=P	6
DBKCMP<>P	6
DBKCMP>P	6
DBKCMP<=P	6
DBKCMP<P	6
DBKCMP>=P	6
DBKCMP=_U	6
DBKCMP<>_U	6
DBKCMP>_U	6
DBKCMP<=_U	6
DBKCMP<_U	6
DBKCMP>=_U	6
DBKCMP=P_U	6
DBKCMP<>P_U	6
DBKCMP>P_U	6

Instruction name	Number of minimum steps
DBKCMP<=P_U	6
DBKCMP<P_U	6
DBKCMP>=P_U	6
+ (s) (d)	5
+P (s) (d)	5
+ (s1) (s2) (d)	5
+P (s1) (s2) (d)	5
+_U (s) (d)	5
+P_U (s) (d)	5
+_U (s1) (s2) (d)	5
+P_U (s1) (s2) (d)	5
ADD	5
ADDP	5
ADD_U	5
ADDP_U	5
- (s) (d)	5
-P (s) (d)	5
- (s1) (s2) (d)	5
-P (s1) (s2) (d)	5
-_U (s) (d)	5
-P_U (s) (d)	5
-_U (s1) (s2) (d)	5
-P_U (s1) (s2) (d)	5
SUB	5
SUBP	5
SUB_U	5
SUBP_U	5
D+ (s) (d)	5
D+P (s) (d)	5
D+ (s1) (s2) (d)	5
D+P (s1) (s2) (d)	5
D+_U (s) (d)	5
D+P_U (s) (d)	5
D+_U (s1) (s2) (d)	5
D+P_U (s1) (s2) (d)	5
DADD	5
DADDP	5
DADD_U	5
DADDP_U	5
D- (s) (d)	5
D-P (s) (d)	5
D- (s1) (s2) (d)	5
D-P (s1) (s2) (d)	5
D-_U (s) (d)	5
D-P_U (s) (d)	5
D-_U (s1) (s2) (d)	5
D-P_U (s1) (s2) (d)	5
DSUB	5
DSUBP	5
DSUB_U	5
DSUBP_U	5
*	7
*P	7

Instruction name	Number of minimum steps
*_U	7
*P_U	7
MUL	7
MULP	7
MUL_U	7
MULP_U	7
/	7
/P	7
/_U	7
/P_U	7
DIV	7
DIVP	7
DIV_U	7
DIVP_U	7
D*	7
D*P	7
D*_U	7
D*P_U	7
DMUL	7
DMULP	7
DMUL_U	7
DMULP_U	7
D/	7
D/P	7
D/_U	7
D/P_U	7
DDIV	7
DDIVP	7
DDIV_U	7
DDIVP_U	7
B+ (s) (d)	4
B+P (s) (d)	4
B+ (s1) (s2) (d)	5
B+P (s1) (s2) (d)	5
B- (s) (d)	4
B-P (s) (d)	4
B- (s1) (s2) (d)	5
B-P (s1) (s2) (d)	5
DB+ (s) (d)	4
DB+P (s) (d)	4
DB+ (s1) (s2) (d)	5
DB+P (s1) (s2) (d)	5
DB- (s) (d)	4
DB-P (s) (d)	4
DB- (s1) (s2) (d)	5
DB-P (s1) (s2) (d)	5
B*	7
B*P	7
B/	7
B/P	7
DB*	7
DB*P	7
DB/	7

Instruction name	Number of minimum steps
DB/P	7
BK+	6
BK+P	6
BK+_U	6
BK+P_U	6
BK-	6
BK-P	6
BK-_U	6
BK-P_U	6
DBK+	6
DBK+P	6
DBK+_U	6
DBK+P_U	6
DBK-	6
DBK-P	6
DBK-_U	6
DBK-P_U	6
INC	3
INCP	3
INC_U	3
INCP_U	3
DEC	3
DECP	3
DEC_U	3
DECP_U	3
DINC	3
DINCP	3
DINC_U	3
DINCP_U	3
DDEC	3
DDECP	3
DDEC_U	3
DDECP_U	3
WAND (s) (d)	5
WANDP (s) (d)	5
WAND (s1) (s2) (d)	5
WANDP (s1) (s2) (d)	5
DAND (s) (d)	5
DANDP (s) (d)	5
DAND (s1) (s2) (d)	5
DANDP (s1) (s2) (d)	5
BKAND	6
BKANDP	6
WOR (s) (d)	5
WORP (s) (d)	5
WOR (s1) (s2) (d)	5
WORP (s1) (s2) (d)	5
DOR (s) (d)	5
DORP (s) (d)	5
DOR (s1) (s2) (d)	5
DORP (s1) (s2) (d)	5
BKOR	6
BKORP	6

Instruction name	Number of minimum steps
WXOR (s) (d)	5
WXORP (s) (d)	5
WXOR (s1) (s2) (d)	5
WXORP (s1) (s2) (d)	5
DXOR (s) (d)	5
DXORP (s) (d)	5
DXOR (s1) (s2) (d)	5
DXORP (s1) (s2) (d)	5
BKXOR	6
BKXORP	6
WXNR (s) (d)	5
WXNRP (s) (d)	5
WXNR (s1) (s2) (d)	5
WXNRP (s1) (s2) (d)	5
DXNR (s) (d)	5
DXNRP (s) (d)	5
DXNR (s1) (s2) (d)	5
DXNRP (s1) (s2) (d)	5
BKXNR	6
BKXNRP	6
BSET	4
BSETP	4
BRST	4
BRSTP	4
TEST	5
TESTP	5
DTEST	5
DTESTP	5
BKRST	4
BKRSTP	4
ZRST	4
ZRSTP	4
BCD	4
BCDP	4
DBCD	4
DBCDP	4
BIN	4
BINP	4
DBIN	4
DBINP	4
FLT2INT	4
FLT2INTP	4
FLT2UINT	4
FLT2UINTP	4
FLT2DINT	4
FLT2DINTP	4
FLT2UDINT	4
FLT2UDINTP	4
INT2UINT	4
INT2UINTP	4
INT2DINT	4
INT2DINTP	4
INT2UDINT	4

Instruction name	Number of minimum steps
INT2UDINTP	4
UINT2INT	4
UINT2INTP	4
UINT2DINT	4
UINT2DINTP	4
UINT2UDINT	4
UINT2UDINTP	4
DINT2INT	4
DINT2INTP	4
DINT2UINT	4
DINT2UINTP	4
DINT2UDINT	4
DINT2UDINTP	4
UDINT2INT	4
UDINT2INTP	4
UDINT2UINT	4
UDINT2UINTP	4
UDINT2DINT	4
UDINT2DINTP	4
GRY	4
GRYP	4
GRY_U	4
GRYP_U	4
DGRY	4
DGRYP	4
DGRY_U	4
DGRYP_U	4
GBIN	4
GBINP	4
GBIN_U	4
GBINP_U	4
DGBIN	4
DGBINP	4
DGBIN_U	4
DGBINP_U	4
DABIN	4
DABINP	4
DABIN_U	4
DABINP_U	4
DDABIN	4
DDABINP	4
DDABIN_U	4
DDABINP_U	4
HEXA	5
HEXAP	5
VAL	5
VALP	5
VAL_U	5
VALP_U	5
DVAL	5
DVALP	5
DVAL_U	5
DVALP_U	5

Instruction name	Number of minimum steps
NEG	3
NEGP	3
DNEG	3
DNEGP	3
DECO	5
DECOP	5
ENCO	5
ENCOP	5
SEGD	4
SEGDP	4
SEGL	5
DIS	5
DISP	5
UNI	5
UNIP	5
NDIS	5
NDISP	5
NUNI	5
NUNIP	5
WTOB	5
WTOBP	5
BTOW	5
BTOWP	5
DSW	6
MOV	4
MOVP	4
DMOV	4
DMOVP	4
CML	4
CMLP	4
DCML	4
DCMLP	4
SMOV	7
SMOVP	7
CMLB	4
CMLBP	4
BMOV	5
BMOVP	5
FMOV	5
FMOVP	5
DFMOV	5
DFMOVP	5
XCH	4
XCHP	4
DXCH	4
DXCHP	4
SWAP	3
SWAPP	3
DSWAP	3
DSWAPP	3
MOVB	4
MOVBP	4
PRUN	5

Instruction name	Number of minimum steps
PRUNP	5
DPRUN	5
DPRUNP	5
BLKMOVB	5
BLKMOVBP	5
ROR	4
RORP	4
RCR	4
RCRP	4
DROR	4
DRORP	4
DRCR	4
DRCRP	4
ROL	4
ROLP	4
RCL	4
RCLP	4
DROL	4
DROLP	4
DRCL	4
DRCLP	4
CJ	3
CJP	3
GOEND	1
DI	1
DI (s)	3
EI	1
IMASK	3
SIMASK	5
IRET	1
WDT	1
WDTP	1
FOR	3
NEXT	1
BREAK	5
BREAKP	5
CALL	3
CALLP	3
RET	1
SRET	1
XCALL	3
SFRD	5
SFRDP	5
POP	5
POPP	5
SFWR	5
SFWRP	5
FINS	5
FINSP	5
FDEL	5
FDELP	5
S.DEVLD	8
SP.DEVLD	8

Instruction name	Number of minimum steps
SP.DEVST	10
SP.FREAD	14
SP.FWRITE	14
SP.FDELETE	12
SP.FCOPY	16
SP.FMOVE	16
SP.FRENAME	14
SP.FSTATUS	14
ERREAD	8
ERWRITE	8
ERINIT	4
LD$=	4
LD$<>	4
LD$>	4
LD$<=	4
LD$<	4
LD$>=	4
AND$=	4
AND$<>	4
AND$>	4
AND$<=	4
AND$<	4
AND$>=	4
OR$=	4
OR$<>	4
OR$>	4
OR$<=	4
OR$<	4
OR$>=	4
$+ (s) (d)	4
$+P (s) (d)	4
$+ (s1) (s2) (d)	5
$+P (s1) (s2) (d)	5
$MOV	4
$MOVP	4
$MOV_WS	6
$MOVP_WS	6
BINDA	4
BINDAP	4
BINDA_U	4
BINDAP_U	4
DBINDA	4
DBINDAP	4
DBINDA_U	4
DBINDAP_U	4
ASCI	5
ASCIP	5
STR	5
STRP	5
STR_U	5
STRP_U	5
DSTR	5
DSTRP	5

Instruction name	Number of minimum steps
DSTR_U	5
DSTRP_U	5
ESTR	5
ESTRP	5
DESTR	5
DESTRP	5
WS2SJIS	6
WS2SJISP	6
SJIS2WS	6
SJIS2WSP	6
SJIS2WSB	6
SJIS2WSBP	6
LEN	4
LENP	4
RIGHT	5
RIGHTP	5
LEFT	5
LEFTP	5
MIDR	5
MIDRP	5
MIDW	5
MIDWP	5
INSTR	6
INSTRP	6
STRINS	5
STRINSP	5
STRDEL	5
STRDELP	5
LDE=	4
LDE<>	4
LDE>	4
LDE<=	4
LDE<	4
LDE>=	4
ANDE=	4
ANDE<>	4
ANDE>	4
ANDE<=	4
ANDE<	4
ANDE>=	4
ORE=	4
ORE<>	4
ORE>	4
ORE<=	4
ORE<	4
ORE>=	4
DECMP	5
DECMPP	5
DEZCP	6
DEZCPP	6
E+ (s) (d)	4
E+P (s) (d)	4
E+ (s1) (s2) (d)	5

Instruction name	Number of minimum steps
E+P (s1) (s2) (d)	5
DEADD	5
DEADDP	5
E- (s) (d)	4
E-P (s) (d)	4
E- (s1) (s2) (d)	5
E-P (s1) (s2) (d)	5
DESUB	5
DESUBP	5
E*	5
E*P	5
DEMUL	5
DEMULP	5
E/	5
E/P	5
DEDIV	5
DEDIVP	5
INT2FLT	4
INT2FLTP	4
UINT2FLT	4
UINT2FLTP	4
DINT2FLT	4
DINT2FLTP	4
UDINT2FLT	4
UDINT2FLTP	4
EVAL	4
EVALP	4
DEVAL	4
DEVALP	4
DEBCD	4
DEBCDP	4
DEBIN	4
DEBINP	4
ENEG	3
ENEGP	3
DENEG	3
DENEGP	3
EMOV	4
EMOVP	4
DEMOV	4
DEMOVP	4
SIN	4
SINP	4
DSIN	4
DSINP	4
COS	4
COSP	4
DCOS	4
DCOSP	4
TAN	4
TANP	4
DTAN	4
DTANP	4

Instruction name	Number of minimum steps
ASIN	4
ASINP	4
DASIN	4
DASINP	4
ACOS	4
ACOSP	4
DACOS	4
DACOSP	4
ATAN	4
ATANP	4
DATAN	4
DATANP	4
RAD	4
RADP	4
DRAD	4
DRADP	4
DEG	4
DEGP	4
DDEG	4
DDEGP	4
DESQR	4
DESQRP	4
ESQRT	4
ESQRTP	4
EXP	4
EXPP	4
DEXP	4
DEXPP	4
LOG	4
LOGP	4
DLOGE	4
DLOGEP	4
POW	5
POWP	5
LOG10	4
LOG10P	4
DLOG10	4
DLOG10P	4
EMAX	5
EMAXP	5
EMIN	5
EMINP	5
RND	3
RNDP	3
ZPUSH (d)	3
ZPUSHP (d)	3
ZPUSH (s) (d)	4
ZPUSHP (s) (d)	4
ZPOP (d)	3
ZPOPP (d)	3
ZPOP (s) (d)	4
ZPOPP (s) (d)	4
LIMIT	6

Instruction name	Number of minimum steps
LIMITP	6
LIMIT_U	6
LIMITP_U	6
DLIMIT	6
DLIMITP	6
DLIMIT_U	6
DLIMITP_U	6
BAND	6
BANDP	6
BAND_U	6
BANDP_U	6
DBAND	6
DBANDP	6
DBAND_U	6
DBANDP_U	6
ZONE	6
ZONEP	6
ZONE_U	6
ZONEP_U	6
DZONE	6
DZONEP	6
DZONE_U	6
DZONEP_U	6
SCL	5
SCLP	5
SCL_U	5
SCLP_U	5
DSCL	5
DSCLP	5
DSCL_U	5
DSCLP_U	5
SCL2	5
SCL2P	5
SCL2_U	5
SCL2P_U	5
DSCL2	5
DSCL2P	5
DSCL2_U	5
DSCL2P_U	5
TTMR	4
STMR	7
UDCNTF	5
ROTC	6
RAMPF	6
SPD	5
DSPD	5
PLSY	5
DPLSY	5
PWM	5
DPWM	6
MTR	6
IST	7
ABSD	9

Instruction name	Number of minimum steps
DABSD	9
INCD	9
CCD	5
CCDP	5
SERMM	6
SERMMP	6
DSERMM	6
DSERMMP	6
SUM	4
SUMP	4
DSUM	4
DSUMP	4
BON	5
BONP	5
DBON	5
DBONP	5
MAX	5
MAXP	5
MAX_U	5
MAXP_U	5
DMAX	5
DMAXP	5
DMAX_U	5
DMAXP_U	5
MIN	5
MINP	5
MIN_U	5
MINP_U	5
DMIN	5
DMINP	5
DMIN_U	5
DMINP_U	5
SORTTBL	7
SORTTBL_U	7
SORTTBL2	7
SORTTBL2_U	7
DSORTTBL2	7
DSORTTBL2_U	7
WSUM	7
WSUMP	7
WSUM_U	7
WSUMP_U	7
DWSUM	7
DWSUMP	7
DWSUM_U	7
DWSUMP_U	7
MEAN	5
MEANP	5
MEAN_U	5
MEANP_U	5
DMEAN	5
DMEANP	5
DMEAN_U	5

Instruction name	Number of minimum steps
DMEANP_U	5
SQRT	4
SQRTP	4
DSQRT	4
DSQRTP	4
CRC	5
CRCP	5
ADRSET	4
ADRSETP	4
TRD	3
TRDP	3
TWR	3
TWRP	3
TADD	5
TADDP	5
TSUB	5
TSUBP	5
HTOS	4
HTOSP	4
DHTOS	4
DHTOSP	4
STOH	4
STOHP	4
DSTOH	4
DSTOHP	4
LDDT=	5
LDDT<>	5
LDDT>	5
LDDT<=	5
LDDT<	5
LDDT>=	5
ANDDT=	5
ANDDT<>	5
ANDDT>	5
ANDDT<=	5
ANDDT<	5
ANDDT>=	5
ORDT=	5
ORDT<>	5
ORDT>	5
ORDT<=	5
ORDT<	5
ORDT>=	5
LDTM=	5
LDTM<>	5
LDTM>	5
LDTM<=	5
LDTM<	5
LDTM>=	5
ANDTM=	5
ANDTM<>	5
ANDTM>	5
ANDTM<=	5

Instruction name	Number of minimum steps
ANDTM<	5
ANDTM>=	5
ORTM=	5
ORTM<>	5
ORTM>	5
ORTM<=	5
ORTM<	5
ORTM>=	5
TCMP	7
TCMPP	7
TZCP	6
TZCPP	6
DUTY	5
HOURM	5
DHOURM	5
REF	4
REFP	4
RFS	4
RFSP	4
FROM	6
FROMP	6
DFROM	6
DFROMP	6
TO	6
TOP	6
DTO	6
DTOP	6
FROMD	6
FROMDP	6
DFROMD	6
DFROMDP	6
TOD	6
TODP	6
DTOD	6
DTODP	6
LOGTRG	3
LOGTRGR	3
RTM	2
STL	3
RETSTL	1
PID	6
LD [S□]	2
LD [BL□\S□]	3
LD [BL□]	3
LDI [S□]	2
LDI [BL□\S□]	3
LDI [BL□]	3
AND [S□]	2
AND [BL□\S□]	3
AND [BL□]	3
ANI [S□]	2
ANI [BL□\S□]	3
ANI [BL□]	3

Instruction name	Number of minimum steps
OR [S□]	2
OR [BL□\S□]	3
OR [BL□]	3
ORI [S□]	2
ORI [BL□\S□]	3
ORI [BL□]	3
MOV [KnS□]	5
MOV [BL□\KnS□]	7
MOVP [KnS□]	5
MOVP [BL□\KnS□]	7
DMOV [KnS□]	5
DMOV [BL□\KnS□]	7
DMOVP [KnS□]	5
DMOVP [BL□\KnS□]	7
BMOV [KnS□]	7
BMOV [BL□\KnS□]	9
BMOVP [KnS□]	7
BMOVP [BL□\KnS□]	9
SET [S□]	3
SET [BL□\S□]	3
SET [BL□]	3
RST [S□]	3
RST [BL□\S□]	3
RST [BL□]	3
OUT [S□]	3
OUT [BL□\S□]	3
ZRST [S□]	5
ZRST [BL□\S□]	9
ZRSTP [S□]	5
ZRSTP [BL□\S□]	9
GP.READ	19
GP.SREAD	23
GP.WRITE	20
GP.SWRITE	23
GP.SEND	16
GP.RECV	16
SP.SOCOPEN	10
SP.SOCCLOSE	10
SP.SOCRCV	12
SP.SOCSND	12
SP.SOCCINF	10
S.SOCRDATA	12
SP.SOCRDATA	12
SP.ECPRTCL	7
SP.SLMPSND	13
SP.FTPPUT	13
SP.FTPGET	13
GP.OPEN	10
GP.CLOSE	10
GP.SOCRCV	12
GP.SOCSND	12
G.CCPASET	22
GP.CCPASET	22

Instruction name	Number of minimum steps
G.UINI	14
GP.UINI	14
G.SLMPSND	20
GP.SLMPSND	20
DHSCS	5
DHSCR	5
DHSZ	6
HIOEN	5
HIOENP	5
DHIOEN	5
DHIOENP	5
HCMOV	5
HCMOVP	5
DHCMOV	5
DHCMOVP	5
RS2	7
IVCK	7
IVDR	7
IVRD	7
IVWR	7
IVBWR	7
IVMC	13
ADPRW	13
S.CPRTCL	7
SP.CPRTCL	7
DSZR	6
DDSZR	6
DVIT	6
DDVIT	6
TBL	4
DRVTBL	7
DRVMUL	13
DABS	5
PLSV	5
DPLSV	5
DRVI	6
DDRVI	6
DRVA	6
DDRVA	6
G.ABRST□	15
GP.PSTRT□	15
GP.TEACH□	15
GP.PFWRT	15
GP.PINIT	15
RBFM	7
WBFM	7

Appendix 3 Added and Changed Functions

The functions added or changed with the CPU module and engineering tool, and the supported CPU modules' firmware version and engineering tool software version are given below.

Firmware version can be checked through the module diagnostics (CPU diagnostics). For the module diagnostics (CPU diagnostics), refer to the following manuals.

MELSEC iQ-F FX5S/FX5UJ/FX5U/FX5UC User's Manual (Hardware)

For software version, refer to GX Works3 Operating Manual.

FX5S CPU module

Add/Change Function	Supported CPU module firmware version	Supported engineering tool software version	Reference
FX5S CPU module is supported.	From first production	"1.080J" or later	—
The SFC program instruction is supported.	"1.010" or later	"1.095Z" or later	Page 956 SFC PROGRAM INSTRUCTIONS

FX5UJ CPU module

Add/Change Function	Supported CPU module firmware version	Supported engineering tool software version	Reference
FX5UJ CPU module is supported.	From first production	"1.060N" or later	—
File operation instructions are supported.	"1.030" or later	"1.085P" or later	Page 486 SP.FREAD Page 512 SP.FWRITE Page 535 SP.FDELETE Page 543 SP.FCOPY Page 553 SP.FMOVE Page 563 SP.FRENAME Page 571 SP.FSTATUS
Unicode string data transfer instruction is supported.	"1.030" or later	"1.085P" or later	Page 598 $MOV(P)_WS
Unicode character string to Shift JIS character string convert instruction is supported.	"1.030" or later	"1.085P" or later	Page 628 WS2SJIS(P)
Shift JIS character string to Unicode character string convert instruction (without byte order mark) is supported.	"1.030" or later	"1.085P" or later	Page 631 SJIS2WS(P)
Shift JIS character string to Unicode convert instruction (with byte order mark) is supported.	"1.030" or later	"1.085P" or later	Page 634 SJIS2WSB(P)
The file transfer function instruction is supported.	"1.030" or later	"1.085P" or later	Page 1054 SP.FTPPUT Page 1059 SP.FTPGET
The SFC program instruction is supported.	"1.050" or later	"1.095Z" or later	Page 956 SFC PROGRAM INSTRUCTIONS

FX5U/FX5UC CPU module

Add/Change Function	Supported CPU module firmware version	Supported engineering tool software version	Reference
The ESQRT(P) instruction is supported.	"1.015" and above[*1]	"1.015R" and above	Page 720 DESQR(P)/ESQRT(P)
The number of settable high-speed comparison tables was changed from maximum 4 to 32.	"1.015" and above[*1]	"1.015R" and above	Page 1100 HIOEN(P) Page 1104 DHIOEN(P)
The number of high-speed comparisons was changed from 4 to 32.	"1.015" and above[*1]	"1.015R" and above	Page 1100 HIOEN(P) Page 1104 DHIOEN(P)

Add/Change Function	Supported CPU module firmware version	Supported engineering tool software version	Reference
The high-speed pulse input/output module is supported.	"1.030" and above	"1.025B" and above	Page 809 PWM Page 816 DPWM Page 1100 HIOEN(P) Page 1104 DHIOEN(P) Page 1108 HCMOV(P) Page 1110 DHCMOV(P) Page 1133 POSITIONING INSTRUCTION
The CC-Link IE Field Network intelligent device station is supported.	"1.030" and above	"1.025B" and above	Page 986 NETWORK COMMON INSTRUCTION Page 1085 CC-LINK IE FIELD NETWORK INSTRUCTION
The logging instruction is supported.	"1.040" and above*2	"1.030G" and above	Page 945 LOGTRG Page 946 LOGTRGR
The SLMP frame send instruction is supported.	"1.040" and above	"1.030G" and above	Page 1049 SP.SLMPSND
The positioning module instruction is supported.	"1.050" and above	"1.035M" and above	Page 1180 G.ABRST□ Page 1184 GP.PSTRT□ Page 1187 GP.TEACH□ Page 1190 GP.PFWRT Page 1193 GP.PINIT
Reading/writing data instructions.	"1.060" and above	"1.040S" and above	Page 481 S(P).DEVLD Page 483 SP.DEVST
Extended file register operation instruction.	"1.060" and above*2	"1.040S" and above	Page 580 ERREAD Page 583 ERWRITE Page 586 ERINIT
Real-time monitor function instruction.	"1.060" and above	"1.040S" and above	Page 947 RTM
Slave station No. (MODBUS address number) that can be set in FX5 master station was changed from 32 to 247.	"1.060" and above	"1.040S" and above	Page 1127 ADPRW
Add U□\G□ to applicable devices of BLKMOV(P) instruction.	"1.060" and above	"1.040S" and above	Page 424 BLKMOVB(P)
FBD/LD language and ST language of step ladder instructions is supported.	—	"1.045X" and above	Page 949 STL, RETSTL
The Ethernet module instruction is supported.	"1.110" and above	"1.050C" and above	Page 1064 GP.OPEN Page 1067 GP.CLOSE Page 1069 GP.SOCRCV Page 1071 GP.SOCSND
Down counting of LC0 to LC34 is supported.	"1.201" and above	"1.060N" and above	Page 774 UDCNTF
The file transfer function instruction is supported.	"1.210" and above	"1.065T" and above	Page 1054 SP.FTPPUT
The CC-Link IE TSN instruction is supported.	"1.210" and above	"1.065T" and above	Page 1073 G(P).UINI
The BL (SFC block device) is supported.	"1.220" and above	"1.070Y" and above	Page 17
The SFC control instruction is supported.	"1.220" and above	"1.070Y" and above	Page 956 LD, LDI, AND, ANI, OR, ORI [S□/BL□\S□] Page 958 LD, LDI, AND, ANI, OR, ORI [BL□] Page 960 MOV(P) [KnS□/BL□\KnS□] Page 963 DMOV(P) [KnS□/BL□\KnS□] Page 966 BMOV(P) [KnS□/BL□\KnS□] Page 970 SET [BL□] Page 971 RST [BL□] Page 976 SET [S□/BL□\S□] Page 978 RST [S□/BL□\S□] Page 980 OUT [S□/BL□\S□] Page 982 ZRST(P) [S□/BL□\S□]
The SFC dedicated instruction is supported.	"1.220" and above	"1.070Y" and above	Page 984 TRAN

Add/Change Function	Supported CPU module firmware version	Supported engineering tool software version	Reference
File operation instructions is supported.	"1.240" and above	"1.075D" and above	Page 486 SP.FREAD Page 512 SP.FWRITE Page 535 SP.FDELETE Page 543 SP.FCOPY Page 553 SP.FMOVE Page 563 SP.FRENAME Page 571 SP.FSTATUS
Unicode string data transfer instruction is supported.	"1.240" and above	"1.075D" and above	Page 598 $MOV(P)_WS
Unicode character string to Shift JIS character string convert instruction is supported.	"1.240" and above	"1.075D" and above	Page 628 WS2SJIS(P)
Shift JIS character string to Unicode character string convert instruction (without byte order mark) is supported.	"1.240" and above	"1.075D" and above	Page 631 SJIS2WS(P)
Shift JIS character string to Unicode convert instruction (with byte order mark) is supported.	"1.240" and above	"1.075D" and above	Page 634 SJIS2WSB(P)
Retrieving FTP client files instruction is supported.	"1.240" and above	"1.075D" and above	Page 1059 SP.FTPGET
The SFC control instruction is supported.	"1.290" and above	"1.095Z" and above	Page 972 PAUSE [BL□] Page 974 RSTART [BL□]

*1 Supported with CPU module serial No. 158**** and above.
*2 Supported with CPU module serial No. 16Y**** and above.

INSTRUCTION INDEX

Symbols

-(P)(_U) 214,216
*(P)(_U) 232
/(P)(_U) 236
+(P)(_U) 208,210
$+(P) 592,594
$MOV(P) 596
$MOV(P)_WS 598

A

ABS(_E) 1283
ABSD 838
ACOS(_E) 1296
ACOS(P) 711
ADD(_E) 1300
ADD(P)(_U) 212
ADD_TIME(_E) 1354
ADPRW 1127
ADRSET(P) 890
ALT(P) 157
ANB 123
AND 114,956,958
AND(_E) 1322
AND<(_U) 189
AND<=(_U) 189
AND<>(_U) 189
AND=(_U) 189
AND>(_U) 189
AND>=(_U) 189
AND$< 589
AND$<= 589
AND$<> 589
AND$= 589
AND$> 589
AND$>= 589
ANDD<(_U) 192
ANDD<=(_U) 192
ANDD<>(_U) 192
ANDD=(_U) 192
ANDD>(_U) 192
ANDD>=(_U) 192
ANDD_EQ(_U) 192
ANDD_GE(_U) 192
ANDD_GT(_U) 192
ANDD_LE(_U) 192
ANDD_LT(_U) 192
ANDD_NE(_U) 192
ANDDT< 911
ANDDT<= 911
ANDDT<> 911
ANDDT= 911
ANDDT> 911
ANDDT>= 911
ANDDT_EQ 911
ANDDT_GE 911
ANDDT_GT 911
ANDDT_LE 911
ANDDT_LT 911
ANDDT_NE 911
ANDE< 659
ANDE<= 659
ANDE<> 659
ANDE= 659
ANDE> 659
ANDE>= 659
ANDE_EQ 659
ANDE_GE 659
ANDE_GT 659
ANDE_LE 659
ANDE_LT 659
ANDE_NE 659
AND_EQ(_U) 189
ANDF 117
ANDFI 121
AND_GE(_U) 189
AND_GT(_U) 189
AND_LE(_U) 189
AND_LT(_U) 189
AND_NE(_U) 189
ANDP 117
ANDPI 121
ANDSTRING_EQ 589
ANDSTRING_GE 589
ANDSTRING_GT 589
ANDSTRING_LE 589
ANDSTRING_LT 589
ANDSTRING_NE 589
ANDTM< 914
ANDTM<= 914
ANDTM<> 914
ANDTM= 914
ANDTM> 914
ANDTM>= 914
ANDTM_EQ 914
ANDTM_GE 914
ANDTM_GT 914
ANDTM_LE 914
ANDTM_LT 914
ANDTM_NE 914
ANI 114,956,958
ANR(P) 151
ANS 149
ASCI(P) 611
ASIN(_E) 1294
ASIN(P) 708
ATAN(_E) 1298
ATAN(P) 714

B

B-(P) 250,251
B*(P) 259
B/(P) 261
B+(P) 248,249
BAND(P)(_U) 747
BCD(P) 318
BCD_TO_DINT(_E) 1249
BCD_TO_INT(_E) 1247
BDIVISION(P) 261
BIN(P) 322
BINDA(P)(_U) 600
BITARR_TO_DINT(_E) 1275

BITARR_TO_INT(_E) 1274
BK-(P)(_U) 269
BK+(P)(_U) 267
BKAND(P) 285
BKCMP<(P)(_U) 202
BKCMP<=(P)(_U) 202
BKCMP<>(P)(_U) 202
BKCMP=(P)(_U) 202
BKCMP>(P)(_U) 202
BKCMP>=(P)(_U) 202
BKCMP_EQ(P)(_U) 202
BKCMP_GE(P)(_U) 202
BKCMP_GT(P)(_U) 202
BKCMP_LE(P)(_U) 202
BKCMP_LT(P)(_U) 202
BKCMP_NE(P)(_U) 202
BKMINUS(P)(_U) 269
BKOR(P) 292
BKPLUS(P)(_U) 267
BKRST(P) 314
BKXNR(P) 306
BKXOR(P) 299
BLKMOVB(P) 424
BMINUS(P) 250,251
BMOV(P) 407,966
BMULTI(P) 259
BON(P) 855
BOOL_TO_DINT(_E) 1208
BOOL_TO_DWORD(_E) 1206
BOOL_TO_INT(_E) 1207
BOOL_TO_STRING(_E) 1210
BOOL_TO_TIME(_E) 1209
BOOL_TO_WORD(_E) 1204
BPLUS(P) 248,249
BREAK(P) 458
BRST(P) 309
BSET(P) 308
BSFL(P) 167
BSFR(P) 165
BTOW(P) 392

C

CALL(P) 460
CCD(P) 845
CJ(P) 436
CML(P) 401
CMLB(P) 406
CMP(P)(_U) 194
CONCAT(_E) 1343
COS(_E) 1291
COS(P) 704
COUNTER_FB_M 1379
CPY_BITARR(_E) 1278
CPY_BIT_OF_INT(_E) 1281
CRC(P) 887
CTD(_E) 1374
CTU(_E) 1372
CTUD(_E) 1376

D

D-(P)(_U) 226,228
D*(P)(_U) 240
D/(P)(_U) 244
D+(P)(_U) 220,222
DABIN(P)(_U) 352
DABS 1154
DABSD 840
DACOS(P) 711
DADD(P)(_U) 224
DAND(P) 282,283
DASIN(P) 708
DATAN(P) 714
DB-(P) 256,257
DB*(P) 263
DB/(P) 265
DB+(P) 253,254
DBAND(P)(_U) 749
DBCD(P) 320
DBDIVISION(P) 265
DBIN(P) 324
DBINDA(P)(_U) 605
DBK-(P)(_U) 273
DBK+(P)(_U) 271
DBKCMP<(P)(_U) 205
DBKCMP<=(P)(_U) 205
DBKCMP<>(P)(_U) 205
DBKCMP=(P)(_U) 205
DBKCMP>(P)(_U) 205
DBKCMP>=(P)(_U) 205
DBKCMP_EQ(P)(_U) 205
DBKCMP_GE(P)(_U) 205
DBKCMP_GT(P)(_U) 205
DBKCMP_LE(P)(_U) 205
DBKCMP_LT(P)(_U) 205
DBKCMP_NE(P)(_U) 205
DBKMINUS(P)(_U) 273
DBKPLUS(P)(_U) 271
DBMINUS(P) 256,257
DBMULTI(P) 263
DBON(P) 857
DBPLUS(P) 253,254
DCML(P) 403
DCMP(P)(_U) 196
DCOS(P) 704
DDABIN(P)(_U) 356
DDEC(P)(_U) 279
DDEG(P) 718
DDIV(P)(_U) 246
DDIVISION(P)(_U) 244
DDRVA 1176,1178
DDRVI 1168,1170
DDSZR 1138
DDVIT 1144,1146
DEADD(P) 673
DEBCD(P) 694
DEBIN(P) 696
DEC(P)(_U) 277
DECMP(P) 661
DECO(P) 373
DEDIV(P) 683
DEG(P) 718
DELETE(_E) 1347
DEMOV(P) 700
DEMUL(P) 681
DENEG(P) 698
DESQR(P) 720
DESTR(P) 621
DESUB(P) 675
DEVAL(P) 689
DEXP(P) 722

DEZCP(P) 663
DFMOV(P) 411
DFROM(P) 932,935
DFROMD(P) 939,941
DGBIN(P)(_U) 350
DGRY(P)(_U) 347
DHCMOV(P) 1110
DHIOEN(P) 1104
DHOURM 928
DHSCR 1094
DHSCS 1091
DHSCS_I 1091
DHSZ 1097
DHTOS(P) 905
DI 441,443
DI_1 443
DINC(P)(_U) 278
DINT2FLT(P) 687
DINT2INT(P) 340
DINT2UDINT(P) 342
DINT2UINT(P) 341
DINT_TO_BCD(_E) 1241
DINT_TO_BITARR(_E) 1277
DINT_TO_BOOL(_E) 1235
DINT_TO_DWORD(_E) 1238
DINT_TO_INT(_E) 1239
DINT_TO_REAL(_E) 1243
DINT_TO_STRING(_E) 1245
DINT_TO_TIME(_E) 1244
DINT_TO_WORD(_E) 1236
DIS(P) 382
DIV(_E) 1306
DIV(P)(_U) 238
DIVISION(P)(_U) 236
DIV_TIME(_E) 1360
DLIMIT(P)(_U) 745
DLOG10(P) 728
DLOGE(P) 724
DMAX(P)(_U) 861
DMEAN(P)(_U) 882
DMIN(P)(_U) 865
DMINUS(P)(_U) 226,228
DMOV(P) 399,963
DMUL(P)(_U) 242
DMULTI(P)(_U) 240
DNEG(P) 372
DOR(P) 289,290
DPLSV 1160,1162
DPLSY 801,805
DPLUS(P)(_U) 220,222
DPRUN(P) 422
DPWM 816
DRAD(P) 716
DRCL(P) 434
DRCR(P) 432
DROL(P) 434
DROR(P) 432
DRVA 1172,1174
DRVI 1164,1166
DRVMUL 1152
DRVTBL 1150
DSCL(P)(_U) 759
DSCL2(P)(_U) 766
DSERMM(P) 850
DSFL(P) 170
DSFR(P) 169
DSIN(P) 702
DSORTTBL2(_U) 873
DSPD 788
DSQRT(P) 886
DSTOH(P) 909
DSTR(P)(_U) 618
DSUB(P)(_U) 230
DSUM(P) 854
DSW 395
DSWAP(P) 418
DSZR 1134,1136
DTAN(P) 706
DTEST(P) 312
DTO(P) 935,936
DTOD(P) 941,942
DUTY 923
DVAL(P)(_U) 367
DVIT 1140,1142
DWORD_TO_BOOL(_E) 1217
DWORD_TO_DINT(_E) 1222
DWORD_TO_INT(_E) 1220
DWORD_TO_TIME(_E) 1223
DWORD_TO_WORD(_E) 1218
DWSUM(P)(_U) 878
DXCH(P) 415
DXNR(P) 303,304
DXOR(P) 296,297
DZCP(P)(_U) 200
DZONE(P)(_U) 754

E

E-(P) 669,671
E*(P) 677
E/(P) 679
E+(P) 665,667
EDIVISION(P) 679
EI 441
EMAX(P) 730
EMIN(P) 732
EMINUS(P) 669,671
EMOV(P) 700
EMULTI(P) 677
ENCO(P) 375
END 186
ENEG(P) 698
EPLUS(P) 665,667
EQ(_E) 1333
ERINIT 586
ERREAD 580
ERWRITE 583
ESQRT(P) 720
ESTR(P) 621
EVAL(P) 689
EXP(_E) 1289
EXP(P) 722
EXPT(_E) 1310

F

FDEL(P) 478
FEND 183
FF 156
FIND(_E) 1352
FINS(P) 476
FLT2DINT(P) 330

FLT2INT(P) 326
FLT2UDINT(P) 332
FLT2UINT(P) 328
FMOV(P) 409
FOR 455
FROM(P) 932,935
FROMD(P) 939,941
F_TRIG(_E) 1370

G

G.ABRST1 1180
G.ABRST2 1180
G(P).CCPASET 1085
G(P).SLMPSND 1076
G(P).UINI 1073,1088
G(P)_CCPASET 1085
G(P)_SLMPSND 1076
G(P)_UINI 1073,1088
G_ABRST1 1180
G_ABRST2 1180
GBIN(P)(_U) 349
GE(_E) 1333
GET_BIT_OF_INT(_E) 1279
GET_BOOL_ADDR 1282
GET_INT_ADDR 1282
GET_WORD_ADDR 1282
GOEND 440
GP.CLOSE 1067
GP.OPEN 1064
GP.PFWRT 1190
GP.PINIT 1193
GP.PSTRT1 1184
GP.PSTRT2 1184
GP.READ 988
GP.RECV 1024
GP.SEND 1016
GP.SOCRCV 1069
GP.SOCSND 1071
GP.SREAD 994
GP.SWRITE 1008
GP.TEACH1 1187
GP.TEACH2 1187
GP.WRITE 1000
GP_CLOSE 1067
GP_OPEN 1064
GP_PFWRT 1190
GP_PINIT 1193
GP_PSTRT1 1184
GP_PSTRT2 1184
GP_READ 988
GP_RECV 1024
GP_SEND 1016
GP_SOCRCV 1069
GP_SOCSND 1071
GP_SREAD 994
GP_SWRITE 1008
GP_TEACH1 1187
GP_TEACH2 1187
GP_WRITE 1000
GRY(P)(_U) 346
GT(_E) 1333

H

HCMOV(P) 1108
HEXA(P) 360
HIOEN(P) 1100
HOURM 926
HTOS(P) 903

I

IMASK 447
INC(P)(_U) 275
INCD 842
INSERT(_E) 1345
INSTR(P) 652
INT2DINT(P) 335
INT2FLT(P) 685
INT2UDINT(P) 336
INT2UINT(P) 334
INT_TO_BCD(_E) 1229
INT_TO_BITARR(_E) 1276
INT_TO_BOOL(_E) 1224
INT_TO_DINT(_E) 1228
INT_TO_DWORD(_E) 1226
INT_TO_REAL(_E) 1231
INT_TO_STRING(_E) 1233
INT_TO_TIME(_E) 1232
INT_TO_WORD(_E) 1225
INV 128
IRET 451
IST 827
IVBWR 1123
IVCK 1115
IVDR 1117
IVMC 1125
IVRD 1119
IVWR 1121

L

LD 114,956,958
LD<(_U) 189
LD<=(_U) 189
LD<>(_U) 189
LD=(_U) 189
LD>(_U) 189
LD>=(_U) 189
LD$< 589
LD$<= 589
LD$<> 589
LD$= 589
LD$> 589
LD$>= 589
LDD<(_U) 192
LDD<=(_U) 192
LDD<>(_U) 192
LDD=(_U) 192
LDD>(_U) 192
LDD>=(_U) 192
LDD_EQ(_U) 192
LDD_GE(_U) 192
LDD_GT(_U) 192
LDD_LE(_U) 192
LDD_LT(_U) 192
LDD_NE(_U) 192
LDDT< 911
LDDT<= 911

LDDT<> 911
LDDT= 911
LDDT> 911
LDDT>= 911
LDDT_EQ 911
LDDT_GE 911
LDDT_GT 911
LDDT_LE 911
LDDT_LT 911
LDDT_NE 911
LDE< 659
LDE<= 659
LDE<> 659
LDE= 659
LDE> 659
LDE>= 659
LDE_EQ 659
LDE_GE 659
LDE_GT 659
LDE_LE 659
LDE_LT 659
LDE_NE 659
LD_EQ(_U) 189
LDF 117
LDFI 121
LD_GE(_U) 189
LD_GT(_U) 189
LDI 114,956,958
LD_LE(_U) 189
LD_LT(_U) 189
LD_NE(_U) 189
LDP 117
LDPI 121
LDSTRING_EQ 589
LDSTRING_GE 589
LDSTRING_GT 589
LDSTRING_LE 589
LDSTRING_LT 589
LDSTRING_NE 589
LDTM< 914
LDTM<= 914
LDTM<> 914
LDTM= 914
LDTM> 914
LDTM>= 914
LDTM_EQ 914
LDTM_GE 914
LDTM_GT 914
LDTM_LE 914
LDTM_LT 914
LDTM_NE 914
LE(_E) 1333
LEFT(_E) 1339
LEFT(P) 642
LEN(_E) 1337
LEN(P) 637
LIMIT(_E) 1329
LIMIT(P)(_U) 743
LN(_E) 1286
LOG(_E) 1287
LOG(P) 724
LOG10(P) 728
LOGTRG 945
LOGTRGR 946
LT(_E) 1333

M

MAX(_E) 1327
MAX(P)(_U) 859
MC 179
MCR 179
MEAN(P)(_U) 880
MEF 129
MEP 129
MID(_E) 1341
MIDR(P) 645
MIDW(P) 648
MIN(_E) 1327
MIN(P)(_U) 863
MINUS(P)(_U) 214,216
MOD(_E) 1308
MOV(P) 397,960
MOVB(P) 419
MOVE(_E) 1312
MPP 125
MPS 125
MRD 125
MTR 823
MUL(_E) 1302
MUL(P)(_U) 234
MULTI(P)(_U) 232
MUL_TIME(_E) 1358
MUX(_E) 1331

N

NDIS(P) 386
NE(_E) 1335
NEG(P) 370
NEXT 455
NOT(_E) 1324
NUNI(P) 388

O

OR 114,956,958
OR(_E) 1322
OR<(_U) 189
OR<=(_U) 189
OR<>(_U) 189
OR=(_U) 189
OR>(_U) 189
OR>=(_U) 189
OR$< 589
OR$<= 589
OR$<> 589
OR$= 589
OR$> 589
OR$>= 589
ORB 123
ORD<(_U) 192
ORD<=(_U) 192
ORD<>(_U) 192
ORD=(_U) 192
ORD>(_U) 192
ORD>=(_U) 192
ORD_EQ(_U) 192
ORD_GE(_U) 192
ORD_GT(_U) 192
ORD_LE(_U) 192
ORD_LT(_U) 192
ORD_NE(_U) 192

ORDT< 911
ORDT<= 911
ORDT<> 911
ORDT= 911
ORDT> 911
ORDT>= 911
ORDT_EQ 911
ORDT_GE 911
ORDT_GT 911
ORDT_LE 911
ORDT_LT 911
ORDT_NE 911
ORE< 659
ORE<= 659
ORE<> 659
ORE= 659
ORE> 659
ORE>= 659
ORE_EQ 659
ORE_GE 659
ORE_GT 659
ORE_LE 659
ORE_LT 659
ORE_NE 659
OR_EQ(_U) 189
ORF 117
ORFI 121
OR_GE(_U) 189
OR_GT(_U) 189
ORI 114,956,958
OR_LE(_U) 189
OR_LT(_U) 189
OR_NE(_U) 189
ORP 117
ORPI 121
ORSTRING_EQ 589
ORSTRING_GE 589
ORSTRING_GT 589
ORSTRING_LE 589
ORSTRING_LT 589
ORSTRING_NE 589
ORTM< 914
ORTM<= 914
ORTM<> 914
ORTM= 914
ORTM> 914
ORTM>= 914
ORTM_EQ 914
ORTM_GE 914
ORTM_GT 914
ORTM_LE 914
ORTM_LT 914
ORTM_NE 914
OUT 130,139,980
OUT C 135
OUT F 139
OUT LC 137
OUT ST 132
OUT T 132
OUT_C 135,137
OUTH 132
OUTH ST 132
OUTH T 132
OUTHS 132
OUTHS ST 132
OUTHS T 132
OUT_T 132

P

PAUSE 972
PID 953
PLF 154
PLS 152
PLSV 1156,1158
PLSY 793,797
PLUS(P)(_U) 208,210
POP(P) 471
POW(P) 726
PRUN(P) 420
PWM 809

R

RAD(P) 716
RAMPF 779
RBFM 1196
RCL(P) 429
RCR(P) 426
REAL_TO_DINT(_E) 1253
REAL_TO_INT(_E) 1251
REAL_TO_STRING(_E) 1255
REF(P) 930
REPLACE(_E) 1349
RET 465
RETSTL 949
RFS(P) 930
RIGHT(_E) 1339
RIGHT(P) 639
RND(P) 734
ROL(_E) 1318
ROL(P) 429
ROR(_E) 1320
ROR(P) 426
ROTC 776
RS(_E) 1366
RS2 1113
RST 143,147,971,978
RST F 147
RSTART 974
RTM 947
R_TRIG(_E) 1368

S

S(P).CPRTCL 1129
S(P).DEVLD 481
S(P).SOCRDATA 1043
S(P)_CPRTCL 1129
S(P)_DEVLD 481
S(P)_SOCRDATA 1043
SCL(P)(_U) 756
SCL2(P)(_U) 762
SEGD(P) 377
SEGL 379
SEL(_E) 1325
SERMM(P) 848
SET 141,145,970,976
SET F 145
SET_BIT_OF_INT(_E) 1280
SFL(P) 163
SFR(P) 161

SFRD(P) 468
SFT(P) 159
SFTL(P) 173
SFTR(P) 171
SFWR(P) 474
SHL(_E) 1314
SHR(_E) 1316
SIMASK 449
SIN(_E) 1290
SIN(P) 702
SJIS2WS(P) 631
SJIS2WSB(P) 634
SMOV(P) 404
SORTTBL(_U) 867
SORTTBL2(_U) 870
SP.DEVST 483
SP.ECPRTCL 1045
SP.FCOPY 543
SP.FDELETE 535
SP.FMOVE 553
SP.FREAD 486
SP.FRENAME 563
SP.FSTATUS 571
SP.FTPGET 1059
SP.FTPPUT 1054
SP.FWRITE 512
SP.SLMPSND 1049
SP.SOCCINF 1041
SP.SOCCLOSE 1033
SP.SOCOPEN 1030
SP.SOCRCV 1035
SP.SOCSND 1038
SPD 782
SP_DEVST 483
SP_ECPRTCL 1045
SP_FCOPY 543
SP_FDELETE 535
SP_FMOVE 553
SP_FREAD 486
SP_FRENAME 563
SP_FSTATUS 571
SP_FTPGET 1059
SP_FTPPUT 1054
SP_FWRITE 512
SP_SLMPSND 1049
SP_SOCCINF 1041
SP_SOCCLOSE 1033
SP_SOCOPEN 1030
SP_SOCRCV 1035
SP_SOCSND 1038
SQRT(_E) 1285
SQRT(P) 884
SR(_E) 1364
SRET 465
STL 949
STMR 772
STOH(P) 907
STOP 188
STR(P)(_U) 615
STRDEL(P) 657
STRINGMOV(P) 596
STRINGMOV(P)_WS 598
STRINGPLUS(P) 592,594
STRING_TO_BOOL(_E) 1265
STRING_TO_DINT(_E) 1268
STRING_TO_INT(_E) 1266
STRING_TO_REAL(_E) 1270
STRING_TO_TIME(_E) 1273
STRINS(P) 655
SUB(_E) 1304
SUB(P)(_U) 218
SUB_TIME(_E) 1356
SUM(P) 852
SWAP(P) 417

T

TADD(P) 897
TAN(_E) 1292
TAN(P) 706
TBL 1148,1149
TCMP(P) 917
TEST(P) 310
TIMER_100_FB_M 1387
TIMER_10_FB_M 1387
TIMER_1_FB_M 1387
TIMER_CONT_FB_M 1387
TIMER_CONTHS_FB_M 1387
TIME_TO_BOOL(_E) 1258
TIME_TO_DINT(_E) 1262
TIME_TO_DWORD(_E) 1260
TIME_TO_INT(_E) 1261
TIME_TO_STRING(_E) 1263
TIME_TO_WORD(_E) 1259
TO(P) 935,936
TOD(P) 941,942
TOF(_E) 1385
TOF_10(_E) 1385
TON(_E) 1383
TON_10(_E) 1383
TP(_E) 1381
TP_10(_E) 1381
TRAN 984
TRD(P) 892
TSUB(P) 900
TTMR 769
TWR(P) 894
TZCP(P) 920

U

UDCNTF 774
UDINT2DINT(P) 345
UDINT2FLT(P) 688
UDINT2INT(P) 343
UDINT2UINT(P) 344
UINT2DINT(P) 338
UINT2FLT(P) 686
UINT2INT(P) 337
UINT2UDINT(P) 339
UNI(P) 384

V

VAL(P)(_U) 364

W

WAND(P) 280,281
WBFM 1200
WDT(P) 454
WOR(P) 287,288

I

WORD_TO_BOOL(_E) 1211
WORD_TO_DINT(_E) 1214
WORD_TO_DWORD(_E) 1212
WORD_TO_INT(_E) 1213
WORD_TO_TIME(_E) 1216
WS2SJIS(P) 628
WSFL(P) 177
WSFR(P) 175
WSUM(P)(_U) 876
WTOB(P) 390
WXNR(P) 301,302
WXOR(P) 294,295

X

XCALL 466
XCH(P) 413
XOR(_E) 1322

Z

ZCP(P)(_U) 198
ZONE(P)(_U) 752
ZPOP(P) 739,742
ZPOP(P)_2 742
ZPUSH(P) 736,740
ZPUSH(P)_2 740
ZRST(P) 315,982

REVISIONS

Revision date	Revision	Description
October 2014	A	First Edition
January 2015	B	Added or modified parts Section 6.5, 6.6, 7.12, 7.16, 7.17, Chapter 8, 9, 10, 11, 12, 13, Appendix 1 "FBD/LD language" (for each instruction) Error correction
February 2015	C	Error correction
April 2015	D	■A part of the cover design is changed.
August 2015	E	■Added or modified parts RELEVANT MANUALS, Section 8.24, 10.1, Chapter 12, Appendix 3 Section 1.4 was changed to Chapter 2.
May 2016	F	■Added or modified parts RELEVANT MANUALS, TERMS, Section 4.1, 4.2, 8.2, 8.15, 8.24, Chapter 12, 13, 14, 16
October 2016	G	■Added or modified parts TEAMS, 3.3, 3.5, 6.3, 8.25, 10.3, 10.4, 14.1, Appendix 2, 3
April 2017	H	■Added or modified parts RELEVANT MANUALS, TERMS, Section 4.6, Chapter 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 24, 30, Appendix 2, 3
October 2017	J	■Added or modified parts RELEVANT MANUALS, TERMS, Section 3.3, 7.7, 8.6, 8.7, 8.28, 10.2, 14.1, 23.2, 23.3, Chapter 21, 24, Appendix 2, 3
April 2018	K	■Added or modified parts Section 1.2, 2.4, 6.3, 6.5, 7.4, 7.5, 8.6, 8.7, 8.8, 8.9, 8.13, 8.14, 8.17, 8.21, 8.22, 8.23, 8.24, 8.25, 8.26, Chapter 9, Section 10.1, 10.2, 10.4, Chapter 12, 13, Section 14.1, 15.3, Chapter 16, Appendix 3
July 2018	L	■Added or modified parts TERMS, Section 1.4, 6.3, 18.35, 18.36, 20.5, Appendix 1
October 2018	M	■Added or modified parts RELEVANT MANUALS,TERMS, Section 1.2, 3.5, 4.2, 12.1, 12.2, 12.3, 12.4, 12.5, Appendix 2, 3
May 2019	N	■Added or modified parts TERMS, Section 6.1, 6.2, 6.3, 6.4, 6.6, 7.1, 7.2, 7.3, 7.4, 7.5, 7.7, 8.2, 8.3, 8.4, 8.5, 8.8, 8.9, 8.10, 8.11, 8.12, 8.13, 8.17, 8.22, 8.24, 8.25, 8.26, 14.1, 17.1
October 2019	P	■Added models FX5UJ CPU module Added or modified parts RELEVANT MANUALS, TERMS, HOW TO READ THIS MANUAL, Section 1.4, Chapter 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, Appendix 1, 3
May 2020	Q	■Added or modified parts RELEVANT MANUALS, TERMS, Section 1.2, 1.3, Chapter 4, Section 8.4, 8.18, Chapter11, Section 12.4, 12.5, Chapter 13, Appendix 1, 2, 3, TRADEMARKS
October 2020	R	■Added or modified parts RELEVANT MANUALS, HOW TO READ THIS MANUAL, Section 3.3, 3.6, 7.5, 8.9, 8.14, 8.17, 8.28, 9.1, Chapter 11, Section 25.2, Appendix 1, 2, 3
April 2021	S	■Added or modified parts RELEVANT MANUALS, TERMS, Section 1.2, 3.3, 4.2, 8.7, 8.9, 9.1, 13.5, 20.35, 20.36, 32.1, 32.2, 32.3, Appendix 1, 2, 3
October 2021	T	■Added or modified parts RELEVANT MANUALS, TERMS, GENERIC TERMS AND ABBREVIATIONS, Section 4.3, 14.1, 14.3, Appendix 2
April 2022	U	■Added model FX5S CPU module ■Added or modified parts RELEVANT MANUALS, GENERIC TERMS AND ABBREVIATIONS, HOW TO READ THIS MANUAL, Section 1.2, 1.3, 4.3, 6.1, 6.2, 7.5, 7.6, 8.3, 8.7, 8.8, 8.9, 8.12, 8.13, 8.18, 8.19, 8.25, 8.26, 9.1, 12.1, 13.5, 14.1, 16.2, 17.1, 17.2, 17.4, 18.1, 22.1, 22.2, 22.3, 22.4, 25.4, 26.1, Appendix 3

Revision date	Revision	Description
October 2022	V	■Added or modified parts Section 7.5, 8.9
April 2023	W	■Added or modified parts Section 3.6, Chapter 11, Appendix 3
October 2023	X	■Added or modified parts GENERIC TERMS AND ABBREVIATIONS, Section 3.2, 7.5, 8.7, 8.18, 12.1, Chapter 13, 14, 15, 17, 18

Japanese manual number: JY997D54701-X

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses.
Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2014 MITSUBISHI ELECTRIC CORPORATION

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company. However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
   1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
   2. Failure caused by unapproved modifications, etc., to the product by the user.
   3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
   4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
   5. Relay failure or output contact failure caused by usage beyond the specified life of contact (cycles).
   6. Failure caused by external irresistible forces such as fires or abnormal voltages, and failure caused by force majeure such as earthquakes, lightning, wind and water damage.
   7. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
   8. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued.
Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

(1) Damages caused by any cause found not to be the responsibility of Mitsubishi.

(2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.

(3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.

(4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

6. Product application

(1) In using the Mitsubishi MELSEC programmable controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.

(2) The Mitsubishi programmable controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for railway companies or public service purposes shall be excluded from the programmable controller applications.
In addition, applications in which human life or property that could be greatly affected, such as in aircraft, medical applications, incineration and fuel devices, manned transportation, equipment for recreation and amusement, and safety devices, shall also be excluded from the programmable controller range of applications.
However, in certain cases, some applications may be possible, providing the user consults their local Mitsubishi representative outlining the special requirements of the project, and providing that all parties concerned agree to the special circumstances, solely at the user's discretion.

(3) Mitsubishi shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

TRADEMARKS

Microsoft, Microsoft Access, ActiveX, Excel, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, Windows NT, Windows Server, Windows Vista, and Windows XP are trademarks of the Microsoft group of companies.
Anywire and AnyWireASLINK are either registered trademarks or trademarks of Anywire Corporation.
Unicode is either a registered trademark or a trademark of Unicode, Inc. in the United States and 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.

Manual number: JY997D55801X
Model: FX5-P-MF-E
Model code: 09R539

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.