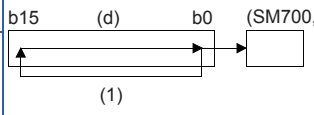
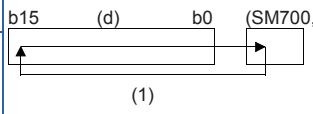
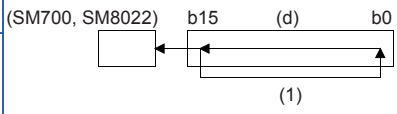
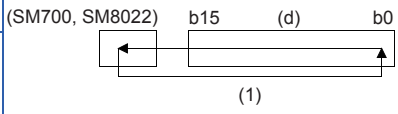
3.3 Application Instruction

Rotation instruction

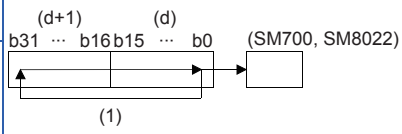
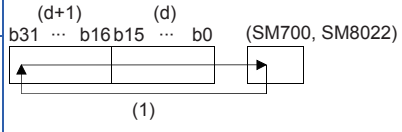
■Rotating 16-bit data to the right

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	

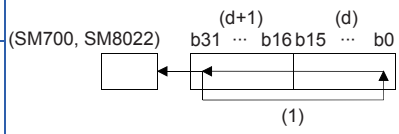
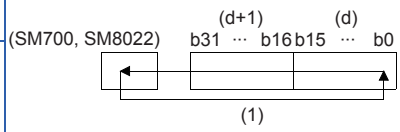
■Rotating 16-bit data to the left

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

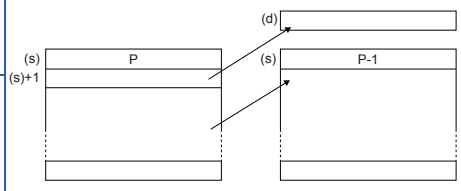
Instruction symbol	Description	Reference
RET	Returns from the subroutine program.	Page 465
SRET		

■Calling a subroutine program

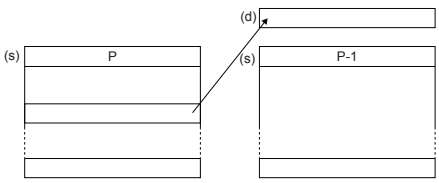
Instruction symbol	Description	Reference
XCALL	Executes a subroutine program specified by (P) when the input condition is met. Carry out non-execution processing for the subroutine program (P), when input conditions are not met.	Page 466

Data table operation instruction

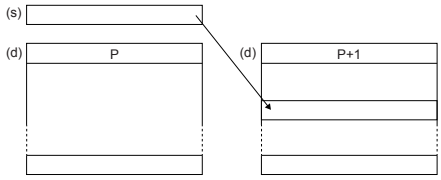
■Reading the oldest data from the data table

Instruction symbol	Description	Reference
SFRD	 P: Pointer	Page 468
SFRDP		

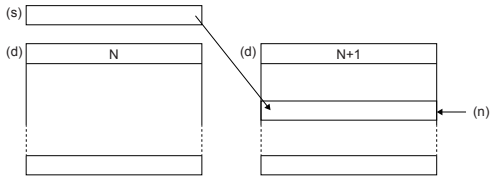
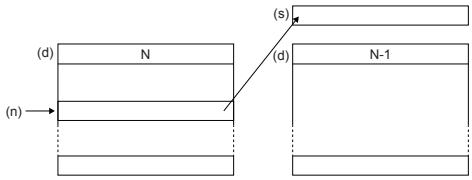
■Reading the newest data from the data table

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		

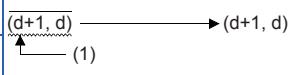
■Converting binary floating point to decimal floating point

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)

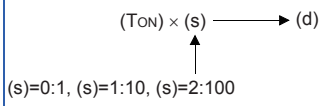
Instruction symbol	Description	Reference
SCL SCLP SCL_U SCLP_U	Executes scaling using the scaling conversion data (16-bit data units) specified by (s2) for the input value specified by (s1), and then stores the result in the device specified by (d). The scaling conversion is executed based on the scaling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

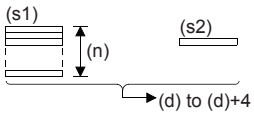
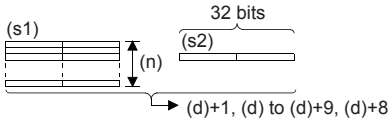
Check code

■Check code

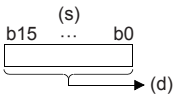
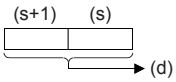
Instruction symbol	Description	Reference
CCD	This instruction calculates the sum data and horizontal parity value of data stored in (s) to (s)+(n)-1. The sum data is stored in (d), and the horizontal parity value is stored in (d)+1.	Page 845
CCDP		

Data operation instruction

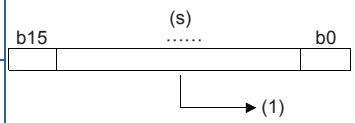
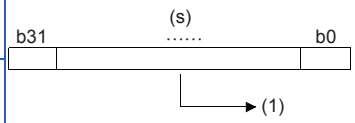
■Searching 16-bit/32-bit data

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

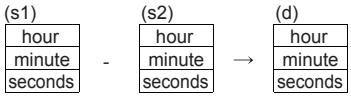
■Writing clock data

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

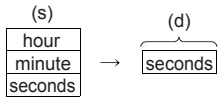
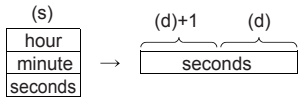
■Adding clock data

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

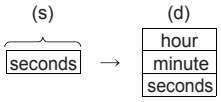
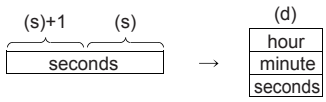
■Subtracting clock data

Instruction symbol	Description	Reference
TSUB TSUBP		Page 900

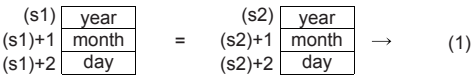
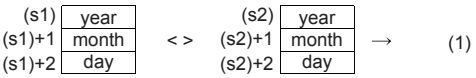
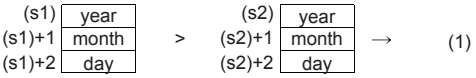
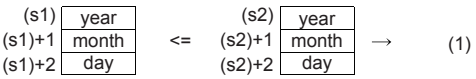
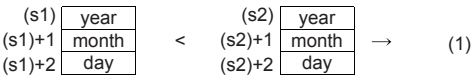
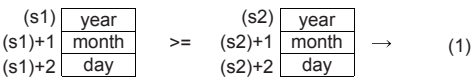
■Converting time data from hour/minute/second to seconds in 16 bits/32 bits

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

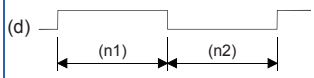
Instruction symbol	Description	Reference
TCMP TCMPP	(s1) hour / (s2) minute / (s3) seconds > (s4) hour / (s4)+1 minute / (s4)+2 seconds ⇨ (d) = ON (s1) hour / (s2) minute / (s3) seconds = (s4) hour / (s4)+1 minute / (s4)+2 seconds ⇨ (d)+1 = ON (s1) hour / (s2) minute / (s3) seconds < (s4) hour / (s4)+1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