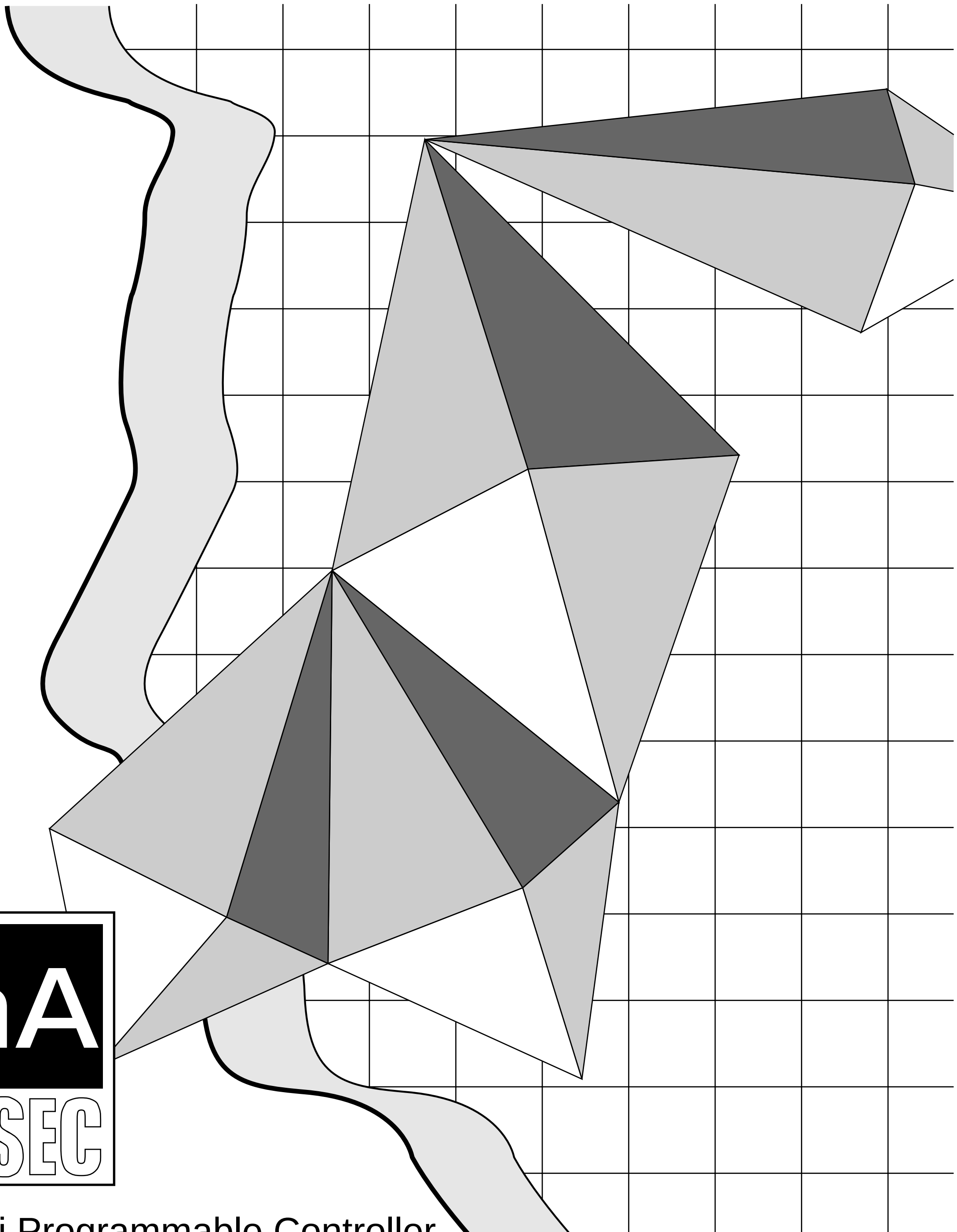
MITSUBISHI

QnA SERIES

QnACPU

PROGRAMMING MANUAL (Common Instructions)



QnA

MELSEC

Mitsubishi Programmable Controller

SAFETY CAUTIONS

(You must read these cautions before using the product)

In connection with the use of this product, in addition to carefully reading both this manual and the related manuals indicated in this manual, it is also essential to pay due attention to safety and handle the product correctly.
The safety cautions given here apply to this product in isolation. For information on the safety of the PC system as a whole, refer to the CPU module User's Manual.
Store this manual carefully in a place where it is accessible for reference whenever necessary, and forward a copy of the manual to the end user.

CONTENTS

REVISIONS

*The manual number is given on the bottom left of the back cover.

Print Date	*Manual Number	Revision
Jun., 1996	IB (NA) 66615-A	First edition
Sep., 1998	IB (NA) 66615-B	Addition of Models Q4ARCPU, Q2ASHCPU, Q2ASHCPU-S1 Correction SAFETY CAUTIONS, Contents, Section 1.1, 2.2, 2.5.1, 2.5.6, 2.5.14, 2.5.17, 2.5.18, 3.2.2, 3.2.3, 3.3, 3.4, 3.6, 3.8, 5.2.4, 5.2.5, 5.3.2, 5.3.10, 6.2.11, 6.4.5, 6.4.6, 7.1.2, 7.1.4, 7.1.6, 7.1.8, 7.5.4, 7.5.10, 7.5.11, 7.6.2, 7.6.3, 7.6.5, 7.8.2, 7.11.4, 7.11.12, 7.16.2, Chapter 8, Section 8.2.1, 8.2.2, 8.2.3, 8.2.4, 8.2.5, 8.2.7, 8.3.1, 8.3.2, 8.3.3, 8.3.4, 8.3.5, 8.3.6, 9.2, 9.3, APPENDICES, APP1.2, APP2.1, APP2.4, APP3, APP4
Feb., 1999	IB (NA) 66615-C	Correction Contents, Section 3.2.2, 3.2.3, Chapter 4, Section 6.2.9, Section 7.9.5, 7.17, Section 8.2.1 to 8.2.9, 8.3.1 to 8.3.6, Section 9.2, APPENDICES, APP 1.2, APP3, APP4

About Manuals

The manuals related to the QnACPU are listed in the table below.
Please order those you require.

Related Manuals

Manual Name	Manual Number
QnACPU Guidebook Aimed at people using QnACPU for the first time. Describes procedures for everything from creating programs and writing created programs to the CPU, to debugging. Also describes how to use the QnACPU most effectively.	IB-66606
Q2A(S1)/Q3A/Q4ACPU User's Manual Describes the performance, functions, and handling of the Q2ACPU(S1), Q3ACPU, and Q4ACPU, and the specifications and handling of memory cards and base units. (Purchased separately)	IB-66608
Type Q2AS(H)CPU(S1) User's Manual This manual explains performance, functions, and handling of the Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, and Q2ASHCPU-S1, power supply, memory card, specifications, and handling of the base unit. (Sold separately)	SH-3599
A4ARCPU User's Manual (Detailed Version) Describes the Q4ARCPU features, functions, and usage. Also describes the specification and usage of the bus switching module, system management module, power module, memory card, and base module. (Sold separately)	IB-66685
QnACPU Programming Manual (Fundamentals) Describes how to create programs, the names of devices, parameters, and types of program. (Purchased separately)	IB-66614
QnACPU Programming Manual (Special Function) Describes the dedicated instructions for special function modules available when using the Q2ACPU(S1), Q3ACPU, and Q4ACPU. (Purchased separately)	IB-66616
QnACPU Programming Manual (AD57 Instructions) Describes the dedicated instructions for controlling an AD57(S1) type CRT controller module available when using the Q2ACPU(S1), Q3ACPU, or Q4ACPU. (Purchased separately)	IB-66617
QnACPU Programming Manual (PID Control Instructions) Describes the dedicated instructions for PID control available when using the Q2ACPU(S1), Q3ACPU, or Q4ACPU. (Purchased separately)	IB-66618
QnACPU Programming Manual (SFC) Describes the performance specifications, functions, programming, debugging, and error codes, for SFC program. (Purchased separately)	IB-66619
MELSECNET/10 Network System (for QnA) Reference Manual Describes the general concept, specifications, and part names and settings, for MELSECNET/10. (Purchased separately)	IB-66620
MELSECNET, MELSECNET/B Data Link System Reference Manual Describes the general concept, specifications, and part names and settings, for MELSECNET (II) and MELSECNET/B. (Purchased separately)	IB-66350
GPP Function software for Windows SW2D5C-GPPW-E SW2D5F-GPPW-E Operating Manual Describes the online functions of SW2D5 -GPPW-E including the programming procedure, printing out procedure, monitoring pro□edure, and debugging procedure. (Package with the product)	IB-66877-A
Type SW2IVD-GPPQ GPP Function Operating Manual (OFFLINE) Describes to create programs and print out data when using SW2IVD-GPPQ, and the offline functions of SW2IVD-GPPQ such as file maintenance. (Supplied with the product)	IB-66774
Type SW2IVD-GPPQ GPP Function Operating Manual (ONLINE) Describes the online functions of SW2IVD-GPPQ, including the methods for monitoring and debugging. (Supplied with the product)	IB-66775
Type SW2IVD-GPPQ GPP Function Operating Manual (SFC) Describes SFC functions such as SFC program editing and monitoring. (Supplied with the product)	IB-66776

1. GENERAL DESCRIPTION

This manual contains information on the common instructions for QnACPU, Q2AS(H)CPU(S1) that are required when programming with a QnACPU, Q2AS(H)CPU(S1).
Common instructions are all instructions except those used for special function modules such as AJ71QC24, AJ71PT32-S3, etc.; the instructions for AD57; the instructions for PID control, and those for MELSAP3.

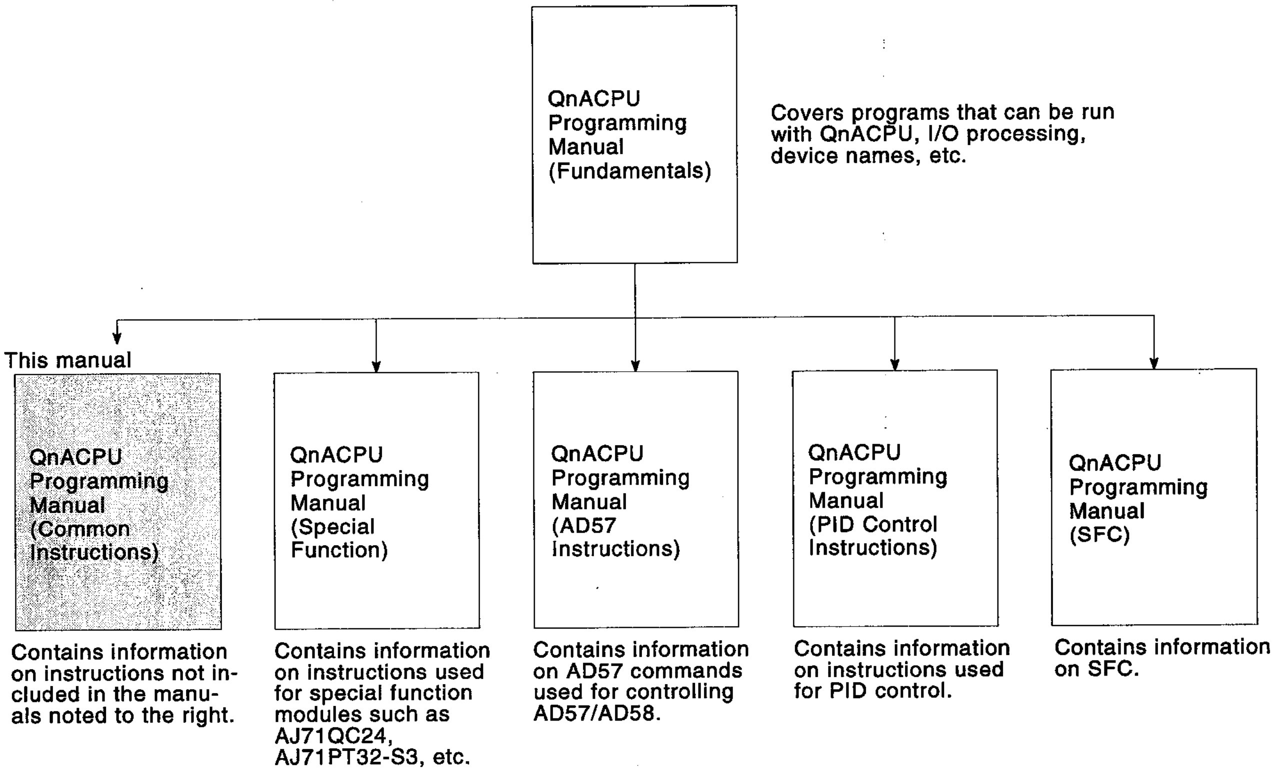
A list of the instructions dealt with in Common Instructions is included in Sections 2.3 through 2.6.
See Sections 2.3 through 2.6 for instructions which can be used.

1.1 Related Programming Manuals

In addition to this manual, there are five other programming manuals for QnACPU. These are as follows:

- QnACPU Programming Manual (Fundamentals)
- QnACPU Programming Manual (Special Function)
- QnACPU Programming Manual (PID Control Instructions)
- QnACPU Programming Manual (AD57 Instructions)
- QnACPU Programming Manual (SFC)

Before reading this manual, it is advisable to review the QnACPU Programming Manual (Fundamentals) to confirm what programs can be used with QnACPU, input/output processing, and basic information on devices.



Covers programs that can be run with QnACPU, I/O processing, device names, etc.

Contains information on instructions not included in the manuals noted to the right.

Contains information on instructions used for special function modules such as AJ71QC24, AJ71PT32-S3, etc.

Contains information on AD57 commands used for controlling AD57/AD58.

Contains information on instructions used for PID control.

Contains information on SFC.

REMARK

In this manual, QnACPU and Q2AS(H)CPU(S1) are all referred to as QnACPU.

2. INSTRUCTION TABLES

2.1 Types of Instructions

The major types of QnACPU instructions are sequence instructions, basic instructions, application instructions, and instructions used for data links. These types are listed in Table 2.1 below.

Table 2.1 Types of Instructions

Types of Instructions		Meaning	See for Description
Sequence instructions	Contact instruction	Operation start, series connection, parallel connection	5-2 to 5-7
	Connection instructions	Ladder block connection, creation of pulses from operation results, store/read operation results	5-8 to 5-17
	Output instruction	Bit device output, pulse output, output reversal	5-18 to 5-38
	Shift instruction	Bit device shift	5-39 to 5-40
	Master control instruction	Master control	5-41 to 5-44
	Termination instruction	Program termination	5-45 to 5-48
	Other instructions	Program stop, instructions such as no operation which do not fit in the above categories	5-49 to 5-54
Basic instructions	Comparison operation instruction	Comparisons such as =, >, <	6-2 to 6-19
	Arithmetic operation instruction	Addition, subtraction, multiplication or division of BIN or BCD	6-20 to 6-63
	BCD ↔ BIN conversion instruction	Conversion from BCD to BIN and from BIN to BCD	6-64 to 6-87
	Data transfer instruction	Transmits designated data	6-88 to 6-105
	Program branch instruction	Program jumps	6-106 to 6-109
	Program run control instruction	Enable or prohibit interrupt programs	6-110 to 6-114
	I/O refresh	Run partial refresh	6-115 to 6-116
	Other convenient instructions	Instructions for: Counter increment/decrement, teaching timer, special function timer, rotary table shortest direction control, etc.	6-117 to 6-140
Application instructions	Logical operation instructions	Logical operations such as logical sum, logical product, etc.	7-2 to 7-33
	Rotation instruction	Rotation of designated data	7-34 to 7-41
	Shift instruction	Shift of designated data	7-42 to 7-47
	Bit processing instructions	Bit set and reset, bit test, batch reset of bit devices	7-48 to 7-53
	Data processing instructions	16-bit data searches, data processing such as decoding and encoding	7-54 to 7-90
	Structure creation instructions	Repeated operation, subroutine program calls, index qualification in ladder units	7-91 to 7-118
	Table operation instruction	Read/Write of FIFO table	7-119 to 7-127
	Buffer memory access instruction	Data read/write for special function modules	7-128 to 7-133
	Display instructions	Print ASCII code, LED character display, etc.	7-134 to 7-144
	Debugging and failure diagnosis instructions	Check, status check, sampling trace, program trace	7-145 to 7-160
	Character string processing instructions	Conversion between BIN/BCD and ASCII; conversion between BIN and character string; conversion between floating decimal point data and character strings, character string processing, etc.	7-161 to 7-221
	Special function instructions	Trigonometric functions, conversion between angles and radians, exponential operations, automatic logarithms, square roots	7-222 to 7-262
	Data control instructions	Upper and lower limit controls, dead band controls, zone controls	7-263 to 7-271

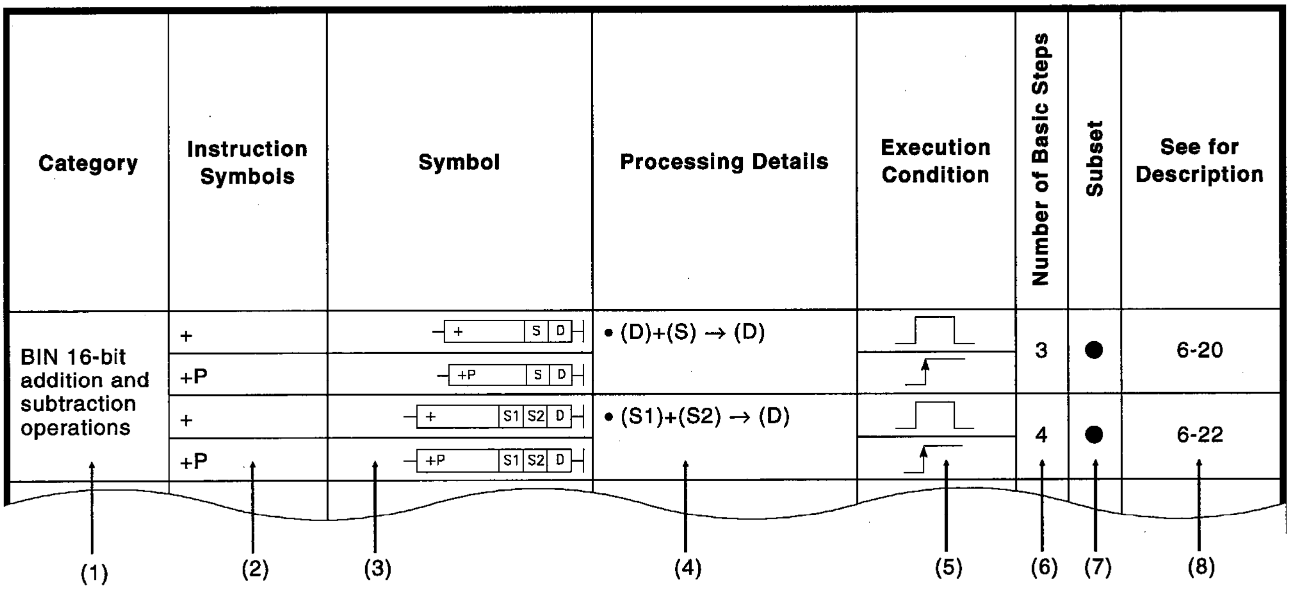
Table 2.1 Types of Instructions (Continued)

Types of Instructions		Meaning	See for Description
Application instructions	Switching instructions	File register block No. switches, designation of file registers and comment files	7-272 to 7-277
	Clock instructions	Read/write of year, month, day, hour, minute, second, and day of the week; conversion between time statement (hour, minute, second) and seconds	7-278 to 7-288
	Peripheral device instructions	Input/Output to peripheral devices	7-289 to 7-292
	Program instructions	Instructions to switch program execution conditions	7-293 to 7-300
	Other instructions	Instructions that do not fit in the above categories, such as watchdog timer reset instructions and timing clock instructions	7-301 to 7-316
Data link instructions	Link refresh instructions	Designated network refresh	8-6 to 8-8
	Instructions dedicated to QnA links	Read/write of data from other stations; data transmission signals to other stations; processing requests to other stations	8-9 to 8-57
	Instructions for A-series-compatible link	Read/write for designated station word device, read/write data from remote I/O station special function module	8-58 to 8-81

2.2 How to Read Instruction Tables

The instruction tables found from Section 2.3 to 2.6 have been made according to the following format:

Table 2.2 How to Read Instruction Tables



Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
BIN 16-bit addition and subtraction operations	+	+ S D	• (D)+(S) → (D)	⎍	3	●	6-20
	+P	+P S D		↑			
	+	+ S1 S2 D	• (S1)+(S2) → (D)	⎍	4	●	6-22
	+P	+P S1 S2 D		↑			

Description

(1) .... Classifies instructions according to their application

(2) .... Indicates the instruction symbol used to enter the instruction in a program
Instruction code is built around the 16-bit instruction, with the following notations used to mark 32-bit instructions, instructions executed only at the leading edge of OFF to ON, real number instructions, and character string instructions:

- 32-bit instruction.......... The letter "D" is added to the first line of the instruction

Example + ——→ D+
16-bit instruction → 32-bit instruction

- Instructions executed only at the leading edge of OFF to ON .... The letter "P" is appended to the end of the instruction

Example + ——→ +P
Instructions executed when ON → Instructions executed only at the leading edge of OFF to ON

- Real number instructions .. The letter "E" is added to the first line of the instruction

Example + ——→ E+
Real number instructions

- Character string instructions
  .... A dollar sign $ is added to the first line of the instruction

Example + ——→ $+
↑
Character string instructions

(3) .... Shows symbol drawing on the ladder

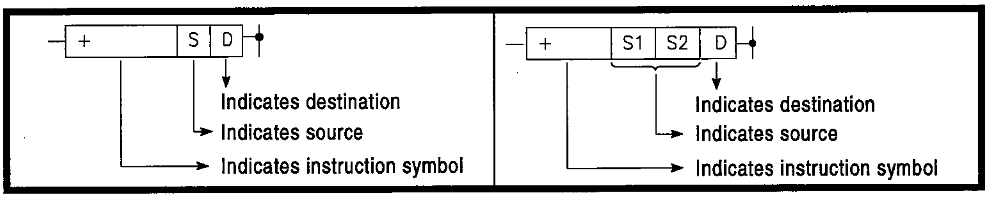


Fig. 2.1 Shows Symbol Drawing on the Ladder

Destination .... Indicates where data will be sent following operation

Source............ Stores data prior to operation

(4) .... Indicates the type of processing that is performed by individual instructions

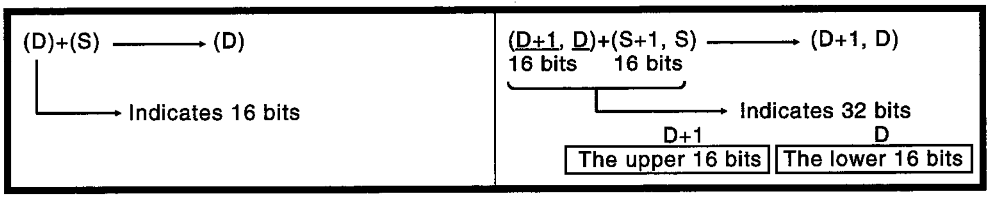
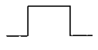
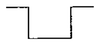
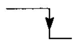


Fig. 2.2 Type of Processing Performed by Individual Instructions

(5) .... The details of conditions for the execution of individual instructions are as follows:

Symbol	Execution Condition
No symbol recorded	Instruction executed under normal circumstances, with no regard to the ON/OFF status of conditions prior to the instruction. If the preconditions is OFF, the instruction will conduct OFF processing.
	Executed during ON; instruction is executed only while the precondition is ON.If the preconditions is OFF, the instruction is not executed, and no processing is conducted.
	Executed once at ON; instruction executed only at leading edge when precondition goes from OFF to ON. Following execution, instruction will not be executed and no processing conducted even if condition remains ON.
	Executed during OFF; instruction is executed only while the precondition is OFF. If the precondition is ON, the instruction is not executed, and no processing is conducted.
	Executed once at OFF; instruction executed only at trailing edge when precondition goes from ON to OFF. Following execution, instruction will not be executed and no processing conducted even if condition remains OFF.

(6) .... Indicates the basic number of steps for individual instructions. See Section 3.8 for a description of the number of steps.

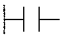
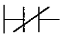
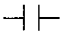
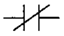
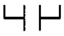
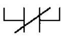
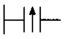
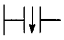
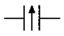
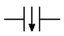
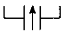
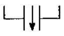
(7) .... The ● mark indicates instructions for which subset processing is possible.
See Section 3.5 for details on subset processing.

(8) .... Indicates the page numbers where the individual instructions are discussed.

2.3 Sequence Instructions

2.3.1 Contact instruction

Table 2.3 Contact Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Contact	LD		• Starts logic operation (Starts A contact logic operation)		*1		5-2
	LDI		• Starts logical NOT operation (Starts B contact logic operation)				
	AND		• Logical product (A contact series connection)				
	ANI		• Logical product NOT (B contact series connection)				
	OR		• Logical sum (A contact parallel connection)				
	ORI		• Logical sum NOT (B contact parallel connection)				
	LDP		• Starts leading edge pulse operation		*2		5-5
	LDF		• Starts trailing edge pulse operation				
	ANDP		• Leading edge pulse series connection				
	ANDF		• Trailing edge pulse series connection				
	ORP		• Leading edge pulse parallel connection				
	ORF		• Trailing edge pulse parallel connection				

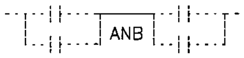
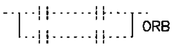
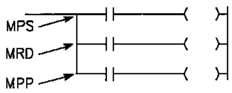
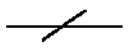
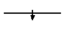
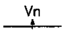
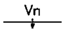
REMARKS

1) *1: The number of steps differs depending on the device in use.
- When internal device or file register (R0 to R32767) is in use :1
- When direct access input (DX) is in use :2
- When some other device is in use :3

2) *2: The number of steps differs depending on the device in use.
- When internal device or file register (R0 to R32767) is in use :2
- When direct access input (DX) is in use :3
- When some other device is in use :4

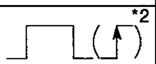
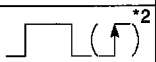
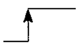
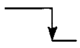
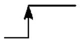
2.3.2 Connection instructions

Table 2.4 Connection Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Connection	ANB		• AND between logical blocks (Series connection between logical blocks)		1		5-8
	ORB		• OR between logical blocks (Series connection between logical blocks)				
	MPS		• Memory storage of operation results		1		5-10
	MRD		• Read of operation results stored with MPS instruction				
	MPP		• Read and reset of operation results stored with MPS instruction				
	INV		• Inversion of operation result		1		5-13
	MEP		• Conversion of operation result to leading edge pulse		1		5-14
	MEF		• Conversion of operation result to trailing edge pulse				
	EGP		• Conversion of operation result to leading edge pulse (Stored at Vn)		1		5-16
	EGF		• Conversion of operation result to trailing edge pulse (Stored at Vn)				

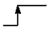
2.3.3 Output instructions

Table 2.5 Output Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Output	OUT	—( )—	• Device output		*1		5-18
	SET	—[SET D]—	• Set device	 *2	*1		5-27 5-31
	RST	—[RST D]—	• Reset device	 *2	*1		5-29 5-31
	PLS	—[PLS D]—	• Generates 1 cycle program pulse at leading edge of input signal		2		5-33
	PLF	—[PLF D]—	• Generates 1 cycle program pulse at trailing edge of input signal				
	FF	—[FF D]—	• Reversal of device output		2		5-35
	DELTA	—[DELTA D]—	• Pulse conversion of direct output		2		5-37
	DELTAP	—[DELTAP D]—					

REMARKS

1) *1: The number of steps differs depending on the device in use.
See description pages of individual instructions for number of steps.

2) *2: The  execution condition applies only when an annunciator (F) is in use.

2.3.4 Shift instructions

Table 2.6 Shift Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Shift	SFT	—[SFT D]—	• 1-bit shift of device		2		5-39
	SFTP	—[SFTP D]—					

2.3.5 Master control instructions

Table 2.7 Master Control Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Master control	MC	MC n D	• Starts master control		2		5-41
	MCR	MCR n	• Resets master control		1		

2.3.6 Termination instruction

Table 2.8 Termination Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Program end	FEND	FEND	• Termination of main program		1		5-45
	END	END	• Termination of sequence program				5-47

2.3.7 Other instructions

Table 2.9 Other Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Stop	STOP	STOP	• Terminates sequence operation after input condition has been met • Sequence program is executed by placing the RUN/STOP key switch back in the RUN position	⎍	1		5-49
Ignored	NOP	—	• Ignored (For program deletion or space)		1		5-51
	NOPLF	NOPLF	• Ignored (To change pages during printouts)				
	PAGE	PAGE n	• Ignored (Subsequent programs will be controlled from step 0 of page n)				

2.4 Basic Instructions

2.4.1 Comparison operation instruction

Table 2.10 Comparison Operation Instruction

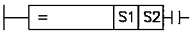
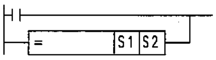
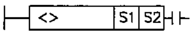
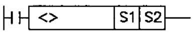
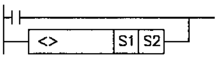
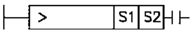
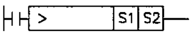
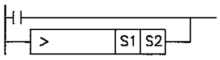
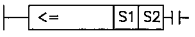
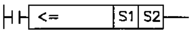
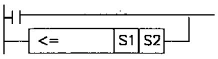
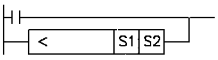
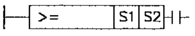
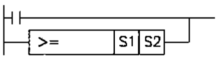
Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
16-bit data comparisons	LD=		• Conductive state when (S1) = (S2) • Non-conductive state when (S1) ≠ (S2)		3	●	6-2
	AND=						
	OR=						
	LD<>		• Conductive state when (S1) ≠ (S2) • Non-conductive state when (S1) = (S2)		3	●	6-2
	AND<>						
	OR<>						
	LD>		• Conductive state when (S1) > (S2) • Non-conductive state when (S1) ≤ (S2)		3	●	6-2
	AND>						
	OR>						
	LD<=		• Conductive state when (S1) ≤ (S2) • Non-conductive state when (S1) > (S2)		3	●	6-2
	AND<=						
	OR<=						
	LD<		• Conductive state when (S1) < (S2) • Non-conductive state when (S1) ≥ (S2)		3	●	6-2
	AND<						
	OR<						
	LD>=		• Conductive state when (S1) ≥ (S2) • Non-conductive state when (S1) < (S2)		3	●	6-2
	AND>=						
	OR>=						

Table 2.10 Comparison Operation Instructions (Continued)

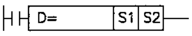
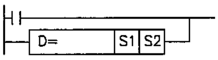
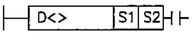
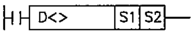
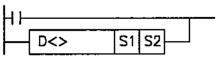
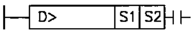
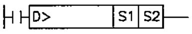
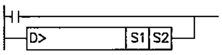
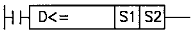
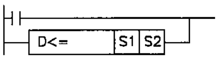
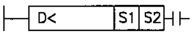
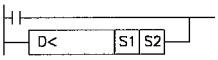
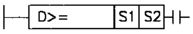
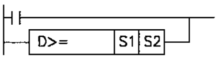
Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
32-bit data comparisons	LDD=		• Conductive state when (S1+1, S1) = (S2+1, S2) • Non-Conductive state when (S1+1, S1) ≠ (S2+1, S2)		3	●	6-5
	ANDD=						
	ORD=						
	LDD<>		• Conductive state when (S1+1, S1) ≠ (S2+1, S2) • Non-Conductive state when (S1+1, S1) = (S2+1,S2)		3	●	6-5
	ANDD<>						
	ORD<>						
	LDD>		• Conductive state when (S1+1, S1) > (S2+1, S2) • Non-Conductive state when (S1+1, S1) ≤ (S2+1, S2)		3	●	6-5
	ANDD>						
	ORD>						
	LDD<=		• Conductive state when (S1+1, S1) ≤ (S2+1, S2) • Non-Conductive state when (S1+1, S1) > (S2+1, S2)		3	●	6-5
	ANDD<=						
	ORD<=						
	LDD<		• Conductive state when (S1+1, S1) < (S2+1, S2) • Non-Conductive state when (S1+1, S1) ≥ (S2+1, S2)		3	●	6-5
	ANDD<						
	ORD<						
	LDD>=		• Conductive state when (S1+1, S1) ≥ (S2+1, S2) • Non-Conductive state when (S1+1, S1) < (S2+1, S2)		3	●	6-5
	ANDD>=						
	ORD>=						

Table 2.10 Comparison Operation Instructions (Continued)

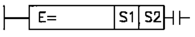
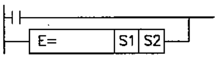
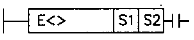
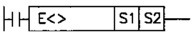
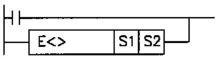
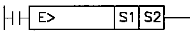
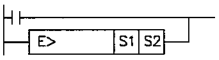
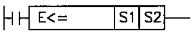
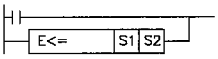
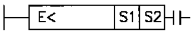
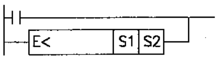
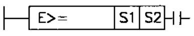
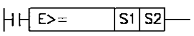
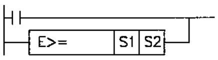
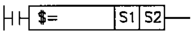
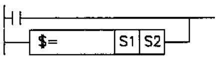
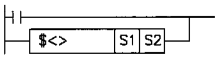
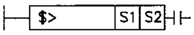
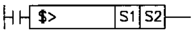
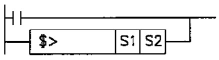
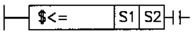
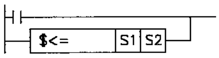
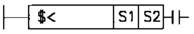
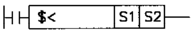
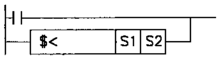
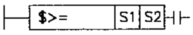
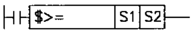
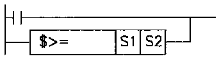
Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Real number data comparisons	LDE=		• Conductive state when (S1+1, S1) = (S2+1, S2) • Non-Conductive state when (S1+1, S1) ≠ (S2+1, S2)		3		6-8
	ANDE=						
	ORE=						
	LDE<>		• Conductive state when (S1+1, S1) ≠ (S2+1, S2) • Non-Conductive state when (S1+1, S1) = (S2+1,S2)		3		6-8
	ANDE<>						
	ORE<>						
	LDE>		• Conductive state when (S1+1, S1) > (S2+1, S2) • Non-Conductive state when (S1+1, S1) ≤ (S2+1, S2)		3		6-8
	ANDE>						
	ORE>						
	LDE<=		• Conductive state when (S1+1, S1) ≤ (S2+1, S2) • Non-Conductive state when (S1+1, S1) > (S2+1, S2)		3		6-8
	ANDE<=						
	ORE<=						
	LDE<		• Conductive state when (S1+1, S1) < (S2+1, S2) • Non-Conductive state when (S1+1, S1) ≥ (S2+1, S2)		3		6-8
	ANDE<						
	ORE<						
	LDE>=		• Conductive state when (S1+1, S1) ≥ (S2+1, S2) • Non-Conductive state when (S1+1, S1) < (S2+1, S2)		3		6-8
	ANDE>=						
	ORE>=						

Table 2.10 Comparison Operation Instructions (Continued)

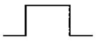
Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Character string data comparisons	LD$=		• Compares character string S1 and character string S2 one character at a time. * • Conductive state when (character string S1) = (character string S2) • Non-Conductive state when (character string S1) ≠ (character string S2)		3		6-11
	AND$=						
	OR$=						
	LD$<>		• Compares character string S1 and character string S2 one character at a time. * • Conductive state when (character string S1) ≠ (character string S2) • Non-Conductive state when (character string S1) = (character string S2)		3		6-11
	AND$<>						
	OR$<>						
	LD$>		• Compares character string S1 and character string S2 one character at a time. * • Conductive state when (character string S1) > (character string S2) • Non-Conductive state when (character string S1) ≤ (character string S2)		3		6-11
	AND$>						
	OR$>						
	LD$<=		• Compares character string S1 and character string S2 one character at a time. * • Conductive state when (character string S1) ≤ (character string S2) • Non-Conductive state when (character string S1) > (character string S2)		3		6-11
	AND$<=						
	OR$<=						
	LD$<		• Compares character string S1 and character string S2 one character at a time. * • Conductive state when (character string S1) < (character string S2) • Non-Conductive state when (character string S1) ≥ (character string S2)		3		6-11
	AND$<						
	OR$<						
	LD$>=		• Compares character string S1 and character string S2 one character at a time. * • Conductive state when (character string S1) ≥ (character string S2) • Non-Conductive state when (character string S1) < (character string S2)		3		6-11
	AND$>=						
	OR$>=						

REMARK

*: The conditions under which character string comparisons can be made are as shown below:

- Match : All characters in the strings must match
- Larger string : If character strings are different, determines the string with the largest number of character codes
  If the lengths of the character strings are different, determines the longest character string
- Smaller string : If the character strings are different, determines the string with the smallest number of character codes
  If the lengths of the character strings are different, determines the shortest character string

Table 2.10 Comparison Operation Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Block data comparisons	BKCMP=	BKCMP= S1 S2 D n	• Compares n points of data from S1 to n points of data from S2 in 1-word units, and stores the results of the comparison at n points from the bit device designated by (D).		5		6-16
	BKCMP<>	BKCMP<> S1 S2 D n					
	BKCMP>	BKCMP> S1 S2 D n					
	BKCMP<=	BKCMP<= S1 S2 D n					
	BKCMP<	BKCMP< S1 S2 D n					
	BKCMP>=	BKCMP>= S1 S2 D n					
	BKCMP=P	BKCMP=P S1 S2 D n					
	BKCMP<>P	BKCMP<>P S1 S2 D n					
	BKCMP>P	BKCMP>P S1 S2 D n					
	BKCMP<=P	BKCMP<=P S1 S2 D n					
	BKCMP<P	BKCMP<P S1 S2 D n					
	BKCMP>=P	BKCMP>=P S1 S2 D n					

2.4.2 Arithmetic operation instructions

Table 2.11 Arithmetic Operation Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
BIN 16-bit addition and subtraction operations	+	+ S D	• (D)+(S) → (D)	⎍	3	●	6-20
	+P	+P S D		↑			
	+	+ S1 S2 D	• (S1)+(S2) → (D)	⎍	4	●	6-22
	+P	+P S1 S2 D		↑			
	-	− S D	• (D)-(S) → (D)	⎍	3	●	6-20
	-P	−P S D		↑			
	-	− S1 S2 D	• (S1)-(S2) → (D)	⎍	4	●	6-22
	-P	−P S1 S2 D		↑			
BIN 32-bit addition and subtraction operations	D+	D+ S D	• (D+1, D)+(S+1, S) → (D+1, D)	⎍	3	●	6-24
	D+P	D+P S D		↑			
	D+	D+ S1 S2 D	• (S1+1, S1)+(S2+1, S2) → (D+1, D)	⎍	4	●	6-26
	D+P	D+P S1 S2 D		↑			
	D-	D− S D	• (D+1, D)-(S+1, S) → (D+1, D)	⎍	3	●	6-24
	D-P	D−P S D		↑			
	D-	D− S1 S2 D	• (S1+1, S1)-(S2+1, S2) → (D+1, D)	⎍	4	●	6-26
	D-P	D−P S1 S2 D		↑			
BIN 16-bit multiplication and division operations	*	* S1 S2 D	• (S1)*(S2) → (D+1, D)	⎍	4	●	6-28
	*P	*P S1 S2 D		↑			
	/	/ S1 S2 D	• (S1)/(S2) → Quotient (D), Remainder (D+1)	⎍	4	●	6-28
	/P	/P S1 S2 D		↑			
BIN 32-bit multiplication and division operations	D*	D* S1 S2 D	• (S1+1, S1)*(S2+1, S2) → (D+3, D+2, D+1, D)	⎍	4	●	6-30
	D*P	D*P S1 S2 D		↑			
	D/	D/ S1 S2 D	• (S1+1, S1)/(S2+1, S2) → Quotient (D+1, D), Remainder (D+3, D+2)	⎍	4	●	6-30
	D/P	D/P S1 S2 D		↑			

Table 2.11 Arithmetic Operation Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
BCD 4-digit addition and subtraction operations	B+	–[B+ | S | D]–	• (D)+(S) → (D)	⎍	3	●	6-33
	B+P	–[B+P | S | D]–		↑			
	B+	–[B+ | S1 | S2 | D]–	• (S1)+(S2) → (D)	⎍	4		6-35
	B+P	–[B+P | S1 | S2 | D]–		↑			
	B-	–[B– | S | D]–	• (D)-(S) → (D)	⎍	3	●	6-33
	B-P	–[B–P | S | D]–		↑			
	B-	–[B– | S1 | S2 | D]–	• (S1)-(S2) → (D)	⎍	4		6-35
	B-P	–[B–P | S1 | S2 | D]–		↑			
BCD 8-digit addition and subtraction operations	DB+	–[DB+ | S | D]–	• (D+1, D)+(S+1, S) → (D+1, D)	⎍	3		6-37
	DB+P	–[DB+P | S | D]–		↑			
	DB+	–[DB+ | S1 | S2 | D]–	• (S1+1, S1)+(S2+1, S2) → (D+1, D)	⎍	4		6-40
	DB+P	–[DB+P | S1 | S2 | D]–		↑			
	DB-	–[DB– | S | D]–	• (D+1, D)-(S+1, S) → (D+1, D)	⎍	3		6-37
	DB-P	–[DB–P | S | D]–		↑			
	DB-	–[DB– | S1 | S2 | D]–	• (S1+1, S1)-(S2+1, S2) → (D+1, D)	⎍	4		6-40
	DB-P	–[DB–P | S1 | S2 | D]–		↑			
BCD 4-digit multiplication and division operations	B*	–[B* | S1 | S2 | D]–	• (S1)*(S2) → (D+1, D)	⎍	4	●	6-42
	B*P	–[B*P | S1 | S2 | D]–		↑			
	B/	–[B/ | S1 | S2 | D]–	• (S1)/(S2) → Quotient (D), Remainder (D+1)	⎍	4	●	6-42
	B/P	–[B/P | S1 | S2 | D]–		↑			
BCD 8-digit multiplication and division operations	DB*	–[DB* | S1 | S2 | D]–	• (S1+1, S1)*(S2+1, S2) → (D+3, D+2, D+1, D)	⎍	4		6-44
	DB*P	–[DB*P | S1 | S2 | D]–		↑			
	DB/	–[DB/ | S1 | S2 | D]–	• (S1+1, S1)/(S2+1, S2) → Quotient (D+1, D), Remainder (D+3, D+2)	⎍	4	●	6-44
	DB/P	–[DB/P | S1 | S2 | D]–		↑			

Table 2.11 Arithmetic Operation Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Floating decimal point data addition and subtraction operations	E+	E+ | S | D	• (D+1, D)+(S+1, S) ⟶ (D+1, D)	⎍	3		6-47
	E+P	E+P | S | D		↑			
	E+	E+ | S1 | S2 | D	• (S1+1, S1)+(S2+1, S2) ⟶ (D+1, D)	⎍	4		6-49
	E+P	E+P | S1 | S2 | D		↑			
	E-	E− | S | D	• (D+1, D)-(S+1, S) ⟶ (D+1, D)	⎍	3		6-47
	E-P	E−P | S | D		↑			
	E-	E− | S1 | S2 | D	• (S1+1, S1)-(S2+1, S2) ⟶ (D+1, D)	⎍	4		6-49
	E-P	E−P | S1 | S2 | D		↑			
Floating decimal point data multiplication and division operations	E*	E* | S1 | S2 | D	• (S1+1, S1)*(S2+1, S2) ⟶ (D+1, D)	⎍	4		6-51
	E*P	E*P | S1 | S2 | D		↑			
	E/	E/ | S1 | S2 | D	• (S1+1, S1)/(S2+1, S2) ⟶ Quotient (D+1, D)	⎍	4		6-51
	E/P	E/P | S1 | S2 | D		↑			
Character string data combinations	$+	$+ | S | D	• Links character string designated with (S) to character string designated with (D), and stores the result from (D) onward.	⎍	3		6-56
	$+P	$+P | S | D		↑			
	$+	$+ | S1 | S2 | D	• Links character string designated with (S2) to character string designated with (S1), and stores the result from (D) onward.	⎍	4		6-58
	$+P	$+P | S1 | S2 | D		↑			
BIN block addition and subtraction operations	BK+	BK+ | S1 | S2 | D | n	• Adds data n points from (S1) to data n points from (S2) in batch.	⎍	5		6-53
	BK+P	BK+P | S1 | S2 | D | n		↑			
	BK-	BK− | S1 | S2 | D | n	• Subtracts data n points from (S1) from data n points from (S2) in batch.	⎍	5		6-53
	BK-P	BK−P | S1 | S2 | D | n		↑			
BIN data increment	INC	INC | D	• (D)+1 ⟶ (D)	⎍	2	●	6-60
	INCP	INCP | D		↑			
	DINC	DINC | D	• (D+1, D)+1 ⟶ (D+1, D)	⎍	2	●	6-62
	DINCP	DINCP | D		↑			

Table 2.11 Arithmetic Operation Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
BIN data decrement	DEC	–[DEC D]–	• (D-1) ──→ (D)	⎍	2	●	6-60
	DECP	–[DECP D]–		↑			
	DDEC	–[DDEC D]–	• (D+1, D)-1 ──→ (D+1, D)	⎍	2	●	6-62
	DDECP	–[DDECP D]–		↑			

2.4.3 Data conversion instructions

Table 2.12 Data Conversion Instructions

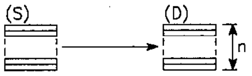
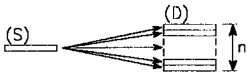
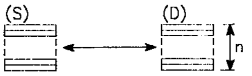
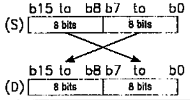
Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
BCD conversions	BCD	–[BCD S D]–	• (S) ──BCD conversion──→ (D) ↑── BIN (0 to 9999)	⎍	3	●	6-64
	BCDP	–[BCDP S D]–		↑			
	DBCD	–[DBCD S D]–	BCD conversion • (S+1, S) ──→ (D+1, D) ↑── BIN (0 to 99999999)	⎍	3	●	6-64
	DBCDP	–[DBCDP S D]–		↑			
BIN conversions	BIN	–[BIN S D]–	• (S) ──BIN conversion──→ (D) ↑── BCD (0 to 9999)	⎍	3	●	6-67
	BINP	–[BINP S D]–		↑			
	DBIN	–[DBIN S D]–	BIN conversion • (S+1, S) ──→ (D+1, D) ↑── BCD (0 to 99999999)	⎍	3	●	6-67
	DBINP	–[DBINP S D]–		↑			
Conversion from floating decimal point to BIN	INT	–[INT S D]–	Conversion to BIN • (S+1, S) ──→ (D) ↑── Real number (-32768 to 32767)	⎍	3		6-72
	INTP	–[INTP S D]–		↑			
	DINT	–[DINT S D]–	Conversion to BIN • (S+1, S) ──→ (D+1, D) ↑── Real number (-2147483648 to 2147483647)	⎍	3		6-72
	DINTP	–[DINTP S D]–		↑			
Conversion from BIN to floating decimal point	FLT	–[FLT S D]–	Conversion to floating decimal point • (S+1, S) ──→ (D) ↑──BIN (-32768 to 32767)	⎍	3		6-70
	FLTP	–[FLTP S D]–		↑			

Table 2.12 Data Conversion Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Conversion from BIN to floating decimal point	DFLT	DFLT S D	Conversion to floating decimal point • (S+1, S) → (D+1, D) └ Real number (-2147483648 to 2147483647)	⎍	3		6-70
	DFLTP	DFLTP S D		↑			
Conversion between BIN 16-bit and 32-bit	DBL	DBL S D	• (S) Conversion → (D+1, D) └ BIN (-32768 to 32767)	⎍	3		6-75
	DBLP	DBLP S D		↑			
	WORD	WORD S D	• (S+1, S) Conversion → (D) └ BIN (-32768 to 32767)	⎍	3		6-76
	WORDP	WORDP S D		↑			
Conversion from BIN to gray code	GRY	GRY S D	Conversion to gray code • (S) → (D) └ BIN (-32768 to 32767)	⎍	3		6-77
	GRYP	GRYP S D		↑			
	DGRY	DGRY S D	Conversion to gray code • (S+1, S) → (D+1, D) └ BIN (-2147483648 to 2147483647)	⎍	3		6-77
	DGRYP	DGRYP S D		↑			
Conversion from gray code to BIN	GBIN	GBIN S D	Conversion to gray code • (S) → (D) └ Gray code (-32768 to 32767)	⎍	3		6-79
	GBINP	GBINP S D		↑			
	DGBIN	DGBIN S D	Conversion to gray code • (S+1, S) → (D+1, D) └ Gray code (-2147483648 to 2147483647)	⎍	3		6-79
	DGBINP	DGBINP S D		↑			
Complement to 2	NEG	NEG D	• (D) → (D) └ BIN data	⎍	2		6-81
	NEGP	NEGP D		↑			
	DNEG	DNEG D	• (D+1, D) → (D+1, D) └ BIN data	⎍	2		6-81
	DNEGP	DNEGP D		↑			
	ENEG	ENEG D	• (D+1, D) → (D+1, D) └ Real number data	⎍	2		6-83
	ENEGP	ENEGP D		↑			
Block conversions	BKBCD	BKBCD S D n	• Batch converts BIN data n points from (S) to BCD data and stores the result from (D) onward.	⎍	4		6-84
	BKBCDP	BKBCDP S D n		↑			
	BKBIN	BKBIN S D n	• Batch converts BCD data n points from (S) to BIN data and stores the result from (D) onward.	⎍	4		6-86
	BKBINP	BKBINP S D n		↑			

2.4.4 Data transfer instructions

Table 2.13 Data Transfer Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
16-bit data transfer	MOV	MOV S D	• (S) → (D)	⎍	3	●	6-88
	MOVP	MOVP S D		↑			
32-bit data transfer	DMOV	DMOV S D	• (S+1, S) → (D+1, D)	⎍	3		6-88
	DMOVP	DMOVP S D		↑			
Floating decimal point data transfer	EMOV	EMOV S D	• (S+1, S) → (D+1, D) └ Real number data	⎍	3		6-90
	EMOVP	EMOVP S D		↑			
Character string data transfer	$MOV	$MOV S D	• Transfers character string designated by (S) to device designated by (D) onward.	⎍	3		6-92
	$MOVP	$MOVP S D		↑			
16-bit data negation transfer	CML	CML S D	• $\overline{(S)}$ → (D)	⎍	3	●	6-94
	CMLP	CMLP S D		↑			
32-bit data negation transfer	DCML	DCML S D	• $\overline{(S+1, S)}$ → (D+1, D)	⎍	3	●	6-94
	DCMLP	DCMLP S D		↑			
Block transfer	BMOV	BMOV S D n		⎍	4	●	6-97
	BMOVP	BMOVP S D n		↑			
Multiple transfers of same data block	FMOV	FMOV S D n		⎍	4	●	6-99
	FMOVP	FMOVP S D n		↑			
16-bit data exchange	XCH	XCH S D	• (S) ↔ (D)	⎍	3	●	6-101
	XCHP	XCHP S D		↑			
32-bit data exchange	DXCH	DXCH S D	• (S+1, S) ↔ (D+1, D)	⎍	3	●	6-101
	DXCHP	DXCHP S D		↑			
Block data exchange	BXCH	BXCH S D n		⎍	4		6-103
	BXCHP	BXCHP S D n		↑			
Exchange of upper and lower bytes	SWAP	SWAP S D		⎍	3		6-105
	SWAPP	SWAPP S D		↑			

2.4.5 Program branch instruction

Table 2.14 Program Branch Instruction

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Jump	CJ	–[CJ │Pn]–	• Jumps to Pn when input conditions are met	⎍	2	●	6-106
	SCJ	–[SCJ │Pn]–	• Jumps to Pn from the scan after the meeting of input condition	⎍	2	●	6-106
	JMP	├──[JMP │Pn]–	• Jumps unconditionally to Pn		2	●	6-106
	GOEND	–[GOEND]–	• Jumps to END instruction when input condition is met	⎍	1		6-109

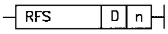
2.4.6 Program execution control instructions

Table 2.15 Program Execution Control Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Disable interrupts	DI	–[DI]–	• Prohibits the running of an interrupt program		1		6-110
Enable interrupts	EI	–[EI]–	• Resets interrupt program execution prohibition		1		6-110
Interrupt disable/enable setting	IMASK	–[IMASK │S]–	• Prohibits or permits interrupts for each interrupt program		2		6-110
Return	IRET	–[IRET]–	• Returns to sequence program following an interrupt program		1		6-113

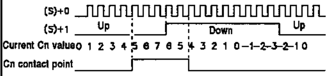
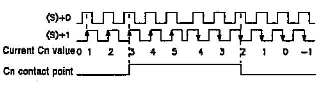
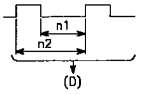
2.4.7 I/O refresh instructions

Table 2.16 I/O Refresh Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
I/O Refresh	RFS		• Refreshes the relevant I/O area during scan	↑	3		6-115

2.4.8 Other convenient instructions

Table 2.17 Other Convenient Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Up/Down Counter	UDCNT1	UDCNT1 S D n		⎍	4		6-117
	UDCNT2	UDCNT2 S D n		⎍	4		6-120
Teaching timer	TTMR	TTMR D n	• (Time that TTMR is ON)*n ⟶ (D) n=0:1, n=0:10, n=2:100	⎍	3		6-123
Special timer	STMR	STMR S n D	• The 4 points from the bit device designated 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 timer output		3		6-125
Nearest path control	ROTC	ROTC S n1 n2 D	• Rotates a rotary table with n1 divisions from the stop position to the position designated by (S+1) by the nearest path.	⎍	5		6-128
Ramp signal	RAMP	RAMP n1 n2 D1 n3 D2	• Changes device data designated by D1 from n1 to n2 in n3 scans.	⎍	6		6-131
Pulse density	SPD	SPD S n D	• Counts pulse input from device designated by D1 for time designated by n, and stores the count at device designated by (D).	⎍	4		6-133
Pulse output	PLSY	PLSY n1 n2 D	• (n1)Hz ⟶ (D) Output n2 times	⎍	4		6-135
Pulse width modulation	PWM	PWM n1 n2 D		⎍	4		6-137
Matrix input	MTR	MTR S D1 D2 n	• Successively inputs data of 16 points times n strings starting from the device designated by S1 and stores it from the device designated by D2 onward.	⎍	5		6-139

2.5 Application Instructions

2.5.1 Logical operation instructions

Table 2.18 Logical Operation Instructions

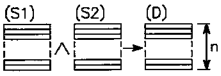
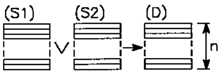
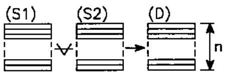
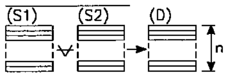
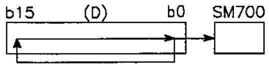
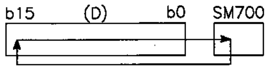
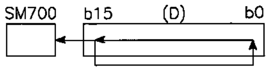
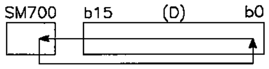
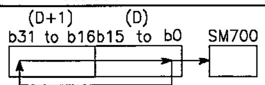
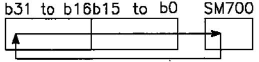
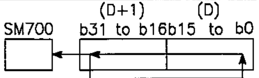
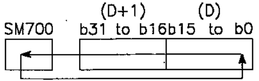
Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Logical product	WAND	WAND S D	• (D)∧(S) → (D)	⎍	3	●	7-3
	WANDP	WANDP S D		↑			
	WAND	WAND S1 S2 D	• (S1)∧(S2) → (D)	⎍	4		7-6
	WANDP	WANDP S1 S2 D		↑			
	DAND	DAND S D	• (D+1, D)∧(S+1, S) → (D+1, D)	⎍	3	●	7-3
	DANDP	DANDP S D		↑			
	DAND	DAND S1 S2 D	• (S1+1, S1)∧(S2+1, S2) → (D+1, D)	⎍	4		7-6
	DANDP	DANDP S1 S2 D		↑			
	BKAND	BKAND S1 S2 D n		⎍	5		7-9
	BKANDP	BKANDP S1 S2 D n		↑			
Logical sum	WOR	WOR S D	• (D)∨(S) → (D)	⎍	3	●	7-11
	WORP	WORP S D		↑			
	WOR	WOR S1 S2 D	• (S1)∨(S2) → (D)	⎍	4		7-14
	WORP	WORP S1 S2 D		↑			
	DOR	DOR S D	• (D+1, D)∨(S+1, S) → (D+1, D)	⎍	3	●	7-11
	DORP	DORP S D		↑			
	DOR	DOR S1 S2 D	• (S1+1, S1)∨(S2+1, S2) → (D+1, D)	⎍	4		7-14
	DORP	DORP S1 S2 D		↑			
	BKOR	BKOR S1 S2 D n		⎍	5		7-17
	BKORP	BKORP S1 S2 D n		↑			

Table 2.18 Logical Operation Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Exclusive OR	WXOR	-[WXOR S D]-	• $(D)\veebar(S)\longrightarrow(D)$	⎍	3	●	7-19
	WXORP	-[WXORP S D]-		↑			
	WXOR	-[WXOR S1 S2 D]-	• $(S1)\veebar(S2)\rightarrow(D)$	⎍	4		7-22
	WXORP	-[WXORP S1 S2 D]-		↑			
	DXOR	-[DXOR S D]-	• $(D+1, D)\veebar(S+1, S)\longrightarrow(D+1, D)$	⎍	3	●	7-19
	DXORP	-[DXORP S D]-		↑			
	DXOR	-[DXOR S1 S2 D]-	• $(S1+1, S1)\veebar(S2+1, S2)\longrightarrow(D+1, D)$	⎍	4		7-22
	DXORP	-[DXORP S1 S2 D]-		↑			
	BKXOR	-[BKXOR S1 S2 D n]-		⎍	5		7-25
	BKXORP	-[BKXORP S1 S2 D n]-		↑			
NON exclusive logical sum	WXNR	-[WXNR S D]-	• $\overline{(D)\veebar(S)}\longrightarrow(D)$	⎍	3	●	7-27
	WXNRP	-[WXNRP S D]-		↑			
	WXNR	-[WXNR S1 S2 D]-	• $\overline{(S1)\veebar(S2)}\rightarrow(D)$	⎍	4		7-29
	WXNRP	-[WXNRP S1 S2 D]-		↑			
	DXNR	-[DXNR S D]-	• $\overline{(D+1, D)\veebar(S+1, S)}\longrightarrow(D+1, D)$	⎍	3	●	7-27
	DXNRP	-[DXNRP S D]-		↑			
	DXNR	-[DXNR S1 S2 D]-	• $\overline{(S1+1, S1)\veebar(S2+1, S2)}\longrightarrow(D+1, D)$	⎍	4		7-29
	DXNRP	-[DXNRP S1 S2 D]-		↑			
	BKNXR	-[BKNXR S1 S2 D n]-		⎍	5		7-32
	BKNXRP	-[BKNXRP S1 S2 D n]-		↑			

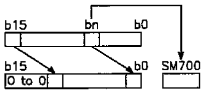
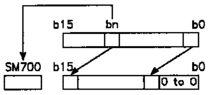
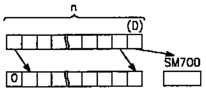
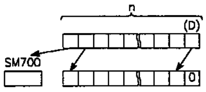
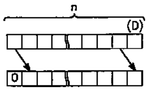
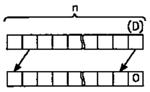
2.5.2 Rotation instructions

Table 2.19 Rotation Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Right rotation	ROR	ROR D n	 Rotates n bits to the right	⎍	3	●	7-34
	RORP	RORP D n		↑			
	RCR	RCR D n	 Rotates n bits to the right	⎍	3	●	7-34
	RCRP	RCRP D n		↑			
Left rotation	ROL	ROL D n	 Rotates n bits to the left	⎍	3	●	7-36
	ROLP	ROLP D n		↑			
	RCL	RCL D n	 Rotates n bits to the left	⎍	3	●	7-36
	RCLP	RCLP D n		↑			
Right rotation	DROR	DROR D n	 Rotates n bits to the right	⎍	3	●	7-38
	DRORP	DRORP D n		↑			
	DRCR	DRCR D n	 Rotates n bits to the right	⎍	3	●	7-38
	DRCRP	DRCRP D n		↑			
Left rotation	DROL	DROL D n	 Rotates n bits to the left	⎍	3	●	7-40
	DROLP	DROLP D n		↑			
	DRCL	DRCL D n	 Rotates n bits to the left	⎍	3	●	7-40
	DRCLP	DRCLP D n		↑			

2.5.3 Shift instructions

Table 2.20 Shift Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
n-bit shift	SFR	SFR D n		⎍	3	●	7-42
	SFRP	SFRP D n		↑			
	SFL	SFL D n		⎍	3	●	7-42
	SFLP	SFLP D n		↑			
1-bit shift	BSFR	BSFR D n		⎍	3		7-44
	BSFRP	BSFRP D n		↑			
	BSFL	BSFL D n		⎍	3		7-44
	BSFLP	BSFLP D n		↑			
1-word shift	DSFR	DSFR D n		⎍	3	●	7-46
	DSFRP	DSFRP D n		↑			
	DSFL	DSFL D n		⎍	3	●	7-46
	DSFLP	DSFLP D n		↑			

2.5.4 Bit processing instructions

Table 2.21 Bit processing instructions

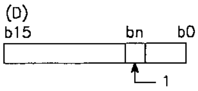
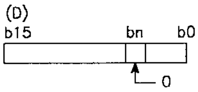
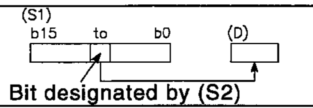
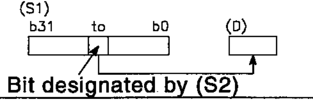
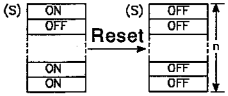
Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Bit set and reset	BSET	BSET D n		⎍	3	●	7-48
	BSETP	BSETP D n		↑			
	BRST	BRST D n		⎍	3	●	7-48
	BRSTP	BRSTP D n		↑			

Table 2.21 Bit processing Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Bit tests	TEST	TEST S1 S2 D	 Bit designated by (S2)	⎍	4		7-50
	TESTP	TESTP S1 S2 D		↑			
	DTEST	DTEST S1 S2 D	 Bit designated by (S2)	⎍	4		7-50
	DTESTP	DTESTP S1 S2 D		↑			
Batch reset of bit devices	BKRST	BKRST S n		⎍	3		7-52
	BKRSTP	BKRSTP S n		↑			

2.5.5 Data processing instructions

Table 2.22 Data Processing Instructions

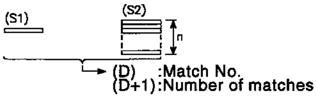
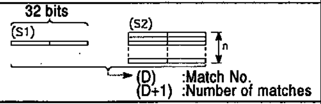
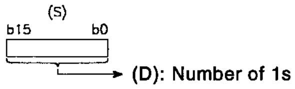
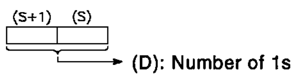
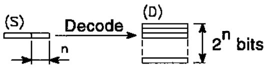
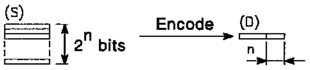
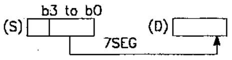
Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Data searches	SER	SER S1 S2 D n		⎍	5		7-54
	SERP	SERP S1 S2 D n		↑			
	DSER	DSER S1 S2 D n		⎍	5		7-54
	DSERP	DSERP S1 S2 D n		↑			
Bit checks	SUM	SUM S D		⎍	3	●	7-58
	SUMP	SUMP S D		↑			
	DSUM	DSUM S D		⎍	3	●	7-58
	DSUMP	DSUMP S D		↑			
Decode	DECO	DECO S D n	Decode from 8 to 256 	⎍	4		7-60
	DECOP	DECOP S D n		↑			
Encode	ENCO	ENCO S D n	Decode from 256 to 8 	⎍	4		7-62
	ENCOP	ENCOP S D n		↑			
7-segment decode	SEG	SEG S D		⎍	3	●	7-64
	SEGP	SEGP S D		↑			

Table 2.22 Data Processing Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Separating and linking	DIS	–[DIS | S | D | n]–	• Separates 16-bit data designated by (S) into 4-bit units, and stores at the lower 4 bits n points from (D). (n≤4)	⎍	4		7-66
	DISP	–[DISP | S | D | n]–		↑			
	UNI	–[UNI | S | D | n]–	• Links the lower 4 bits n points from the device designated by (S) and stores at the device designated by (D). (n≤4)	⎍	4		7-68
	UNIP	–[UNIP | S | D | n]–		↑			
	NDIS	–[NDIS | S1 | D | S2]–	• Separates the data at the devices below that designated by (S) into bits designated below (S2) and stores in sequence from the device designated by (D).	⎍	4		7-70
	NDISP	–[NDISP | S1 | D | S2]–		↑			
	NUNI	–[NUNI | S1 | D | S2]–	• Links the data at the devices below that designated by (S) in the bits designated below (S2) and stores in sequence from the device designated by (D).	⎍			
	NUNIP	–[NUNIP | S1 | D | S2]–		↑			
	WTOB	–[WTOB | S | D | n]–	• Breaks n-points of 16-bit data from the device designated by (S) into 8-bit units, and stores in sequence at the device designated by (D).	⎍	4		7-75
	WTOBP	–[WTOBP | S | D | n]–		↑			
	BTOW	–[BTOW | S | D | n]–	• Links the lower 8 bits of n-points of 16-bit data from the device designated by (S) into 16-bit units, and stores in sequence at the device designated by (D).	⎍			
	BTOWP	–[BTOWP | S | D | n]–		↑			

Table 2.22 Data Processing Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Search	MAX	MAX S D n	• Searches the data n-points from the device designated by (S) in 16-bit units, and stores the maximum value at the device designated by (D).	⎍	4		7-78
	MAXP	MAXP S D n		↑			
	MIN	MIN S D n	• Searches the data n-points from the device designated by (S) in 16-bit units, and stores the minimum value at the device designated by (D).	⎍			7-78
	MINP	MINP S D n		↑			
	DMAX	DMAX S D n	• Searches the data 2xn-points from the device designated by (S) in 32-bit units, and stores the maximum value at the device designated by (D).	⎍	4		7-81
	DMAXP	DMAXP S D n		↑			
	DMIN	DMIN S D n	• Searches the data 2xn-points from the device designated by (S) in 32-bit units, and stores the minimum value at the device designated by (D).	⎍			7-81
	DMINP	DMINP S D n		↑			
Sort	SORT	SORT S1 n S2 D1 D2 S2: Number of comparisons made during one run D1: Device to turn ON when sort is completed D2: For system use	• Sorts data n-points from device designated by (S1) in 16-bit units. ({n x (n-1)}/2 scans required)	⎍	6		7-84
	DSORT	DSORT S1 n S2 D1 D2 S2: Number of comparisons made during one run D1: Device to turn ON when sort is completed D2: For system use	• Sorts data 2xn-points from device designated by (S1) in 32-bit units. ({n x (n+1)}/2 scans required)				
Total value calculations	WSUM	WSUM S D n		⎍	4		7-87
	WSUMP	WSUMP S D n		↑			
	DWSUM	DWSUM S D n		⎍			7-89
	DWSUMP	DWSUMP S D n		↑			

2.5.6 Structure creation instructions

Table 2.23 Structure Creation Instructions

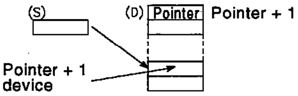
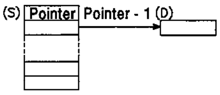
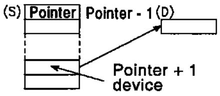
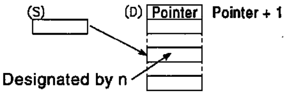
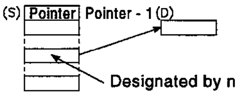
Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Number of repeats	FOR	FOR n	• Executes n times between FOR and NEXT		2		7-91
	NEXT	NEXT			1		
	BREAK	BREAK D Pn	• Forcibly ends the execution of the FOR to NEXT cycle and jumps pointer to Pn.	⎍	3		7-93
	BREAKP	BREAKP D Pn		↑			
Sub-routine program calls	CALL	CALL Pn S1 to Sn	• Executes sub-routine program Pn when input condition is met. (S1 to Sn are arguments sent to sub-routine program. 0≤n≤5)	⎍	*1 2+n		7-95
	CALLP	CALLP Pn S1 to Sn		↑			
	RET	RET	• Returns from sub-routine program		1		7-98
	FCALL	FCALL Pn S1 to Sn	• Performs non-execution processing on sub-routine program Pn if input conditions have not been met.	⊔⎍	*1 2+n		7-99
	FCALLP	FCALLP Pn S1 to Sn		↓			
	ECALL	ECALL * Pn S1 to Sn *: Program Name	• Executes sub-routine program Pn from within designated program name when input condition is met. (S1 to Sn are arguments sent to sub-routine program. 0≤n≤5)	⎍	*2 3+n		7-103
	ECALLP	ECALLP * Pn S1 to Sn *: Program Name		↑			
	EFCALL	EFCALL * Pn S1 to Sn *: Program Name	• Performs non-execution processing of sub-routine program Pn from within designated program name if input condition is not met.	⎍	*2 3+n		7-106
	EFCALLP	EFCALLP * Pn S1 to Sn *: Program Name		↑			
	COM	COM	• Performs link refresh and general data processing.		1		7-110
Fixed index qualification	IX	IX S Device qualification ladder	• Conducts index qualification for individual devices used in device qualification ladder.		2		7-113
	IXEND	IXEND			1		
	IXDEV	IXDEV	• Stores qualification value used for index qualification performed between IX and IXEND in the device below that designated by D.	↑	1		7-117
	IXSET	IXSET Pn D Designates qualification value			3		

*1: n indicates number of arguments for sub-routine program.

*2: n indicates the total of the number of arguments used in the sub-routine program and the number of program name steps. The number of program name steps is calculated as "number of characters in the program / 2" (decimal fraction is rounded up).

2.5.7 Table operation instructions

Table 2.24 Table Operation Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Table processing	FIFW	FIFW S D		⊓	3		7-119
	FIFWP	FIFWP S D		↑			
	FIFR	FIFR S D		⊓	3		7-121
	FIFRP	FIFRP S D		↑			
	FPOP	FPOP S D		⊓	3		7-123
	FPOPP	FPOPP S D		↑			
	FINS	FINS S D		⊓	4		7-125
	FINSP	FINSP S D		↑			
	FDEL	FDEL S D n		⊓	4		7-125
	FDELP	FDELP S D n		↑			

2.5.8 Buffer memory access instructions

Table 2.25 Buffer Memory Access Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Data read	FROM	FROM n1 n2 D n3	• Reads data in 16-bit units from special function module	⎍	5		7-128
	FROMP	FROMP n1 n2 D n3		↑			
	DFRO	DFRO n1 n2 D n3	• Reads data in 32-bit units from special function module	⎍	5		7-128
	DFROP	DFROP n1 n2 D n3		↑			
Data write	TO	TO n1 n2 S n3	• Writes data in 16-bit units to special function module	⎍	5		7-131
	TOP	TOP n1 n2 S n3		↑			
	DTO	DTO n1 n2 S n3	• Writes data in 32-bit units to special function module	⎍	5		7-131
	DTOP	DTOP n1 n2 S n3		↑			

2.5.9 Display instructions

Table 2.26 Display Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
ASCII print	PR	* SM701 When OFF PR S D	• Outputs ASCII code 8 points (16 characters) from device designated by (S) to output module.	↑	3		7-134
	PRC	* SM701 When ON PR S D	• Outputs ASCII code from device designated by (S) to 00_H to output module.				7-137
	PRC	PRC S D	• Converts comments from device designated by (S) to ASCII code and outputs to output module.				

Table 2.26 Display Instructions (continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Display	LED	-[LED S]-	• Displays ASCII code 8 points (16 characters) from the device designated by (S) at the LED display device on the front of the CPU	⎍	2		7-139
	LEDC	-[LEDC S]-	• Displays the comments from the device designated by (S) at the LED display device on the front of the CPU.				
Reset	LEDR	-[LEDR]-	• Resets annunciator and display unit display.	⎍	1		7-143

2.5.10 Debugging and failure diagnosis instructions

Table 2.27 Debugging and Failure Diagnosis Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Checks	CHKST	–[CHKST]–	• CHK instruction is executed when CHKST is executable. • Jumps to the step following the CHK instruction when CHKST is in a non-executable state.		1		7-145
	CHK	├┤├–┤├──[CHK]┤ Check Condition	• During normal conditions→ SM80: OFF, SD80: 0 • During abnormal conditions → SM80: ON, SD80: Failure No.				
	CHKCIR	–[CHKCIR]–	• Starts update in ladder pattern being checked by CHK instruction		1		7-150
	CHKEND	–[CHKEND]–	• Ends update in ladder pattern being checked by CHK instruction				
Status latch	SLT	–[SLT]–	• Executes status latch	↑	1		7-155
	SLTR	–[SLTR]–	• Resets status latch to enable re-execution				
Sampling trace	STRA	–[STRA]–	• Applies trigger to sampling trace	↑	1		7-157
	STRAR	–[STRAR]–	• Resets sampling trace to enable re-execution				
Program trace	PTRA	–[PTRA]–	• Applies trigger to program trace	↑	1		7-159
	PTRAR	–[PTRAR]–	• Resets program trace to enable re-execution				
	PTRAEXE	–[PTRAEXE]–	• Executes program trace	↑	1		7-159
	PTRAEXEP	–[PTRAEXEP]–		↑			

2.5.11 Character string processing instructions

Table 2.28 Character String Processing Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
From BIN to decimal ASCII	BINDA	─[BINDA | S | D]─	• Converts 1-word BIN value designated by (S) to a 5-digit, decimal ASCII value, and stores it at the word device designated by (D).	⎍	3		7-161
	BINDAP	─[BINDAP | S | D]─		↑			
	DBINDA	─[DBINDA | S | D]─	• Converts 2-word BIN value designated by (S) to a 10-digit, decimal ASCII value, and stores it at word devices following the word device number designated by (D).	⎍	3		7-161
	DBINDAP	─[DBINDAP | S | D]─		↑			
From BIN to hexadecimal ASCII	BINHA	─[BINHA | S | D]─	• Converts 1-word BIN value designated by (S) to a 4-digit, hexadecimal ASCII value, and stores it at a word device following the word device designated by (D).	⎍	3		7-164
	BINHAP	─[BINHAP | S | D]─		↑			
	DBINHA	─[DBINHA | S | D]─	• Converts 2-word BIN value designated by (S) to an 8-digit, hexadecimal ASCII value, and stores it at word devices following the word device designated by (D).	⎍	3		7-164
	DBINHAP	─[DBINHAP | S | D]─		↑			
From BCD to decimal ASCII	BCDDA	─[BCDDA | S | D]─	• Converts 1-word BCD value designated by (S) to a 4-digit, decimal ASCII value, and stores it at a word device following the word device designated by (D).	⎍	3		7-167
	BCDDAP	─[BCDDAP | S | D]─		↑			
	DBCDDA	─[DBCDDA | S | D]─	• Converts 2-word BCD value designated by (S) to an 8-digit, decimal ASCII value, and stores it at word devices following the word device number designated by (D).	⎍	3		7-167
	DBCDDAP	─[DBCDDAP | S | D]─		↑			
From decimal ASCII to BIN	DABIN	─[DABIN | S | D]─	• Converts a 5-digit, decimal ASCII value designated by (S) to a 1-word BIN value, and stores it at a word device number designated by (D).	⎍	3		7-170
	DABINP	─[DABINP | S | D]─		↑			
	DDABIN	─[DDABIN | S | D]─	• Converts a 10-digit, decimal ASCII value designated by (S) to a 2-word BIN value, and stores it at a word device number designated by (D).	⎍	3		7-170
	DDABINP	─[DDABINP | S | D]─		↑			

Table 2.28 Character String Processing Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
From hexadecimal ASCII to BIN	HABIN	HABIN S D	• Converts a 4-digit, hexadecimal ASCII value designated by (S) to a 1-word BIN value, and stores it at a word device number designated by (D).	⎍	3		7-173
	HABINP	HABINP S D		↑			
	DHABIN	DHABIN S D	• Converts an 8-digit, hexadecimal ASCII designated by (S) value to a 2-word BIN value, and stores it at a word device number designated by (D).	⎍	3		7-173
	DHABINP	DHABINP S D		↑			
From decimal ASCII to BCD	DABCD	DABCD S D	• Converts a 4-digit, decimal ASCII value designated by (S) to a 1-word BCD value, and stores it at a word device number designated by (D).	⎍	3		7-176
	DABCDP	DABCDP S D		↑			
	DDABCD	DDABCD S D	• Converts an 8-digit, decimal ASCII designated by (S) value to a 2-word BCD value, and stores it at a word device number designated by (D).	⎍	3		7-176
	DDABCDP	DDABCDP S D		↑			
Device comment read operation	COMRD	COMRD S D	• Stores comment from device designated by (S) at a device designated by (D).	⎍	3		7-179
	COMRDP	COMRDP S D		↑			
Character string length detection	LEN	LEN S D	• Stores data length (number of characters) in character string designated by (S) at a device designated by (D).	⎍	3		7-181
	LENP	LENP S D		↑			
BIN to decimal character string	STR	STR S1 S2 D	• Converts a 1-word BIN value designated by S2 to a decimal character string with the total number of digits and the number of decimal fraction digits designated by S1 and stores this at a device designated by (D).	⎍	4		7-183
	STRP	STRP S1 S2 D		↑			
	DSTR	DSTR S1 S2 D	• Converts a 2-word BIN value designated by S2 to a decimal character string with the total number of digits and the number of decimal fraction digits designated by S1 and stores this at a device designated by (D).	⎍	4		7-183
	DSRTP	DSTRP S1 S2 D		↑			

Table 2.28 Character String Processing Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Decimal character string to BIN	VAL	─[VAL │ S │ D1 │ D2]─	• Converts a character string including decimal point designated by (S) to a 1-word BIN value and the number of decimal fraction digits, and stores at devices designated by D1 and D2.	⎍	4		7-189
	VALP	─[VALP │ S │ D1 │ D2]─		↑			
	DVAL	─[DVAL │ S │ D1 │ D2]─	• Converts a character string including decimal point designated by (S) to a 2-word BIN value and the number of decimal fraction digits, and stores at devices designated by D1 and D2.	⎍	4		7-189
	DVALP	─[DVALP │ S │ D1 │ D2]─		↑			
Floating decimal point to character string	ESTR	─[ESTR │ S1 │ S2 │ D]─	• Converts floating decimal point data designated by (S) to character string, and stores in a device designated by (D).	⎍	4		7-194
	ESTRP	─[ESTRP │ S1 │ S2 │ D]─		↑			
Character string to floating decimal point	EVAL	─[EVAL │ S │ D]─	• Converts character string designated by (S) to floating decimal point data, and stores in a device designated by (D).	⎍	3		7-201
	EVALP	─[EVALP │ S │ D]─		↑			
Hexadecimal BIN to ASCII	ASC	─[ASC │ S │ D │ n]─	• Converts 1-word BIN value following device number designated by (S) to hexadecimal ASCII, and stores designated number of characters following a word device number designated by (D).	⎍	4		7-205
	ASCP	─[ASCP │ S │ D │ n]─		↑			
ASCII to hexadecimal BIN	HEX	─[HEX │ S │ D │ n]─	• Converts just the number of characters designated by n from hexadecimal ASCII data following word device designated by (S) to BIN value, and stores following device number designated by (D).	⎍	4		7-207
	HEXP	─[HEXP │ S │ D │ n]─		↑			

Table 2.28 Character String Processing Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Character string processing	RIGHT	–[RIGHT S D n]–	• Stores n number of characters from the end of a character string designated by (S) at a device designated by (D).	⎍	4		7-209
	RIGHTP	–[RIGHTP S D n]–		↑			
	LEFT	–[LEFT S D n]–	• Stores n number of characters from the beginning of a character string designated by (S) at a device designated by (D).	⎍			
	LEFTP	–[LEFTP S D n]–		↑			
	MIDR	–[MIDR S1 D S2]–	• Stores a designated number of characters from the position designated by (S2) of a character string designated by (S1) to a device designated by (D).	⎍	4		7-212
	MIDRP	–[MIDRP S1 D S2]–		↑			
	MIDW	–[MIDW S1 D S2]–	• Stores a character string designated by (S1) a designated number of characters from the position designated by (S2) of a device designated by (D).	⎍			
	MIDWP	–[MIDWP S1 D S2]–		↑			
	INSTR	–[INSTR S1 S2 D n]–	• Searches character string (S1) from the nth character of character string (S2), and stores matched positions at (D).	⎍	5		7-216
	INSTRP	–[INSTRP S1 S2 D n]–		↑			
Floating decimal point to BCD	EMOD	–[EMOD S1 S2 D]–	• Converts floating decimal point data (S1) to BCD data with number of decimal fraction digits designated by (S2) , and stores at device designated by (D).	⎍	4		7-218
	EMODP	–[EMODP S1 S2 D]–		↑			
BCD to floating decimal point data	EREXP	–[EREXP S D]–	• Converts BCD data (S1) to floating decimal point data with the number of decimal fraction digits designated by (S2), and stores at device designated by (D).	⎍	3		7-220
	EREXPP	–[EREXPP S D]–		↑			

2.5.12 Special function instructions

Table 2.29 Special Function Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Trigonometric functions (Floating decimal point data)	SIN	SIN S D	• Sin(S+1, S) ⟶ (D+1, D)	⎍	3		7-222
	SINP	SINP S D		↑			
	COS	COS S D	• Cos(S+1, S) ⟶ (D+1, D)	⎍	3		7-224
	COSP	COSP S D		↑			
	TAN	TAN S D	• Tan(S+1, S) ⟶ (D+1, D)	⎍	3		7-226
	TANP	TANP S D		↑			
	ASIN	ASIN S D	• Sin^{-1}(S+1, S) ⟶ (D+1, D)	⎍	3		7-228
	ASINP	ASINP S D		↑			
	ACOS	ACOS S D	• Cos^{-1}(S+1, S) ⟶ (D+1, D)	⎍	3		7-230
	ACOSP	ACOSP S D		↑			
	ATAN	ATAN S D	• Tan^{-1}(S+1, S) ⟶ (D+1, D)	⎍	3		7-232
	ATANP	ATANP S D		↑			
Conversion between angles and radians	RAD	RAD S D	• (S+1, S) ⟶ (D+1, D) Conversion from angles to radians	⎍	3		7-234
	RADP	RADP S D		↑			
	DEG	DEG S D	• (S+1, S) ⟶ (D+1, D) Conversion from radians to angles	⎍	3		7-236
	DEGP	DEGP S D		↑			
Square root	SQR	SQR S D	• $\sqrt{(S+1, S)}$ ⟶ (D+1, D)	⎍	3		7-238
	SQRP	SQRP S D		↑			
Exponent operations	EXP	EXP S D	• $e^{(S+1, S)}$ ⟶ (D+1, D)	⎍	3		7-240
	EXPP	EXPP S D		↑			
Natural logarithms	LOG	LOG S D	• Log e(S1+1, S) ⟶ (D+1, D)	⎍	3		7-242
	LOGP	LOGP S D		↑			

Table 2.29 Special Function Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Square root	BSQR	BSQR S D	• $\sqrt{(S)}$ → (D)+0 Integer part +1 Decimal fraction part	⎍	3		7-246
	BSQRP	BSQRP S D		↑			
	BDSQR	BDSQR S D	• $\sqrt{(S+1, S)}$ → (D)+0 Integer part +1 Decimal fraction part	⎍	3		7-246
	BDSQRP	BDSQRP S D		↑			
Trigonometric function	BSIN	BSIN S D	• Sin(S) → (D)+0 Sign +1 Integer part +2 Decimal fraction part	⎍	3		7-249
	BSINP	BSINP S D		↑			
	BCOS	BCOS S D	• Cos(S) → (D)+0 Sign +1 Integer part +2 Decimal fraction part	⎍	3		7-251
	BCOSP	BCOSP S D		↑			
	BTAN	BTAN S D	• Tan(S) → (D)+0 Sign +1 Integer part +2 Decimal fraction part	⎍	3		7-253
	BTANP	BTANP S D		↑			
	BASIN	BASIN S D	• $\mathrm{Sin}^{-1}(S)$ → (D)+0 Sign +1 Integer part +2 Decimal fraction part	⎍	3		7-256
	BASINP	BASINP S D		↑			
	BACOS	BACOS S D	• $\mathrm{Cos}^{-1}(S)$ → (D)+0 Sign +1 Integer part +2 Decimal fraction part	⎍	3		7-258
	BACOSP	BACOSP S D		↑			
	BATAN	BATAN S D	• $\mathrm{Tan}^{-1}(S)$ → (D)+0 Sign +1 Integer part +2 Decimal fraction part	⎍	3		7-261
	BATANP	BATANP S D		↑			

2.5.13 Data control instructions

Table 2.30 Data Control Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Upper and lower limit controls	LIMIT	LIMIT S1 S2 S3 D	• When S3<S1 ... Store value of S1 at (D) • When S1≤S3≤S2 ... Store value of S3 at (D) • When S2<S3 ... Store value of S2 at (D)	⎍	5		7-263
	LIMITP	LIMITP S1 S2 S3 D		↑			
	DLIMIT	DLIMIT S1 S2 S3 D	• When (S3+1, S3)<(S1+1, S1) ... Store value of (S1+1, S1) at ((D)+1, (D)) • When (S1+1, S1)≤(S3+1, S3)<(S2+1, S2) ... Store value of (S3+1, S3) at ((D)+1, (D)) • When (S2, S2+1)<(S3, S3+1) ... Store value of (S2+1, S2) at ((D)+1, (D))	⎍	5		7-263
	DLIMITP	DLIMITP S1 S2 S3 D		↑			
Dead band controls	BAND	BAND S1 S2 S3 D	• When S1≤S3≤S2 . . . 0→(D) • When S3<S1 . . . . S3→S1→(D) • When S2<S3 . . . . S3→S2→(D)	⎍	5		7-266
	BANDP	BANDP S1 S2 S3 D		↑			
	DBAND	DBAND S1 S2 S3 D	• 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))	⎍	5		7-266
	DBANDP	DBANDP S1 S2 S3 D		↑			
Zone controls	ZONE	ZONE S1 S2 S3 D	• When S3=0... 0→(D) • When S3>0... S3+S2→(D) • When S3<0... S3→S1→(D)	⎍	5		7-269
	ZONEP	ZONEP S1 S2 S3 D		↑			
	DZONE	DZONE S1 S2 S3 D	• 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))	⎍	5		7-269
	DZONEP	DZONEP S1 S2 S3 D		↑			

2.5.14 Switching instructions

Table 2.31 Switching Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Block number designations	RSET	RSET | S	• Converts extension file register block number to number designated by (S).	⎍	2+n*		7-272
	RSETP	RSETP | S		↑			
	QDRSET	QDRSET | File Name	• Sets file names used as file registers.	⎍	2+n*		7-274
	QDRSETP	QDRSETP | File Name		↑			
	QCDSET	QCDSET | File Name	• Sets file names used as comment files.	⎍			7-276
	QCDSETP	QCDSETP | File Name		↑			

*: n ([number of file name characters] / 2) indicates a step. (Decimal fractions are rounded up.)

2.5.15 Clock instructions

Table 2.32 Clock Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Read/write clock data	DATERD	DATERD D	• (Clock device)→ (D)+0 Year +1 Month +2 Day +3 Hour +4 Minute +5 Sec. +6 Day of week	⎍	2		7-278
	DATERDP	DATERDP D		↑			
	DATEWR	DATEWR D	• (D)+0 Year → (Clock device) +1 Month +2 Day +3 Hour +4 Minute +5 Sec. +6 Day of week	⎍	2		7-280
	DATEWRP	DATEWRP D		↑			
	DATE+	DATE+ S1 S2 D	(S1) Hour Minute Sec. + (S2) Hour Minute Sec. → (D) Hour Minute Sec.	⎍	4		7-282
	DATE+P	DATE+P S1 S2 D		↑			
	DATE-	DATE– S1 S2 D	(S1) Hour Minute Sec. - (S2) Hour Minute Sec. → (D) Hour Minute Sec.	⎍	4		7-284
	DATE-P	DATE–P S1 S2 D		↑			
	SECOND	SECOND S D	(S) Hour Minute Sec. → (D) Sec. (lower level) Sec. (upper level)	⎍	3		7-286
	SECONDP	SECONDP S D		↑			
	HOUR	HOUR S D	(S) Sec. (lower level) Sec. (upper level) → (D) Hour Minute Sec.	⎍			
	HOURP	HOURP S D		↑			

2.5.16 Peripheral device instructions

Table 2.33 Peripheral Device Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Input/output to peripheral devices	MSG	–[MSG S]–	• Stores message designated by (S) at QnACPU. This message is displayed at the peripheral device	⎍	2		7-289
	PKEY	–[PKEY D]–	• Data input from the peripheral device is stored at device designated by (D).	⎍	2		7-291

2.5.17 Program instructions

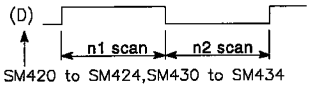
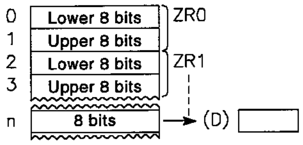
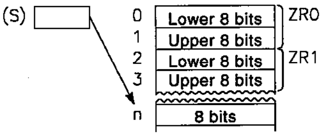
Table 2.34 Program Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
	PSTOP	–[PSTOP Program Name]–	• Places designated program in standby status	⎍	2+n*		7-293
	PSTOPP	–[PSTOPP Program Name]–		↑			
	POFF	–[POFF Program Name]–	• Turns OUT instruction coil of designated program OFF, and places program in standby status.	⎍	2+n*		7-295
	POFFP	–[POFFP Program Name]–		↑			
	PSCAN	–[PSCAN Program Name]–	• Registers designated program as scan execution program.	⎍	3+n*		7-297
	PSCANP	–[PSCANP Program Name]–		↑			
	PLOW	–[PLOW Program Name]–	• Registers designated program as low-speed execution program.	⎍	3+n*		7-299
	PLOWP	–[PLOWP Program Name]–		↑			

*: n ([number of program name characters] / 2) indicates a step. (Decimal fractions are rounded up.)

2.5.18 Other instructions

Table 2.35 Other Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
WDT reset	WDT	WDT	• Resets watchdog timer during sequence program	⎍	1		7-301
	WDTP	WDTP		↑			
Timing clock	DUTY	DUTY n1 n2 D		↑	4		7-303
Direct read/write operations in 1-byte units	ZRRDB	ZRRDB n D		⎍	3		7-305
	ZRRDBP	ZRRDBP n D		↑			
	ZRWRB	ZRWRB n D		⎍	3		7-307
	ZRWRBP	ZRWRBP n D		↑			
	ADRSET	ADRSET S D	(S) ——→ (D) Indirect address of designated device Device name	⎍	3		7-309
	ADRSETP	ADRSETP S D		↑			
Numerical key input from keyboard	KEY	KEY S n D1 D2	• Takes in ASCII data for 8 points of input unit designated by (S), converts to hexadecimal value following device number designated by D1, and stores.	⎍	5		7-310

3. CONFIGURATION OF INSTRUCTIONS

3.1 Configuration of Instructions

The majority of the instructions that can be used with QnACPU can be broken into an instruction part and a device part.

The instruction part and device part have the following applications:

- Instruction part .... Indicates the function of the instruction
- Device part .... Indicates the data that is to be used with the instruction.

The device part is classified into source data, destination data, and number of devices.

(1) Source (S)

(a) Source data is the data that is used for operations.

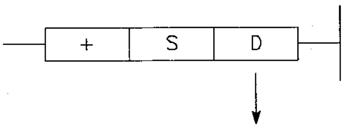
(b) The following types of source are possible for the various instru ctions, depending on the device that has been designated:

- Constants
  .... Designates the numerical value that will be used in the operation.
  This is set when the program iswritten, and cannot be changed during the execution of the program.
  Any variable data that is to be used as a constant should be indexed.
- Bit devices and Word devices
  .... Designates the device where the data to be used by the operation is being stored.
  Data must be stored in the designated device up to the point that the operation is executed.
  Changing the data that is storedin a designated device during the execution of a program makes it possible to change the data which will be used by the instruction in question.

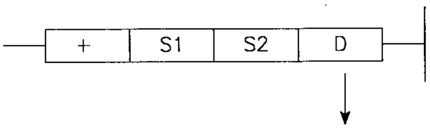
(2) Destination (D)

(a) The destination indicates where data will be stored after the operation has been conducted.
However, some instructions also require the storing of data that will be used in an operation at the destination prior to the execution of the operation.

Example: An addition instruction involving BIN 16-bit data



Stores data needed for the actual operation prior to its use.



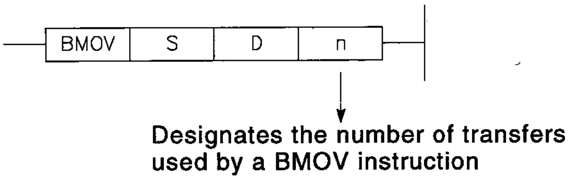
Stores results of the operation.

(b) A device for the storage of data must always be set for the destination.

(3) Number of devices and number of transfers (n)

(a) The number of devices and number of transfers designate the numbers of devices and transfers used by instructions involving multiple devices.

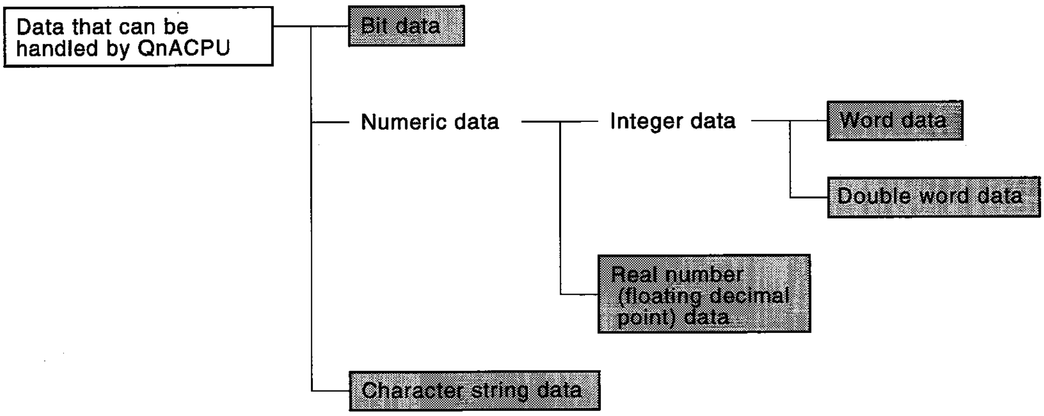
Example: Block transfer instruction



(b) The number of devices or number of transfers can be set between 0 and 32767.
However, if the number is 0, the instruction will be a no-operation instruction.

3.2 Designating Data

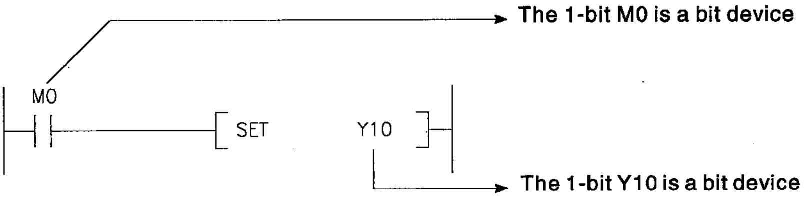
The following five types of data can be used with QnACPU instructions:



3.2.1 Using bit data

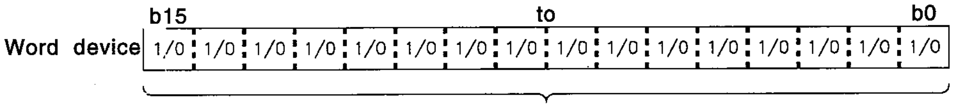
Bit data is data used in one-bit units, such as for contact points or coils. "Bit devices" and "Bit designated word devices" can be used as bit data.

(1) When using bit devices
Bit devices are designated in one-point units.



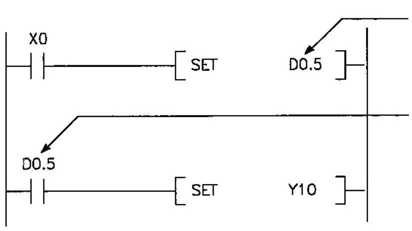
(2) Using word devices

(a) Word devices enable the use of a designated bit number 1/0 as bit data by the designation of that bit number.



Each bit can be used as 1 for ON and 0 for OFF.

(b) Word device bit designation is done by designating Word Device Bit No. (Designation of bit numbers is done in hexadecimal.)
For example, bit 5 (b5) of D0 is designated as D0.5, and bit 10 (b10) of D0 is designated as D0.A.
However, there can be no bit designation for timers (T), retentive timers (ST), or counters (C).



Word device bit designation. (Bit 5 (b5) of D0 designated as ON (1).)

Word device bit designation. (Turned ON or OFF according to the 1/O status of bit 5 (b5) of D0.)

3.2.2 Using word (16 bits) data

Word data is 16-bit numeric data used by basic instructions and application instructions.
The following two types of word data can be used with QnACPU:
- Decimal constants .... K-32768 to K32767
- Hexadecimal constants........ H0000 to HFFFF

Word devices and bit devices designated by digit can be used as word data.
For direct access input (DX) and direct access output (DY), designation of word data is not possible by digit designation. For details on using word data for direct access input and direct access output, refer to QnACPU programming Manual (Fundamentals).

(1) When using bit devices

(a) Bit devices can deal with word data when digits are desigated. Digit designation of bit devices is done by designating Number of digits and Initial number of bit device.
Digit designation of bit devices can be done in 4-point (4-bit) units, and designation can be made for K1 through K4.
(For link direct devices, designation is done by J Network No. / Digit designation Initial number of bit device. When X100 through X10F are designated for Network No.2, it is done by J2/K4X100.)
For example, if X0 is designated for digit designation, the following points would be designated:

- K1X0...... The 4 points X0 through X3 are designated
- K2X0...... The 8 points X0 through X7 are designated
- K3X0...... The 12 points X0 through XB are designated
- K4X0...... The 16 points X0 through XF are designated

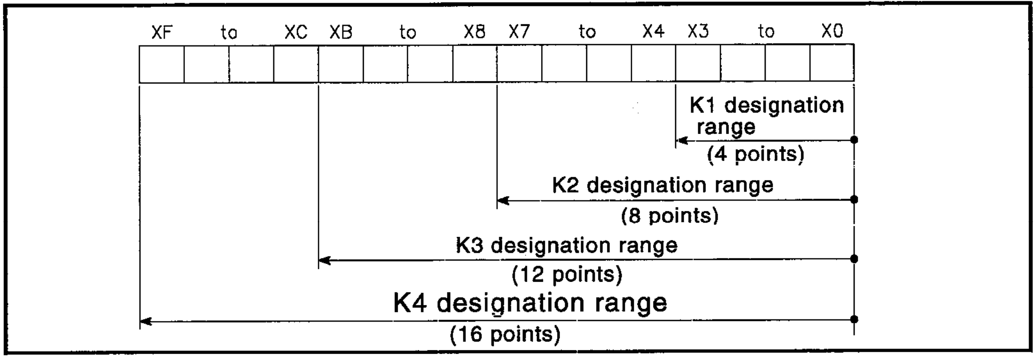


Fig 3.1 Digit Designation Setting Range for 16-Bit Instruction

(b) In cases where digit designation has been made at the source (S), the numeric values shown in Table 3.1 are those which can be dealt with as source data.

Table 3.1 List of Numeric Values that Can Be Dealt with as Digit Designation

Number of Digits Designated	With 16-Bit Instruction
K1 (4 points)	0 to 15
K2 (8 points)	0 to 255
K3 (12 points)	0 to 4095
K4 (16 points)	-32768 to 32767

In cases where the source is a bit device designated by digit designation, and the destination is a word device, the word device for the destination becomes 0 following the bit designated by digit designation at the source.

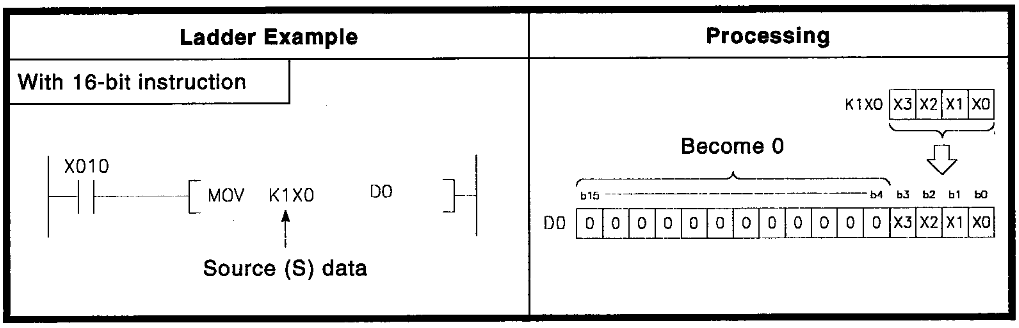


Fig 3.2 Ladder Example and Processing Conducted

(c) In cases where digit designation is made at the destination (D), the number of points designated are used as the destination. Bit devices below the number of points designated as digits do not change.

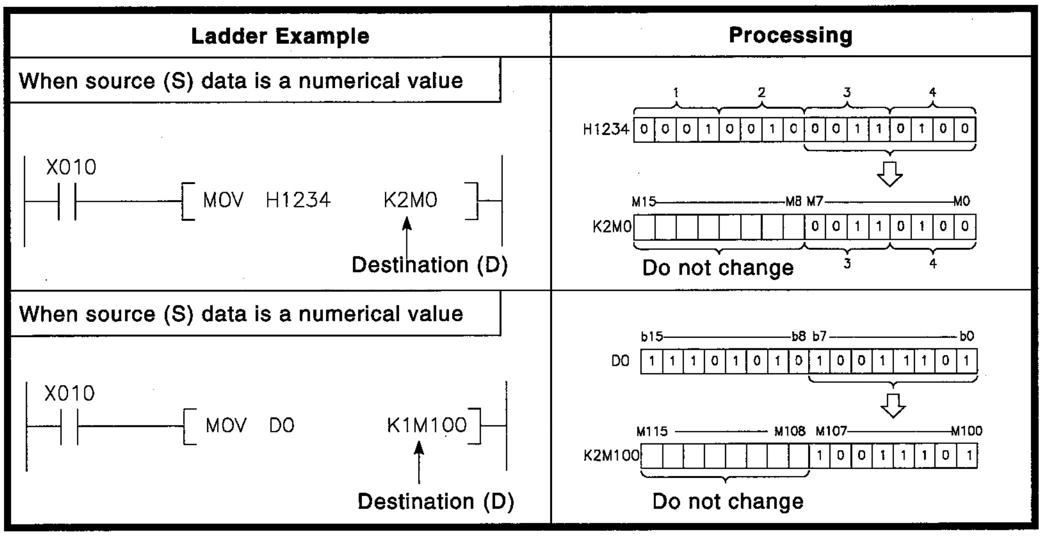
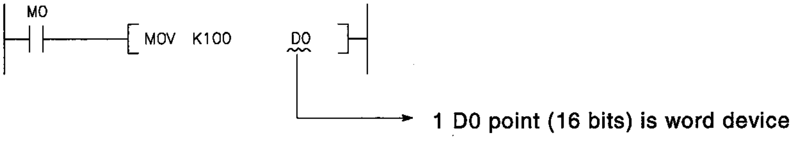


Fig 3.3 Ladder Example and Processing Conducted

(2) When using word devices
Word devices are designated in 1-point (16 bits) units.



POINT

(1) When digit designation processing is conducted, a random value can be used for the bit device initial device number.

(2) Digit designation cannot be made for the direct device designation DX and DY.

3.2.3 Using double word data (32 bits)

Double word data is 32-bit numerical data used by basic instructions and application instructions.
The two types of double word data that can be dealt with by QnACPU are as follows:
- Decimal constants....... K-2147483648 to K2147483647
- Hexadecimal constants... H00000000 to HFFFFFFFF

Word devices and bit devices designated by digit designation can be used as double word data.
For direct access input (DX) and direct access output (DY), designation of double word data is not possible by digit designation.

(1) When using bit devices

(a) Digit designation can be used to enable a bit device to deal with double word data.
Digit designation of bit devices is done by designating Number of digits and Initial number of bit device.
Digit designation of bit devices can be done in 4-point (4-bit) units, and designation can be made for K1 through K8.
(For link direct devices, designation is done by J Network No. / Digit designation Initial number of bit device.When X100 through X11F are designated for Network No.2, it is done by J2/K8X100.)
For example, if X0 is designated for digit designation, the following points would be designated:
- K1X0....... The 4 points X0 through X3 are designated
- K2X0....... The 8 points X0 through X7 are designated
- K3X0....... The 12 points X0 through XB are designated
- K4X0....... The16 points X0 through XF are designated
- K5X0....... The 20 points X0 through X13 are designated
- K6X0....... The 24 points X0 through X17 are designated
- K7X0....... The 28 points X0 through X1B are designated
- K8X0....... The 32 points X0 through X1F are designated

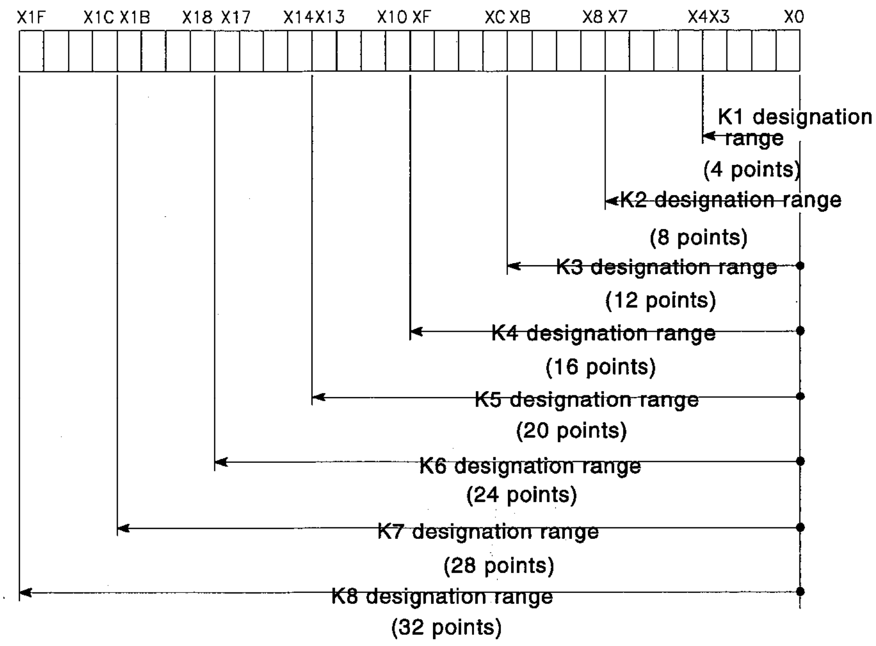


Fig 3.4 Digit Designation Setting Range for 32-Bit Instructions

(b) In cases where digit designation has been made at the source (S), the numeric values shown in Table 3.2 are those which can be dealt with as source data.

Table 3.2 List of Numeric Values that Can Be Dealt with as Digit Designation

Number of Digits Designated	With 32-bit Instructions	Number of Digits Designated	With 32-bit Instructions
K1 (4 points)	0 to 15	K5 (20 points)	0 to 1048575
K2 (8 points)	0 to 255	K6 (24 points)	0 to 16777215
K3 (12 points)	0 to 4095	K7 (28 points)	0 to 268435455
K4 (16 points)	0 to 65535	K8 (32 points)	-2147483648 to 2147483647

In cases where the source is a bit device designated by digit designation, and the destination is a word device, the word device for the destination becomes 0 following the bit designated by digit designation at the source.

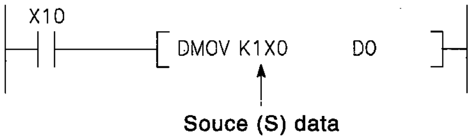
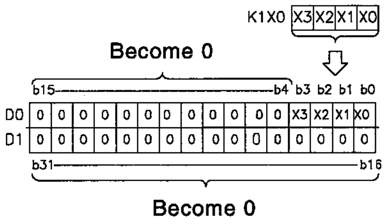
Ladder Example	Processing
With 32-bit instruction 	

Fig 3.5 Ladder Example and Processing Conducted

(c) In cases where digit designation is made at the destination (D), the number of points designated are used as the destination.
Bit devices after the number of points designated as digits do not change.

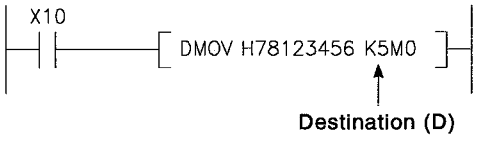
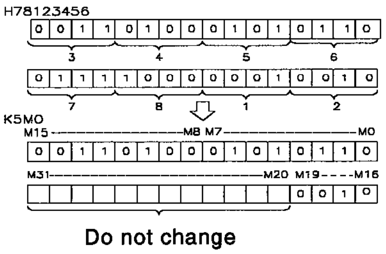
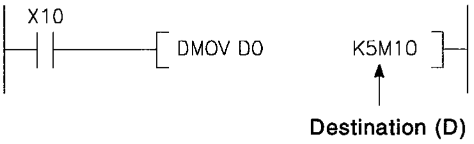
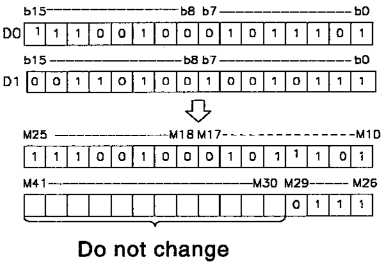
Ladder Example	Processing
When source (S) data is a numerical value 	
When source (S) data is a word device 	

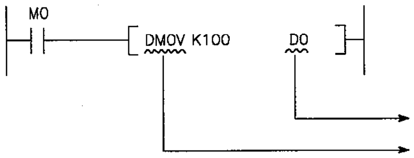
Fig 3.6 Ladder Example and Processing Conducted

POINT

(1)When digit designation processing is conducted, a random value can be used for the bit device initial device number.

(2)Digit designation cannot be made for the direct device designation DX and DY.

(2) When using word devices
A word device designates devices used by the lower 16 bits of data.
A 32-bit instruction uses (designation device number) and (designation device number + 1).

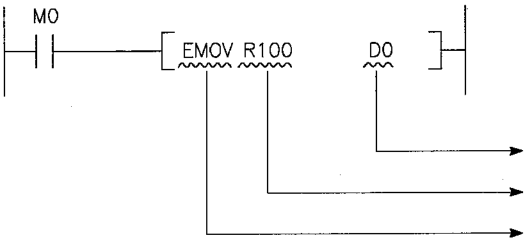


The 2 points D0 and D1 are used
32-bit data transfer instruction

3.2.4 Using real number data

Real number data is 32-bit floating decimal point data used with basic instructions and application instructions.
Only word devices are capable of storing real number data.

Instructions which deal with real numbers designate devices which are used for the lower 16 bits of data.
Real numbers are stored in the 32 bits which make up (designated device number) and (designated device number + 1).



The 2 points D0 and D1 (32 bits) are used
The 2 points R100 and R101 (32 bits) are used
Real number data transfer

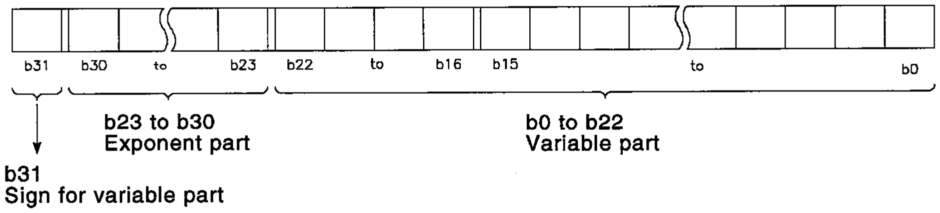
REMARK

In sequence programs, real numbers are designated by E□.

Floating decimal point data uses two word devices and is expressed in the following manner:

1. [Variable part] x 2 $^{\text{[exponent part]}}$

The bit configuration and meaning of the internal representation of floating decimal point data is as follows:



- Sign for variable part The sign for the variable part is represented at b31.
  0: Positive
  1: Negative

- Exponent part The n of 2^n is represented from b23 to b30. Depending on the BIN value of b23 through b30, the value of n is as follows.

b23 to b30	FF_H	FE_H	FD_H			81_H	80_H	$7F_H$	$7E_H$			02_H	01_H	00_H
n	Not in use	127	126			2	1	0	-1			-125	-126	Not in use

- Exponent part The 23 bits from b0 through b22, represents the XXXXXX... at binary 1.XXXXXX....

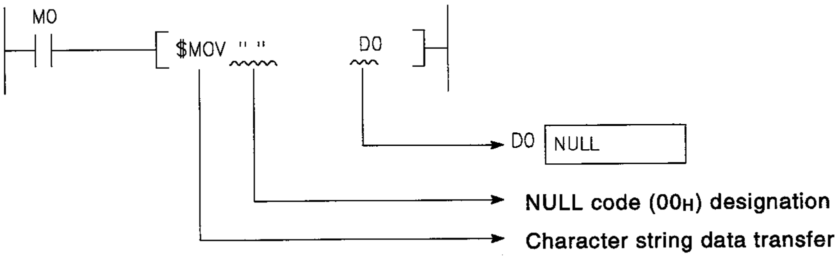
POINT

- The QnACPU floating decimal point data can be monitored using the monitoring function of a peripheral device.
- When this is expressed as 0, all data from b0 to b31 will be 0.

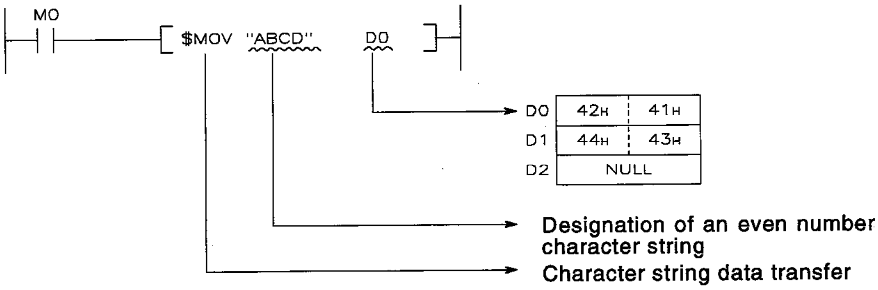
3.2.5 Using character string data

Character string data is character data used by basic instructions and application instructions.
It encompasses all data from the designated character to the NULL code (00H).

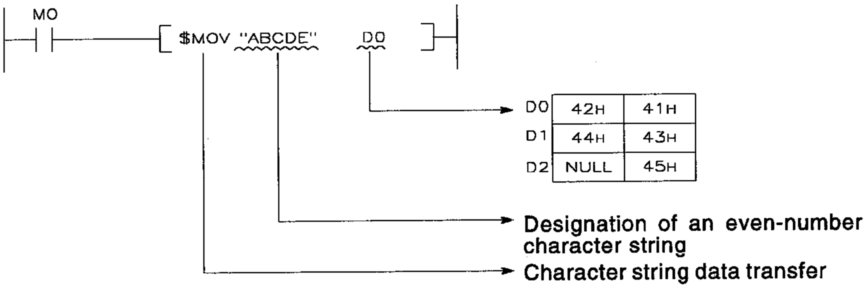
(1) When designated character is the NULL code.
One word is used to store the NULL code.



(2) When character string is even
Uses (number of characters/2 + 1) words, and stores character string and NULL code.
For example, if "ABCD" is transferred to D0, the character string ABCD is stored at D0 and D1, and the NULL code is stored at D2.



(3) When number of characters is odd
Uses (number of characters/2) words (rounds up decimal fractions) and stores the character string and NULL code.
For example, if "ABCD" is transferred to D0, the character string (ABCDE) and the NULL code are stored from D0 to D2.



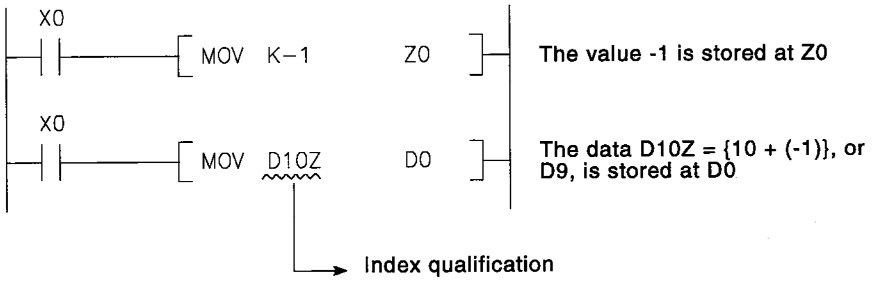
3.3 Index Qualification

(1) Index qualification

(a) Index qualification is an indirect setting made by using an index register.
When an index qualification is used in a sequence program, the device to be used will become the device number designated directly plus the contents of the index register.
For example, if D0Z2 has been designated, and the content of Z2 is 3, D(2+3), or D5, will become the designated device.

(b) There are 16 index registers, from Z0 to Z15.
Each index register can be set between -32768 and 32767.

(2) Index qualification is as follows:



Example

A case where index qualification has been performed, and the actual process device, would be as follows: (When Z0=20 and Z1=-5)

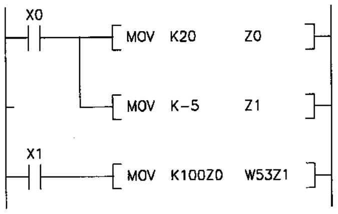
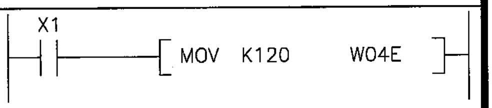
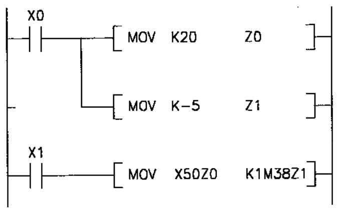
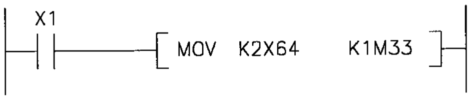
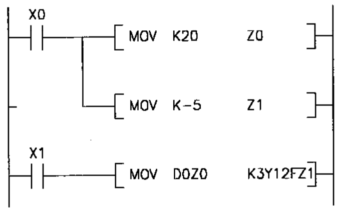
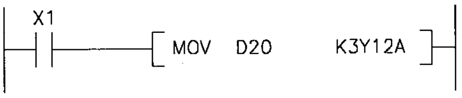
Ladder Example	Actual Process Device
	 Description K100Z .... K (100+20)=K120 W53 ...... W (53-5)=W4E Hexadecimal number
	 Description K2X50Z... K2X (50+14)=K2X64 K20 is converted to hexadecimal K1M38.... K1M (38-5)=K1M33
	 Description D0Z ...... D(0+20)=D20 K3Y12F... K3Y (12F-5)=K3Y12A Hexadecimal number

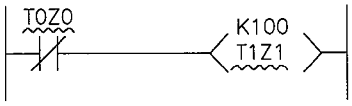
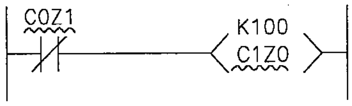
Fig. 3.7 Ladder Example and Actual Process Device

(3) Devices which can be index-qualified
With the exception of the restrictions noted below, index qualification can be used with devices used with contacts, coils, basic instructions, and application instructions.

(a) Devices which cannot use index qualification

Device	Meaning
E	Floating decimal point data
$	Character-string data
□.□	Bit designated for word device
FX, FY, FD	Function devices
P	Pointers used as labels
I	Interrupt pointers used as labels
Z	Index register
S	Step relay
TR	SFC transfer devices *
BL	SFC block devices *
T, ST	Value set for timer
C	Value set for counter

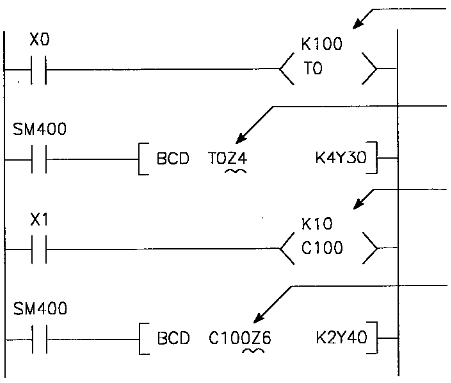
(b) Devices with limits for use with index registers

Device	Meaning	Application Example
T	• Only Z0 and Z1 can be used for timer contacts and coils	
C	• Only Z0 and Z1 can be used for counter contacts and coils	

REMARKS

(1) *: SFC transfer devices and SFC block devices are devices for SFC use.
See the QnACPU Programming Manual (SFC) for information on how to use these devices.

(2) For timer and counter present values, there are no limits on index register numbers used.



Set value of timer
(Index qualification not possible)

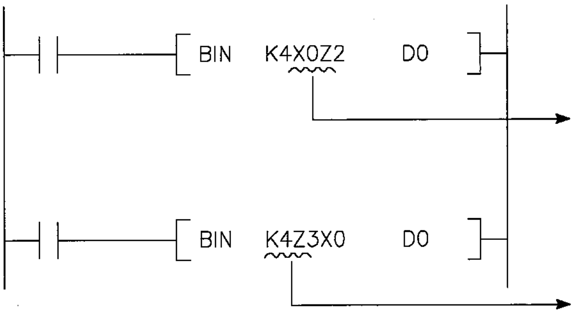
Present value of timer

Set value of counter
(Index qualification not possible)

Present value of counter

(c) Other

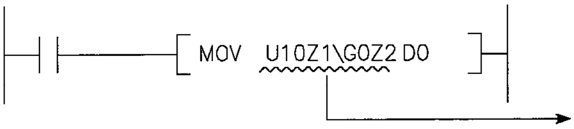
1) Bit data
Device numbers can be index qualified when performing digit designation.
However, index qualification is not possible by digit designation.



Setting that enables device number index qualification If Z2=3, then X(0+3)=X3.

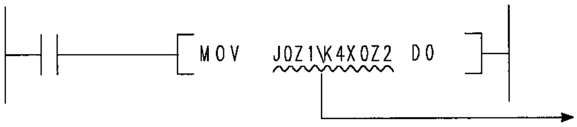
Setting that cannot enable digit designation index qualification

2) Both I/O numbers and buffer memories can be index qualified with special function module devices.



If Z1=2 and Z2=8, then U(10+2)\G(0+8)=U12\X8.

3) Both network numbers and device numbers can be index qualified with link direct devices.



If Z1=2 and Z2=8, then J(0+2)\X(0+8)=J2\X8.

REMARKS

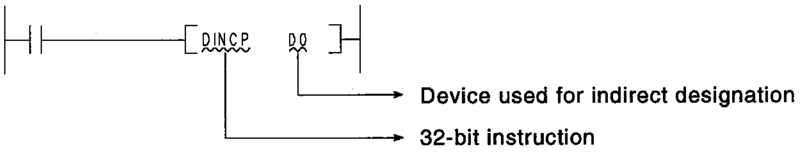
(1) See QnACPU Programming Manual (Fundamentals), Section 4.5, for information on special function modules.

(2) See QnACPU Programming Manual (Fundamentals), Section 4.4, for information on link direct devices.

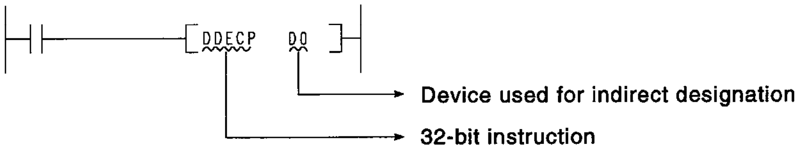
(4) Cautions

When designating an address indirectly, use two words.
Therefore, if indirect designation is used instead of index qualification, execute addition/subtraction of 32-bit data.
The circuit block shown below is used for addition/subtraction of the address, stored in D1 and D0, of the device that is designated indirectly.

[To add "1" to the address of the device to be designated indirectly]



[To subtract "1" from the address of the device to be designated indirectly]



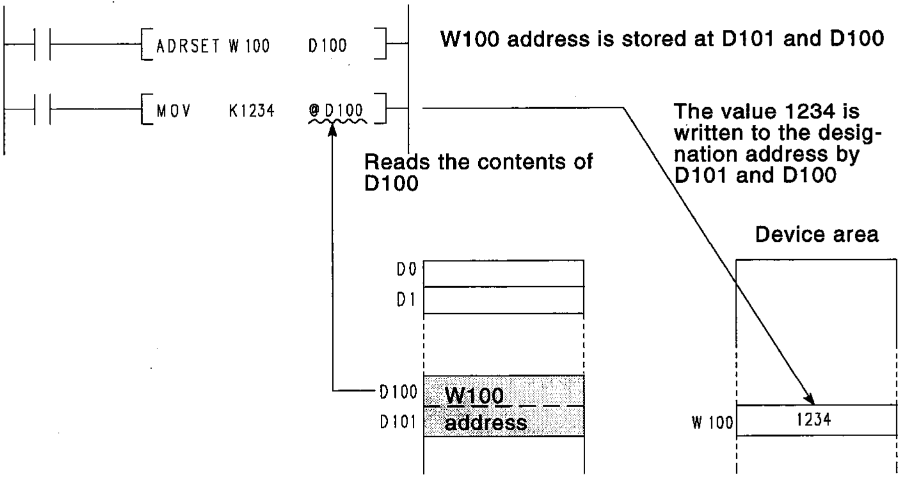
3.4 Indirect Designation

(1) Indirect Designation

(a) Indirect designation is a way of using a word device to designate a device address that will be used in a sequence program.
This method can be used when the index register is insufficient.

(b) The device which designates the designated device address is designated by "@+(word device number)".
For example, designation of @D100 will make the contents of D100 D101 the device address.

(c) The address of the device performing indirect designation can be confirmed with the ADRSET instruction.



(2) Devices Capable of Indirect Designation
A list of QnACPU devices which are capable of indirect designation is included in Table 3.3

Table 3.3 List of Devices Capable of Indirect Designation

Device Type		Capable/ Incapable of Indirect Designation	Example of Indirect Designation
Internal user devices	Bit devices[*1]	Incapable	——
	Word devices[*1]	Capable	• @D100 • @D100Z2[*2]
Link direct devices	Bit devices[*1]	Incapable	——
	Word devices[*1]	Capable	• @J1\W10 • @J1Z1\W10Z2[*2]
Special direct devices		Capable	• @U10\G0 • @U10Z1\G0Z2[*2]
Index register		Incapable	——
File register		Capable	• @R0, @ZR20000 • @R0Z1, @ZR20000Z1[*2]
Nesting		Incapable	——
Pointer			——
Constants			——
Other	SFC block devices		——
	Devices below SFC		
	Network No.		
	I/O No.		

REMARKS

(1) *1: See QnACPU Programming Manual (Fundamentals) for more information on device names.

(2) *2: Indicates index qualification by index register

3.5 Subset Processing

Subset processing is used to place limits on bit devices used by basic instructions and application instructions in order to increase processing speed.
However, the instruction symbol does not change.
To shorten scans, run instructions under the conditions indicated below.

(1) Conditions which each device must meet for subset processing

(a) When using word data

Device	Condition
Bit device	• Designates a bit device number in a factor of 16 • Only K4 can be designated for digit designation • Does not conduct index qualification
Word device	• No limitations
Constants	• No limitations

(b) When using double word data

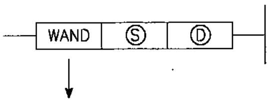
Device	Condition
Bit device	• Designates a bit device number in a factor of 32 • Only K8 can be designated for digit designation • Does not conduct index qualification
Word device	• No limitations
Constants	• No limitations

(2) Instructions for which subset processing can be used

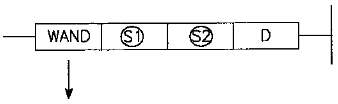
Types of Instructions	Instruction Symbols
Comparison instructions	• =, <>, <, <=, >, >=, D=, D<>, D<, D<=, D>, D>=
Basic arithmetic operations (addition, subtraction, multiplication, and division)	• +, -, *, /, INC, DEC, D+, D-, D*, D/, DINC, DDEC • B+, B-, B*, B/
Data conversion instructions	• BCD, BIN, DBCD, DBIN
Data transfer instruction	• MOV, DMOV, CML, DCML, XCH, DXCH • FMOV, BMOV
Program branch instruction	• CJ, SCJ, JMP
Logic operations *	• WAND, DAND, WOR, DOR, WXOR, DXOR, WXNOR, DXNOR
Rotation instruction	• RCL, DRCL, RCR, DRCR, ROL, DROL, ROR, DROR
Shift instructions	• SFL, DSFL, SFR, DSFR
Data processing instructions	• SUM, SEG
Structured program instructions	• FOR, CALL

REMARK

Subset processing for the logic operation instructions WAND, WOR, WXOR, and WXNOR is conducted when there are 2 devices.



Subset processing possible



Subset processing not possible

3.6 Cautions on Programming (Operation Errors)

Operation errors are returned in the following cases when executing basic instructions and application instructions with QnACPU:

- An error listed on the explanatory page for the individual instruction occurred.
- No special function module was installed at the designated I/O No. position when using the buffer register.
- The relevant network does not exist when using a link device.
- No network module is installed at the designated I/O No. when using a link device.

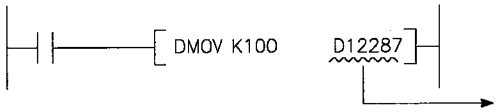
POINT

When a file register setting has been made but no memory card has been installed, or when no file register setting has been made, no error will be returned even if an attempt is made to write to the file register. However, "FFFF_H" will be stored if an attempt is made to read from the file register at which this write operation was attempted.

(1) Device range check

Device range checks for the devices used by basic instructions and application instructions in QnACPU are as indicated below:

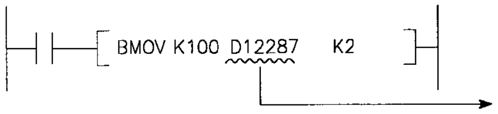
(a) No device range check is made for instructions dealing with fixed-length devices (MOV, DMOV, etc.).
In cases where the corresponding device range is exceeded, data is written to other devices. *
For example, in a case where the data register has been allocated 12 k points, there will be no error even if it exceeds D12287.



D12287 and D12288 have been indicated here, but because D12288 does not exist, the contents of some other device will be destroyed.

Device range checks are not conducted also in cases where index qualification is being performed.

(b) Device range checks are conducted for instructions dealing with variable-length devices (BMOV, FMOV, and others which designate transfer numbers).
In cases where the corresponding device range has been exceeded, an operation error will be returned.
For example, in a case where the data register has been allocated 12 k points, there will be an error if it exceeds D12287.

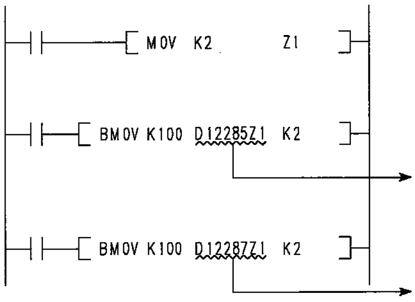


D12287 and D12288 have been indicated here, but because D12288 does not exist, an operation error is returned.

REMARK

*: See Section 3.4 (3) for the internal user device allocation order.

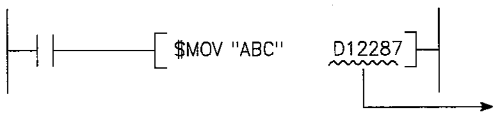
Device range checks are also conducted when index qualification is performed.
However, if index qualification has been conducted, there will be no error returned if the initial device number exceeds the relevant device range.



D12287 and D12288 have been indicated here, but because D12288 does not exist, an operation error is returned.

Because the initial device number is D12289 and that exceeds the device range, the initial device number is made W0, the operation is conducted, and no error is returned.

(c) Because all character string data is of variable length, device range checks are performed.
In cases where the corresponding device range has been exceeded, an operation error will be returned.
For example, in a case where the data register has been allocated 12 k points, there will be an error if it exceeds D12287.



D12287 and D12288 have been indicated here, but because D12288 does not exist, an operation error is returned.

Note that an operation error does not occur even if the head device number exceeds the device range as the result of index qualification.

(d) Device range checks are conducted when index qualification is performed by direct access output (DY).

(2) Device data check

Device data checks for the devices used by basic instructions and application instructions in QnACPU are as indicated below:

(a) When using BIN data

- No error is returned even if the operation results in overflow or underflow.
  The carry flag does not go on at such times, either.

(b) When using BCD data

1) Each digit is check for BCD value (0/ to 9).
   An operation error is returned if individual digits are outside the 0 to 9 (A to F) range.
2) No error is returned even if the operation results in overflow or underflow.
   The carry flag does not go on at such times, either.

(c) When using floating decimal point data

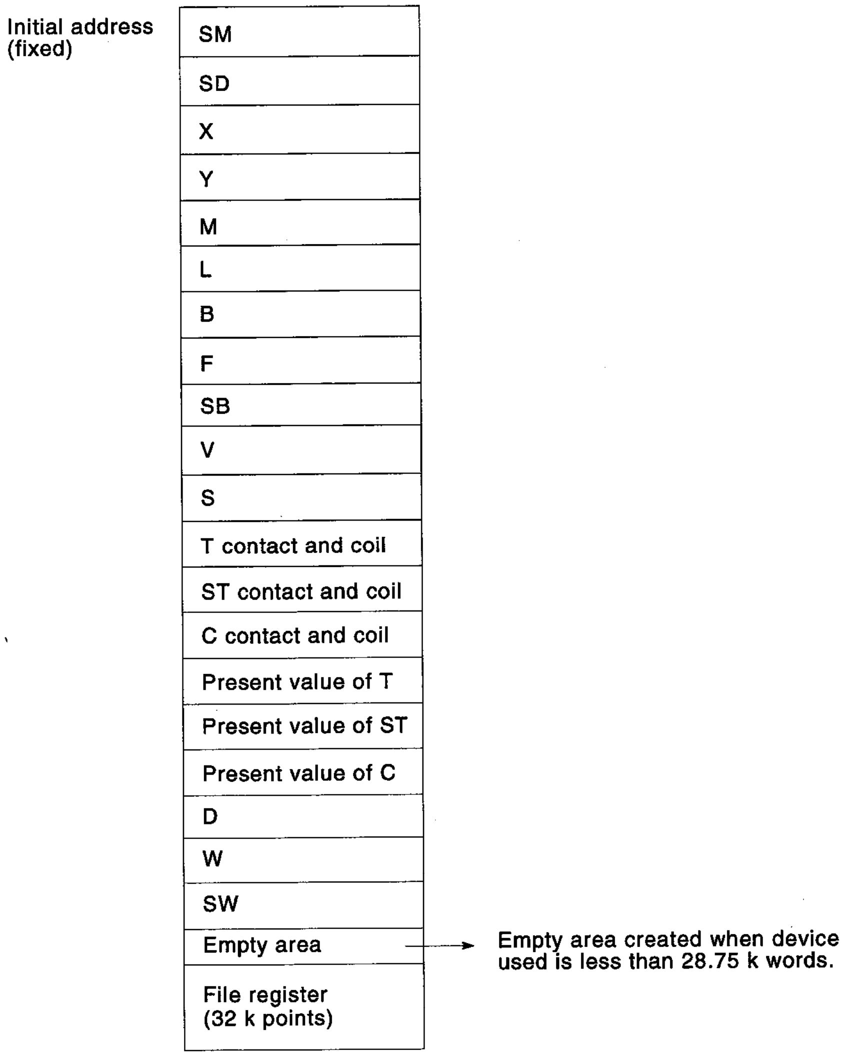
Operation errors are returned in the following cases:

- When value of floating decimal point data is 0
- When the absolute value of the floating decimal point data is 1.0×2^{-127} or lower
- When absolute value of floating decimal point data is 1.0×2^{128} or higher

(d) When using character string data

No data check is conducted.

(3) If internal user device allocation is changed by parameter device allocation, such allocations are made in the device order indicated below:
If the allocation of the device used is less than 28.75 k words, the area following the device used will be empty.



REMARK

See the QnACPU Programming Manual (Fundamentals) for more information on changing internal user allocations.

3.7 Conditions for Execution of Instructions

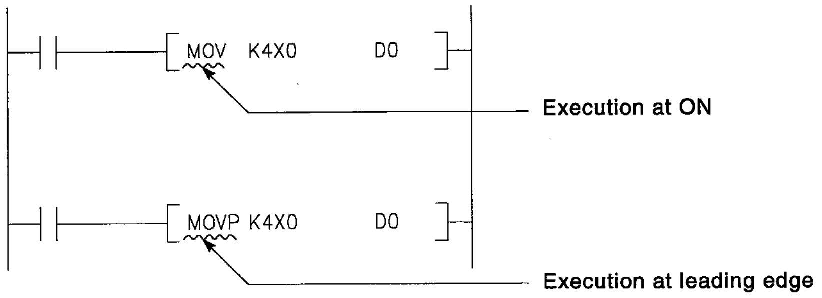
The following four types of execution conditions exist for the execution of QnACPU sequence instructions, basic instructions, and application instructions:

- Non-conditional execution
  ..... Instructions executed without regard to the ON/OFF status of the device
  Example: LD X0, OUT Y10
- Executed at ON
  ..... Instructions executed while input condition is ON
  Example: MOV instruction, FROM instruction
- Executed at leading edge
  ..... Instructions executed only at the leading edge of the input condition (when it goes from OFF to ON)
  Example: PLS instruction; MOVP instruction
- Executed at trailing edge
  ..... Instructions executed only at the trailing edge of the input condition (when it goes from ON to OFF)
  Example: PLF instruction

For coil or equivalent basic instructions or application instructions, where the same instruction can be designated for either execution at ON or leading edge execution, a "P" is added after the instruction name to specify the condition for execution.

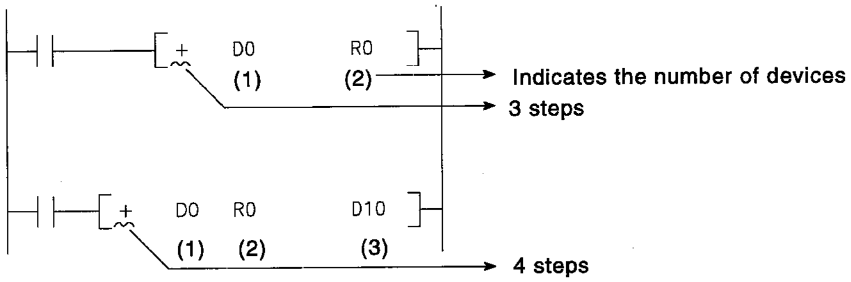
- Instruction to be executed at ON $\boxed{\text{Instruction name}}$
- Instruction to be executed at leading edge $\boxed{\text{Instruction name}}$+ P

Execution at ON and execution at leading edge for the MOV instruction are designated as follow:



3.8 Counting Step Number

The number of steps in QnACPU instructions, basic instructions, and application instructions differs depending on whether indirect setting of the device used is possible or not.
The basic number of steps for basic instructions and application instructions is calculated by adding the device number and 1.
For example, the "+ instruction" would be calculated as follows:



(1) Conditions for increasing the number of steps
The number of steps is increased over the number of basic steps in cases where a device is used that is designated indirectly or for which the number of steps is increased.

(a) When device is designated indirectly

In cases where indirect designation is done by @[ ], the number of steps is increased 1 step over the number of basic steps.
For example, when a 3-step MOV instruction is designated indirectly (example: MOV K4X0 @D0), one step is added and the instruction becomes 4 steps.

(b) Devices where number of steps increases

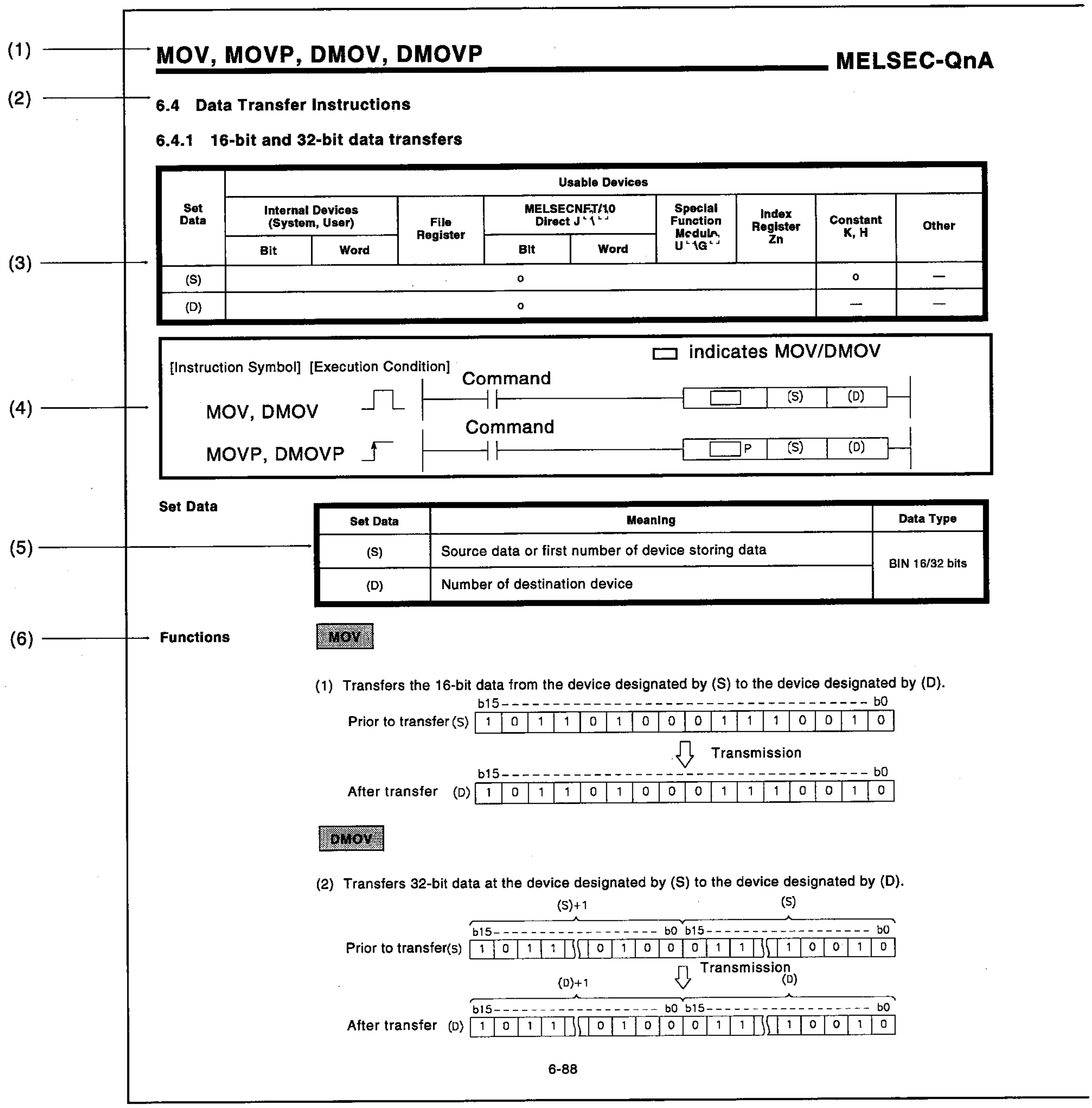
Devices Where Number of Steps Increases	Added Steps	Example
Special-function module device	1	MOV U4\G10 D0
Link direct devices		MOV J3\B20 D0
Serial number access file registers		MOV ZR123 D0
32-bit constants		DMOV K123 D0
Real number constant		EMOV E0.1 D0
Character string constant	For odd numbers: number of characters/2 For even numbers (number of characters + 1)/2-1	$MOV "123" D0

(c) In cases where the conditions described in (a) and (b) above overlap, the number of steps becomes a culmination of the two.
For example, if MOV U1\G10 ZR123 has been designated, 1 step is added for buffer register designation and 1 step is added for serial number access file register designation, making a total of 2 steps added.

MEMO

4. HOW TO READ INSTRUCTIONS

The description of instructions that are contained in the following chapters are presented in the following format.



MOV, MOVP, DMOV, DMOVP MELSEC-QnA

6.4 Data Transfer Instructions

6.4.1 16-bit and 32-bit data transfers

Set Data	Usable Devices							
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

□ indicates MOV/DMOV

[Instruction Symbol] [Execution Condition]

MOV, DMOV — Command — [□ (S) (D)]

MOVP, DMOVP — Command — [□P (S) (D)]

Set Data

Set Data	Meaning	Data Type
(S)	Source data or first number of device storing data	BIN 16/32 bits
(D)	Number of destination device	

Functions

MOV

(1) Transfers the 16-bit data from the device designated by (S) to the device designated by (D).

	b15															b0
Prior to transfer (S)	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	0
After transfer (D)	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	0

Transmission

DMOV

(2) Transfers 32-bit data at the device designated by (S) to the device designated by (D).

(S)+1 (S)

b15 b0 b15 b0

Prior to transfer(S) 1 0 1 1 … 0 1 0 0 0 1 1 … 1 0 0 1 0

Transmission

(D)+1 (D)

b15 b0 b15 b0

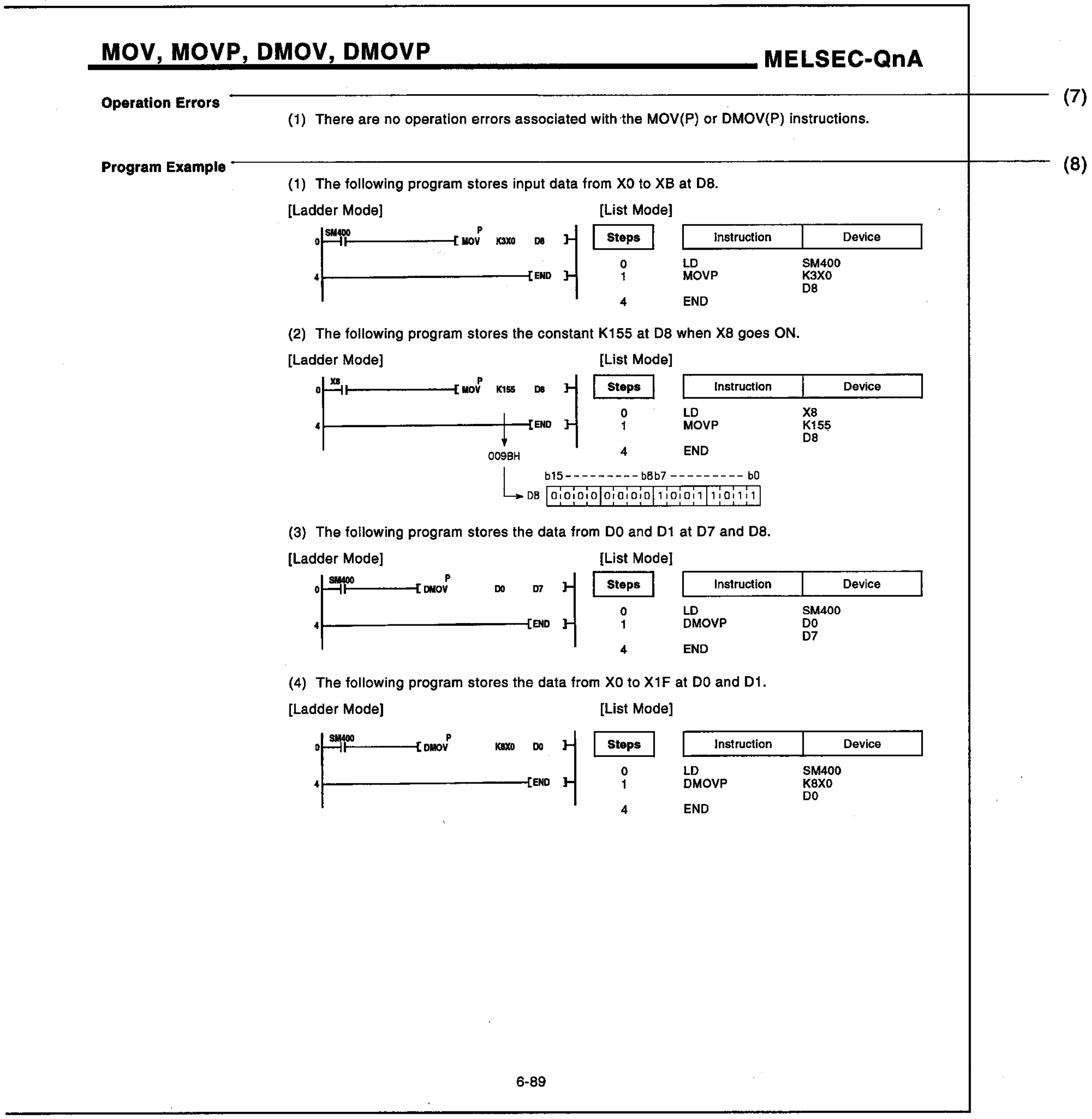
After transfer (D) 1 0 1 1 … 0 1 0 0 0 1 1 … 1 0 0 1 0

6-88

(1) Code used to write instruction (instruction symbol).
(2) Section number and general category of instructions being discussed.
(3) Devices which can be used by the instruction in question are indicated with circle. The types of devices that can be used are as indicated below:

*4 Device Type	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant *1	Other *1
	Bit	Word		Bit	Word				
*4 Usable devices	X, Y, M, L, SM, F, V, B, SB, FX, FY *2	T, ST, C, *3 D, W, SD, SW, FD	R, ZR	J□\X J□\Y J□\B J□\SB	J□\W J□\SW	U□\G□	Z	Decimal constants Hexadecimal constants Real number constant Character string constant	P, I, J, U, DX, DY, N, BL, TR, BL\S

*1: Devices which can be set are recorded in the "Constant" and the "Other" columns.
*2: FX and FY can be used only for bit data, and FD only for word data.



MOV, MOVP, DMOV, DMOVP

MELSEC-QnA

Operation Errors (7)

(1) There are no operation errors associated with the MOV(P) or DMOV(P) instructions.

Program Example (8)

(1) The following program stores input data from X0 to XB at D8.

[Ladder Mode]

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	MOVP	K3X0 D8
4	END	

(2) The following program stores the constant K155 at D8 when X8 goes ON.

[Ladder Mode]

[List Mode]

Steps	Instruction	Device
0	LD	X8
1	MOVP	K155 D8
4	END	

(3) The following program stores the data from D0 and D1 at D7 and D8.

[Ladder Mode]

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DMOVP	D0 D7
4	END	

(4) The following program stores the data from X0 to X1F at D0 and D1.

[Ladder Mode]

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DMOVP	K8X0 D0
4	END	

6-89

*3 When T, ST and C are used for other than the instructions below, only word data can be used. (Bit data cannot be used .)
[Instructions that can be used with bit data]
LD, LDI, AND, ANI, OR, ORI, LDP, LDF, ANDP, ANDF, ORP, ORF, OUT, RST

*4 See QnA Programming Manual for the explanation of each device.

(4) Indicates ladder mode expressions and execution conditions for instructions.

Execution Condition	**Non-conditional Execution**	**Executed while ON**	**Executed One Time at ON**	**Executed while OFF**	**Executed One Time at OFF**
Code recorded on description page	No symbol recorded	⊓	↑	⊔	↓

(5) Discusses the data set for each instruction and the data type.

Data Type	**Meaning**
Bit	Bit data or first number in bit data
BIN 16 bits	BIN 16-bit data or first number in word device
BIN 32 bits	BIN 32-bit data or first number in double word device
BCD 4 digits	4-digit BCD data
BCD 8 digits	8-digit BCD data
Real number	Floating decimal point data
Character string	Character string data
Device name	Dvice name data

(6) Indicates the function of the instruction.

(7) Indicates conditions under which error is returned, and error number.
See Section 3.6 for errors not included here.

(8) Indicates both ladder and list for simple program example.
Also indicates the types of individual devices used when the program is executed.

5. QNA SEQUENCE INSTRUCTIONS

Sequence instructions include instructions for relay control ladders and the like. They are divided into the following categories:

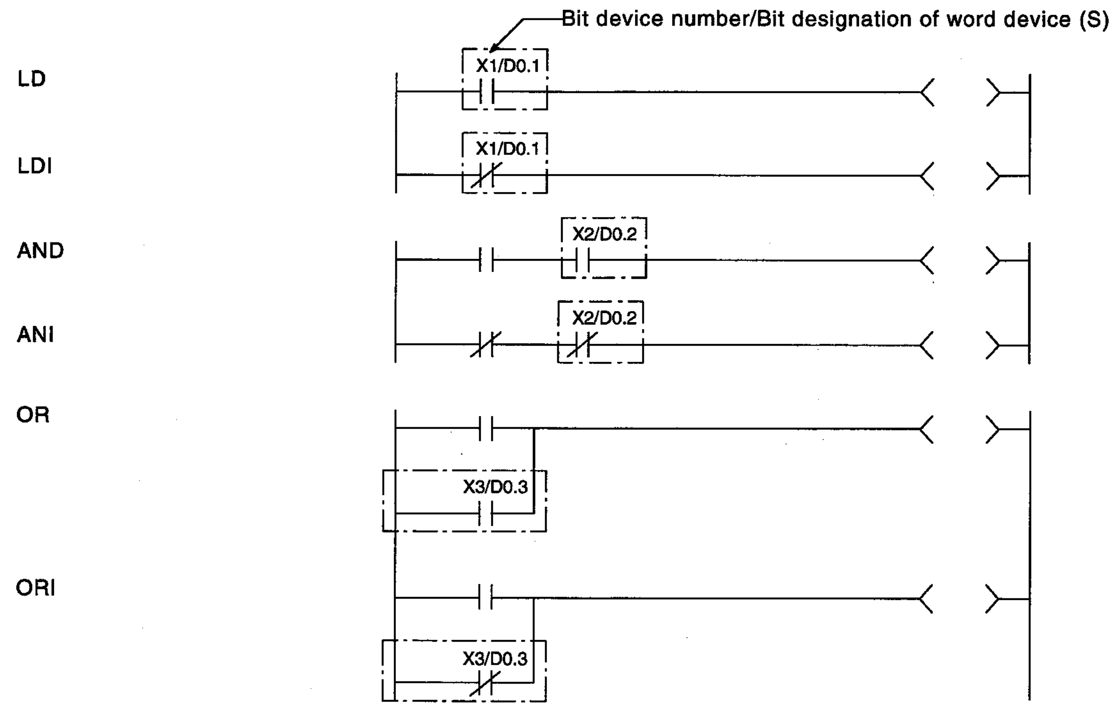
Instruction	Meaning	Reference
Contact instruction	Operation start, series connection, parallel connection	Section 5.1
Connection Instruction	Ladder block connection, creation of pulses from operation results, store/read operation results	Section 5.2
Output instruction	Bit device output, pulse output, output reversal	Section 5.3
Shift instruction	Bit device shift	Section 5.4
Master control instruction	Master control	Section 5.5
Termination instruction	Program termination	Section 5.6
Other instructions	Program stop, instructions such as no operation which do not fit in the above categories	Section 5.7

5.1 Contact Instructions

5.1.1 Operation start, series connection, parallel connection

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				DX, BL
(S)	o						—		o

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	Devices used as connections	Bit

Functions

LD, LDI

(1) LD is the A contact operation start instruction, and LDI is the B contact operation start instruction. They read ON/OFF information from the designated device (if a word device bit has been designated, this becomes the 1/0 status of the designated bit), and use that as an operation result.

AND, ANI

(1) AND is the A contact series connection instruction, and ANI is the B contact series connection instruction. They read the ON/OFF data of the designated bit device (if a bit designation has been made for a word device, the 1/0 status of the designated bit is read), perform an AND operation on that data and the operation result to that point, and take this value as the operation result.

(2) There are no restrictions on the use of AND or ANI, but the following applies with a peripheral device used in the ladder mode:

(a) Write ...... When AND and ANI are connected in series, a ladder with up to 21 stages can be generated.

(b) Read ...... When AND and ANI are connected in series, a ladder with up to 24 stages can be displayed.
If the number exceeds 24 stages, up to 24 will be displayed.

OR, ORI

(1) OR is the A contact single parallel connection instruction, and ORI is the B contact single parallel connection instruction. They read ON/OFF information from the designated device (if a word device bit has been designated, this becomes the 1/0 status of the designated bit), and perform an OR operation with the operation results to that point, and use the resulting value as the operation result.

(2) There are no limits on the use of OR or ORI, but the following applies with a peripheral device used in the ladder mode.

(a) Write ...... OR and ORI can be used to create connections of up to 23 ladders.

(b) Read ...... Up to 23 ladders connected with OR or ORI can be displayed.
The 24th or subsequent ladders cannot be displayed properly.

REMARK

Word device bit designations are made in hexadecimal.
Bit b11 of D0 would be D0.0B.
See Section 3.2.1 for more information on word device bit designation.

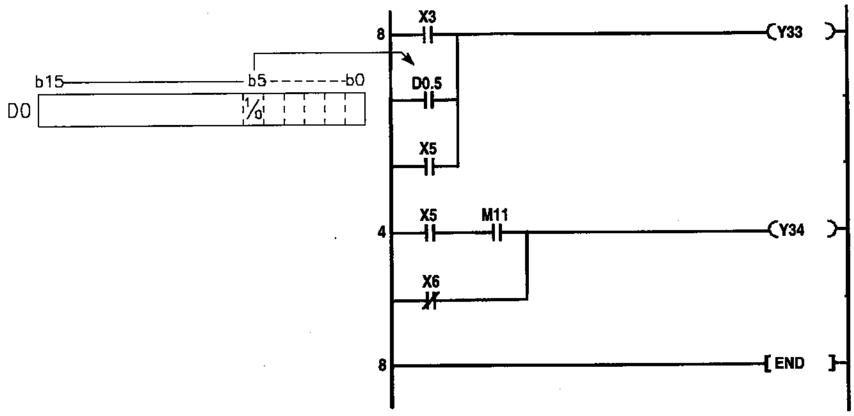
Operation Errors

(1) There are no operation errors with LD, LDI, AND, ANI, OR, or ORI instructions.

Program Example

(1) A program using LD, AND, OR, and ORI instructions.

[Ladder Mode]

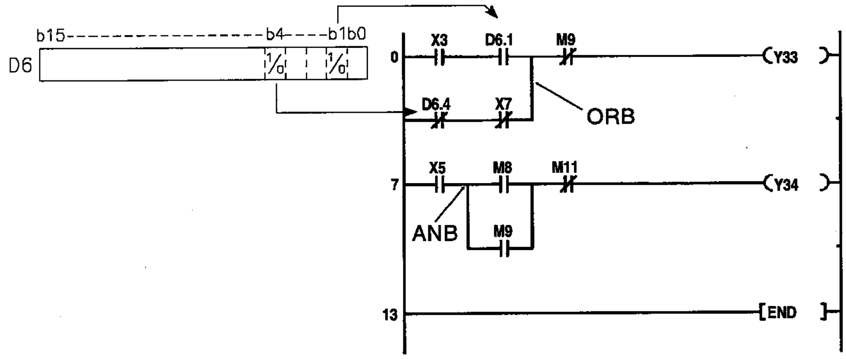


[List Mode]

Steps	Instruction	Device
0	LD	X3
1	OR	D0.5.... Word device bit designation
2	OR	X5
3	OUT	Y33
4	LD	X5
5	AND	M11
6	ORI	X6
7	OUT	Y34
8	END	

(2) A program linking contact points established through the use of ANB and ORB instructions.

[Ladder Mode]

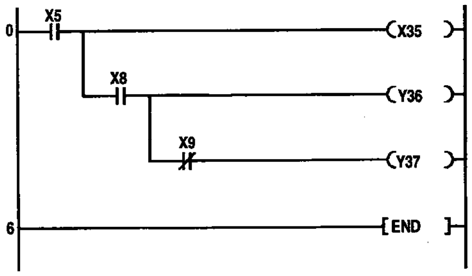


[List Mode]

Steps	Instruction	Device
0	LD	X3
1	AND	D6.1 .... Word device bit designation
2	LDI	D6.4
3	ANI	X7
4	ORB	
5	ANI	M9
6	OUT	Y33
7	LD	X5
8	LD	M8
9	OR	M9
10	ANB	
11	ANI	M11
12	OUT	Y34
13	END	

(3) A parallel program with OUT instruction

[Ladder Mode]



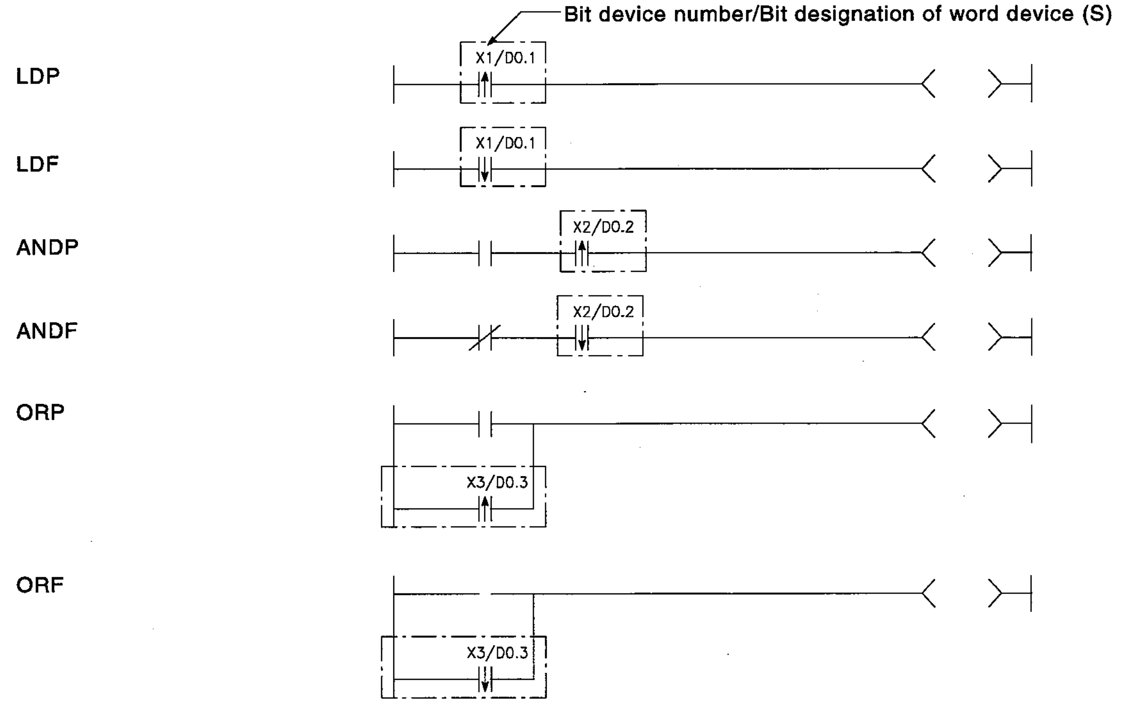
[List Mode]

Steps	Instruction	Device
0	LD	X5
1	OUT	X35
2	AND	X8
3	OUT	Y36
4	ANI	X9
5	OUT	Y37
6	END	

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

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				DX
(S)	○						—		○

[Instruction Symbol] [Execution Condition]



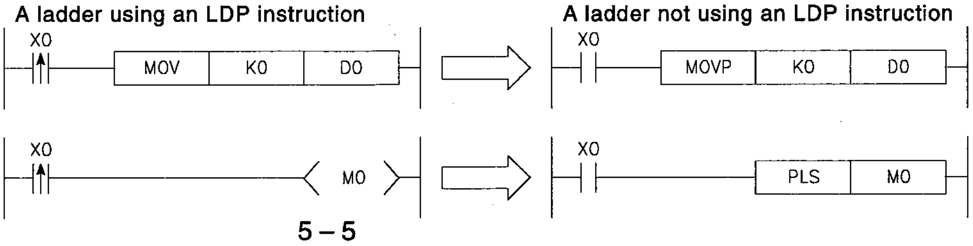
Set Data

Set Data	Meaning	Data Type
(S)	Devices used as contacts	Bit

Functions

LDP, LDF

(1) LDP is the leading edge pulse operation start instruction, and is ON only at the leading edge of the designated bit device (when it goes from OFF to ON).
If a word device has been designated, it is ON only when the designated bit changes from 0 to 1.
In cases where there is only an LDP instruction, it acts identically to instructions for the creation of a pulse that are executed during ON (□□□P).



(2) LDF is the trailing edge pulse operation start instruction, and is ON only at the trailing edge of the designated bit device (when it goes from ON to OFF).
If a word device has been designated, it is ON only when the designated bit changes from 1 to 0.

ANDP, ANDF

(1) ANDP is a leading edge pulse series connection instruction, and ANDF is a trailing edge pulse series connection instruction. They perform an AND operation with the operation result to that point, and take the resulting value as the operation result.
The ON/OFF data used by ANDP and ANDF are indicated in the table below:

Devices Designated by ANDP		ANDP State	Devices Designated by ANDF		ANDF State
Bit Device	Word Device Bit Designation		Bit Device	Word Device Bit Designation	
OFF→ON	0→1	ON	OFF→ON	0→1	OFF
OFF	0	OFF	OFF	0	
ON	1		ON	1	
ON→OFF	1→0		ON→OFF	1→0	ON

ORP, ORF

(1) ORP is a leading edge pulse parallel connection instruction, and ORF is a trailing edge pulse parallel connection instruction. They perform an OR operation with the operation result to that point, and take the resulting value as the operation result.

Devices Designated by ORP		ORP State	Devices Designated by ORF		ORF State
Bit Device	Word Device Bit Designation		Bit Device	Word Device Bit Designation	
OFF→ON	0→1	ON	OFF→ON	0→1	OFF
OFF	0	OFF	OFF	0	
ON	1		ON	1	
ON→OFF	1→0		ON→OFF	1→0	ON

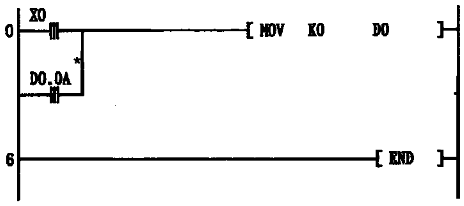
Operation Errors

(1) There are no operation errors with LDP, LDF, ANDP, ANDF, ORP, or ORF instructions.

Program Example

(1) The following program executes the MOV instruction at input X0, or at the leading edge of b11 (bit 11) of data register D0:

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LDP	X0
2	ORP	D0.0A
4	MOV	K0 D0
6	END	

REMARK

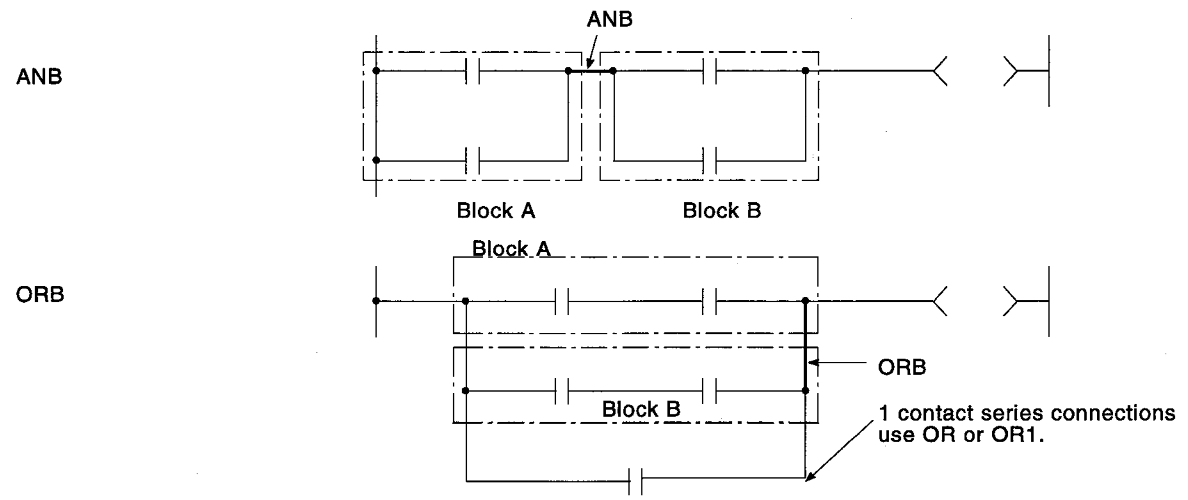
*: Word device bit designations are performed in hexadecimal.
Bit b11 of D0 would be D0.0A.

5.2 Connection Instructions

5.2.1 Ladder block series connections and parallel connections

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]



Functions

ANB

(1) Performs an AND operation on block A and block B, and takes the resulting value as the operation result.

(2) The symbol for ANB is not the contact symbol, but rather is the connection symbol.

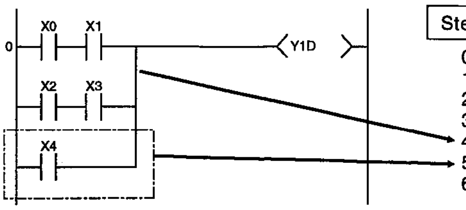
(3) When programming in the list mode, up to 15 ANB instructions (16 blocks) can be written consecutively.

ORB

(1) Conducts an OR operation on Block A and Block B, and takes the resulting value as the operation result.

(2) ORB is used to perform parallel connections for ladder blocks with two or more contacts.
For ladder blocks with only one contact, use OR or ORI; there is no need for ORB in such cases.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	AND	X1
2	LD	X2
3	AND	X3
4	ORB	
5	OR	X4
6	OUT	X10

(3) The ORB symbol is not the contact symbol, but rather is the connection symbol.

(4) When programming in the list mode, it is possible to use up to 15 ORB instructions successively (16 blocks).

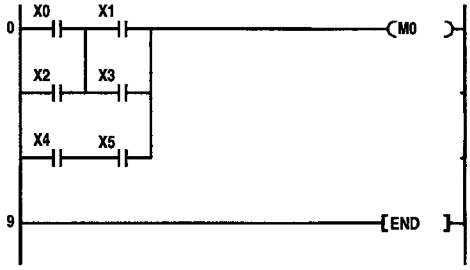
Operation Errors

(1) There are no operation errors associated with ANB or ORB instructions.

Program Example

(1) A program using ANB and ORB instructions

[Ladder Mode]



[List Mode]

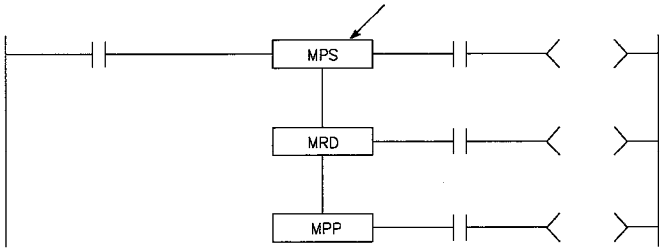
Steps	Instruction	Device
0	LD	X0
1	OR	X2
2	LD	X1
3	OR	X3
4	ANB	
5	LD	X4
6	AND	X5
7	ORB	
8	OUT	M0
9	END	

5.2.2 Operation results push, read, pop

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]

MPS, MRD, and MPP are not displayed as a part of the ladder display.



Functions

MPS

(1) Stores in memory the operation result (ON or OFF) immediately prior to the MPS instruction.

(2) Up to 16 MPS instructions can be used successively.
However, only up to 11 can be created in the ladder mode.
If an MPP instruction is used during this process, the number of uses calculated for the MPS instruction will be decremented by one.

MRD

(1) Reads the operation result stored for the MPS instruction, and uses that result to perform the operation in the next step.

MPP

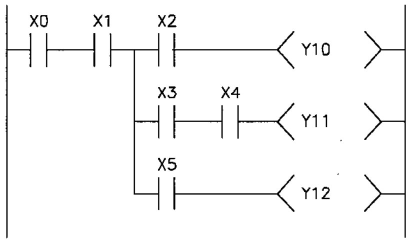
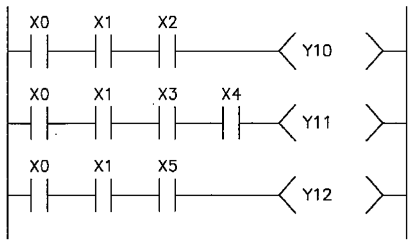
(1) Reads the operation result stored for the MPS instruction, and uses that result to perform the operation in the next step.

(2) Clears the operation results stored by the MPS instruction.

(3) Subtracts 1 from the number of MPS instruction times of use.

POINTS

(1) The following shows ladders both using and not using the MPS, MRD, and MPP instructions.

Ladder Using the MPS, MRD and MPP Instruction.	Ladder not Using MPS, MRD, and MPP Instructions.
	

(2) The number of uses for the MPS and MPP instructions should be equal.
If they are different, the peripheral device will not be able to correctly display the ladder in the ladder mode.

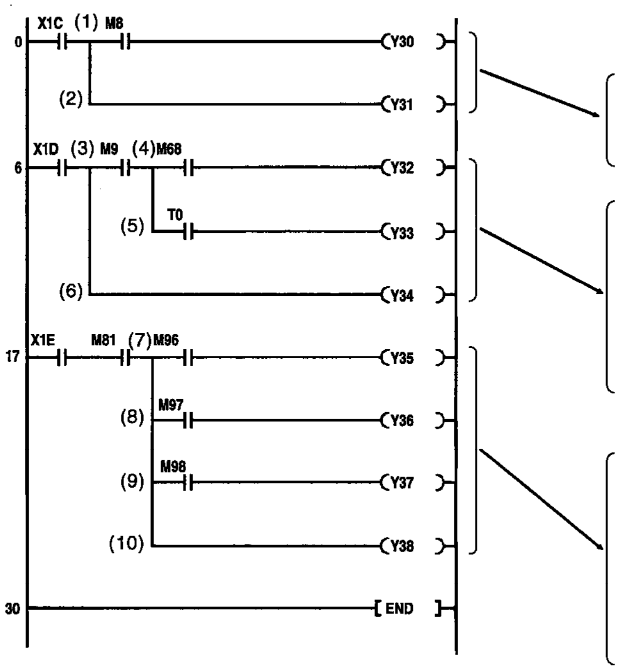
Operation Errors

(1) There are no errors associated with the MPS, MRD, or MPP instructions.

Program Example

(1) A program using the MPS, MRD, and MPP instructions.

[Ladder Mode]

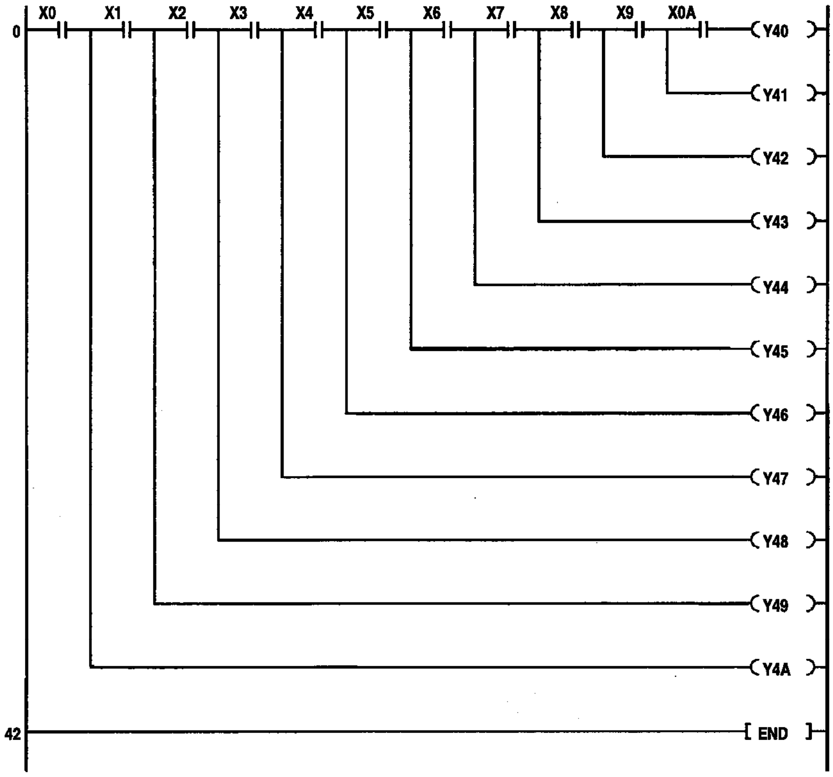


[List Mode]

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

(2) A program using MPS and MPP instructions successively.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MPS	
2	AND	X1
3	MPS	
4	AND	X2
5	MPS	
6	AND	X3
7	MPS	
8	AND	X4
9	MPS	
10	AND	X5
11	MPS	
12	AND	X6
13	MPS	
14	AND	X7
15	MPS	
16	AND	X8
17	MPS	
18	AND	X9
19	MPS	
20	AND	X0A
21	OUT	Y40
22	MPP	
23	OUT	Y41
24	MPP	
25	OUT	Y42
26	MPP	
27	OUT	Y43
28	MPP	
29	OUT	Y44
30	MPP	
31	OUT	Y45
32	MPP	
33	OUT	Y46
34	MPP	
35	OUT	Y47
36	MPP	
37	OUT	Y48
38	MPP	
39	OUT	Y49
40	MPP	
41	OUT	Y4A
42	END	

5.2.3 Operation results inversion

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
—	—								

[Instruction Symbol] [Execution Condition]

INV

Functions

Inverts the operation result immediately prior to the INV instruction.

Operation Result Immediately Prior to the INV Instruction.	Operation Result Following the Execution of the INV Instruction.
OFF	ON
ON	OFF

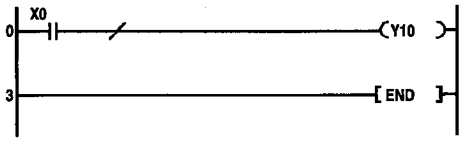
Operation Errors

(1) There are no operation errors associated with the INV instruction.

Program Example

(1) A program which inverts the X0 ON/OFF data, and outputs from Y10

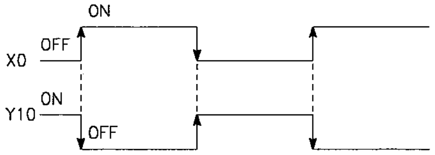
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	INV	
2	OUT	Y10
3	END	

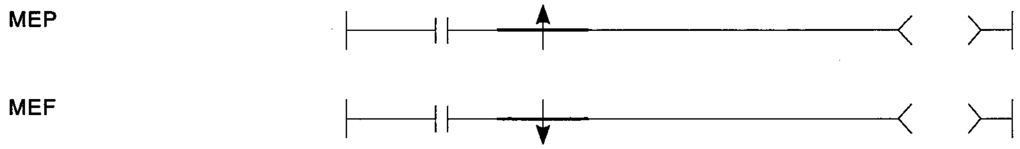
[Timing Chart]



5.2.4 Operation result pulse conversion

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
—	—								

[Instruction Symbol] [Execution Condition]



Functions

MEP

(1) If operation results up to MEP instruction are leading edge (from OFF to ON), goes ON (continuity state).
If operation results up to MEP instruction are anything other than leading edge, goes OFF (non-continuity state).

(2) Use of the MEP instruction simplifies pulse conversion processing when multiple contacts are connected in series.

MEF

(1) If operation results up to MEF instruction are trailing edge (from ON to OFF), goes ON (continuity state).
If operation results up to MEF instruction are anything other than trailing edge, goes OFF (non-continuity state).

(2) Use of the MEF instruction simplifies pulse conversion processing when multiple contacts are connected in series.

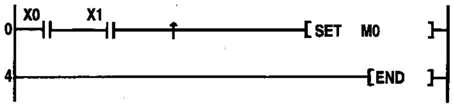
Operation Errors

(1) There are no operation errors associated with the MEP or MEF instructions.

Program Example

(1) A program which performs pulse conversion on the operation results of X0 and X1:

[Ladder Mode]



[List Mode]

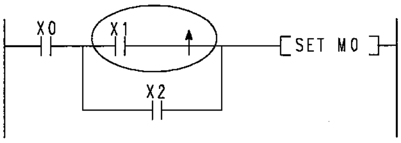
Steps	Instruction	Device
0	LD	X0
1	AND	X1
2	MEP	
3	SET	M0
4	END	

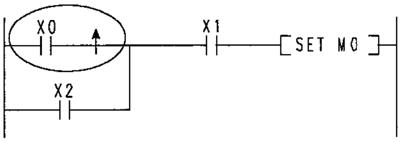
POINTS

(1) The MEP and MEF instructions will occasionally not function properly when pulse conversion is conducted for a contact that has been indexed by a sub-routine program or by the FOR to NEXT instructions.
If pulse conversion is to be conducted for a contact that has been indexed by a sub-routine program or by the FOR to NEXT instructions, use the EGP/EGF instructions.
(See Section 5.2.5)

(2) Because the MEP and MEF instructions operate with the operation results immediately prior to the MEP and MEF instructions, the AND instruction (see Section 5.1.1) should be used at the same position. The MEP and MEF instructions cannot be used at the LD or OR position.

(3) MEP and MEF instructions cannot be used at the circuit block positions shown below.

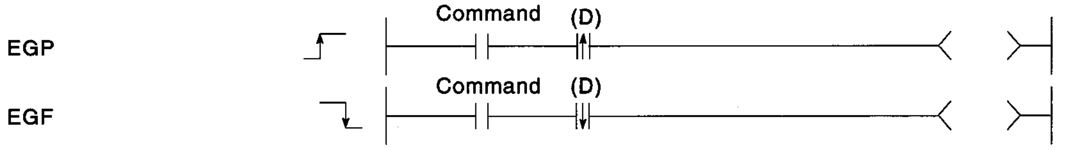




5.2.5 Pulse conversion of edge relay operation results

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
(D)	o (Only V)	—							

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(D)	Edge relay number where operation results are stored	Bit

Functions

EGP

(1) Operation results up to the EGP instruction are stored in memory by the edge relay (V).

(2) Goes ON (continuity state) at the leading edge (OFF to ON) of the operation result up to the EGP instruction.
If the operation result up to the EGP instruction is other than a leading edge (i.e., from ON to ON, ON to OFF, or OFF to OFF), it goes OFF (non-continuity state).

(3) The EGP instruction is used for subroutine programs, and for conducting pulse operations for programs designated by index qualification between FOR and NEXT instructions.

(4) The EGP instruction can be used like an AND instruction.

EGF

(1) Operation results up to the EGF instruction are stored in memory by the edge relay (V).

(2) Goes ON at the trailing edge (from ON to OFF) of the operation result up to the EGF instruction.
If the operation result up to the EGP instruction is other than a trailing edge (i.e., from OFF to ON, ON to ON, or OFF to OFF), it goes OFF (non-continuity state).

(3) The EGF instruction is used for subroutine programs, and for conducting pulse operations for programs designated by index qualification between FOR and NEXT instructions.

(4) The EGP instruction can be used like an AND instruction.

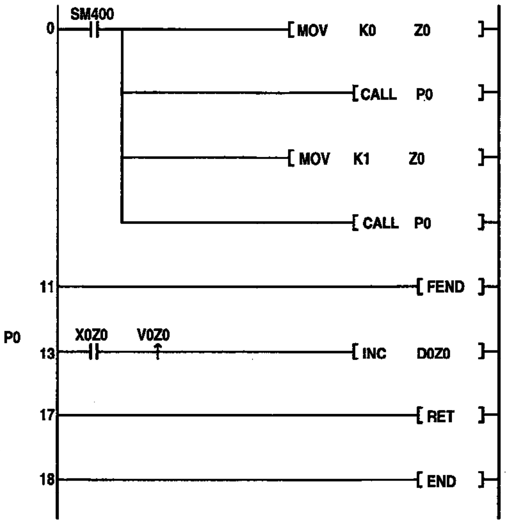
Operation Errors

(1) There are no operation errors associated with the EGP or EGF instructions.

Program Example

(1) A program containing a subroutine program using an EGP instruction

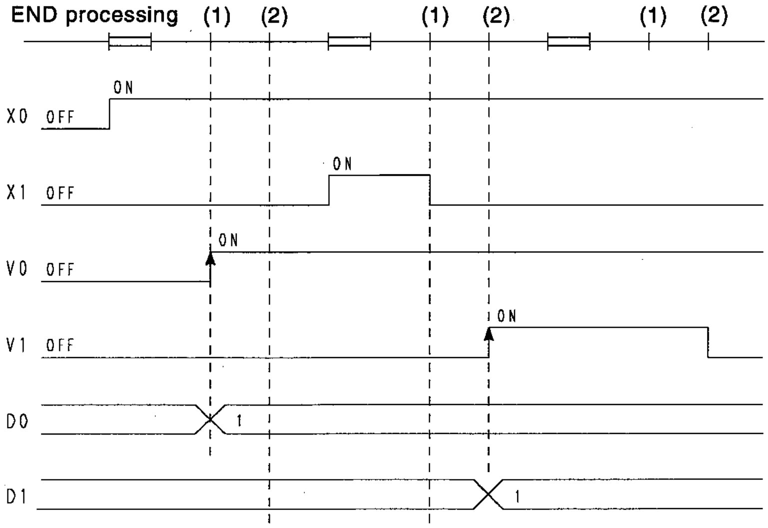
[Ladder Mode]



[List Mode]

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

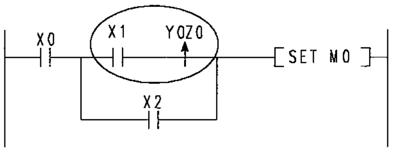
[Operation]

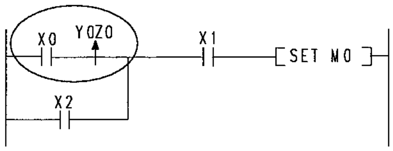


POINTS

(1) Since EGP and EGF instructions are executed according to the results of operation performed immediately before the EGP/EGF instruction, these instructions must be used in the same position as the AND instruction (refer to 5.1.1.).
An EGP and EGF instruction cannot be used at the position of an LD or OR instruction.

(2) EGP and EGF instructions cannot be used at the circuit block positions shown below.





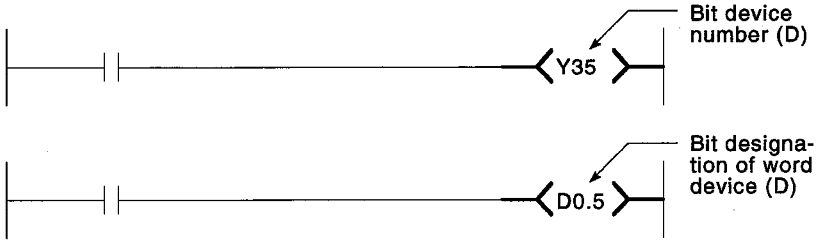
5.3 Out Instructions

5.3.1 Out instructions (excluding timers, counters, and annunciators)

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				DY
(D)	o (Other than T, C, or F)	o					—		o

[Instruction Symbol] [Execution Condition]

OUT



Set Data

Set Data	Meaning	Data Type
(D)	Number of device to be turned ON and OFF	Bit

Functions

(1) Operation results up to the OUT instruction are output to the designated device.

Operation Results	When Using Bit Devices			When Bit Designation has been Made for Word Device
	Coil	Contact		Bit Designated
		A Contact	B Contact	
OFF	OFF	Non-continuity	Continuity	0
ON	ON	Continuity	Non-continuity	1

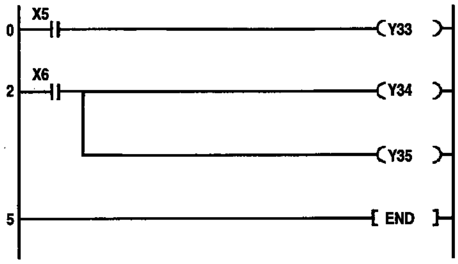
Operation Errors

(1) See Section 3.6 for information regarding errors during index qualification.

Program Example

(1) When bit device is in use

[Ladder Mode]

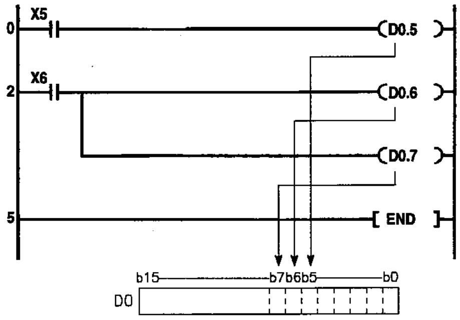


[List Mode]

Steps	Instruction	Device
0	LD	X5
1	OUT	Y33
2	LD	X6
3	OUT	Y34
4	OUT	Y35
5	END	

(2) When bit designation has been made for word device

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X5
1	OUT	D0.5
2	LD	X6
3	OUT	D0.6
4	OUT	D0.7
5	END	

REMARK

The number of basic steps for OUT instructions is as follows:

- When using internal device or file register (R): 1
- When using direct access output (DY) : 2
- When using any other device : 3
  (Including serial number access file register)

5.3.2 Timers

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K	Other
	Bit	Word		Bit	Word				
(D)	o (Only T)	—	—					—	—
Set value	—	o (Other than TC)	o	—	o		—	o *1	—

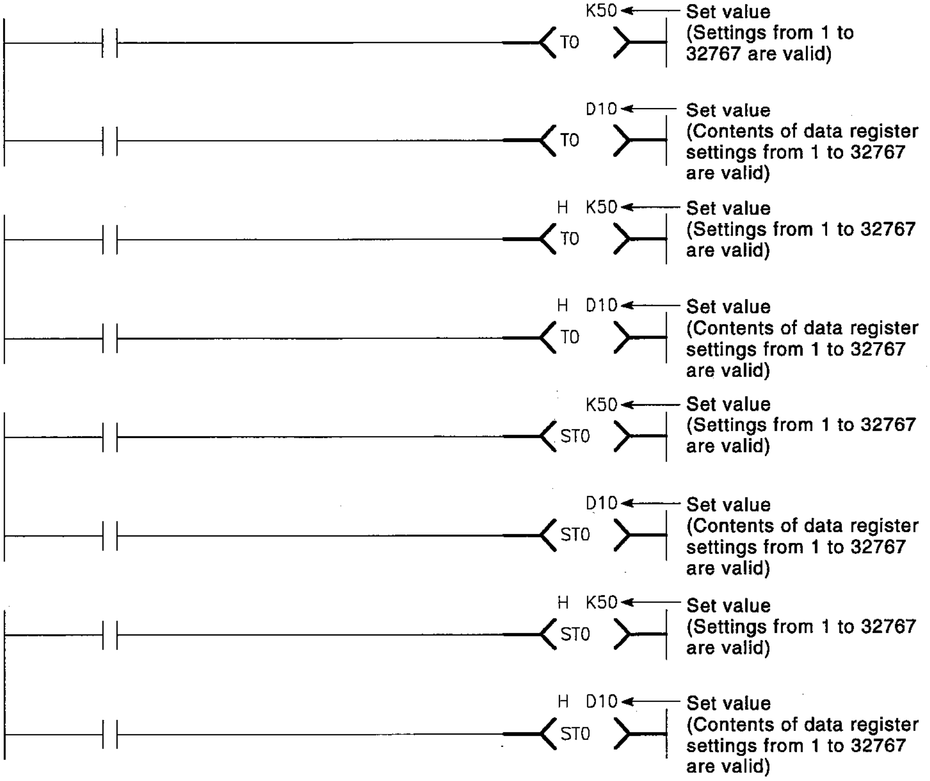
[Instruction Symbol] [Execution Condition]

OUT T□
(Low speed timer)

OUTH T□
(High speed timer)

OUT ST□
(Low speed retentive timer)

OUTH ST□
(High speed retentive timer)

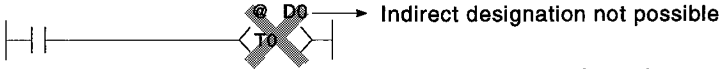


Set Data

Set Data	Meaning	Data Type
(D)	Timer number	Bit
Set value	Value set for timer	BIN 16 bits *2

POINT

*2: The value setting for the timer cannot be designated indirectly.



See Section 3.4 for further information on indirect designations.

REMARK

*1: Timer values can be set only as a decimal constant (K).
Hexadecimal constants cannot be used for timer settings.

Functions

(1) When the operation results up to the OUT instruction are ON, the timer coil goes ON and the timer counts up to the value that has been set; when the time up state (total numeric value is equal to or greater than the setting value), the contact responds as follows:

A contact	Continuity
B contact	Non-continuity

(2) The contact responds as follows when the operation result up to the OUT instruction is a change from ON to OFF:

Type of Timer	Timer Coil	Present Value of Timer	Prior to Time Up		After Time Up	
			A Contact	B Contact	A Contact	B Contact
Low speed timer	OFF	0	Non-continuity	Continuity	Non-continuity	Continuity
High speed timer						
Low speed retentive timer	OFF	Maintains the present value	Non-continuity	Continuity	Continuity	Non-continuity
High speed retentive timer						

* The present value is cleared from low speed retentive timers and high speed retentive timers, and the contact is reset, by use of the RST instruction.

(3) To clear the present value of a retentive timer and turn the contact OFF after time up, use the RST instruction.

(4) A negative number (-32768 to -1) cannot be set as the setting value for the timer.
If the setting value is 0, the timer will time out when the time the OUT instruction is executed.

(5) The following processing is conducted when the OUT instruction is executed:
- OUT T□ coil turned ON or OFF
- OUT T□ contact turned ON or OFF
- OUT T□ present value updated

In cases where a JMP instruction or the like is used to jump to an OUT T□ instruction while the OUT T□ instruction is ON, no present value update or contact ON/OFF operation is conducted. Also, if the same OUT T□ instruction is conducted two or more times during the same scan, the present value of the number of repetitions executed will be updated.

(6) Index qualification for timer coils or contacts can be conducted only by Z0 or Z1.
Index qualification cannot be conducted for the set value for the timer.

REMARKS

(1) The default value for the low speed timer and low speed retentive timer time limit is 100 ms.
The time limit for the low speed timer and low speed retentive timer can be set in the parameter mode "PC system settings" area in increments of 10 ms between the limits of 10 ms to 1 second.

(2) The default value for time limits for the high speed timer and the high speed retentive timer is 10 ms.
The time limits for the high speed timer and the high speed retentive timer can be set in the parameter mode "PC system settings" area in increments of 1 ms between the limits of 10 ms and 100 ms.

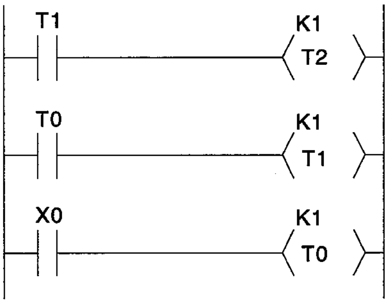
(3) See the QnACPU Programming Manual (Fundamentals), Section 4.2.10, for information on timer counting methods.

Cautions

(1) When creating a program in which the operation of the timer contacttriggers the operation of other timer, create the program according to the operation order of the timers - create the program for the timer that operates later first.
In the following cases, all timers go ON at the same scan if the program is created in the order the timers operate,
- If the set value is smaller than a scan time.
- If "1" is set.

Example

- For timers T0 to T2, the program is created in the order the timer operates later.

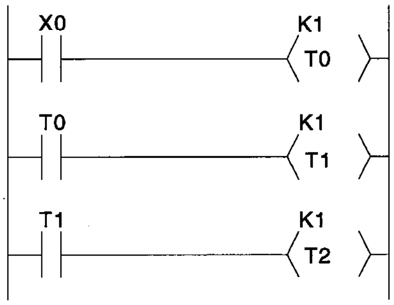


T2 timer starts counting from the next scan after the timer T1 contact is turned ON.

T1 timer starts counting from the next scan after the timer T0 contact is turned ON.

T0 timer starts counting if X0 is turned ON.

- For timers T0 to T2, the program is created in the order of timer operation.



T0 timer starts counting if X0 is turned ON.

T1 and T2 timer contacts are turned ON if the contact of T0 is turned ON.

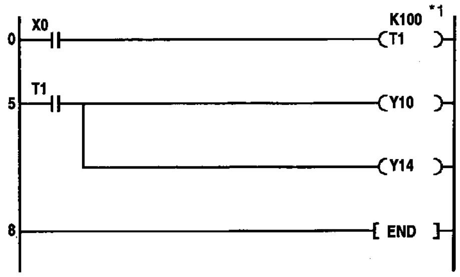
Operation Errors

(1) There are no operation errors associated with the OUT T□ instruction.

Program Example

(1) The following program turns Y10 and Y14 ON 10 seconds after X0 has gone ON

[Ladder Mode]

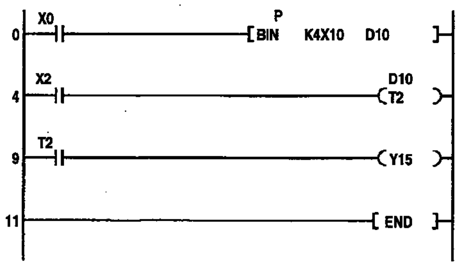


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	OUT	T1 K100
5	LD	T1
6	OUT	Y10
7	OUT	Y14
8	END	

(2) The following program uses the BCD data at X10 to 1F as the timer's set value

[Ladder Mode]



Converts BCD data at X10 to 1F to BIN and stores at D10

When X2 goes ON, the data stored at D10 is calculated as the set value.

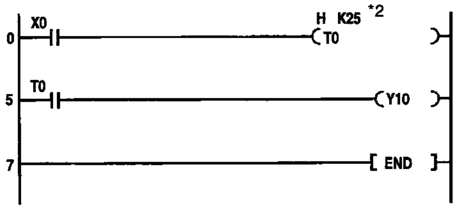
Y15 goes ON when T2 counts out.

[List Mode]

Steps	Instruction	Device
0	LD	X0
1	BINP	K4X10 D10
4	LD	X2
5	OUT	T2 D10
9	LD	T2
10	OUT	Y15
11	END	

(3) The following program turns Y10 ON 250 m after X0 goes ON

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	OUTH	T0 K25
5	LD	T0
6	OUT	Y10
7	END	

REMARKS

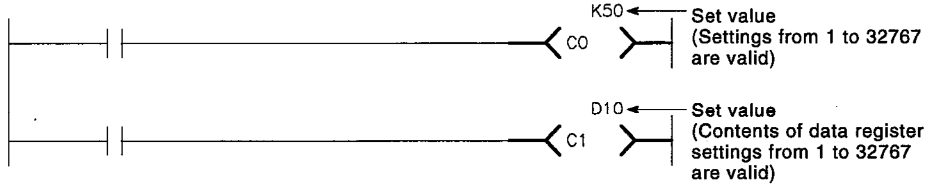
(1) *1: The set value of the low speed timer indicates its default time limit (100 ms).
(2) *2: The set value of the high speed timer indicates its default time limit (10 ms).
(3) The number of basic steps of the OUT T□ instruction is 4.

5.3.3 Counters

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K	Other
	Bit	Word		Bit	Word				U
(D)	o (Only C)	—	—					—	—
Set value	—	o (Other than TC)	o	—	o		—	o *1	—

[Instruction Symbol] [Execution Condition]

OUT C□



Set Data

Set Data	Meaning	Data Type
(D)	Counter number	Bit
Set value	Counter set value	BIN 16 bits *2

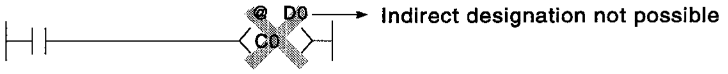
Functions

(1) When the operation results up to the OUT instruction change from OFF to ON, 1 is added to the present value (count value) and the count up state (present value = set value), and the contacts respond as follows:

A contact	Continuity
B contact	Non-continuity

POINTS

*2: Counter value cannot be set by indirect designation.



See Section 3.4 for further information on indirect designations.

REMARKS

(1) See the QnACPU Programming Manual (Fundamentals), Section 4.2.11, for information on counter counting methods.

(2) *1: Counter value can be set only with a decimal constant (K).
A hexadecimal constant (H) or a real number cannot be used for the counter value setting.

(2) No count is conducted with the operation results at ON. (There is no need to perform pulse conversion on count input.)

(3) After the count up state is reached, there is no change in the count value or the contacts until the RST instruction is executed.

(4) A negative number (-32768 to -1) cannot be set as the setting value for the timer.
If the set value is 0, the processing is identical to that which takes place for 1.

(5) Index qualification for the counter coil and contact can use only Z0 and Z1.
Index qualification cannot be performed for the counter setting.

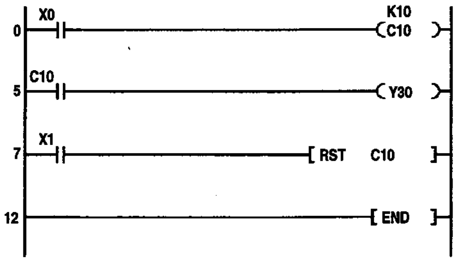
Operation Errors

(1) There are no operation errors associated with the OUT C□ instruction.

Program Example

(1) The following program turns Y30 ON after X0 has gone ON 10 times, and resets the counter when X1 goes ON

[Ladder Mode]

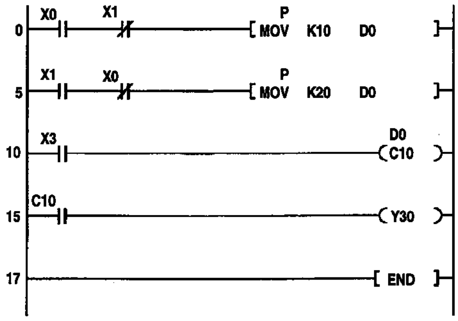


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	OUT	C10 K10
5	LD	C10
6	OUT	Y30
7	LD	X1
8	RST	C10
12	END	

(2) The following program sets the value for C10 at 10 when X0 goes ON, and at 20 when X1 goes ON

[Ladder Mode]



Stores 10 at D0 when X0 goes ON

Stores 20 at D0 when X1 goes ON

C10 takes data stored at D0 as set value, and counts

Y30 goes ON when C10 reaches count out state

[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ANI	X1
2	MOVP	K10 D0
5	LD	X1
6	ANI	X0
7	MOVP	K20 D0
10	LD	X3
11	OUT	C10 D0
15	LD	C10
16	OUT	Y30
17	END	

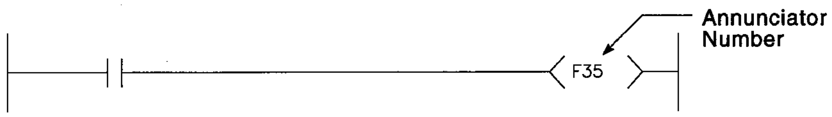
REMARK

The number of basic steps of the OUT C□ instruction is 4.

5.3.4 Annunciator output

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	o (Only F)	—							

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(D)	Number of annunciator to be turned ON	Bit

Functions

(1) Operation results up to the OUT instruction are output to the designated annunciator.

(2) The following responses occur when the annunciator (F) goes ON:
- The annunciator number is displayed at the LED display device on the front of the CPU module, and the "USER" LED goes ON.
- The numbers of the annunciators which are ON (F numbers) are stored from SD64 to SD79.
- The value of SD63 is incremented by 1.

(3) If the value of SD63 is 16 (which happens when 16 annunciators are already ON), even if a new annunciator is turned ON, its number will not be stored at SD64 through SD79.

(4) If an annunciator is turned OFF by the OUT instruction, the coil goes OFF, but there are no changes in the LED display device in the front of the CPU, the state of the "USER" LED, or the contents of the values stored at SD63 through SD79.
To delete an annunciator for which the LED display device on the front of the CPU and the "USER" LED have gone off, and which was turned OFF by the OUT F□ instruction from SD63 to SD79, use the RST F□ instruction.

Operation Errors

(1) There are no operation errors associated with the OUT F□ instruction.

REMARKS

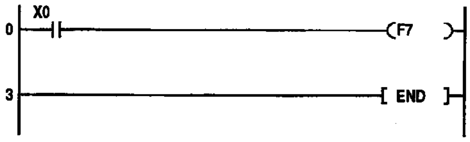
(1) For further information on annunciators, see the QnACPU Programming Manual (Fundamentals).

(2) The number of basic steps for the OUT F□ instruction is 4.

Program Example

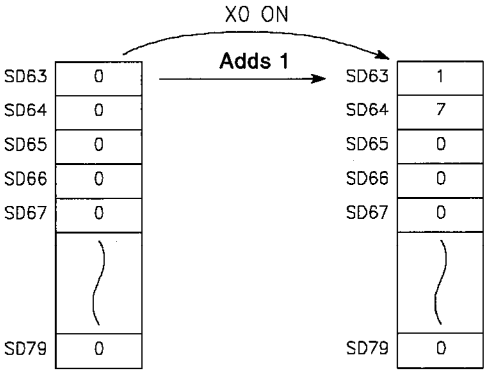
(1) The following program turns F7 ON when X0 goes ON, and stores the value 7 from SD64 to SD79

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	OUT	F7
3	END	

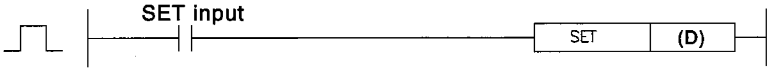


5.3.5 Setting devices (except for annunciators)

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other	
	Bit	Word		Bit	Word				BL	DY
(D)	o	o		o	o		—	—	o	o

[Instruction Symbol] [Execution Condition]

SET



Set Data

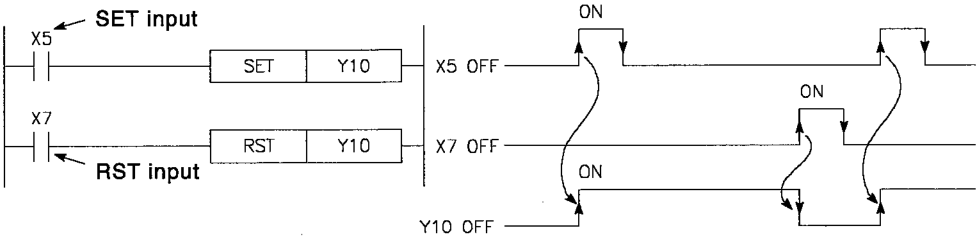
Set Data	Meaning	Data Type
(D)	Bit device number to be set (ON)/Word device bit designation	Bit

Functions

(1) When SET input is ON, the designated devices respond as follows:

Device	Device Status
Bit device	Coils and contacts turned ON
When bit designation has been made for word device	Designation bit set at 1

(2) Devices turned ON will stay ON even if SET input goes to OFF
Devices turned ON by the SET instruction can be turned OFF by the RST instruction



(3) Device status does not change when SET input is OFF.

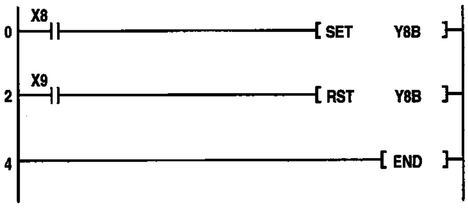
Operation Errors

(1) There are no operation errors associated with the SET instruction.

Program Example

(1) The following program sets Y8B (ON) when X8 goes ON, and resets Y8B (OFF) when X9 goes ON

[Ladder Mode]

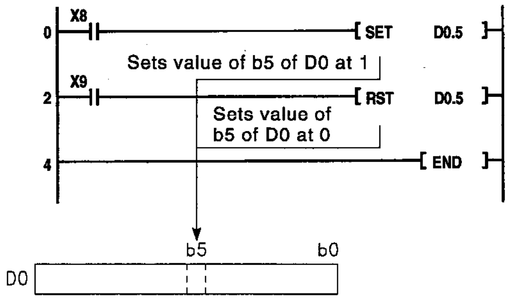


[List Mode]

Steps	Instruction	Device
0	LD	X8
1	SET	Y8B
2	LD	X9
3	RST	Y8B
4	END	

(2) The following program sets the value of D0 bit 5 (b5) at 1 when X8 goes ON, and resets the same bit value to D0 when X9 goes ON

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X8
1	SET	D0.5
2	LD	X9
3	RST	D0.5
4	END	

REMARK

The basic SET instructions are as follows:

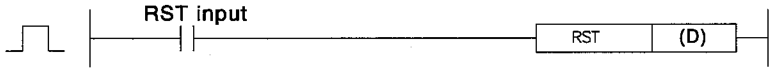
- When internal device or file register (R0 to R32767) are in use : Step 1
- When direct access output (DY) or SFC program device (BL) are in use: Step 2
- When timer (T) or counter (C) are in use : Step 4
- When some other device is in use : Step 3

5.3.6 Resetting devices (except for annunciators)

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other	
	Bit	Word		Bit	Word				BL	DY
(D)	o	o		o	o		o	—	—	o

[Instruction Symbol] [Execution Condition]

RST



Set Data

Set Data	Meaning	Data Type
(D)	Bit device number to be reset/Word device bit designation	Bit
	Word device number to be reset	BIN 16 bits

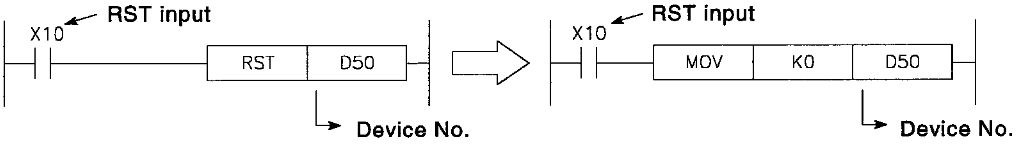
Functions

(1) Designated devices respond as follows when RST input is turned ON:

Device	Device Status
Bit device	Turns coils and contacts OFF
Timers and counters	Sets the present value to 0, and turns coils and contacts OFF
When bit designation has been made for word device	Sets value of designated bit to 0
Word devices other than timers and counters	Sets contact to 0

(2) Device status does not change when RST input goes OFF

(3) The functions of the word devices designated by the RST instruction are identical to the following ladder:

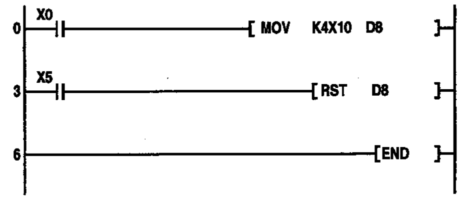


Operation Errors

(1) There are no operation errors associated with the RST instruction.

(2) The following program sets the value of the data register to 0

[Ladder Mode]



When X0 goes ON, the contents of X10 to 1F are stored at D8

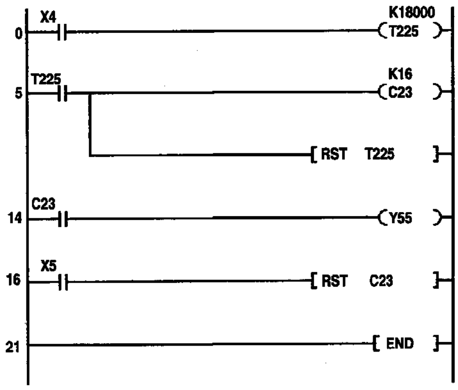
When X5 goes ON, the value of D8 is set to 0

[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MOV	K4X10
		D8
3	LD	X5
4	RST	D8
6	END	

(3) The following program resets the 100 ms retentive timer and counter

[Ladder Mode]



When T225 is set at the retentive timer, the ON time for X4 is 30 minutes, then T225 goes ON.

Counts the number of times T225 goes ON.

When the T225 contact goes ON, the T225 coil, contact, and present value are reset.

When C23 reaches the count out state, Y55 goes ON.

Resets C23 when X5 goes ON.

[List Mode]

Steps	Instruction	Device
0	LD	X4
1	OUT	T225
		K18000
5	LD	T225
6	OUT	C23
		K16
10	RST	T225
14	LD	C23
15	OUT	Y55
16	LD	X5
17	RST	C23
21	END	

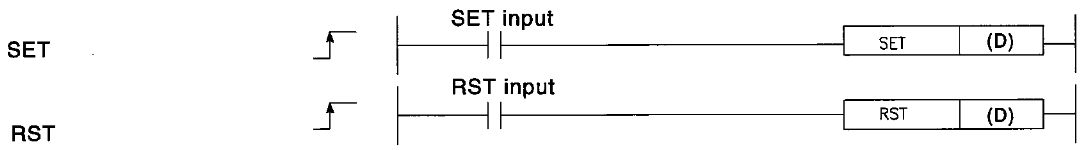
REMARK

The number of basic steps for the RST instruction is 2.

5.3.7 Setting and resetting the annunciators

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
(D)	o (Only F)	—							

[Instruction Symbol] [Execution Condition]



Set Data

Instruction Name	Set Data	Meaning	Data Type
SET	(D)	Number of annunciator to be set (F number)	Bit
RST	(D)	Number of annunciator to be reset (F number)	

Functions

SET

(1) Annunciator designated by (D) goes ON when SET input goes ON.

(2) The following responses occur when the annunciator (F) goes ON:
- The annunciator number is displayed at the LED display device at the front of the CPU, or the "USER" LED goes ON.
- The numbers (F numbers) of the annunciators turned ON are stored successively at the special registers (SD63 to SD79).
- The value of SD63 is incremented by 1.

(3) If the value of SD63 is 16 (which happens when 16 annunciators are already ON), even if a new annunciator is turned ON, its number will not be stored at SD64 through SD79.

RST

(1) Annunciators designated by (D) are turned OFF when RST input goes ON.

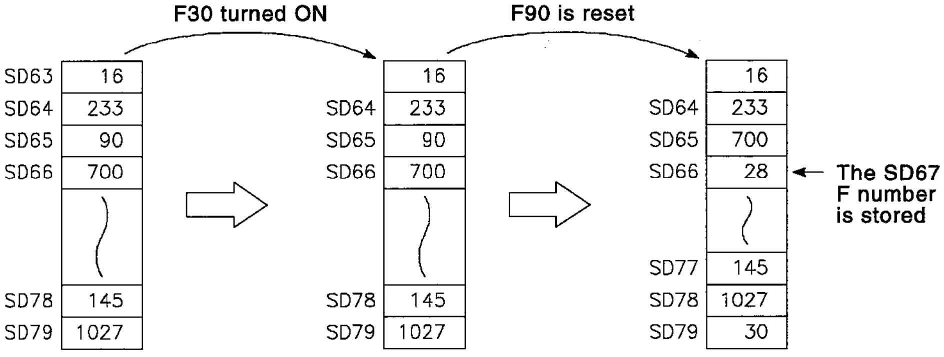
(2) The annunciator numbers (F numbers) of annunciators that have gone OFF are deleted from the special registers (SD64 to SD79), and the value of SD63 is decremented by 1.

REMARK

See the QnACPU Programming Manual (Common Instructions), Section 4.2.5, for further information on annunciators.

(3) If, when the value of SD63 is 16, and annunciator numbers are deleted from SD64 through SD79 by use of the RST instruction, annunciators whose numbers are not registered in SD64 through SD79 are then turned ON, the numbers of these annunciators will be registered. If all annunciator numbers from SD64 through SD79 are turned OFF, the LED display device on the front of the CPU, or the "USER" LED, will be turned OFF.

[Operations which take place when SD63 is 16]



The SD67 F number is stored

The contents of SD63 and SD64 through SD79 are not changed.

F30, which is ON, is stored at SD79

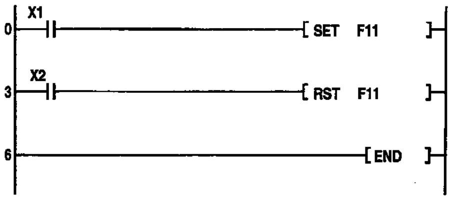
Operation Errors

(1) There are no operation errors associated with the SET F☐ or RST F☐ instructions.

Program Example

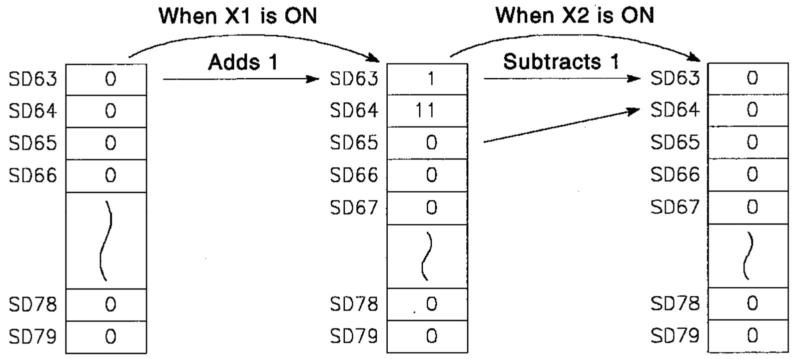
(1) The following program turns annunciator F11 ON when X1 goes ON, and stores the value 11 at the special register (SD64 through SD79). Further, the program resets annunciator F11 if X2 goes ON, and deletes the value 11 from the special registers (SD64 through SD79).

[Ladder Mode]



[List Mode]

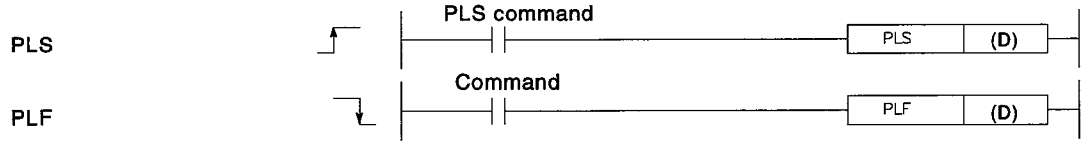
Steps	Instruction	Device
0	LD	X1
1	SET	F11
3	LD	X2
4	RST	F11
6	END	



5.3.8 Leading edge and trailing edge output

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				DY
(D)	o						—		o

[Instruction Symbol] [Execution Condition]



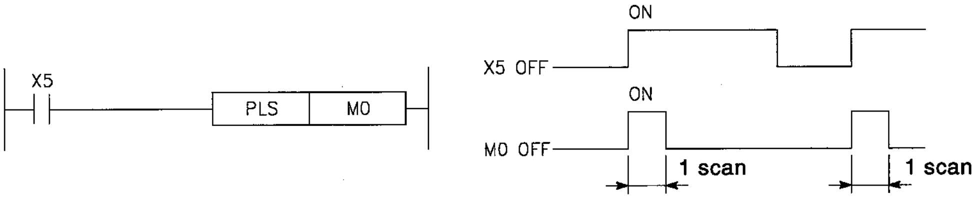
Set Data

Set Data	Meaning	Data Type
(D)	Pulse conversion device	Bit

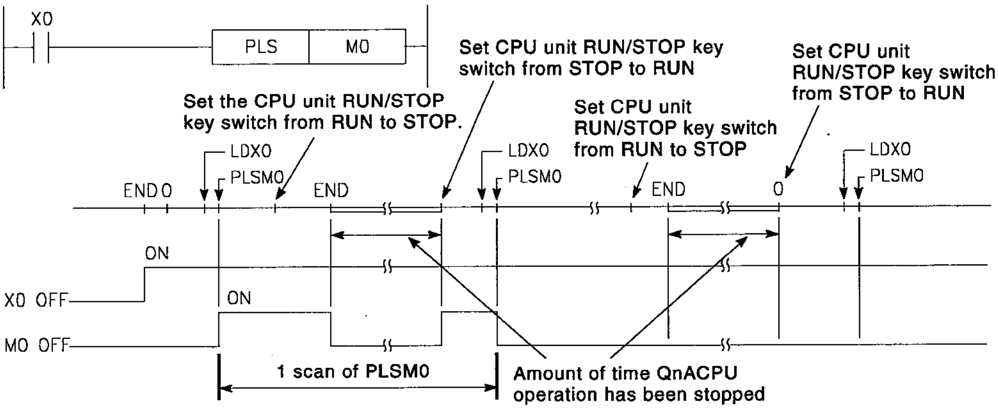
Functions

PLS

(1) When the PLS command goes from OFF to ON, the designation device is scanned once, and turned ON, or turned OFF at any other time (when instruction goes from ON to ON, ON to OFF, or OFF to OFF).



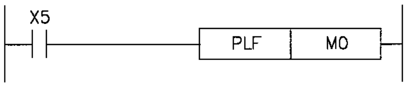
(2) If the RUN/STOP key switch is changed from RUN to STOP after the execution of the PLS instruction, the PLS instruction will not be executed again even if the switch is set back to RUN.

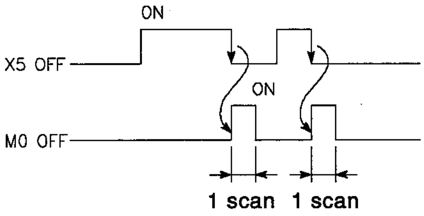


(3) When a latch relay (L) has been designated for the PLS command, if the power is turned OFF while the latch relay (L) is in the ON state, the designated device will come ON for one scan when power is turned back ON.

PLF

(1) When the PLF command goes from ON to OFF, the designated device comes ON for 1 scan ON, and at any other time (from OFF to OFF, from OFF to ON, or from ON to ON), it is turned OFF.





(2) If the RUN/STOP key switch is changed from RUN to STOP after the execution of the PLS instruction, the PLF instruction will not be executed again even if the switch is set back to RUN.

POINT

Note that the device designated by (D) will sometimes be ON for more than one scan if the CJ or a similar instruction has been used to jump to a PLS or PLF instruction.

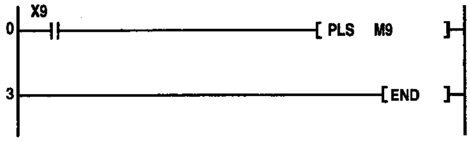
Operation Errors

(1) There are no operation errors associated with the PLS or PLF instructions.

Program Example

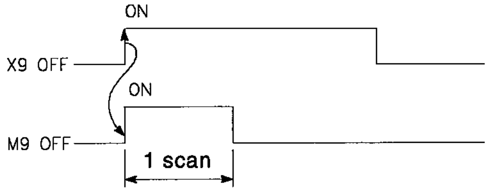
(1) The following program executes the PLS instruction when X9 goes ON.

[Ladder Mode]



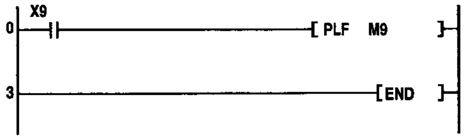
[List Mode]

Steps	Instruction	Device
0	LD	X9
1	PLS	M9
3	END	



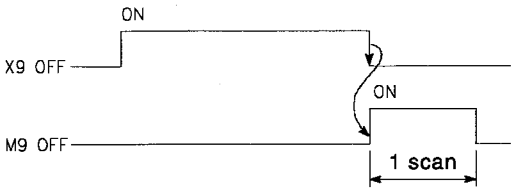
(2) The following program executes the PLF instruction when X9 goes OFF.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X9
1	PLF	M9
3	END	

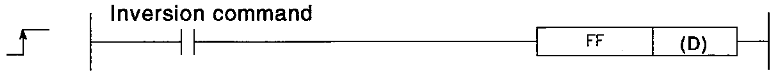


5.3.9 Bit device output inversion

<table>
<tr><th rowspan="3">Set Data</th><th colspan="10">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System, User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th rowspan="2">Constant K, H</th><th>Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th><th>DY</th></tr>
<tr><td>(D)</td><td colspan="6">○</td><td colspan="2">—</td><td>○</td></tr>
</table>

[Instruction Symbol] [Execution Condition]

FF



Set Data

Set Data	Meaning	Data Type
(D)	Device number to invert	Bit

Function

(1) The status of the device designated by (D) is inverted when the inversion command goes from OFF to ON.

Device	Device Status	
	Prior to FF execution	After FF execution
Bit device	OFF	ON
	ON	OFF
Designation of word device	0	1
	1	0

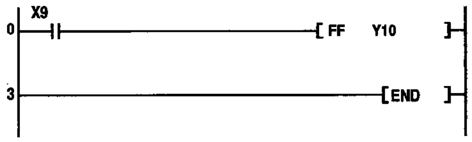
Operation Errors

(1) There are no operation errors associated with the FF instruction.

Program Example

(1) The following program inverts the output of Y10 when X9 goes ON.

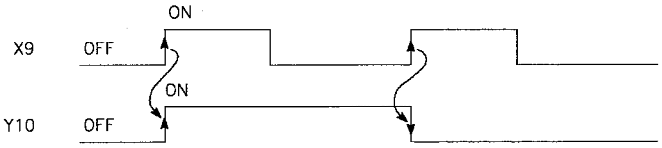
[Ladder Mode]



[List Mode]

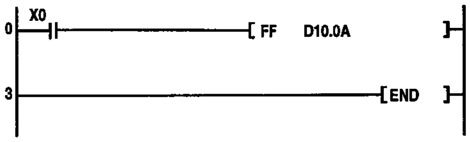
Steps	Instruction	Device
0	LD	X9
1	FF	Y10
3	END	

[Timing Chart]



(2) The following program reverses b10 (bit 10) of D10 when X0 goes ON

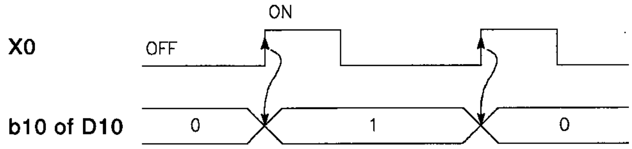
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	FF	D10.0A
3	END	

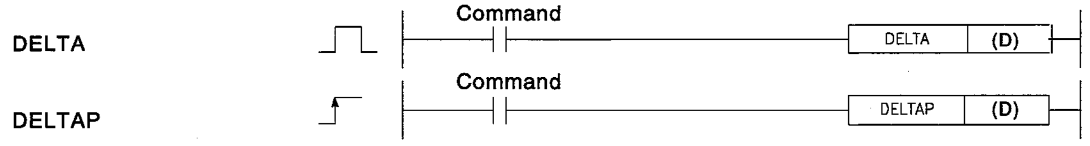
[Timing Chart]



5.3.10 Pulse conversion of contact

<table>
<tr><th rowspan="3">Set Data</th><th colspan="10">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System, User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th rowspan="2">Constant K, H</th><th>Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th><th>DY</th></tr>
<tr><td>(D)</td><td colspan="8">—</td><td>o</td></tr>
</table>

[Instruction Symbol] [Execution Condition]



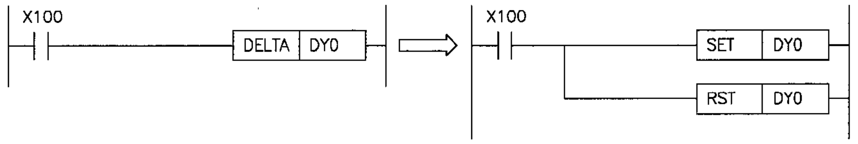
Set Data

Set Data	Meaning	Data Type
(D)	Bit for which pulse conversion is to be conducted	Bit

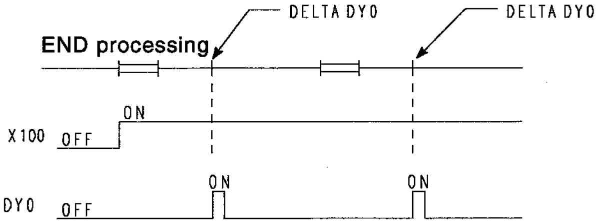
Functions

(1) Conducts pulse output of direct access output (DY) designated by (D). If DELTA DY0 has been designated, the resulting operation will be identical to the ladder shown below, which uses the SET/RST instructions.

[Ladder created by the DELTA instruction] [Ladder using the SET/RST instructions]



[Operation]



(2) The DELTA (P) instruction is used by commands for leading edge execution for a special function module.

Operation Errors

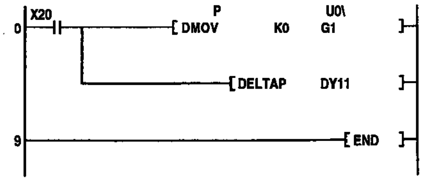
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- A direct access output number designated by (D) has exceeded the QnACPU output range. (Error code: 4101)

Program Example

(1) The following program presets CH1 of the AD61 mounted at slot 0 of the main base unit, when X20 goes ON.

[Ladder Mode]



Stores preset value (0) at addresses 1 and 2 of the AD61 buffer memory

Output of preset instruction

[List Mode]

Steps	Instruction	Device
0	LD	X20
1	DMOVP	K0
		U0\G1
6	DELTAP	DY11
9	END	

5.4 Shift Instruction

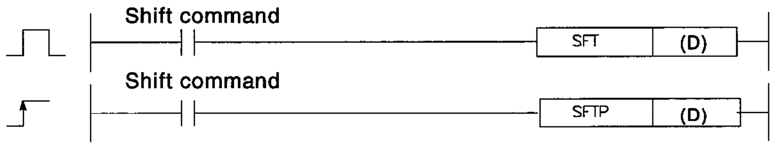
5.4.1 Bit device shift

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant DY	Other
	Bit	Word		Bit	Word				U
(D)	o (Other than TC)						—		o

[Instruction Symbol] [Execution Condition]

SFT

SFTP



Set Data

Set Data	Meaning	Data Type
(D)	Number of device to shift	Bit

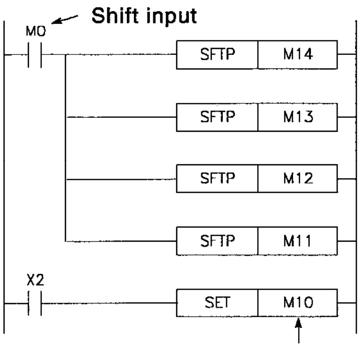
Functions

(1) When bit device is used

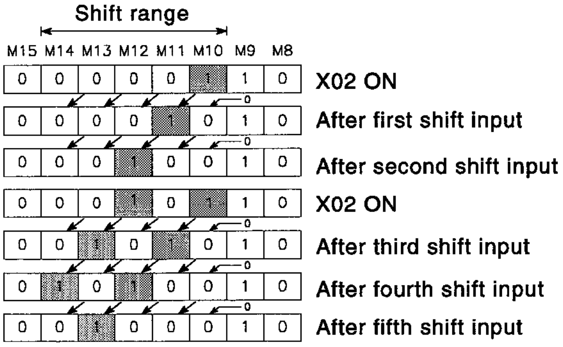
(a) Shifts to a device designated by (D) the ON/OFF status of the device immediately prior to the one designated, and turns the prior device OFF.
For example, if M11 has been designated by the SFT instruction, when the SFT instruction is executed, it will shift the ON/OFF status of M10 to M11, and turn M10 OFF.

(b) Turn the first device to be shifted ON with the SET instruction.

(c) When the SFT and SFTP are to be used consecutively, the program starts from the device with the larger number.



First device to shift

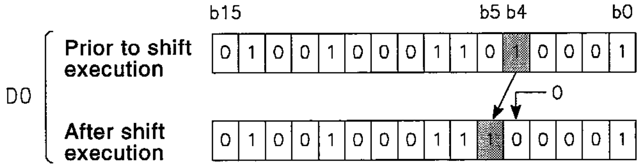


M15	M14	M13	M12	M11	M10	M9	M8	
0	0	0	0	0	1	1	0	X02 ON
0	0	0	0	1	0	1	0	After first shift input
0	0	0	1	0	0	1	0	After second shift input
0	0	0	1	0	1	1	0	X02 ON
0	0	1	0	1	0	1	0	After third shift input
0	1	0	1	0	0	1	0	After fourth shift input
0	0	1	0	0	0	1	0	After fifth shift input

* At M8 to 16, "1" indicates ON and "0" indicates OFF.

(2) When word device bit designation is used

(a) Shifts to a bit in the device designated by (D) the 1/0 status of the bit immediately prior to the one designated, and turns the prior bit to 0.
For example, if D0.5 (bit 5 [b5] of D0) has been designated by the SFT instruction, when the SFT instruction is executed, it will shift the 1/0 status of b4 of D0 to b5, and turn b4 to 0.



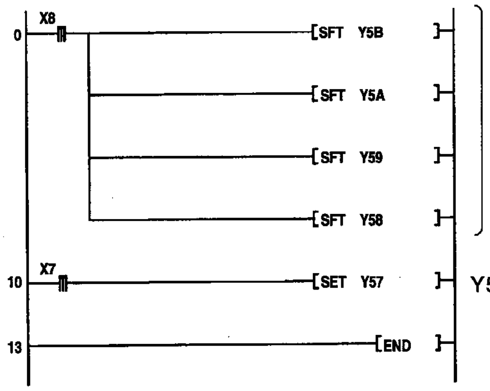
Operation Errors

(1) There are no operation errors associated with the SFT (P) instruction.

Program Example

(1) The following program shifts Y57 to Y5B when X8 goes ON.

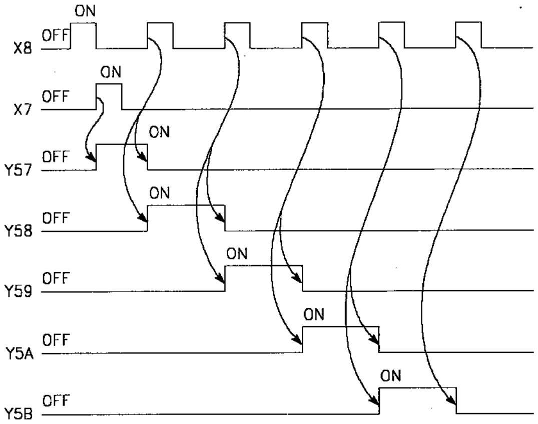
[Ladder Mode]



Shifts Y57 to Y5B when X8 goes ON
(Begin program from larger device number)

Y57 turned ON when X7 goes ON.

[Timing Chart]



[List Mode]

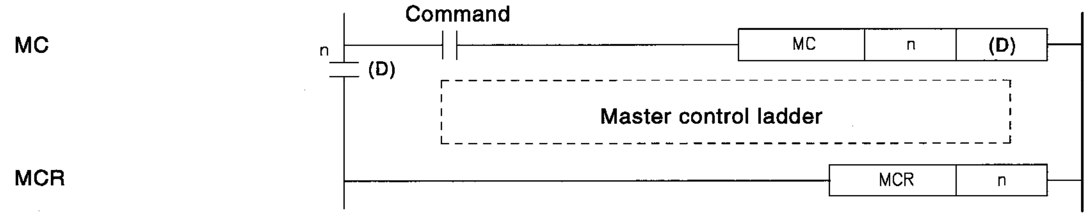
Steps	Instruction	Device
0	LDP	X8
2	SFT	Y5B
4	SFT	Y5A
6	SFT	Y59
8	SFT	Y58
10	LDP	X7
12	SET	Y57
13	END	

5.5 Master Control Instructions

5.5.1 Setting and resetting the master control

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constan K, H	Other	
	Bit	Word		Bit	Word				N	DY
n	—						—		o	—
(D)	o						—		—	o

[Instruction Symbol] [Execution Condition]

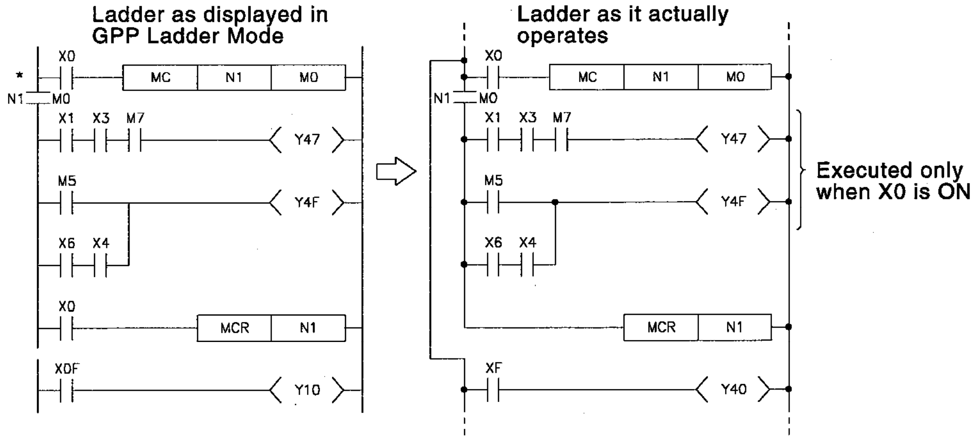


Set Data

Set Data	Meaning	Data Type
n	Nesting (N0 to N14)	Nesting
(D)	Number of device to turn ON	Bit

Functions

The master control instruction is used to enable the creation of highly efficient ladder switching sequence programs, through the opening and closing of a common bus for ladders.
A ladder using the master control would look as shown below:



REMARK

*: When programming in the ladder mode of a peripheral device, it is not necessary to input contacts on the vertical bus.
These will be automatically displayed when the "conversion" operation is conducted after the creation of the ladder.

MC

(1) If the ON/OFF command of the MC instruction is ON when master control is commenced, the operation result between the MC instruction and MCR instruction will be exactly as the instruction (ladder) shows. If the MC ON/OFF indicator is OFF, the operation result between the MC and MCR instructions will be as shown below:

Device	Device Status
High speed timer Low speed timer	Count value goes to 0, coils and contacts all go OFF
High speed retentive timer Low speed retentive timer Counter	Coils go OFF, but counter values and contacts all maintain current status.
Devices in OUT instruction	All turned OFF
SET, RST SFT Basic, Application } Devices in the following instructions:	Maintain current status

(2) Even when the MC instruction is OFF, instructions from the MC instruction to the MCR instruction will be executed, so scan time will not be shortened.

POINT

If there are unnecessary contact instructions (FOR - NEXT, EI, DI, etc.) in ladders which use master controls, the QnACPU will execute these instructions regardless of the ON/OFF state of the MC instruction.

(3) By changing the device designated by (D), the MC instruction can use the same nesting (N) number as often as desired.

(4) Coils from devices designated by (D) are turned ON when the MC instruction is ON.
Further, using these same devices with the OUT instruction or other instructions will cause them to become double coils, so devices designated by (D) should not be used within other instructions.

MCR

(1) This is the instruction for recovery from the master control, and indicates the end of the master control range of operation.

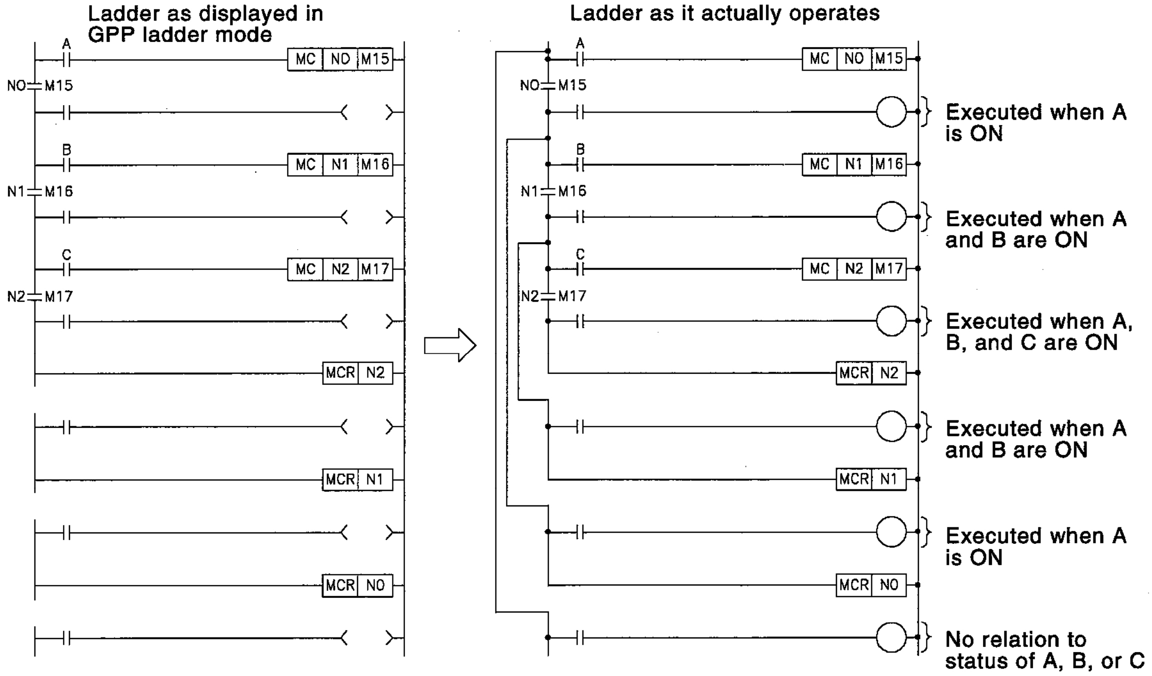
(2) Do not place contact instructions before the MCR instruction.

Operation Errors

(1) There are no operation errors associated with the MC or MCR instructions.

Program Example

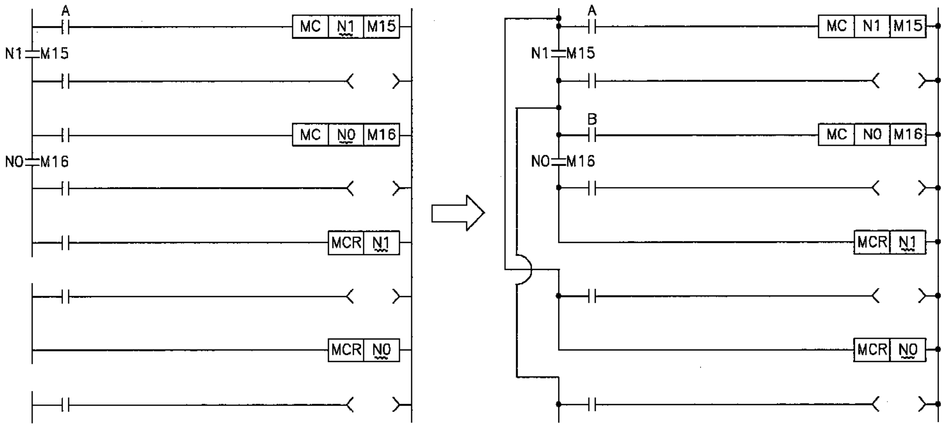
The master control instruction can be used in nesting.
The different master control regions are distinguished by nesting (N).
Nesting can be performed from N0 through N14.
The use of nesting enables the creation of ladders which successively limit the execution condition of the program.
A ladder using nesting would appear as shown below:



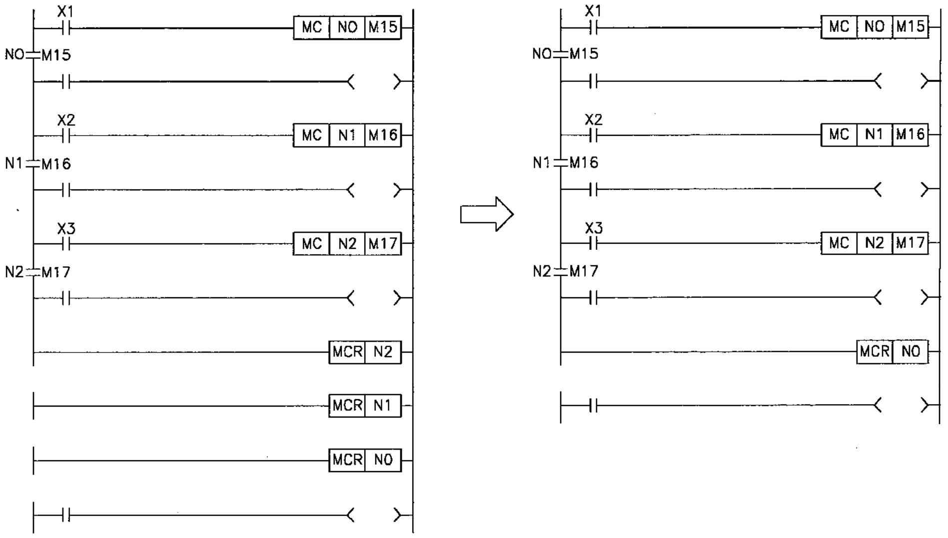
Cautions when Using Nesting Architecture

(1) Nesting can be used up to 15 times (N0 to N14)
When using nesting, nests should be inserted from the lower to higher nesting number (N) with the MC instruction, and from the higher to the lower order with the MCR instruction.
If this order is reversed, there will be no nesting architecture, and the QnACPU will not be capable of performing correct operations.
For example, if nesting is designated in the order N1 to N0 by the MC instruction, and also designated in the N1 to N0 order by the MCR instruction, the vertical bus will intersect and a correct master control ladder will not be produced.

[Ladder as displayed in the GPP ladder mode]   [Ladder as it actually operates]



(2) If the nesting architecture results in MCR instructions concentrated in one location, all master controls can be terminated by use of just the lowest nesting number (N).



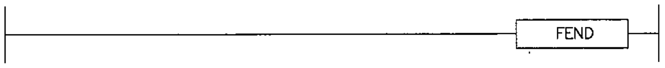
5.6 Termination Instructions

5.6.1 End main routine program

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]

FEND

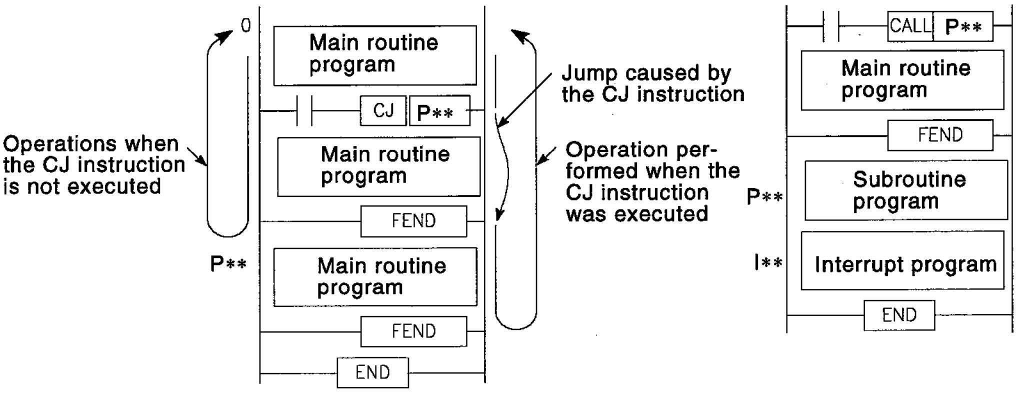


Functions

(1) The FEND instruction is used in cases where the CJ instruction or other instructions are used to cause a branch in the sequence program operations, and in cases where the main routine program is to be split from a subroutine program or an interrupt program.

(2) Execution of the FEND instruction will cause the QnACPU to terminate the program it was executing.

(3) Even sequence programs following the FEND instruction can be displayed in ladder display at a peripheral devices. (Peripheral devices continue to display ladders until encountering an END instruction.)



(a) When using the CJ instruction

(b) When there is a subroutine or interrupt program

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

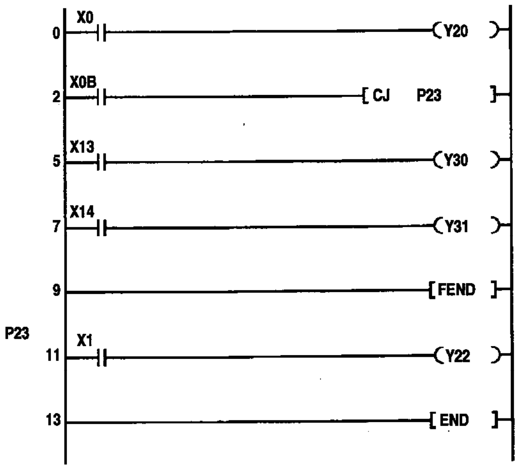
- A FEND instruction is executed after the execution of a CALL, FCALL, ECALL, or EFCALL instruction, and before the execution of the RET instruction. (Error code: 4211)
- A FEND instruction is executed after the execution of a FOR instruction, and before the execution of a NEXT instruction. (Error code: 4200)

- A FEND instruction is executed during an interrupt program, and before the execution of an IRET instruction. (Error code: 4221)
- A FEND instruction is executed between the CHKCIR and CHKEND instructions. (Error code: 4230)
- A FEND instruction is executed between the IX and IXEND instructions. (Error code: 4231)

Program Example

(1) The following program uses the CJ instruction.

[Ladder Mode]



When XB is ON, jumps to label P23; from P23, executes the next step

Executed when XB is OFF

Indicates the termination of the sequence program when XB is OFF

[List Mode]

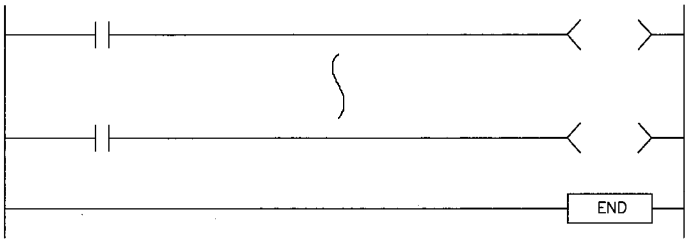
Steps	Instruction	Device
0	LD	X0
1	OUT	Y20
2	LD	X0B
3	CJ	P23
5	LD	X13
6	OUT	Y30
7	LD	X14
8	OUT	Y31
9	FEND	
10		P23
11	LD	X1
12	OUT	Y22
13	END	

5.6.2 End sequence program

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
—	—								

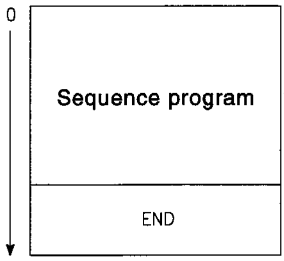
[Instruction Symbol] [Execution Condition]

END



Functions

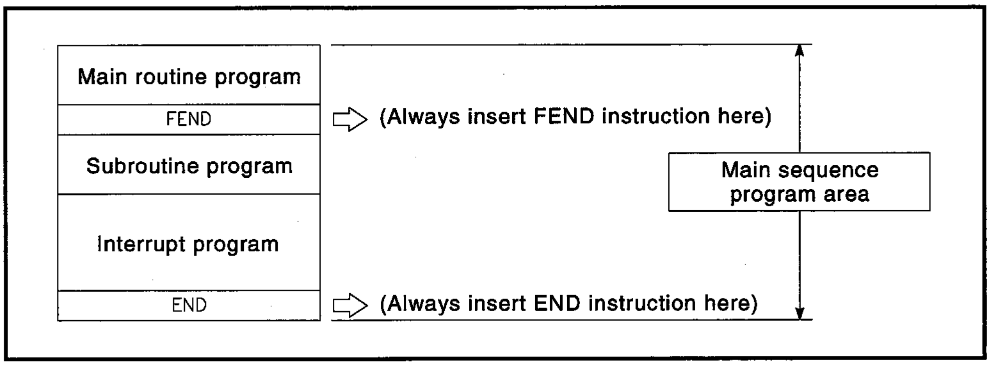
(1) Indicates termination of programs, including main routine program, subroutine program, and interrupt programs.
Execution of the END instruction will cause the QnACPU to terminate the program that was being executed.



(2) An END instruction cannot be used during the execution of the main sequence program.
If it is necessary to perform END processing during the execution of a program, use the FEND instruction.

(3) When programming in the ladder mode of a peripheral device, it is not necessary to input an END instruction.

(4) The use of the END and FEND instructions is broken down as follows for main routine programs, subroutine programs, and interrupt programs:



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- An END instruction was executed before the execution of the RET instruction and after the execution of the CALL, FCALL, ECALL, or EFCALL instruction. (Error code: 4211)
- An END instruction was executed before the execution of a NEXT instruction and after the execution of the FOR instruction. (Error code: 4200)
- An END instruction was executed during an interrupt program prior to the execution of the IRET instruction. (Error code: 4221)
- An END instruction was executed within the CHKCIR to CHKEND instruction loop. (Error code: 4230)
- An END instruction was executed within the IX to IXEND instruction loop. (Error code: 4231)

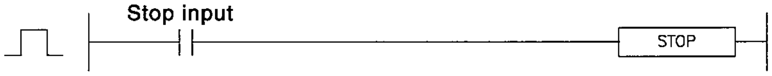
5.7 Other Instructions

5.7.1 Sequence program stop

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
—	—								

[Instruction Symbol] [Execution Condition]

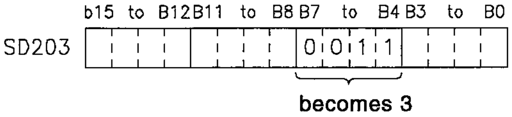
STOP



Functions

(1) When stop input is turned ON, output Y is reset and the QnACPU operations are terminated.
(The same result will take place if the RUN/STOP key switch is turned to the STOP setting.)

(2) Execution of the STOP instruction will cause the value of B4 through B7 of the special register SD203 to become "3".



(3) In order to restart QnACPU operations after the execution of the STOP instruction, return the RUN/STOP key switch, which has been changed from RUN to STOP, back to the RUN position.

Operation Errors

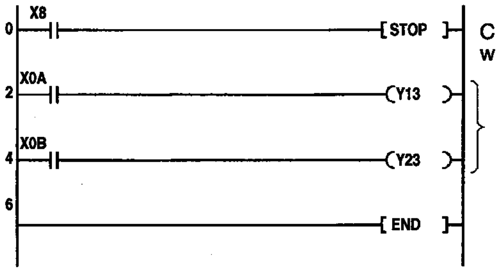
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- A STOP instruction was executed before the execution of the RET instruction and after the execution of the CALL, FCALL, ECALL, or EFCALL instruction. (Error code: 4211)
- A STOP instruction was executed before the execution of a NEXT instruction and after the execution of the FOR instruction. (Error code: 4200)
- A STOP instruction was executed during an interrupt program prior to the execution of the IRET instruction. (Error code: 4221)
- A STOP instruction was executed within the CHKCIR to CHKEND instruction loop. (Error code: 4230)
- A STOP instruction was executed within the IX to IXEND instruction loop. (Error code: 4231)

Program Example

(1) The following program stops the QnACPU when X8 goes ON

[Ladder Mode]



Causes programmable controller to stop when X8 goes ON.

Sequence program

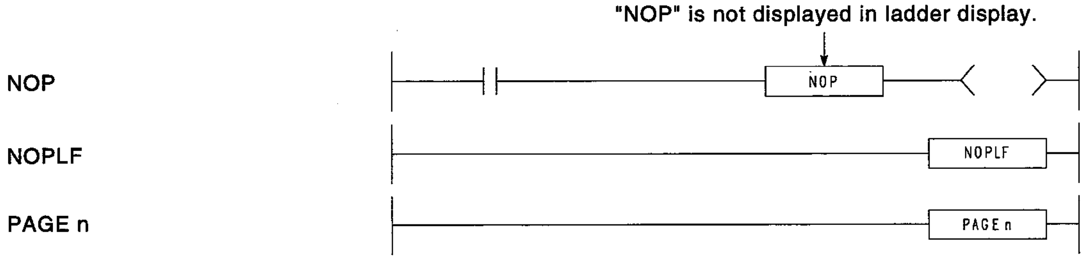
[List Mode]

Steps	Instruction	Device
0	LD	X8
1	STOP	
2	LD	X0A
3	OUT	Y13
4	LD	X0B
5	OUT	Y23
6	END	

5.7.2 No operation

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]



Functions

NOP

(1) This is a no operation instruction that has no impact on any operations up to that point.

(2) The NOP instruction is used in the following cases:

(a) To insert space for sequence program debugging.

(b) To delete an instruction without having to change the number of steps. (Replace the instruction with NOP)

(c) To temporarily delete an instruction.

NOPLF

(1) This is a no operation instruction that has no impact on any operations up to that point.

(2) The NOPLF instruction is used when printing from a peripheral device to force a page change at any desired location.

(a) When printing ladders

1) A page break will be inserted between ladder blocks with the presence of the NOPLF instruction.

2) The ladder cannot be displayed correctly if an NOPLF instruction is inserted in the midst of a ladder block.
Do not insert an NOPLF instruction in the midst of a ladder block.

(b) When printing instruction lists

- The page will be changed after the printing of the NOPLF instruction.

(3) See the Operating Manual for the peripheral device in use for more information regarding printouts from peripheral devices.

PAGEN

(1) This is a no operation instruction that has no impact on any operations up to that point.

(2) Causes processing from step 0 of the designated nth page of the program following the PAGEn instruction. (Peripheral device display, printers, etc.)

(3) If there is no PAGEn instruction, processing begins from page 0.

Operation Errors

(1) There are no errors associated with the NOP, NOPLF, or PAGE instructions.

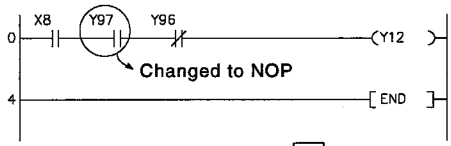
Program Example

NOP

(1) Contact closed......... Deletes AND or ANI instruction

Before change

[Ladder Mode]

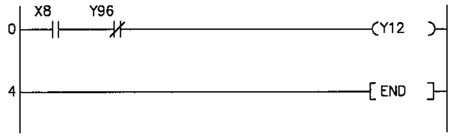


[List Mode]

Steps	Instruction	Device
0	LD	X8
1	AND	Y97
2	ANI	X96
3	OUT	Y12
4	END	

After change

[Ladder Mode]



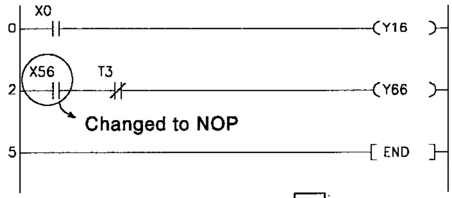
[List Mode]

Steps	Instruction	Device
0	LD	X8
1	NOP	
2	ANI	Y96
3	OUT	Y12
4	END	

(2) Contact closed......... LD, LDI changed to NOP (Note carefully that changing the LD and LDI instructions to NOP completely changes the nature of the ladder.)

Before change

[Ladder Mode]

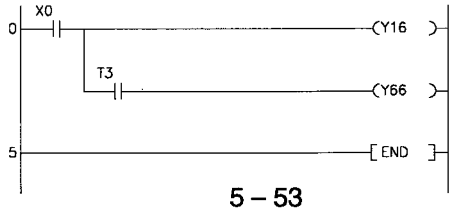


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	OUT	Y16
2	LD	X56
3	AND	T3
4	OUT	Y66
5	END	

After change

[Ladder Mode]

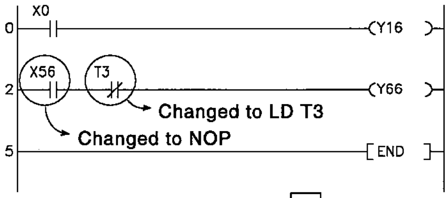


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	OUT	Y16
2	NOT	
3	AND	T3
4	OUT	Y66
5	END	

Before change

[Ladder Mode]

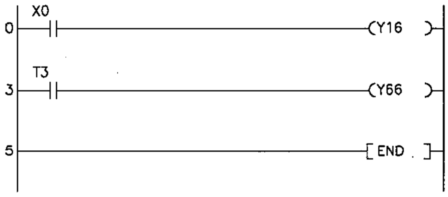


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	OUT	Y16
2	LD	X56
3	AND	T3
4	OUT	Y66
5	END	

After change

[Ladder Mode]

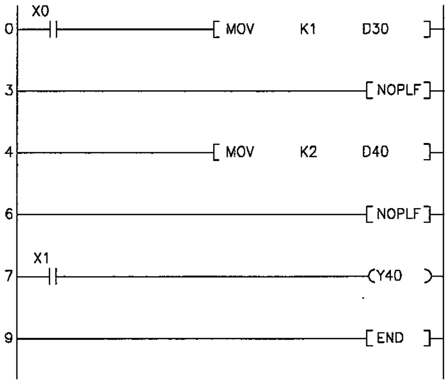


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	OUT	Y16
2	NOP	
3	LD	T3
4	OUT	Y66
5	END	

NOPLF

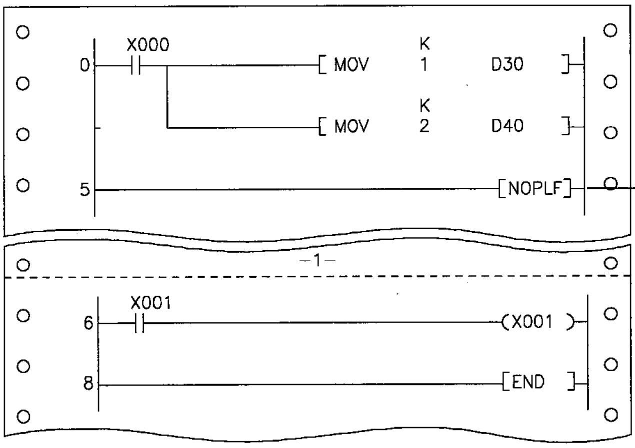
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MOV	K1 D30
3	NOPLF	
4	MOV	K2 D40
6	NOPLF	
7	LD	X1
8	OUT	Y40
9	END	

- Printing the ladder will result in the following:



Page change forced when NOPLF is inserted between two ladder blocks.

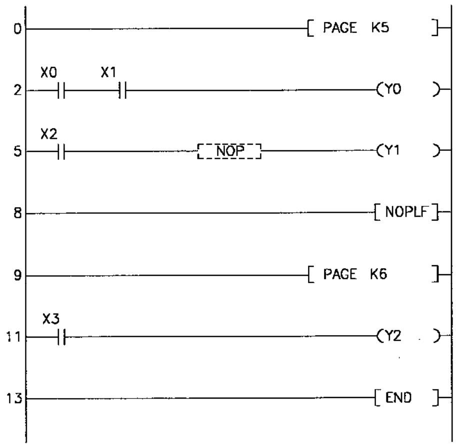
- Printing an instruction list with the NOPLF instruction will result in the following:

```
0    LD       X000
1    MOV      K1
              D30
3    MOV      K2
              D40
5    NOPLF
                -1-
6    LD       X001
7    OUT      Y040
8    END
```

Changes pages after printing NOPLF

PAGE n

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	PAGE	K5
2	LD	X0
3	AND	X1
4	OUT	Y0
5	LD	X2
6	NOP	
7	OUT	Y1
8	NOPLF	
9	PAGE	K6
11	LD	X3
12	OUT	Y2
13	END	

6. BASIC INSTRUCTIONS

The following types of basic instructions are available.

Instruction	Meaning	Reference Section
Comparison operation instruction	Compare data to data	6.1
Arithmetic operation instructions	Adds, subtracts multiplies, divides, increments, or decrements data with other data	6.2
Data conversion instruction	Converts data types	6.3
Data transfer instruction	Transmits designated data	6.4
Program branch instruction	Program jumps	6.5
Program execution control instructions	Enables and disables program interrupts	6.6
Refresh instruction	Refreshes bit devices	6.7

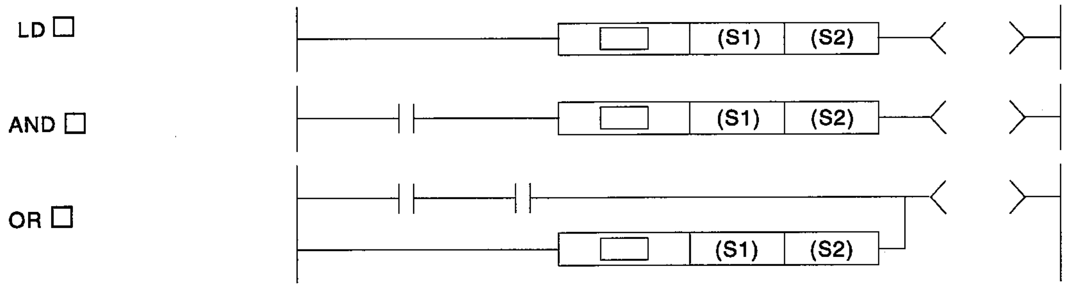
6.1 Comparison Operation Instruction

6.1.1 BIN 16-bit data comparisons

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o								—
(S2)	o								—

[Instruction Symbol] [Execution Condition]

□ indicates the signs =, <>, >, <=, <, >=



Set Data

Set Data	Meaning	Data Type
(S1)	Comparative data, or device number where comparative data is stored	BIN 16 bits
(S2)		

Functions

(1) Treats BIN 16-bit data from device designated by (S1) and BIN 16-bit data from device designated by (S2) as an A contact, and performs comparison operation.

(2) The results of the comparison operations for the individual instructions are as follows:

Instruction Symbol in☐	Condition	Conparison Operation Result	Instruction Symbol in☐	Condition	Comparison Operation Result
=	(S1)=(S2)	Continuity	=	(S1)≠(S2)	Non-continuity
<>	(S1)≠(S2)		<>	(S1)=(S2)	
>	(S1)>(S2)		>	(S1)≤(S2)	
<=	(S1)≤(S2)		<=	(S1)>(S2)	
<	(S1)<(S2)		<	(S1)≥(S2)	
>=	(S1)≥(S2)		>=	(S1)<(S2)	

(3) In cases where hexadecimal constants have been designated by (S1) and (S2), or when a numerical value (8 to F) where the highest bit (b15) will be 1 has been designated, the value will be read as a negative BIN value number for purposes of the comparison.

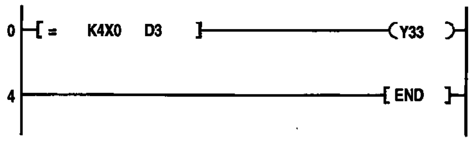
Operation Errors

(1) There are no operation errors associated with the =, <>, >, <=, <, or >= instructions.

Program Example

(1) The following program compares the data at X0 to XF with the data at D3, and turns Y33 ON if the data is identical.

[Ladder Mode]

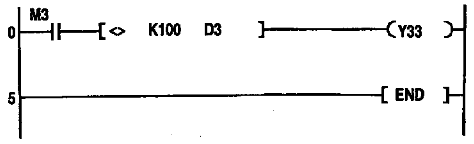


[List Mode]

Steps	Instruction	Device
0	LD=	K4X0
		D3
3	OUT	Y33
4	END	

(2) The following program compares BIN value K100 to the data at D3, and establishes continuity if the data in D3 is something other than 100.

[Ladder Mode]

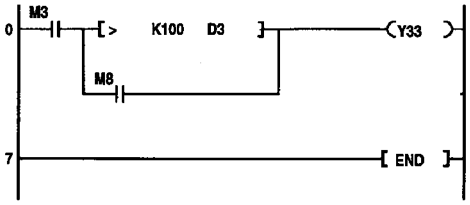


[List Mode]

Steps	Instruction	Device
0	LD	M3
1	AND<>	K100
		D3
4	OUT	Y33
5	END	

(3) The following program compares the BIN value 100 with the data in X0 to XF, and establishes continuity if the D3 data is less than 100.

[Ladder Mode]

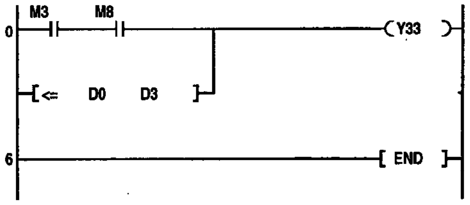


[List Mode]

Steps	Instruction	Device
0	LD	M3
1	LD>	K100
		D3
4	OR	M8
5	ANB	
6	OUT	Y33
7	END	

(4) The following program compares the data in D0 and D3, and if the data in D0 is equal to or less than the data in D3, establishes continuity.

[Ladder Mode]



[List Mode]

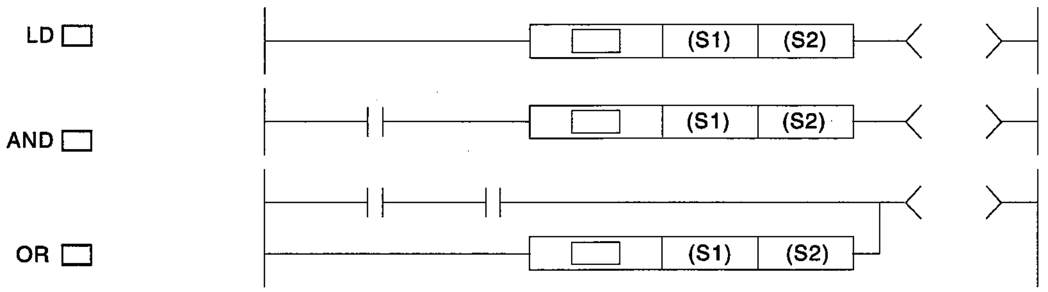
Steps	Instruction	Device
0	LD	M3
1	AND	M8
2	OR<=	D0
		D3
5	OUT	Y33
6	END	

6.1.2 BIN 32-bit data comparisons

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o								—
(S2)	o								—

[Instruction Symbol] [Execution Condition]

□ indicates D=, D<>, D>, D<=, D<, or D>=



Set Data

Set Data	Meaning	Data Type
(S1)	Comparative data, or device number where comparative data is stored	Bit 32 bits
(S2)		

Functions

(1) Treats BIN 32-bit data from device designated by (S1) and BIN 32-bit data from device designated by (S2) as an A contact, and performs comparison operation.

(2) The results of the comparison operations for the individual instructions are as follows:

Instruction Symbol in ☐	Condition	Conparison Operation Result	Instruction Symbol in ☐	Condition	Comparison Operation Result
D=	(S1)=(S2)	Continuity	D=	(S1)≠(S2)	Non-continuity
D<>	(S1)≠(S2)		D<>	(S1)=(S2)	
D>	(S1)>(S2)		D>	(S1)≤(S2)	
D<=	(S1)≤(S2)		D<=	(S1)>(S2)	
D<	(S1)<(S2)		D<	(S1)≥(S2)	
D>=	(S1)≥(S2)		D>=	(S1)<(S2)	

(3) In cases where hexadecimal constants have been designated by (S1) and (S2), or when a numerical value (8 to F) where the highest bit (b31) will be 1 has been designated, the value will be read as a negative BIN value number for the purposes of the comparison.

(4) Data used for comparison should be designated by a 32-bit instruction (DMOV instruction, etc.).
If designation is made with a 16-bit instruction (MOV instruction, etc.), comparisons of large and small values cannot be performed correctly.

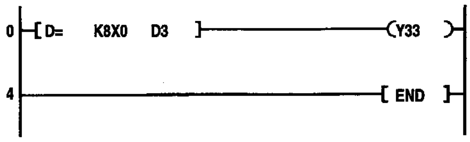
Operation Errors

(1) There are no operation errors associated with the =, <>, >, <=, <, or >= instructions.

Program Example

(1) The following program compares the data at X0 to XF with the data at D3, and turns Y33 ON if the data is identical.

[Ladder Mode]

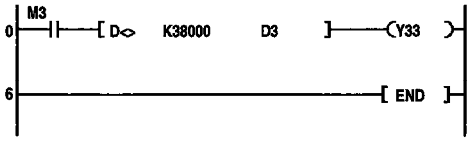


[List Mode]

Steps	Instruction	Device
0	LDD=	K8X0 D3
3	OUT	Y33
4	END	

(2) The following program compares BIN value K38000 to the data at D3 and D4, and establishes continuity if the data in D3 and D4 is something other than 38000.

[Ladder Mode]

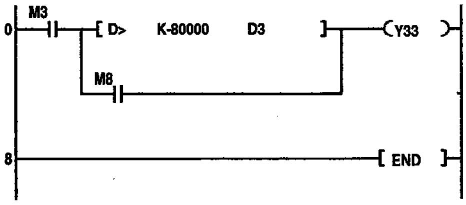


[List Mode]

Steps	Instruction	Device
0	LD	M3
1	ANDD<>	K38000 D3
5	OUT	Y33
6	END	

(3) The following program compares BIN value K-80000 to the data at D3 and D4, and establishes continuity if the data in D3 and D4 is less than -80000.

[Ladder Mode]

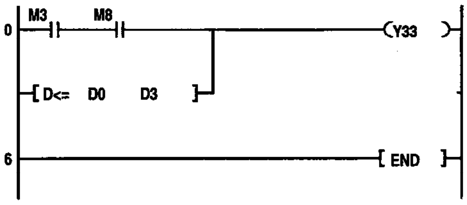


[List Mode]

Steps	Instruction	Device
0	LD	M3
1	LDD>	K-80000
		D3
5	OR	M8
6	ANB	
7	OUT	Y33
8	END	

(4) The following program compares the data in D0 and D1 with the data in D3 and D4, and establishes continuity if the data in D0 and D1 is equal to or less than the data in D3 and D4.

[Ladder Mode]



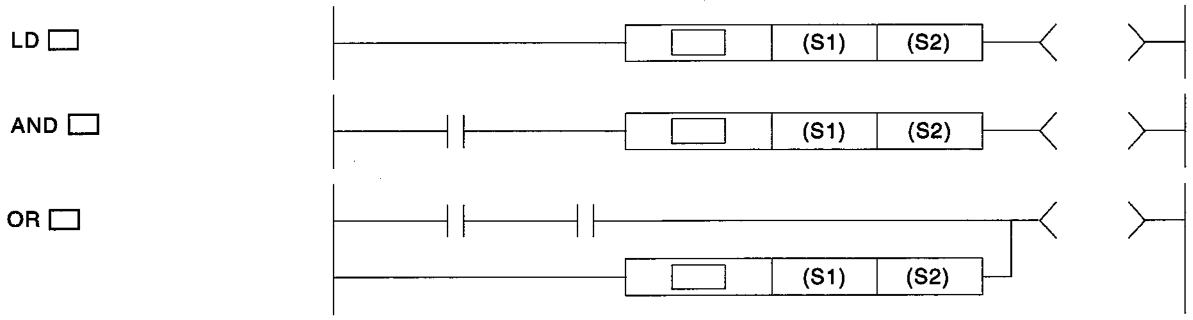
[List Mode]

Steps	Instruction	Device
0	LD	M3
1	AND	M8
2	ORD<=	D0
		D3
5	OUT	Y33
6	END	

6.1.3 Floating decimal point data comparisons

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S1)	—	○		—	○		—	○	—
(S2)	—	○		—	○		—	○	—

[Instruction Symbol] [Execution Condition] □ indicates E=, E<>, E>, E<=, E<, or E>=



Set Data

Set Data	Meaning	Data Type
(S1)	Comparative data, or device number where comparative data is sotred	Real number
(S2)		

Functions

(1) The floating decimal point data from device designated by (S1) and floating decimal point data from device designated by (S2) as A contact, and performs comparison operation.

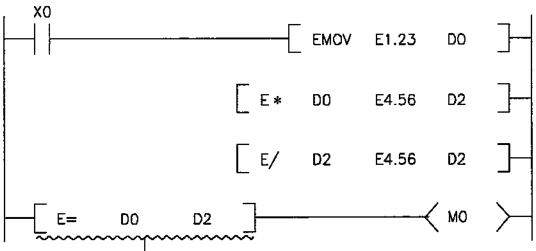
(2) The results of the comparison operations for the individual instructions are as follows.

Instruction Symbol in ☐	**Condition**	**Comparison Operation Result**	**Instruction Symbol in ☐**	**Condition**	**Comparison Operation Result**
E=	(S1)=(S2)	Continuity	E=	(S1)≠(S2)	Non-continuity
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)	

POINT

Note that use of the = instruction can on occasion result in situations where errors cause the two values to not be equal.

Example



Something not equal

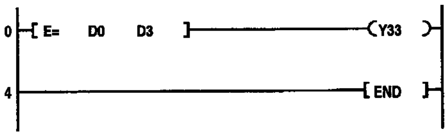
Operation Errors

(1) There are no operation errors associated with the E=, E<>, E>, E<=, E<, or, E>= instructions.

Program Example

(1) The following program compares floating decimal point real number data at D0 and D1 to floating decimal point real number data at D3 and D4.

[LadderMode]

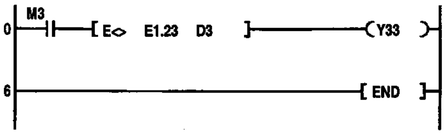


[List Mode]

Steps	Instruction	Device
0	LDE=	D0
		D3
3	OUT	Y33
4	END	

(2) The following program compares the floating decimal point real number 1.23 to the floating decimal point real number data at D3 and D4.

[Ladder Mode]

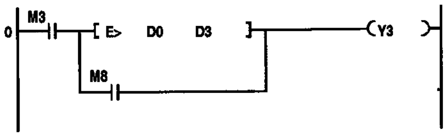


[List Mode]

Steps	Instruction	Device
0	LD	M3
1	ANDE<>	E1.23
		D3
5	OUT	Y33
6	END	

(3) The following program compares floating decimal point real number data at D0 and D1 to floating decimal point real number data at D3 and D4.

[Ladder Mode]

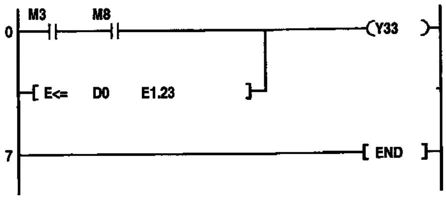


[List Mode]

Steps	Instruction	Device
0	LD	M3
1	LDE>	D0
		D3
4	OR	M8
5	ANB	
6	OUT	Y3
7	END	

(4) The following program compares the floating decimal point data at D0 and D1 to the floating decimal point real number 1.23.

[Ladder Mode]



[List Mode]

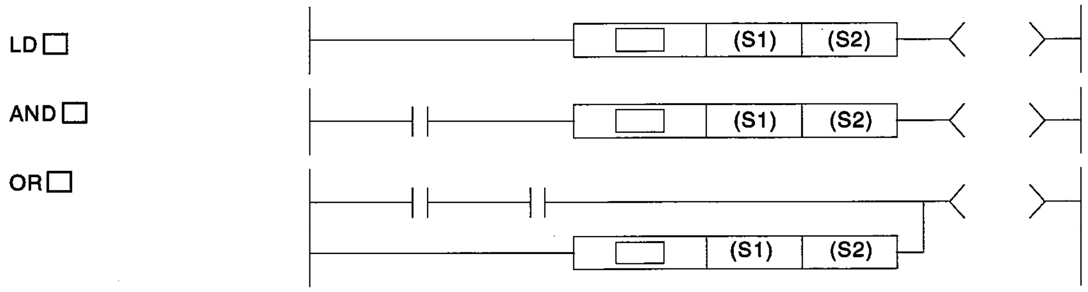
Steps	Instruction	Device
0	LD	M3
1	AND	M8
2	ORE<=	D0
		E1.23
6	OUT	Y33
7	END	

6.1.4 Character string data comparisons

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S1)	—	○		—				○	—
(S2)	—	○		—				○	—

[Instruction Symbol] [Execution Condition]

□ indicates $=, $<>, $>, $<=, $<, or $>=



Set Data

Set Data	Meaning	Data Type
(S1)	First number of comparison data, or of the device where comparison data is being stored.	Character string
(S2)		

Functions

(1) Treats character string data stored following the device designated by (S1) and character string data stored following the device designated by (S2) as A contact, and performs comparison operation.

(2) A comparison operation involves the character-by-character comparison of the ASCII code of the first character in the character string.

(3) The (S1) and (S2) character strings encompass all characters from the designated device number to the next device number storing the code "00H".

(a) If all character strings match, the comparison result will be matched.

	b15 ---- b8	b7 ---- b0
(S1)	42H (B)	41H (A)
(S1)+1	44H (D)	43H (C)
(S1)+2	00H	45H (E)

"ABCDE"

☐

	b15 ---- b8	b7 ---- b0
(S2)	42H (B)	41H (A)
(S2)+1	44H (D)	43H (C)
(S2)+2	00H	45H (E)

"ABCDE"

Instruction Symbol in☐	Comparison Operation Result	Instruction Symbol in☐	Comparison Operation Result
$=	Continuity	$<=	Continuity
$<>	Non-continuity	$<	Non-continuity
$>	Non-continuity	$>=	Continuity

(b) If the character strings are different, the character string with the larger character code will be the larger.

	b15 ---- b8	b7 ---- b0
(S1)	42H (B)	41H (A)
(S1)+1	44H (D)	43H (C)
(S1)+2	00H	46H (F)

"ABCDE"

☐

	b15 ---- b8	b7 ---- b0
(S2)	42H (B)	41H (A)
(S2)+1	44H (D)	43H (C)
(S2)+2	00H	45H (E)

"ABCDE"

Instruction Symbol in☐	Comparison Operation Result	Instruction Symbol in☐	Comparison Operation Result
$=	Non-continuity	$<=	Non-continuity
$<>	Continuity	$<	Non-continuity
$>	Continuity	$>=	Continuity

(c) If the character strings are different, the first different sized character code will determine whether the character string is larger or smaller.

	b15 ---- b8	b7 —— b0
(S1)	32H (2)	31H (A)
(S1)+1	33H (3)	34H (4)
(S1)+2	00H	35H (5)

"12<u>4</u>35"

☐

	b15 ---- b8	b7 —— b0
(S2)	32H (2)	31H (1)
(S2)+1	34H (4)	33H (3)
(S2)+2	00H	35H (5)

"12<u>3</u>46"

Instruction Symbol in☐	Comparison Operation Result	Instruction Symbol in☐	Comparison Operation Result
$=	Non-Continuity	$<=	Non-continuity
$<>	Continuity	$<	Non-continuity
$>	Continuity	$>=	Continuity

(4) If the character strings designated by (S1) and (S2) are of different lengths, the data with the longer character string will be larger.

	b15 ---- b8	b7 ---- b0
(S1)	32H (2)	31H (1)
(S1)+1	34H (4)	33H (3)
(S1)+2	36H (6)	35H (5)
(S1)+3	00H	37H (7)

"123456<u>7</u>"

☐

	b15 ---- b8	b7 ---- b0
(S2)	32H (2)	31H (1)
(S2)+1	34H (4)	33H (3)
(S2)+2	36H (6)	35H (5)
(S2)+3	00H	00H

"123456"

Instruction Symbol in☐	Comparison Operation Result	Instruction Symbol in☐	Comparison Operation Result
$=	Non-continuity	$<=	Continuity
$<>	Continuity	$<	Continuity
$>	Non-continuity	$>=	Non-continuity

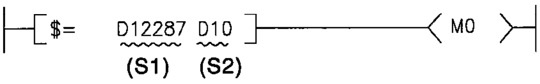
Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.
- The code "00H" does not exist within the relevant device range following the device number designated by (S1) or (S2). (Error code: 4101)

POINT

At the same time that it is conducting a character string comparison, character string data comparison instruction also checks the device range.
For this reason, even in cases where the character string exceeds the device range, character string data is compared and non-matching cases within the device range are detected and the comparison operation results are output without returning an operation error.

Example



(S1) data

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

(S2) data

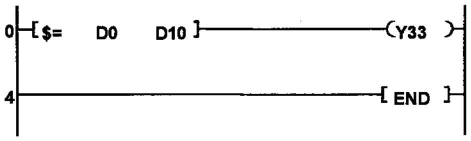
D10	"Z"	"A"
D11	00H	"C"

* In the example shown above, the (S1) character string exceeds the device range, but because its second character is different from that of (S2), the comparison result is "(S1) does not equal (S2)", and the operation result is non-continuity.
In this case, because the non-continuity detection is for D12287 (inside the device range), there will be no operation error returned.

Program Example

(1) The following program compares character strings stored following D0 and characters following D10.

[Ladder Mode]

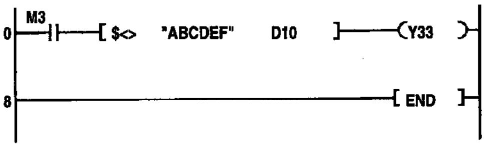


[List Mode]

Steps	Instruction	Device
0	LD$=	D0 D10
3	OUT	Y33
4	END	

(2) The following program compares the character string "ABCDEF" with the character string stored following D10.

[Ladder Mode]

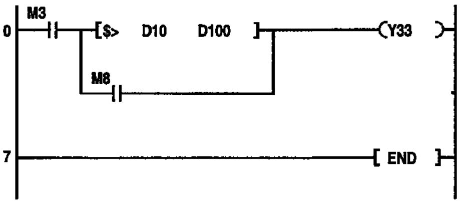


[List Mode]

Steps	Instruction	Device
0	LD	M3
1	AND$<>	"ABCDEF" D10
7	OUT	Y33
8	END	

(3) The following program compares the character string stored following D10 with the character string stored following D100.

[Ladder Mode]

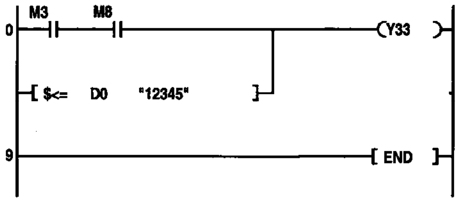


[List Mode]

Steps	Instruction	Device
0	LD	M3
1	LD$>	D10 D100
4	OR	M8
5	ANB	
6	OUT	Y33
7	END	

(4) The following program compares the character string stored following D10 with the character string "12345."

[Ladder Mode]



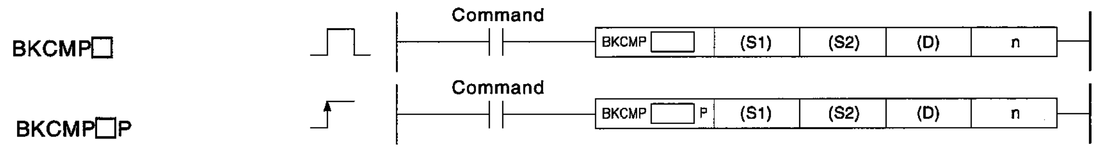
[List Mode]

Steps	Instruction	Device
0	LD	M3
1	AND	M8
2	OR$<=	D0 "12345"
8	OUT	Y33
9	END	

6.1.5 BIN block data comparisons

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—				o	—
(S2)	—	o		—				—	—
(D)	o	o		—				—	—
n	o	o		o				o	—

[Instruction Symbol] [Execution Condition] □ indicates the signs =, <>, >, <=, <, >=



Set Data

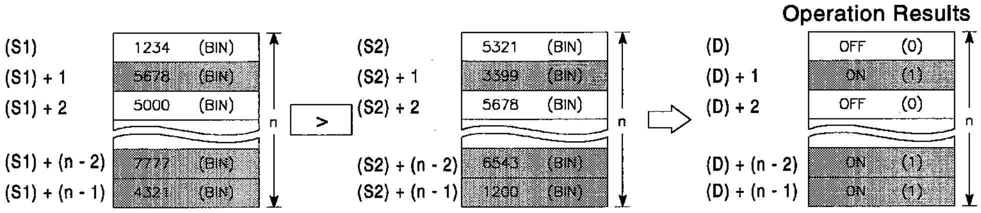
Set Data	Meaning	Data Type
(S1)	Data being compared, or first number of the device where the data being compared is being stored	BIN 16 bits
(S2)	First number of the device where the comparison data is being stored	BIN 16 bits
(D)	First number of the device where the results of the comparison operation are being stored	Bit
n	Number of data blocks compared	BIN 16 bits

Functions

(1) Compares BIN 16-bit data the nth point from the device number designated by (S1) with BIN 16-bit data the nth point from the device number designated by (S2), and stores the result from the device designated by (D) onward.

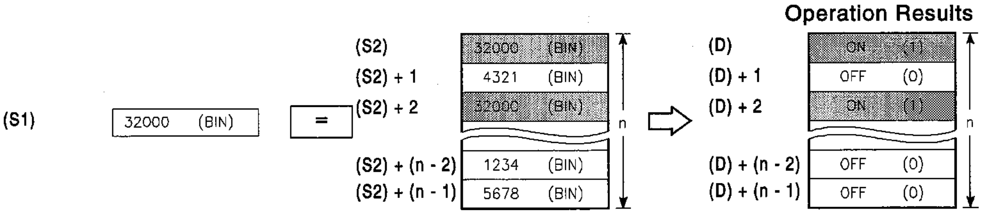
(a) If the comparison condition has been met, the device designated by (D) will be turned ON.

(b) If the comparison condition has not been met, the device designated by (D) will be turned OFF.



(2) The comparison operation is conducted in 16-bit units.

(3) The constant designated by (S1) can be between -32768 and 32767 (BIN 16-bit data).



(4) The results of the comparison operations for the individual instructions are as follows:

Instruction Symbols	Condition	Comparison Operation Result	Instruction Symbols	Condition	Comparison Operation Result
BKCMP=	(S1) = (S2)	ON(1)	BKCMP=	(S1) ≠ (S2)	OFF(0)
BKCMP<>	(S1) ≠ (S2)		BKCMP<>	(S1) = (S2)	
BKCMP>	(S1) > (S2)		BKCMP>	(S1) ≤ (S2)	
BKCMP<=	(S1) ≤ (S2)		BKCMP<=	(S1) > (S2)	
BKCMP<	(S1) < (S2)		BKCMP<	(S1) ≥ (S2)	
BKCMP>=	(S1) ≥ (S2)		BKCMP>=	(S1) < (S2)	

(5) If all comparison results stored n-points from (D) are ON (1), SM704 (block comparison signal) goes ON.

(6) If the device designated by (D) is already ON (1), that device will not change.
If the conditions designated by (S1) and (S2) are changed and the BKCMP (P) instruction is executed, the device designated by (D) should be turned OFF.

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

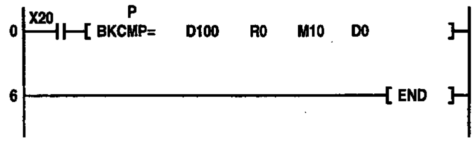
- The range of the device n points from a device designated by (S1), (S2), or (D) exceeds the relevant device. (Error code: 4101)

(2) See Section 3.6 for information regarding errors during index qualification.

Program Example

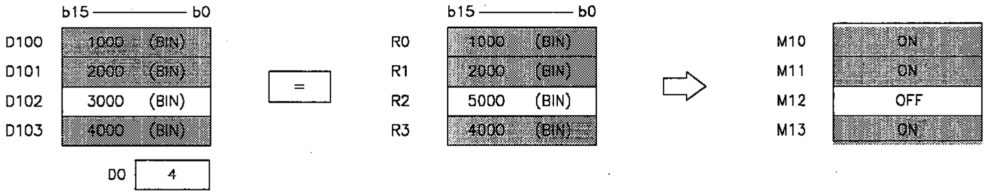
(1) The following program performs a comparison operation when X20 goes ON, comparing the data for the number of points from D100 equivalent to the value stored in D0 with the data the number of points from R0 equivalent to the value stored in D0, and stores the result from M10 onward.

[Ladder Mode]



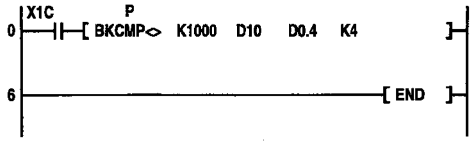
[List Mode]

Steps	Instruction	Device
0	LD	X20
1	BKCMP=P	D100 R0 M10



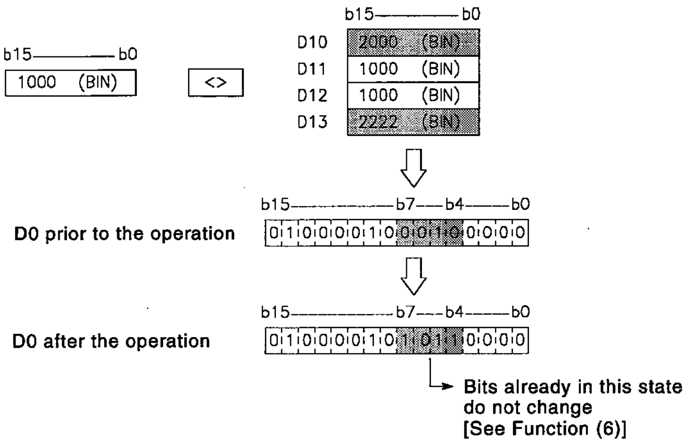
(2) The following program performs a comparison operation when X1C goes ON, comparing the constant K1000 with the data 4 points from D10, and stores the result in b4 through b7 of D0.

[Ladder Mode]



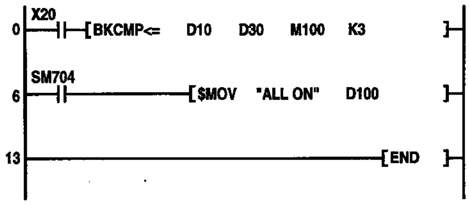
[List Mode]

Steps	Instruction	Device
0	LD	X20
1	BKCMP=P	D100 R0 M10 D0
6	END	



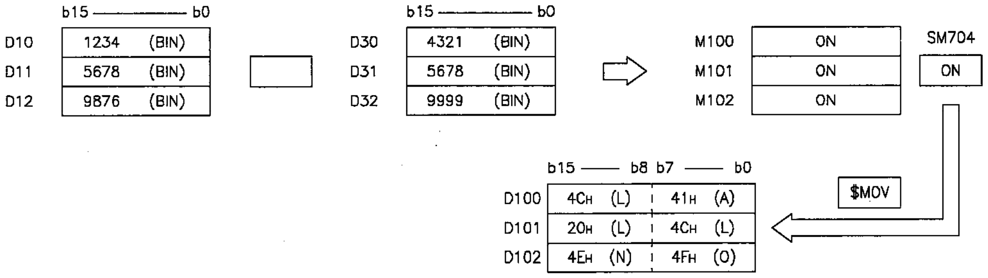
(3) When X20 goes ON, compares the data 3 points from D10 with the data 3 points from D30, and stores the result from M100 onward. The following program transfers the character string "ALL ON" to D100 onward when all devices from M100 onward have reached the 1 "ON" state.

[Ladder Mode]



[List Mode]

```
Steps    Instruction    Device
  0      LD             X20
  1      BKCMP<=        D10
                        D30
                        M100
                        K3
  6      LD             SM704
  7      $MOV           "ALL ON"
                        D100
 13      END
```

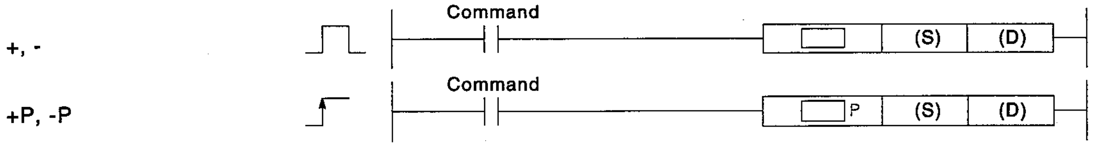


6.2 Arithmetic Operation Instructions

6.2.1 BIN 16-bit addition and subtraction operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition] □ indicates the signs +/-



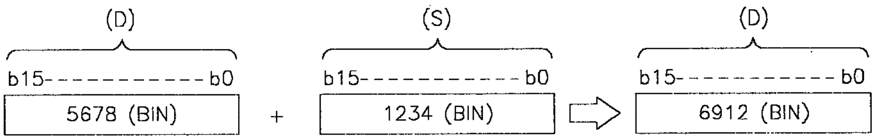
Set Data

Set Data	Meaning	Data Type
(S)	Addition or subtraction data, or first number of device storing addition or subtraction data	BIN 16 bits
(D)	Data to be added to or subtracted from, or first number of device storing such data	

Functions

+

(1) Adds 16-bit BIN data designated by (D) to 16-bit BIN data designated by (S) and stores the result of the addition at the device designated by (D).



(2) Values for (S) and (D) can be designated between -32768 and 32767 (BIN, 16 bits).

(3) The judgment of whether data is positive or negative is made by the most significant bit (b15).

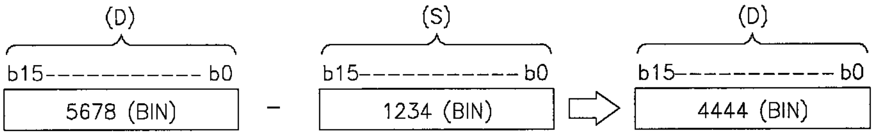
- 0 . . . . Positive
- 1 . . . . Negative

(4) The following will happen when an underflow or overflow is generated in an operation result:
The carry flag in this case does not go ON.

- K32767 + K2 ——→ K-32767 . . . . . A negative value is generated
  (H7FFF) (H0002) (H8001) if b15 is 1.
- K-32768 + K-2 ——→ K-32766 . . . . . A positive value is generated
  (H8000) (HFFFE) (7FFE) if b15 is 0.

−

(1) Subtracts 16-bit BIN data designated by (D) from 16-bit BIN data designated by (S) and stores the result of the subtraction at the device designated by (D).



(2) Values for (S) and (D) can be designated between -32768 and 32767 (BIN, 16 bits).

(3) The judgment of whether data is positive or negative is made by the most significant bit (b15).

- 0 . . . . Positive
- 1 . . . . Negative

(4) The following will happen when an underflow or overflow is generated in an operation result:
The carry flag in this case does not go ON.

- K-32768 - K2 ——→ K32766. . . . . . A positive value is generated
  (H8000) (H0002) (H7FFE) if b15 is 0.
- K32767 - K-2 ——→ K-32767 . . . . . A negative value is generated
  (H7FFF) (H0002) (H8001) if b15 is 1.

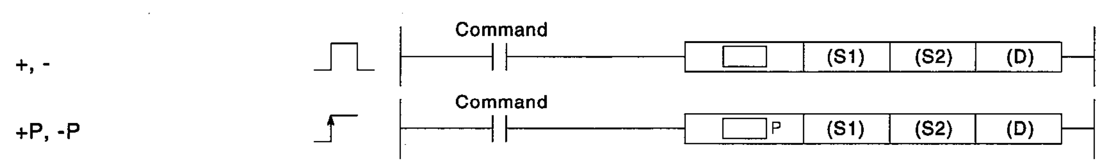
Operation Errors

(1) There are no operation errors associated with the +(P) or -(P) instructions.

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o							o	—
(S2)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]

□ indicates the signs +/-



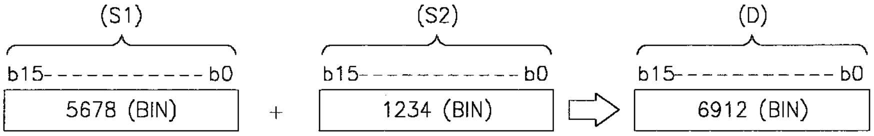
Set Data

Set Data	Meaning	Data Type
(S1)	Data to be added to or subtracted from, or the first number of the device storing such data	BIN 16 bits
(S2)	Addition or subtraction data, or first number of device storing addition or subtraction data	
(D)	First number of device storing addition or subtraction data	

Functions

+

(1) Adds 16-bit BIN data designated by (S1) to 16-bit BIN data designated by (S2) and stores at the device designated by (D).



(2) Values for (S1), (S2) and (D) can be designated from -32768 to 32767 (BIN 16 bits).

(3) The judgment of whether data is positive or negative is made by the most significant bit (b15).

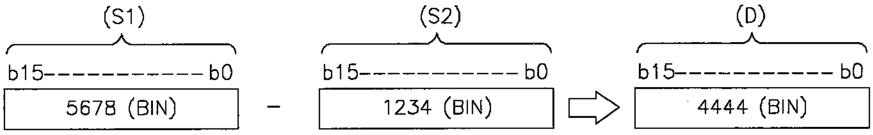
- 0 . . . . Positive
- 1 . . . . Negative

(4) The following will happen when an underflow or overflow is generated in an operation result:
The carry flag in this case does not go ON.

- K32767 + K2 ——→ K-32767. . . . . A negative value is generated
  (H7FFF) (H0002) (H8001) . . . . . if b15 is 1.
- K-32768 + K-2 ——→ K-32766. . . . . A positive value is generated
  (H8000) (HFFFE) (7FFE) if b15 is 0.

−

(1) Subtracts 16-bit BIN data designated by (S1) from 16-bit BIN data designated by (S2) and stores the result of the subtraction at the device designated by (D).



(2) Values for (S1), (S2) and (D) can be designated from -32768 and 32767 (BIN 16 bits).

(3) The judgment of whether data is positive or negative is made by the most significant bit (b15).

- 0 .... Positive
- 1 .... Negative

(4) The following will happen when an underflow or overflow is generated in an operation result:
The carry flag in this case does not go ON.

- K-32768 - K2 ——K32766...... A positive value is generated
  (H8000) (H0002) (7FFE) . if b15 is 0.
- K32767 - K-2 ——K-32767..... A negative value is generated
  (H7FFF) (H0002) (H8001) if b15 is 1.

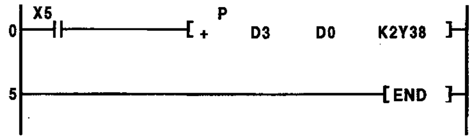
Operation Errors

(1) There are no operation errors associated with the +(P) or -(P) instructions.

Program Example

(1) The following program adds the contents of D3 and the contents of D0 when X5 goes ON, and outputs result to Y30 through 3F.

[Ladder Mode]

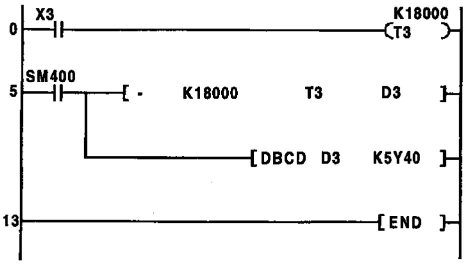


[List Mode]

Steps	Instruction	Device
0	LD	X5
1	+P	D3 D0 K2Y38
5	END	

(2) The following program outputs the difference between the set value for timer T3 and its present value to Y40 to 53.

[Ladder Mode]



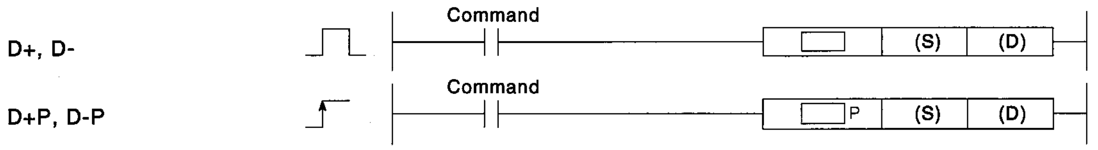
[List Mode]

Steps	Instruction	Device
0	LD	X3
1	OUT	T3 K18000
5	LD	SM400
6	-	K18000 T3 D3
10	DBCD	D3 K5Y40
13	END	

6.2.2 BIN 32-bit addition and subtraction operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition] □ indicates D+/D-



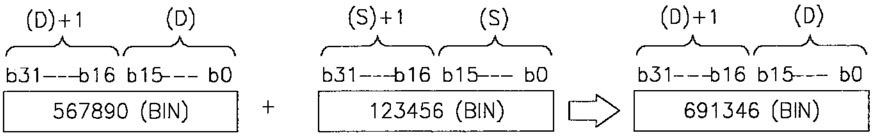
Set Data

Set Data	Meaning	Data Type
(S)	Addition or subtraction data, or first number of device storing addition or subtraction data	BIN 32 bits
(D)	Data to be added to or subtracted from, or first number of device storing such data	

Functions

D+

(1) Adds 32-bit BIN data designated by (D) to 32-bit BIN data designated by (S), and stores the result of the addition at the device designated by (D).



(2) The values for (S) and (D) can be designated at between -2147483648 and 2147483647 (BIN 32 bits).

(3) Judgment of whether the data is positive or negative is made on the basis of the most significant bit (b31).

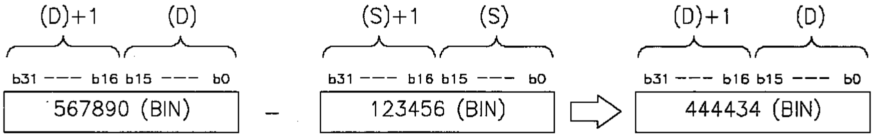
- 0 .... Positive
- 1 .... Negative

(4) The following will happen when an underflow or overflow is generated in an operation result:
The carry flag in this case does not go ON.

- K2147483647 + K2 → K-2147483647 .... Because b31 is 1,
  (H7FFFFFFF) (H2) (H80000001) the value is negative.
- K-2147483648 + K-2 → K2147483646 .... Because b31 is 0,
  (H80000000) (HFFFE) (H7FFFFFFE) the value is positive.

D-

(1) Subtracts 32-bit BIN data designated by (D) from 32-bit BIN data designated by (S) and stores the result of the subtraction at the device designated by (D).



(2) The values for (S) and (D) can be designated at between -2147483648 and 2147483647 (BIN 32 bits).

(3) Judgment of whether the data is positive or negative is made on the basis of the most significant bit (b31).

- 0 .... Positive
- 1 .... Negative

(4) The following will happen when an underflow or overflow is generated in an operation result:
The carry flag in this case does not go ON.

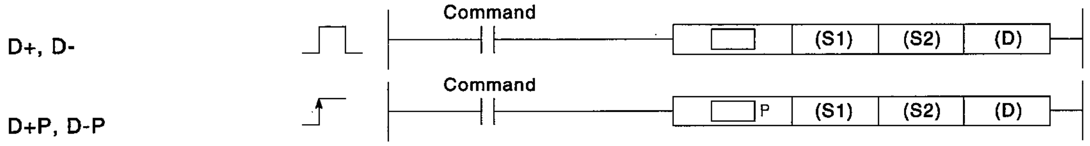
- K-2147483648 - K2 → K2147483646 .... Because b31 is 0,
  (H80000000) (H2) (H7FFFFFFE) the value is positive.
- K2147483647 - K-2 → K-2147483647 .... Because b31 is 1,
  (7FFFFFFF) (HFFFE) (H80000001) the value is negative.

Operation Errors

(1) There are no operation errors associated with the +(P) or -(P) instructions.

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o							o	—
(S2)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition] □ indicates D+/D-



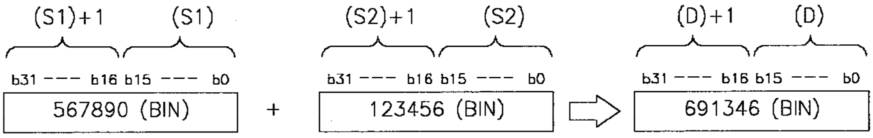
Set Data

Set Data	Meaning	Data Type
(S1)	Data to be added to or subtracted from, or the first number of the device storing such data	BIN 32 bits
(S2)	Addition or subtraction data, or first number of device storing addition or subtraction data	
(D)	First number of device storing addition or subtraction data	

Functions

D+

(1) Adds 32-bit BIN data designated by (S1) to 32-bit BIN data designated by (S2) and stores at the device designated by (D).



(2) The values for (S1), (S2), and (D) can be designated at between -2147483648 and 2147483647 (BIN 32 bits).

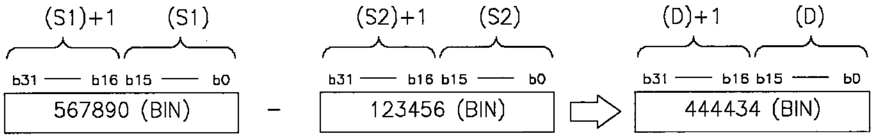
(3) Judgment of whether the data is positive or negative is made on the basis of the most significant bit (b31).

(4) The following will happen when an underflow or overflow is generated in an operation result:
The carry flag in this case does not go ON.

- K2147483647 + K2——K-2147483647 .... Because b31 is 1,
  (H7FFFFFFF) (H2) (H80000001) the value is negative.
- K-2147483648 + K-2—K2147483646 .... Because b31 is 0,
  (H80000000) (HFFFE) (H7FFFFFFE) the value is positive.

D-

(1) Subtracts 32-bit BIN data designated by (S1) from 32-bit BIN data designated by (S2) and stores the result of the subtraction at the device designated by (D).



(2) The values for (S1), (S2), and (D) can be designated at between -2147483648 and 2147483647 (BIN 32 bits).

(3) Judgment of whether the data is positive or negative is made on the basis of the most significant bit (b31).

- 0 . . . . Positive
- 1 . . . . Negative

(4) The following will happen when an underflow or overflow is generated in an operation result:
The carry flag in this case does not go ON.

- K-2147483648 - K2 → K2147483646 . . . . Because b31 is 0,
  (H80000000) (H2) (H7FFFFFFE) the value is positive.
- K2147483647 - K-2 → K-2147483647 . . . . Because b31 is 1,
  (H7FFFFFFF) (HFFFE) (H80000001) the value is negative.

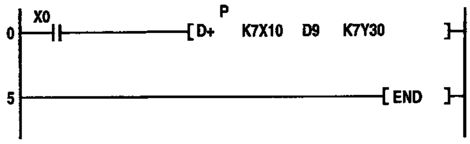
Operation Errors

(1) There are no operation errors associated with the +(P) or -(P) instructions.

Program Example

(1) The following program adds 28-bit data from X10 to X2B to the data at D9 and D10 when X0 goes ON, and outputs the result of the operation to Y30 to Y4B.

[Ladder Mode]

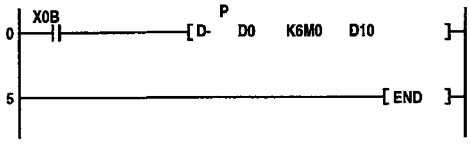


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	D+P	K7X10 D9 K7X30
5	END	

(2) The following program subtracts the data from M0 to M23 from the data at D0 and D1 when XB goes ON, and stores the result at D10 and D11.

[Ladder Mode]



[List Mode]

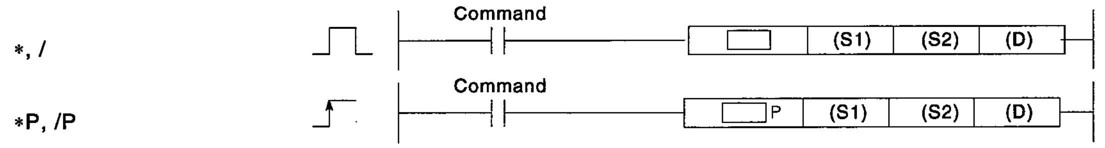
Steps	Instruction	Device
0	LD	X0B
1	D-P	D0 K6M0 D10
5	END	

6.2.3 BIN 16-bit multiplication and division operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o							o	—
(S2)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]

□ indicates * or /



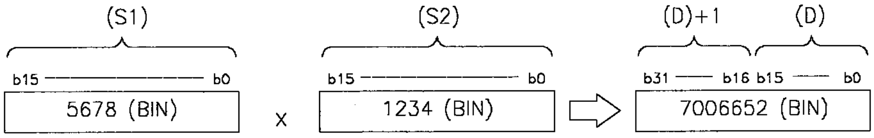
Set Data

Set Data	Meaning	Data Type
(S1)	Data that will be multiplied or divided, or the first number of the device storing data that will be multiplied or divided	BIN 16 bits
(S2)	Data to multiply or divide by, or the first number of device storing such data	
(D)	First number of the device storing the operation results of multiplication or division operation	BIN 32 bits

Functions

*

(1) Multiplies BIN 16-bit data designated by (S1) and BIN 16-bit data designated by (S2), and stores the result in the device designated by (D).



(2) If (D) is a bit device, designation is made from the lower bits.
Example K1 . . . Lower 4 bits (b0 to 3)
K4 . . . Lower 16 bits (b0 to 15)
K8 . . . 32 bits (b0 to 31)

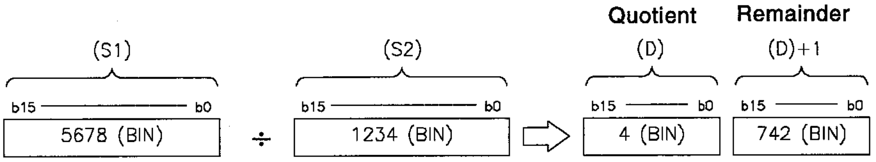
(3) The values for (S1), (S2), and (D) can be designated at between -32768 and 32767 (BIN 16 bits).

(4) Judgments whether (S1), (S2), and (D) are positive or negative are made on the basis of the most significant bit (b15 for (S1) and (S2), for (D) and b31).

- 0 . . . . Positive
- 1 . . . . Negative

/

(1) Divides BIN 16-bit data designated by (S1) and BIN 16-bit data designated by (S2), and stores the result in the device designated by (D).



(2) If a word device has been used, the result of the division operation is stored as 32 bits, and both the quotient and remainder are stored; if a bit device has been used, 16 bits are used and only the quotient is stored.
Quotient..... Stored at the lower 16 bits
Remainder... Stored at the higher 16 bits
(Can be stored only when a word device has been used)

(3) The values for (S1), (S2), and (D) can be designated at between -32768 and 32767 (BIN 16 bits).

(4) Judgment whether values for (S1), (S2), and (D) are positive or negative is made on the basis of the most significant bit (b15 for (S1) and (S2), and b15 for (D)).

- 0 .... Positive
- 1 .... Negative

Operation Errors

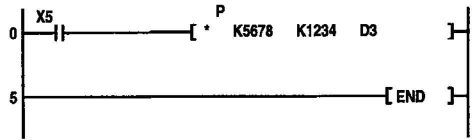
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Attempt to divide (S2) by 0. (Error code: 4100)

Program Example

(1) The following program divides "5678" by "1234" when X5 goes ON, and stores the result at D3 and D4.

[Ladder Mode]

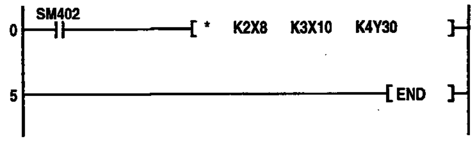


[List Mode]

Steps	Instruction	Device
0	LD	X5
1	*P	K5678 K1234 D3
5	END	

(2) The following program divides BIN data at X8 through XF by BIN data at X10 through X1B, and outputs the result of the division operation to Y30 through Y3F.

[Ladder Mode]

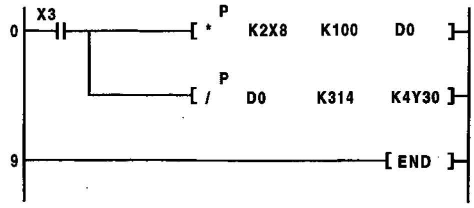


[List Mode]

Steps	Instruction	Device
0	LD	SM402
1	*	K2X8 K3X10 K4Y30
5	END	

(3) The following program outputs the value resulting when the data at X8 through XF is divided by 3.14 to Y30 through Y3F when X3 is ON.

[Ladder Mode]



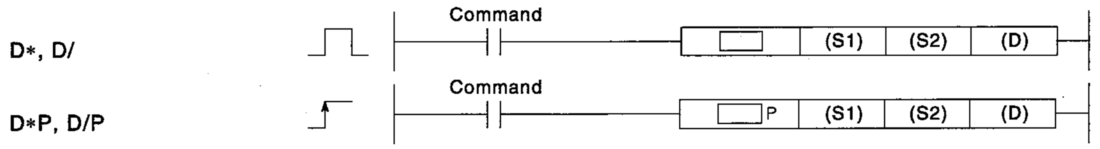
[List Mode]

Steps	Instruction	Device
0	LD	X3
1	*P	K2X8 K100 D0
5	/P	D0 K314 K4Y30
9	END	

6.2.4 BIN 32-bit multiplication and division operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o			o					—
(S2)	o			o					—
(D)	o			—					—

[Instruction Symbol] [Execution Condition] □ indicates D*/D/



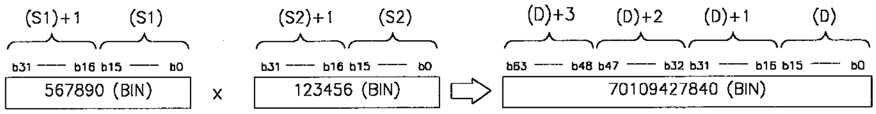
Set Data

Set Data	Meaning	Data Type
(S1)	Data that will be multiplied or divided, or the first number of the device storing data that will be multiplied or divided	BIN 32 bits
(S2)	Data to multiply or divide by, or the first number of device storing such data	
(D)	First number of the device storing the operation results of multiplication or division operation	BIN 64 bits

Functions

D*

(1) Multiplies BIN 32-bit data designated by (S1) and BIN 32-bit data designated by (S2), and stores the result in the device designated by (D).



(2) If (D) is a bit device, only the lower 32 bits of the multiplication result will be considered, and the upper 32 bits cannot be designated.

Example K1 . . . Lower 4 bits (b0 to 3)
K4 . . . Lower 16 bits (b0 to 15)
K8 . . . Lower 32 bits (b0 to 31)

If the upper 32 bits of the bit device are required for the result of the multiplication operation, first temporarily store the data in a word device, then transfer the word device data to the bit device by designating ((D)+2) and ((D)+3) data.

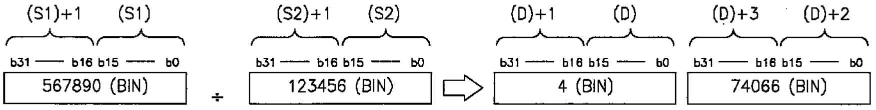
(3) The values for (S1), (S2), and (D) can be designated at between -2147483648 and 2147483647 (BIN 32 bits).

(4) Judgment whether values for (S1), (S2), and (D) are positive or negative are made on the basis of the most significant bit (b31 for (S1) and (S2), and b63 for (D)).

- 0 . . . . Positive
- 1 . . . . Negative

D/

(1) Divides BIN 32-bit data designated by (S1) and BIN 32-bit data designated by (S2), and stores the result in the device designated by (D).



(2) If a word device has been used, the result of the division operation is stored as 64 bits, and both the quotient and remainder are stored; if a bit device has been used, 32 bits are used and only the quotient is stored.
Quotient. . . . . Stored at the lower 32 bits
Remainder. . . Stored at the upper 32 bits
(Can be stored only when a word device has been used)

(3) The values for (S1) and (S2) can be designated at between -2147483648 and 2147483647 (BIN 32 bits).

(4) Judgment whether values for (S1), (S2), (D), and (D)+2 are positive or negative is made on the basis of the most significant bit (b31) (A sign is used with both the quotient and the remainder)

- 0 . . . . Positive
- 1 . . . . Negative

Operation Errors

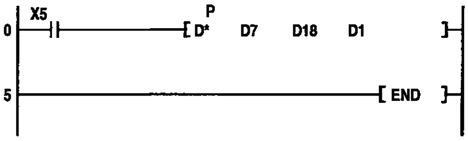
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Attempt to divide (S2) by 0. (Error code: 4100)

Program Example

(1) The following program divides the BIN data at D7 and D8 by the BIN data at D18 and D19 when X5 is ON, and stores the result at D1 through D4.

[Ladder Mode]

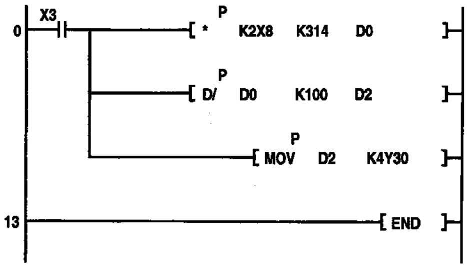


[List Mode]

Steps	Instruction	Device
0	LD	X5
1	D*P	D7 D18 D1
5	END	

(2) The following program outputs the value resulting when the data at X8 through XF is multiplied by 3.14 to Y30 through Y3F when X3 is ON.

[Ladder Mode]



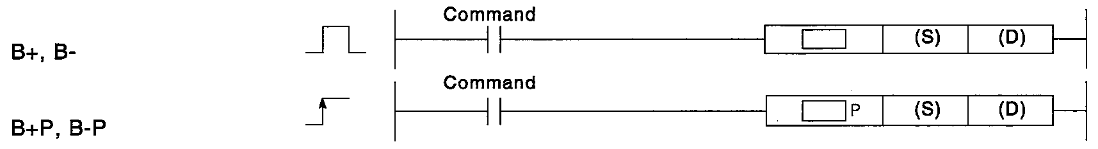
[List Mode]

Steps	Instruction	Device

6.2.5 BCD 4-digit addition and subtraction operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition] ▭ indicates B+/B-



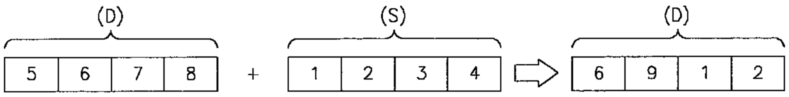
Set Data

Set Data	Meaning	Data Type
(S)	Addition or subtraction data, or first number of device storing addition or subtraction data	BCD 4-digit
(D)	Data to be added to or subtracted from, or first number of device storing such data	

Functions

B+

(1) Adds the BCD 4-digit data designated by (D) and the BCD 4-digit data designated by (S), and stores the result of the addition at the device designated by (D).

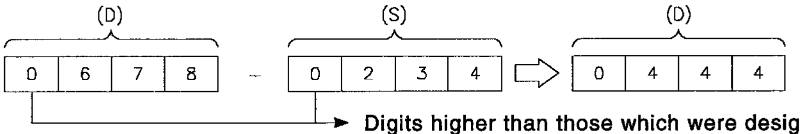


(2) The values for (S) and (D) can be between 0 and 9999 (BCD 4-digit).

(3) If the result of the addition operation exceeds 9999, the higher bits are ignored. The carry flag in this case does not go ON.

B-

(1) Subtracts the BCD 4-digit data designated by (D) and the BCD 4-digit data designated by (S), and stores the result of the subtraction at the device designated by (D).



Digits higher than those which were designated will be read as 0.

(2) The values for (S) and (D) can be between 0 and 9999 (BCD 4-digit).

(3) The following will result if an underflow is generated by the subtraction operation:
The carry flag in this case does not go ON.

0	0	0	1	–	0	0	0	3	⇨	9	9	9	8

Operation Errors

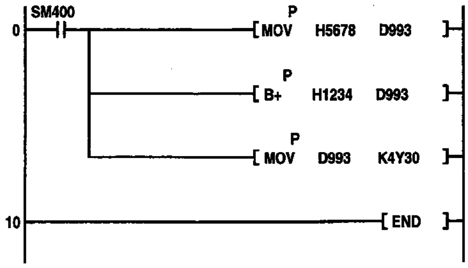
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The (S) or (D) BCD data is outside the 0 to 9999 range.
  (Error code: 4100)

Program Example

(1) The following program adds BCD data 5678 and 1234, stores it at D993, and at the same time outputs it to from Y30 to 3F.

[Ladder Mode]



Stores 5678 at D993

Adds BCD 1234 and D993, and stores the result in D993

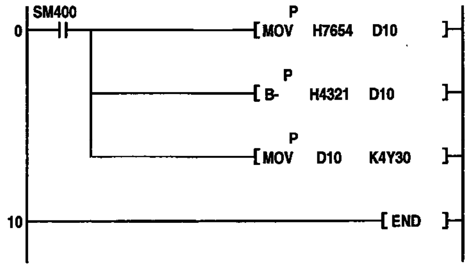
Outputs data at D993 to Y30 to 3F

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	MOVP	H5678
		D993
4	B+P	H1234
		D993
7	MOVP	D993
		K4Y30
10	END	

(2) The following program subtracts the BCD data 4321 from 7654, stores the result at D10, and at the same time outputs it to Y30 to 3F.

[Ladder Mode]



Stores 7654 at D10 as BCD

Subtracts data in D10 from BCD 4321 and stores result at D10

Outputs data at D10 to Y30 to 3F

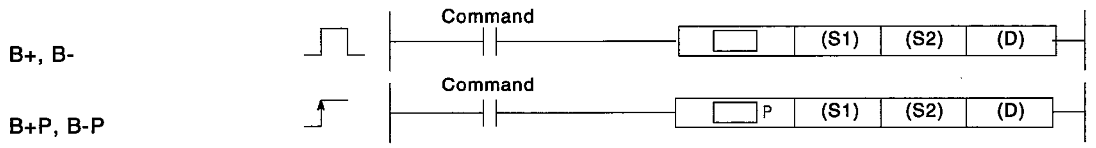
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	MOVP	H7654
		D10
4	B-P	H4321
		D10
7	MOVP	D10
		K4Y30
10	END	

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o							o	—
(S2)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]

□ indicates B+/B-



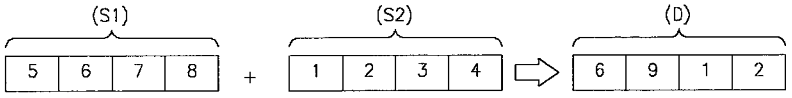
Set Data

Set Data	Meaning	Data Type
(S1)	Data to be added to or subtracted from, or the first number of the device storing such data	BCD 4-digit
(S2)	Addition or subtraction data, or first number of device storing addition or subtraction data	
(D)	First number of device storing addition or subtraction data	

Functions

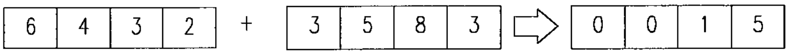
B+

(1) Adds the BCD 4-digit data designated by (S1) and the BCD 4-digit data designated by (S2), and stores the result of the addition at the device designated by (D).



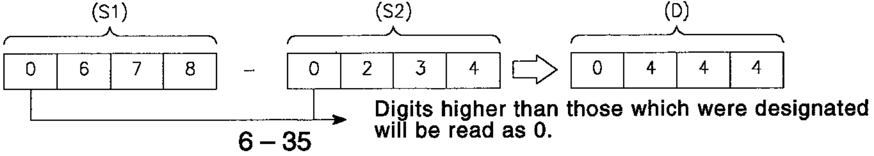
(2) The values for (S1), (S2), and (D) can be between 0 and 9999 (BCD 4-digit data).

(3) If the result of the addition operation exceeds 9999, the higher bits are ignored.
The carry flag in this case does not go ON.



B-

(1) Subtracts the BCD 4-digit data designated by (S1) and the BCD 4-digit data designated by (S2), and stores the result of the subtraction at the device designated by (D).



(2) The values for (S1), (S2), and (D) can be between 0 and 9999 (BCD 4-digit data).

(3) The following will result if an underflow is generated by the subtraction operation:
The carry flag in this case does not go ON.

0	0	0	1	−	0	0	0	3	⇨	9	9	9	8

Operation Errors

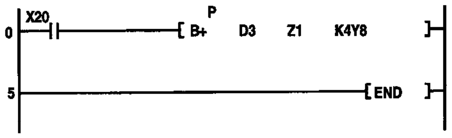
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The (S1), (S2), or (D) BCD data is outside the 0 to 9999 range. (Error code: 4100)

Program Example

(1) The following program adds the D3 BCD data and the Z1 BCD data when X20 goes ON, and outputs the result to Y8 through Y17.

[Ladder Mode]

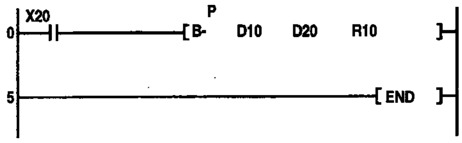


[List Mode]

Steps	Instruction	Device
0	LD	X20
1	B+P	D3 Z1 K4Y8
5	END	

(2) The following program subtracts the BCD data at D20 from the BCD data at D10 when X20 goes ON, and stores the result at R10.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	B-P	D10 D20 R10
5	END	

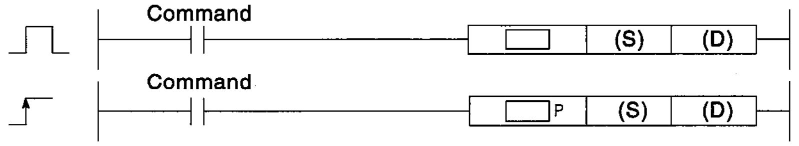
6.2.6 BCD 8-digit addition and subtraction operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]

☐ indicates DB+/DB-

DB+
DB-
DB+P
DB-P



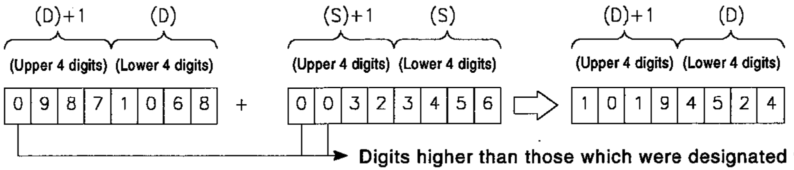
Set Data

Set Data	Meaning	Data Type
(S)	Addition or subtraction data, or first number of device storing addition or subtraction data	BCD 8-digit
(D)	Data to be added to or subtracted from, or first number of device storing such data	

Functions

DB+

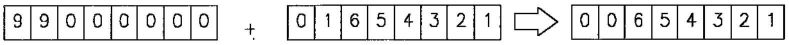
(1) Adds the BCD 8-digit data designated by (D) and the BCD 8-digit data designated by (S), and stores the result of the addition at the device designated by (D).



Digits higher than those which were designated will be read as 0.

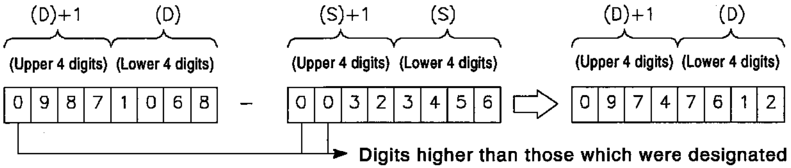
(2) The values for (S) and (D) can be between 0 and 99999999 (BCD 8-digit data).

(3) If the result of the addition operation exceeds 99999999, the upper bits will be ignored.
The carry flag in this case does not go ON.



DB-

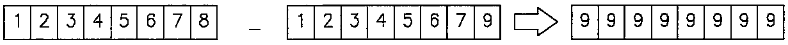
(1) Subtracts the BCD 8-digit data designated by (D) and the BCD 8-digit data designated by (S), and stores the result of the subtraction at the device designated by (D).



Digits higher than those which were designated will be read as 0.

(2) The values for (S) and (D) can be between 0 and 99999999 (BCD 8-digit).

(3) The following will result if an underflow is generated by the subtraction operation:
The carry flag in this case does not go ON.



Operation Errors

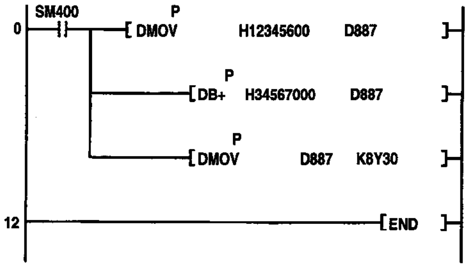
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The (S) or (D) BCD data is outside the 0 to 99999999 range. (Error code: 4100)

Program Example

(1) The following program adds the BCD data 12345600 and 34567000, stores the result at D887 and 888, and at the same time outputs them to from Y30 to 4F.

[Ladder Mode]



Stores 12345600 at D887,888 as BCD

Adds BCD 34567000 and D887,888 and stores the result in D887,888

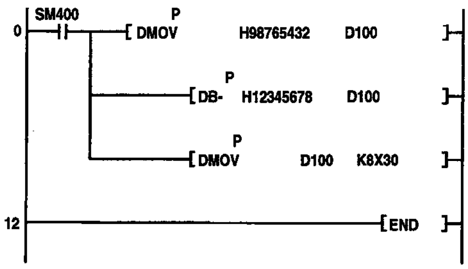
Outputs data at D887,888 to Y30 to 4F

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DMOVP	H12345600 D887
5	DB+P	H34567000 D887
9	DMOVP	D887 K8Y30
12	END	

(2) The following program subtracts the BCD data 98765432 from 12345678, stores the result at D100,101, and at the same time outputs it from Y30 to 4F.

[Ladder Mode]



Stores 98765432 at D100,101 as BCD data.

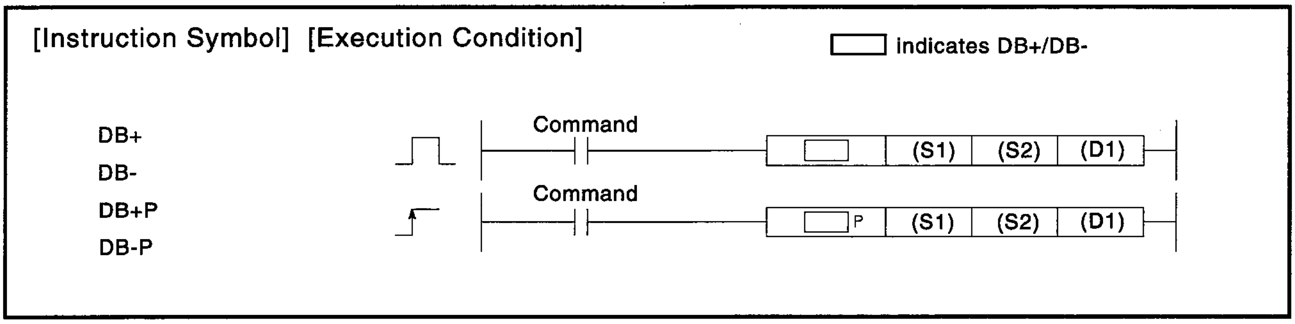
Subtracts data in D12345678 from BCD 100,101 and stores result at D100,101

Outputs data at D100,101 to Y30 to 4F

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DMOVP	H98765432
		D100
5	DB-P	H12345678
		D100
9	DMOVP	D100
		K8X30
12	END	

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o							o	—
(S2)	o							o	—
(D1)	o							—	—



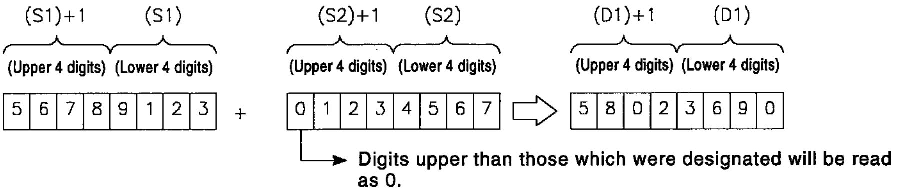
Set Data

Set Data	Meaning	Data Type
(S1)	Data to be added to or subtracted from, or the first number of the device storing such data	BCD 8-digit
(S2)	Addition or subtraction data, or first number of device storing addition or subtraction data	
(D1)	First number of device storing addition or subtraction data	

Functions

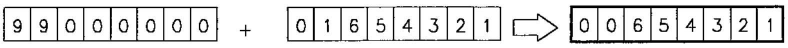
DB+

(1) Adds the BCD 8-digit data designated by (S1) and the BCD 8-digit data designated by (S2), and stores the result of the addition at the device designated by (D1).



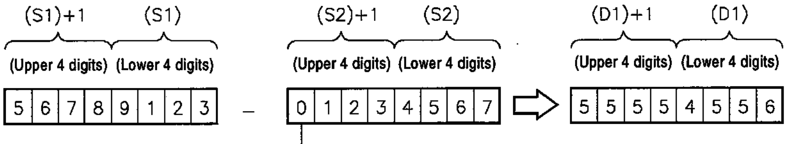
(2) The values for (S1), (S2), and (D) can be between 0 and 99999999 (BCD 8 digits).

(3) If the result of the addition operation exceeds 99999999, the upper bits will be ignored.
The carry flag in this case does not go ON.



DB-

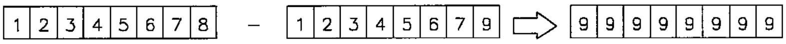
(1) Subtracts the BCD 8-digit data designated by (S1) and the BCD 8-digit data designated by (S2), and stores the result of the subtraction at the device designated by (D1).



Digits higher than those which were designated will be read as 0.

(2) The values for (S1), (S2), and (D) can be between 0 and 99999999 (BCD 8 digits).

(3) The following will result if an underflow is generated by the subtraction operation:
The carry flag in this case does not go ON.



Operation Errors

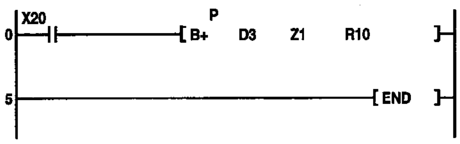
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The (S1), (S2), or (D) BCD data is outside the 0 to 99999999 range. (Error code: 4100)

Program Example

(1) The following program adds the BCD data at D3 and D4 to the BCD data at Z2 and Z2 when X20 goes ON, and stores the result at R10 and R11.

[Ladder Mode]



[List Mode]

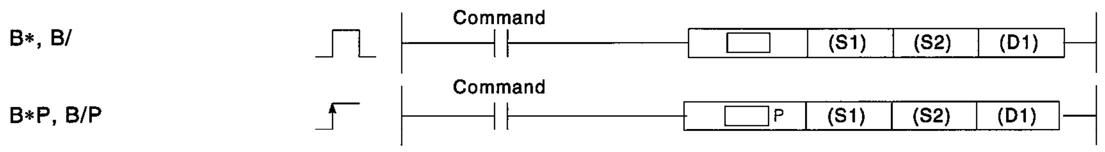
Steps	Instruction	Device
0	LD	X20
1	B+P	D3 Z1 R10
5	END	

6.2.7 BCD 4-digit multiplication and division operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	○							○	—
(S2)	○							○	—
(D)	○							—	—

[Instruction Symbol] [Execution Condition]

▭ indicates B∗/B/



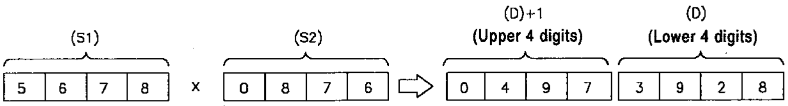
Set Data

Set Data	Meaning	Data Type
(S1)	Data that will be multiplied or divided, or the first number of the device storing data that will be multiplied or divided	BCD 4-digit
(S2)	Data to multiply or divide by, or the first number of device storing such data	
(D)	First number of the device storing the operation results of multiplication or division operation	BCD 8-digit

Functions

B∗

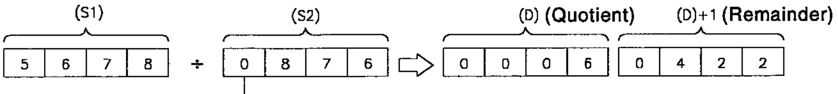
(1) Multiplies BCD data designated by (S1) and BCD data designated by (S2), and stores the result in the device designated by (D).



(2) Values for (S1) and (S2) can be set from 0 to 9999 (BCD 4 digits).

B/

(1) Divides BCD data designated by (S1) and BCD data designated by (S2), and stores the result in the device designated by (D).



Digits higher than those which were designated will be read as 0.

(2) Uses 32 bits to store the result of the division as quotient and remainder

Quotient (BCD 4 digits) . . Stored at the lower 16 bits

Remainder (BCD 4 digits) . . Stored at the upper 16 bits

(3) If (D) has been designated as a bit device, the remainder of the operation will not be stored.

Operation Errors

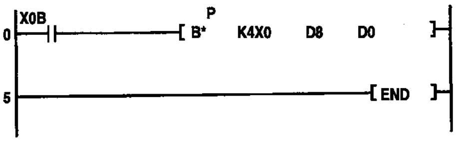
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The (S1) or (S2) data is outside the 0 to 9999 range. (Error code: 4100)
- Attempt to divide (S2) by 0. (Error code: 4100)

Program Example

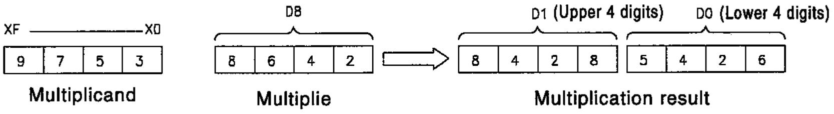
(1) The following program multiplies the BCD data at X0 to F and the BCD data at D8 when X1B goes ON, and stores the result at D0 and D1.

[Ladder Mode]



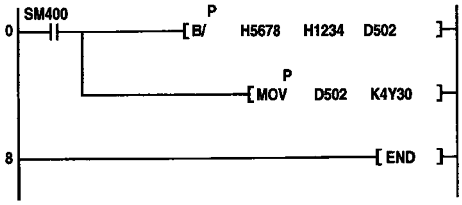
[List Mode]

Steps	Instruction	Device
0	LD	X0B
1	B∗P	K4X0 D8 D0
5	END	



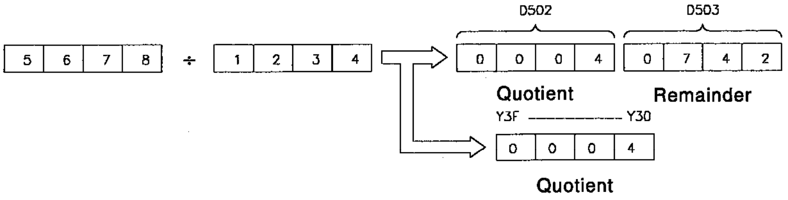
(2) The following program divides 5678 by the BCD data 1234, stores the result at D502 and 503, and at the same time outputs the quotient to Y30 through 3F.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	B/P	H5678 H1234 D502
5	MOVP	D502 K4Y30
8	END	

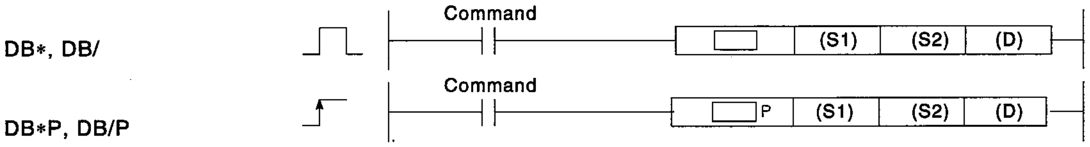


6.2.8 BCD 8-digit multiplication and division operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o			o					—
(S2)	o			o					—
(D)	o			—					—

[Instruction Symbol] [Execution Condition]

□ indicates the signs DB* or DB/



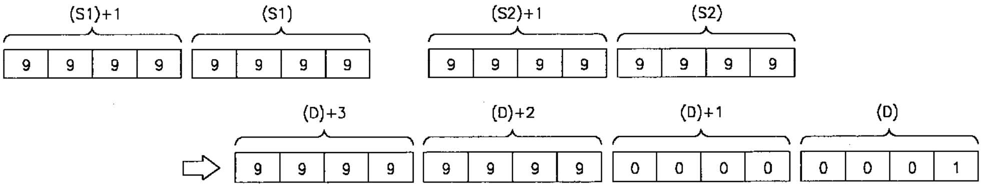
Set Data

Set Data	Meaning	Data Type
(S1)	Data that will be multiplied or divided, or the first number of the device storing data that will be multiplied or divided	BCD 8 digits
(S2)	Data to multiply or divide by, or the first number of device storing such data	
(D)	First number of the device storing the operation results of multiplication or division operation	BCD 16 digits

Functions

DB*

(1) Multiplies the BCD 8-digit data designated by (S1) and the BCD 8-digit data designated by (S2), and stores the product at the device designated by (D).



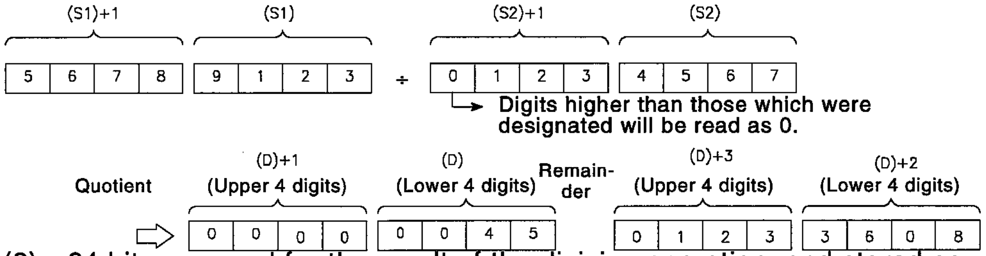
(2) If (D) has designated a bit device, the lower 8 digits (lower 32 bits) will be used for the product, and the higher 8 digits (higher 32 bits) cannot be designated.

K1:Lower 1 digit (B0 through 3)
K4:Lower 4 digits (B0 through 15)
K8:Lower 8 digits (B0 through 31)

(3) The values for (S1) and (S2) can be designated from 0 to 99999999 (8-digit BCD).

DB/

(1) Divides 8-digit BCD data designated by (S1) and 8-digit BCD data designated by (S2), and stores the result in the device designated by (D).



(2) 64 bits are used for the result of the division operation, and stored as quotient and remainder.

Quotient (BCD 8 digits) : Stored at the lower 32 bits
Remainder (BCD 8 digits): Stored at the upper 32 bits

(3) If (D) has been designated as a bit device, the remainder of the operation will not be stored.

Operation Errors

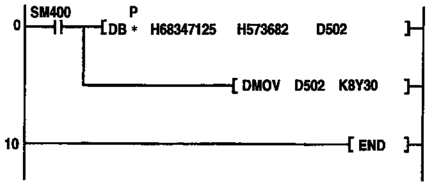
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The (S1) or (S2) data is outside the 0 to 99999999 range. (Error code: 4100)
- Attempt to divide (S2) by 0. (Error code: 4100)

Program Example

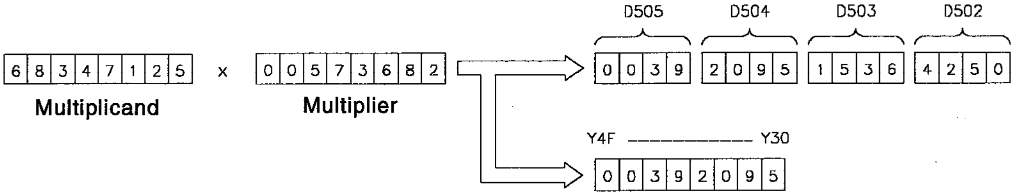
(1) The following program multiplies the BCD data 67347125 and 573682, stores the result from D502 to 505, and at the same time outputs the upper 8 digits to Y30 through 4F.

[Ladder Mode]



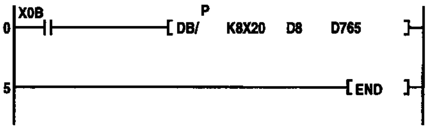
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DB*P	B68347125 B573682 D502
7	DMOV	D504 K8Y30
10	END	



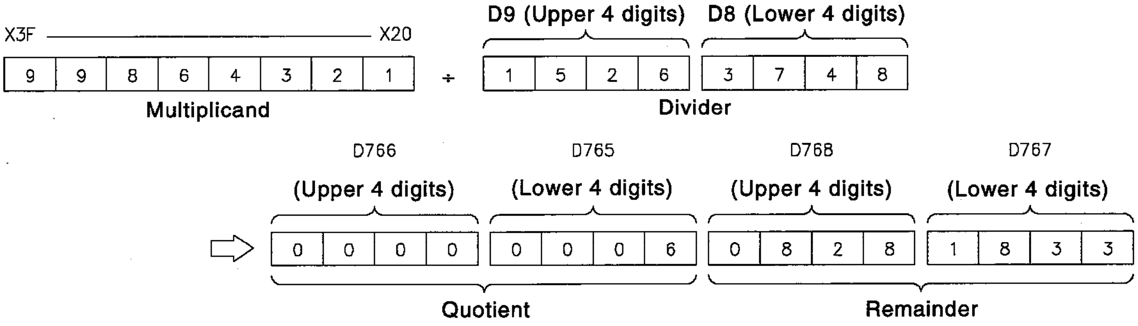
(2) The following program divides the BCD data from X20 to 3F by the BCD data at D8 and D9 when X1B goes ON, and stores the result from D765 to 768.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0B
1	DB/P	K8X20 D8 D765
5	END	

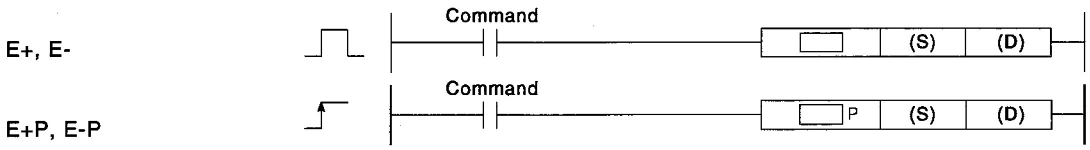


6.2.9 Addition and subtraction operations with floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

□ indicates the signs E+ or E-



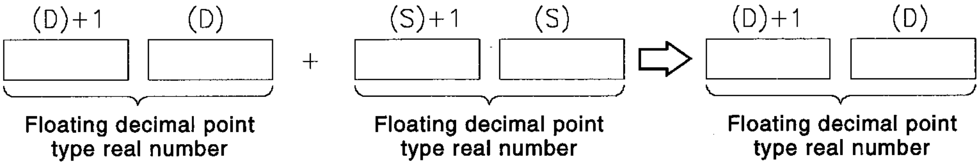
Set Data

Set Data	Meaning	Data Type
(S)	Addition or subtraction data, or first number of device storing addition or subtraction data	Real number
(D)	Data to be added to or subtracted from, or first number of device storing such data	

Functions

E+

(1) Adds the floating decimal point type real number designated at (D) and the floating decimal point type real number designated at (S), and stores the sum in the device designated at (D).

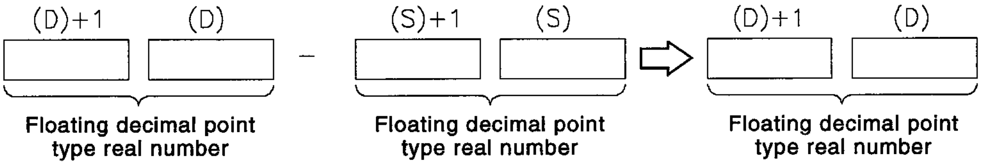


(2) Values which can be designated at (S), (D), and (D1), and which can be stored, are as follows:

$$0, \pm 2^{-127} \leq | \text{Designated value (stored value)} | < \pm 2^{128}$$

E-

(1) Subtracts a floating decimal point real number designated by (D) and a floating decimal point real number designated by (S), and stores the result at a device designated by (D).



(2) Values which can be designated at (S), (D), and (D1), and which can be stored, are as follows:

$0, \pm2^{-127} \leq |$ Designated value (stored value) $| < \pm2^{128}$

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The contents of the designated device or the result of the addition or subtraction operation are not "0", or not within the following range: (Error code: 4100)

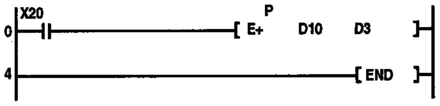
  $\pm2^{-127} \leq |$ Contents of designated device/operation result $| < \pm2^{128}$

- When the specified device contains -0 (Q4ARCPU) (Operation error does not occur even if -0 is stored if SM707 is turned on.) (Error code: 4100)

Program Example

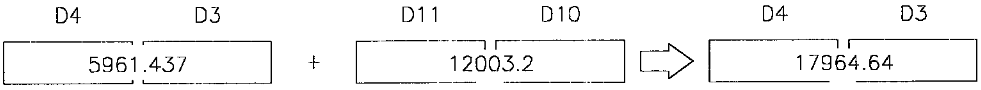
(1) The following program adds the floating decimal point real numbers at D3 and D4 and the floating decimal point real numbers at D10 and D11 when X20 goes ON, and stores the result at D3 and D4.

[Ladder Mode]



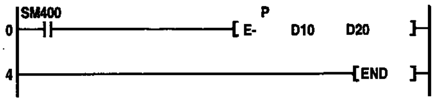
[List Mode]

Steps	Instruction	Device
0	LD	X20
1	E+P	D10 D3
4	END	



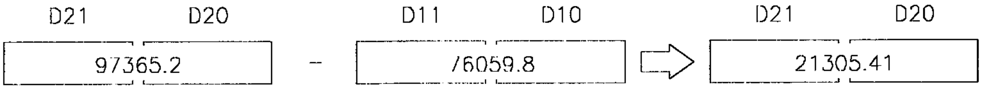
(2) The following program subtracts the floating decimal point real number at D10 and D11 from the floating decimal point real numbers at D20 and D21, and stores the result of the subtraction at D20 and D21.

[Ladder Mode]



[List Mode]

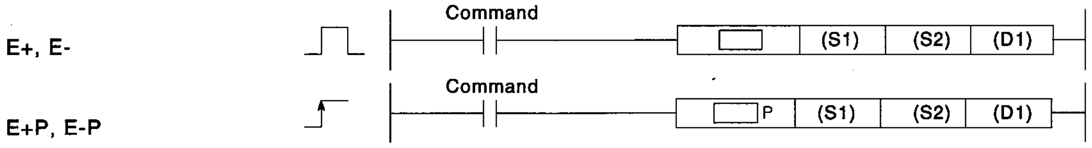
Steps	Instruction	Device
0	LD	SM400
1	E-P	D10 D20
4	END	



Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—	o		—	o	—
(S2)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

□ indicates the signs E+ or E-



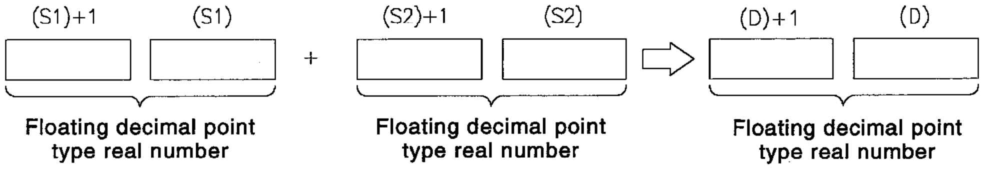
Set Data

Set Data	Meaning	Data Type
(S1)	Data to be added to or subtracted from, or the first number of the device storing such data	Real number
(S2)	Addition or subtraction data, or first number of device storing addition or subtraction data	
(D)	First number of device storing addition or subtraction data	

Functions

E+

(1) Adds the floating decimal point real number designated by (S1) and the floating decimal point real number designated by (S2), and stores the result at the device designated by (D).

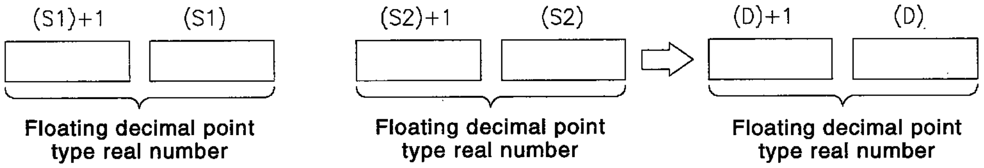


(2) Values that can be designated by (S1), (S2), (D) or (D1), and values that can be stored, are as follows:

$$0, \pm 2^{-127} \leq | \text{Designated value (stored value)} | < \pm 2^{128}$$

E-

(1) Subtracts the floating decimal point real number designated by (S1) from the floating decimal point real number designated by (S2), and stores the result at the device designated by (D1).



(2) Values that can be designated by (S1), (S2), (D) or (D1), and values that can be stored, are as follows:

$$0, \pm 2^{-127} \leq |\text{ Designated value (stored value) }| < \pm 2^{128}$$

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The contents of the designated device or the result of the addition or subtraction operation are not "0", or not within the following range: (Error code: 4100)

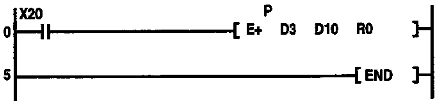
  $$\pm 2^{-127} \leq |\text{ Contents of designated device/results of operation }| < \pm 2^{129}$$

- When the specified device contains -0. (Q4ARCPU) (Operation error does not occur even if -0 is stored if SM707 is turned on.) (Error code: 4100)

Program Example

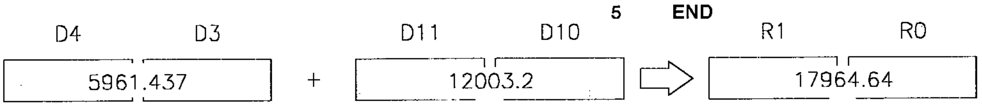
(1) The following program adds the floating decimal point real numbers at D3 and D4 and the floating decimal point real numbers at D10 and D11 when X20 goes ON, and outputs the result to R0 and R1.

[Ladder Mode]



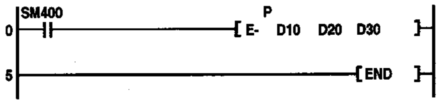
[List Mode]

Steps	Instruction	Device
0	LD	X20
1	E+P	D3 D10 R0
5	END	



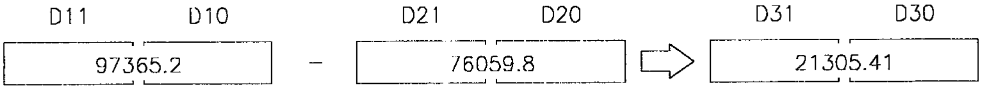
(2) The following programs subtracts the floating decimal point real numbers at D20 and D21 from the floating decimal point numbers at D11 and D10, and stores the result at D30 and D31.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	E-P	D10 D20 D30
5	END	



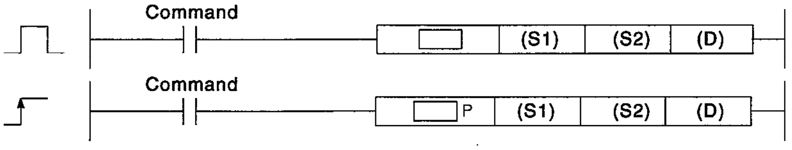
6.2.10 Floating decimal point data multiplication and division

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				U
(S1)	—	o		—	o		—	o	—
(S2)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition] ▭ indicates the signs E* or E/

E*, E/

E*P, E/P



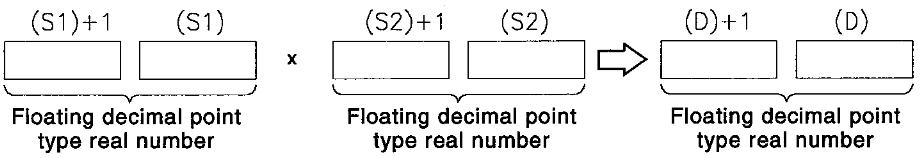
Set Data

Set Data	Meaning	Data Type
(S1)	Data that will be multiplied or divided, or the first number of the device storing data that will be multiplied or divided	Real number
(S2)	Data to multiply or divide by, or the first number of device storing such data	
(D)	First number of the device storing the operation results of multiplication or division operation	

Functions

E*

(1) Multiplies the floating decimal point real numbers designated by (S1) and the floating decimal point real numbers designated by (S2), and stores the product at the device designated by (D).

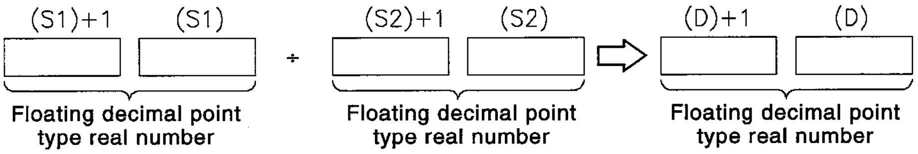


(2) Values that can be designated by (S1), (S2) or (D), and values that can be stored, are as follows:

$$0, \pm 2^{-127} \leq \mid \text{Designated value (stored value)} \mid < \pm 2^{128}$$

E/

(1) Divides floating decimal point real numbers designated by (S1) by floating decimal point real numbers designated by (S2), and stores the result in the device designated by (D).



(2) Values that can be designated by (S1), (S2) or (D), and values that can be stored, are as follows:

0, $\pm 2^{-127} \leq |$ Designated value (stored value) $| < \pm 2^{128}$

Operation Errors

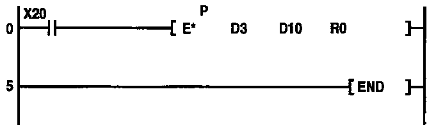
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The contents of the designated device or the result of the multiplication or division operation are not "0", or not within the following range: (Error code: 4100)
  $\pm 2^{-127} \leq |$ Contents of designated device/results of operation $| < \pm 2^{128}$
- When the specified device contains -0 (Q4ARCPU)
  (Operation error does not occur even if -0 is stored if SM707 is turned on.) (Error code: 4100)

Program Example

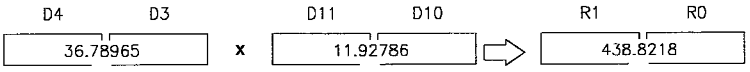
(1) The following program multiplies the floating decimal point real numbers at D3 and D4 and the floating decimal point real numbers at D10 and D11, and stores the result at R0 and R1.

[Ladder Mode]



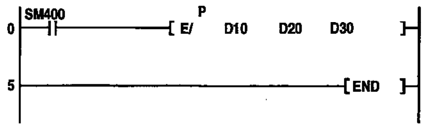
[List Mode]

Steps	Instruction	Device
0	LD	X20
1	BK+P	D100 R0 D200 D0
6	END	



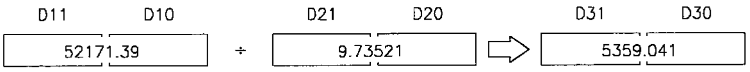
(2) The following program divides the floating decimal point real numbers at D10 and D11 by the floating decimal point real numbers at D20 and D21, and stores the result at D30 and D31.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	E/P	D10 D20 D30
5	END	

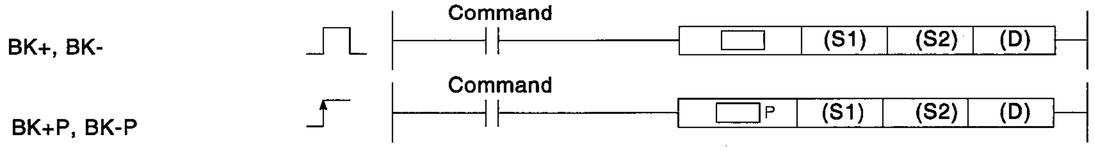


6.2.11 Block addition and subtraction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—				—	—
(S2)	—	o		—				o	—
(D)	—	o		—				—	—
n	o	o		o				o	—

[Instruction Symbol] [Execution Condition]

☐ indicates the signs BK+/BK-



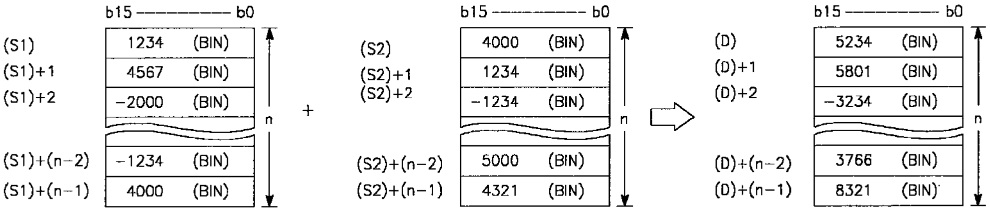
Set Data

Set Data	Meaning	Data Type
(S1)	Data to be added to or subtracted from, or first number of device storing such data	BIN 16 bits
(S2)	Addition or subtraction data, or first number of device storing addition or subtraction data	
(D)	First number of the devices where the operation results are stored	
n	Number of addition/subtraction data blocks	BIN 16 bits

Functions

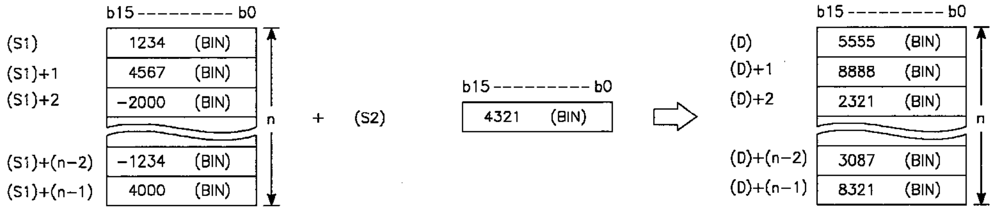
BK+

(1) Adds n-points of BIN data from the device designated by (S1) and n-points of BIN data from the device designated by (S2) and stores the result from the device designated by (D) onward.



(2) Block addition is performed in 16-bit units.

(3) The constant designated by (S2) can be between -32768 and 32767 (BIN 16 bits).

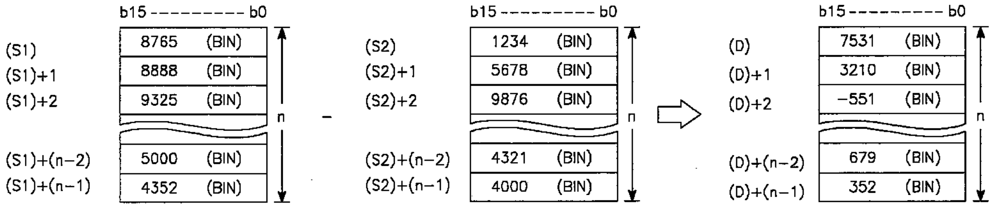


(4) The following happens if an underflow or overflow is generated in the operation results:
The carry flag in this case does not go ON.

- K32767 + K2 ——— K-32767
  (H7FFF) (H0002) (H8001)
- K-32767 + K-2 ——— K32766
  (H8000) (HFFFE) (H7FFE)

BK-

(1) Subtracts n-points of BIN data from the device designated by (S1) and n-points of BIN data from the device designated by (S2) and stores the result from the device designated by (D) onward.



(2) Block subtraction is performed in 16-bit units.

(3) The constant designated by (S2) can be between -32768 and 32767 (BIN 16 bits).

(4) The following happens if an underflow or overflow is generated in the operation results:
The carry flag in this case does not go ON.

- K-32768 - K2 ——— K32766
  (H8000) (H0002) (H7FFF)
- K32767 - K-2 ——— -32766
  (H8000) (HFFFE) (H8001)

Operation Errors

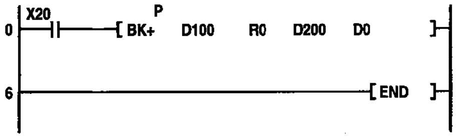
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The n-bit range from the (S1), (S2), or (D) device exceeds the range of that device.

Program Example

(1) The following program adds the data the number of points corresponding to the value stored from D100 to D0 to the data the number of points corresponding to the value stored from R0 to D0 when X20 is ON, and stores the result from D200 onward.

[Ladder Mode]



[List Mode]

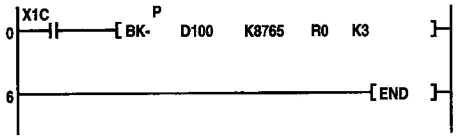
Steps	Instruction	Device
0	LD	X20
1	BK+P	D100 R0 D200 D0
6	END	

	b15 --------- b0			b15 --------- b0			b15 --------- b0
D100	6789 (BIN)		R0	1234 (BIN)		D200	8023 (BIN)
D101	7821 (BIN)	+	R1	2032 (BIN)	⇨	D201	9853 (BIN)
D102	5432 (BIN)		R2	−3252 (BIN)		D202	2180 (BIN)
D103	3520 (BIN)		R3	−1000 (BIN)		D203	2520 (BIN)

D0 4

(2) The following program subtracts the data 3 points from D100 and the constant 8765 when X1C is ON, and stores the result from R0 onward.

[Ladder Mode]



[List Mode]

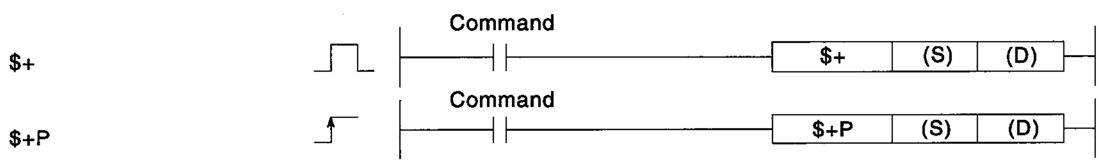
Steps	Instruction	Device
0	LD	X1C
1	BK-P	D100 K8765 R0 K3
6	END	

	b15 --------- b0		b15 --------- b0			b15 --------- b0
D100	12345 (BIN)				R0	3580 (BIN)
D101	8701 (BIN)	−	8765 (BIN)	⇨	R1	−64 (BIN)
D102	3502 (BIN)				R2	−5263 (BIN)

6.2.12 Linking character strings

<table>
<tr><th rowspan="3">Set Data</th><th colspan="9">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System,User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th rowspan="2">Constant $</th><th rowspan="2">Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th></tr>
<tr><td>(S)</td><td>—</td><td colspan="2">o</td><td colspan="4">—</td><td>o</td><td>—</td></tr>
<tr><td>(D)</td><td>—</td><td colspan="2">o</td><td colspan="4">—</td><td>—</td><td>—</td></tr>
</table>

[Instruction Symbol] [Execution Condition]

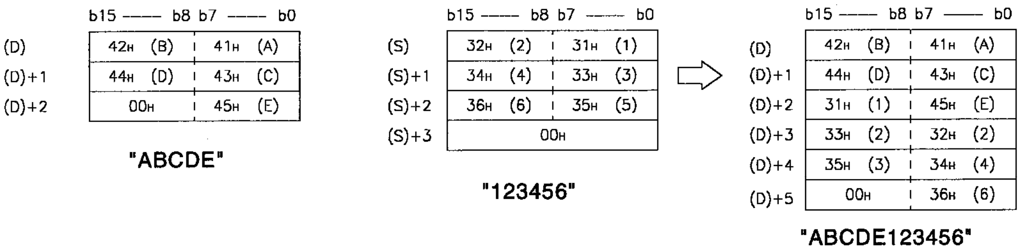


Set Data

Set Data	Meaning	Data Type
(S)	First number of device holding linked data or data	Character string
(D)	First number of device holding data which has been linked	

Functions

(1) Character string data stored in device numbers starting with that designated at (S) will be appended after character string data stored in device numbers starting with that designated at (D), and will be stored in device numbers starting with that designated at (D).
The object of character string data is that character string data stored from device numbers designated at (D) and (S) to that stored at "00H".



	b15 —— b8	b7 —— b0
(D)	42H (B)	41H (A)
(D)+1	44H (D)	43H (C)
(D)+2	00H	45H (E)

"ABCDE"

	b15 —— b8	b7 —— b0
(S)	32H (2)	31H (1)
(S)+1	34H (4)	33H (3)
(S)+2	36H (6)	35H (5)
(S)+3	00H	

"123456"

	b15 —— b8	b7 —— b0
(D)	42H (B)	41H (A)
(D)+1	44H (D)	43H (C)
(D)+2	31H (1)	45H (E)
(D)+3	33H (2)	32H (2)
(D)+4	35H (3)	34H (4)
(D)+5	00H	36H (6)

"ABCDE123456"

(2) When character strings are linked, the "00H", which indicates the end of character string data designated at (D), is ignored, and the character string designated at (S) is appended to the last character of the (D) string.

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD.

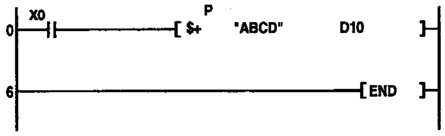
- The entire character string linked from the device number designated by (D) to the final device number of the relevant device cannot be stored.(Error code: 4100)
- The storage device numbers for the character strings designated by (S) and (D) overlap.(Error code: 4101)

(2) See Section 3.6 for information regarding errors during index qualification.

Program Example

(1) The following program links the character string stored from D10 to 12 to the character string "ABCD" when X0 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	$+P	"ABCD" D10
6	END	

	b15 —— b8	b7 —— b0
D10	62H (b)	61H (a)
D11	64H (d)	63H (c)
D12	00H	65H (e)

	b15 —— b8	b7 —— b0
D10	62H (b)	61H (a)
D11	64H (d)	63H (c)
D12	41H (A)	65H (e)
D13	43H (C)	42H (B)
D14	00H	44H (D)

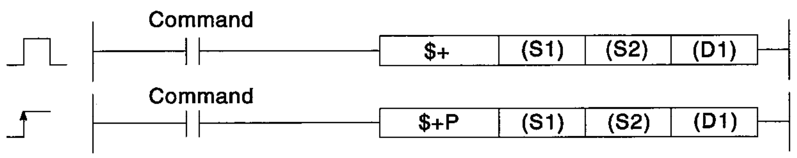
Automatically stores "00H"

<table>
<tr><th rowspan="3">Set Data</th><th colspan="9">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System, User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th rowspan="2">Constant $</th><th rowspan="2">Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th></tr>
<tr><td>(S1)</td><td>—</td><td colspan="2">o</td><td colspan="4">—</td><td>o</td><td>—</td></tr>
<tr><td>(S2)</td><td>—</td><td colspan="2">o</td><td colspan="4">—</td><td>o</td><td>—</td></tr>
<tr><td>(D1)</td><td>—</td><td colspan="2">o</td><td colspan="4">—</td><td>—</td><td>—</td></tr>
</table>

[Instruction Symbol] [Execution Condition]

$+

$+P



Set Data

Set Data	Meaning	Data Type
(S1)	First number of device holding linked data	Character string
(S2)	First number of device holding data which has been linked	
(D1)	First number of device holding results of linking	

Functions

(1) Appends character string data stored from the device number designated by (S2) to the character string data stored from the device number designated by (S1), and stores it from the device number designated by (D1).

	b15 ---- b8	b7 ---- b0
(S1)	46H (F)	48H (H)
(S1)+1	2DH (–)	41H (A)
(S1)+2	00H	

+

	b15 ---- b8	b7 ---- b0
(S2)	35H (5)	31H (1)
(S2)+1	39H (9)	33H (3)
(S2)+2	00H	41H (A)

⇨

	b15 ---- b8	b7 ---- b0
(D)	46H (F)	48H (H)
(D)+1	2DH (–)	41H (A)
(D)+2	35H (5)	31H (1)
(D)+3	39H (9)	33H (3)
(D)+4	00H	41H (A)

(2) When character strings are linked, the "00H" which indicates the end of character string data indicated by (S1), is ignored, and the character string indicated by (S2) is appended to the last character of the (S1) string.

Operation Errors

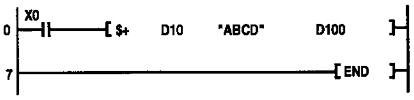
(1) In the following cases an operation error occurs and the error flag goes ON.

- The entire character string linked from the device number designated by (D1) to the final device number of the relevant device cannot be stored.
- The storage device numbers for the character strings designated by (S1) or (S2) overlap with those for (D1).

Program Example

(1) The following program links the character string stored from D10 to 12 with the character string "ABCD" when X0 is ON, and stores them in D100 onwards.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	$+	D10 "ABCD" D100
7	END	

	b15 ---- b8	b7 ---- b0
D10	62_H (b)	61_H (a)
D11	64_H (d)	63_H (c)
D12	00_H	65_H (e)

+ "ABCD"

⇨

	b15 ---- b8	b7 ---- b0
D100	62_H (b)	61_H (a)
D101	64_H (d)	63_H (c)
D102	41_H (A)	65_H (e)
D103	43_H (C)	42_H (B)
D104	00_H	44_H (D)

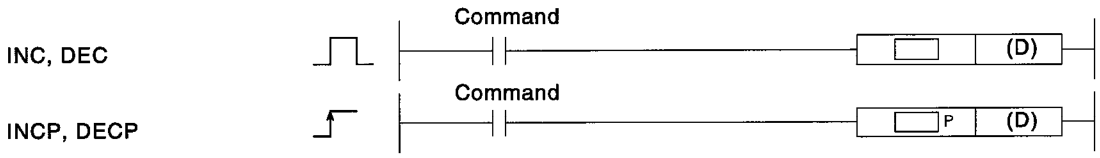
Automatically stores "00_H"

6.2.13 Incrementing and decrementing 16-bit BIN data

Set Data	Usable Devices								
	Internal Devices (System,User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	o							—	

[Instruction Symbol] [Execution Condition]

▭ indicates INC/DEC



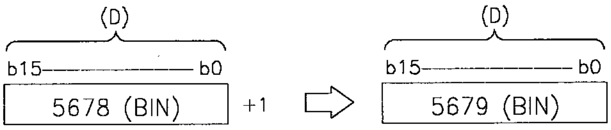
Set Data

Set Data	Meaning	Data Type
(D)	First number of device conducting INC (add 1) or DEC (subtract 1) operation	BIN 16 bits

Functions

INC

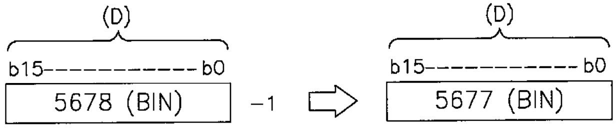
(1) Adds 1 to device designated by (D) (16-bit data).



(2) If the contents of the device designated by (D) were 32767, and the INC or INCP instruction were executed on that device, the value -32768 would be stored in the device designated by (D).

DEC

(1) Subtracts 1 from device designated by (D) (16-bit data).



(2) If the contents of the device designated by (D) were 0, and the DEC or DECP instruction were executed on that device, the value -1 would be stored in the device designated by (D).

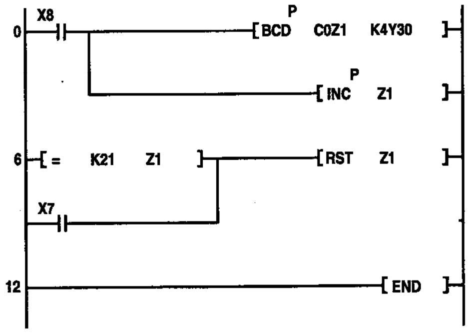
Operation Errors

(1) There are no operation errors associated with the INC, INCP, DEC or DECP instructions.

Program Example

(1) The following program outputs the present value of counter C0 through C20 to Y30 through 3F as BCD data when X8 is ON. (When present value is less than 9999)

[Ladder Mode]



Outputs the present value of C (D+Z1) to Y30 through 3F as BCD.

Executes Z1+1

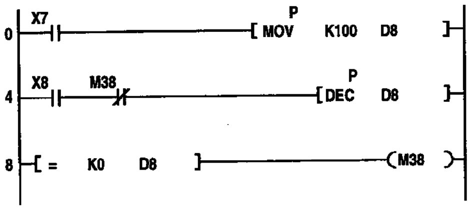
With Z1=1 or X7 (reset input), sets Z to 0

[List Mode]

Steps	Instruction	Device
0	LD	X8
1	BCDP	C0Z1
		K4Y30
4	INCP	Z1
6	LD=	K21
		Z1
9	OR	X7
10	RST	Z1
12	END	

(2) The following is a down counter program.

[Ladder Mode]



Transfers the value 100 to D8 when X7 is ON

When M38 is OFF, X8 goes from OFF to ON, and 1 is decremented from D8.

At D8=0, M30 goes ON

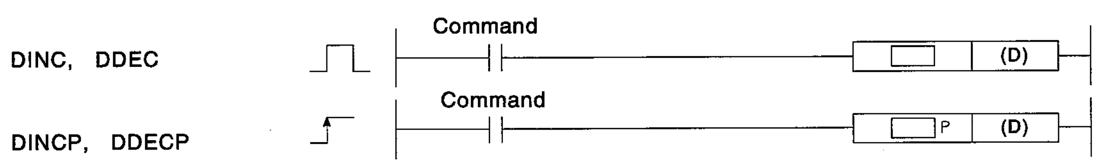
[List Mode]

Steps	Instruction	Device
0	LD	X7
1	MOVP	K100
		D8
4	LD	X8
5	ANI	M38
6	DECP	D8
8	LD=	K0
		D8
11	OUT	M38
12	END	

6.2.14 Incrementing and decrementing 32-bit BIN data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
(D)	o							—	

[Instruction Symbol] [Execution Condition] □ indicates DINC/DDEC



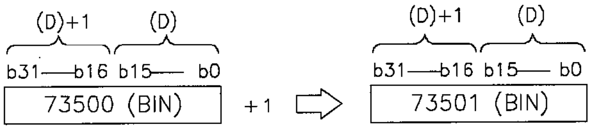
Set Data

Set Data	Meaning	Data Type
(D)	First number of device what will execute the DINC (+1) or DDEC (-1) operation	BIN 32 bits

Functions

DINC

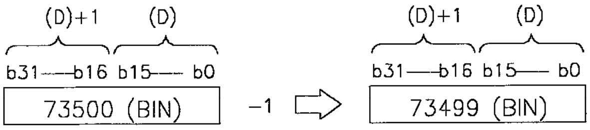
(1) Adds 1 to the device designated by (D) (32-bit data).



(2) If the contents of the device designated by (D) are 2147483647, and the DINC or DINCP instruction is executed, the value -2147483648 will be stored at the device designated by (D).

DDEC

(1) Subtracts 1 from the device designated by (D) (32-bit data).



(2) If the contents of the device designated by (D) are 0, and the DINC or DINCP instruction is executed, the value -1 will be stored at the device designated by (D).

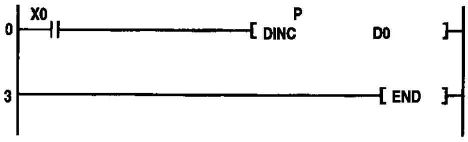
Operation Errors

(1) There are no operation errors associated with DINC(P) or DDEC(P).

Program Example

(1) The following program adds 1 to the data at D0 and D1 when X0 is ON.

[Ladder Mode]

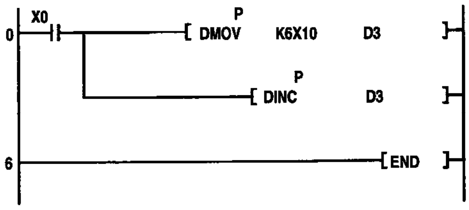


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	DINCP	D0
3	END	

(2) The following program adds 1 to the data set at X10 through X27 when X0 goes ON, and stores the result at D3 and D4.

[Ladder Mode]

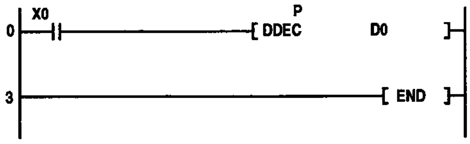


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	DMOVP	K6X10
		D3
4	DINCP	D3
6	END	

(3) The following program subtracts 1 from the data at D0 and D1 when X0 goes ON.

[Ladder Mode]

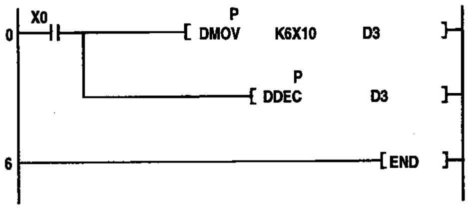


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	DDECP	D0
3	END	

(4) The following program subtracts 1 from the data set at X10 to X27 when X0 goes ON, and stores the result at D3 and D4.

[Ladder Mode]



[List Mode]

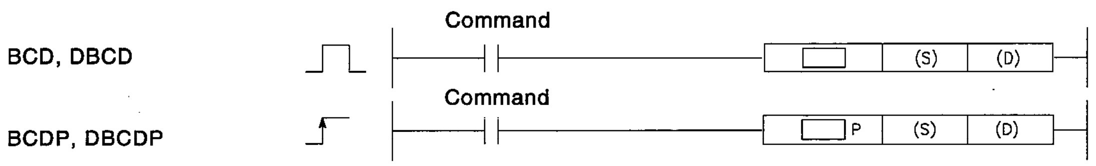
Steps	Instruction	Device
0	LD	X0
1	DMOVP	K6X10
		D3
4	DDECP	D3
6	END	

6.3 Data Conversion Instructions

6.3.1 Conversion from BIN data to 4-digit and 8-digit BCD

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition] ▭ indicates BCD(P) or DBCD(P)



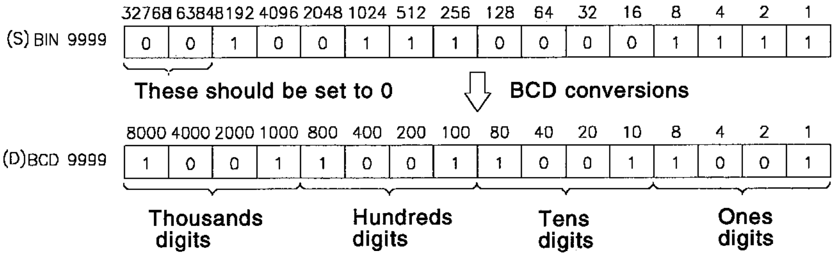
Set Data

Set Data	Meaning	Data Type
(S)	BIN data, or first number of the device where BIN data is stored	BIN 16/ 32 bits
(D)	First number of the device that will store BCD data	BCD 4/8 digits

Functions

BCD

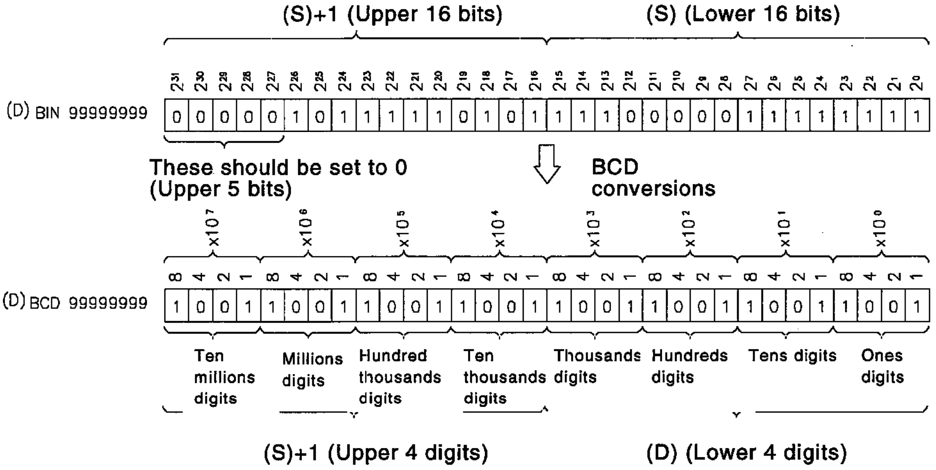
Converts BIN data (0 to 9999) at the device designated by (S) to BCD data, and stores it at the device designated by (D)



	32768	6384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
(S) BIN 9999	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1
	8000	4000	2000	1000	800	400	200	100	80	40	20	10	8	4	2	1
(D)BCD 9999	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1

DBCD

(1) Converts BIN data (0 to 99999999) at the device designated by (S) to BCD data, and stores it at the device designated by (D).



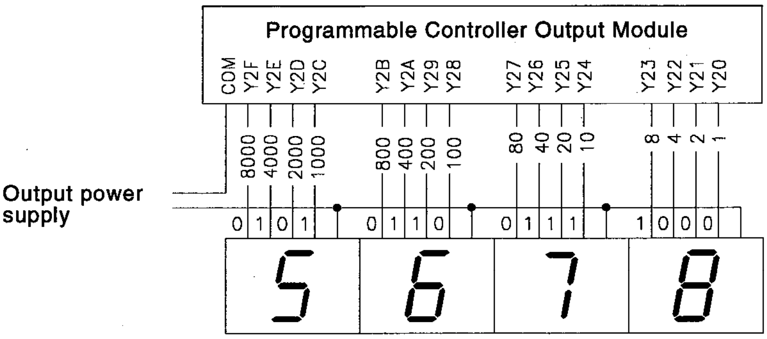
Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data at (S) was not in the 0 to 9999 range when the BCD instruction was issued.(Error code: 4100)
- The data at (S)+1 and (S) was not in the 0 to 99999999 range when the DBCD instruction was issued.(Error code: 4100)

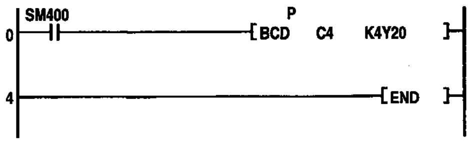
Program Example

(1) The following program outputs the present value of C4 from Y20 to Y2F to the BCD display device.



7-element display unit

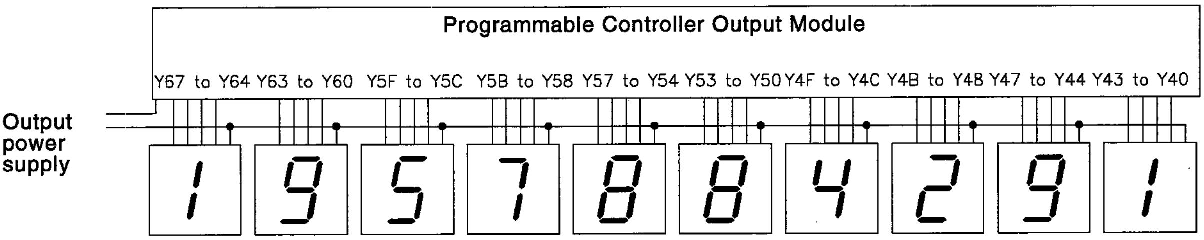
[Ladder Mode]



[List Mode]

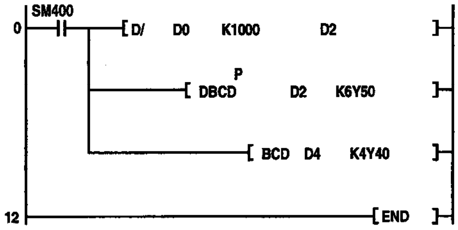
Steps	Instruction	Device
0	LD	SM400
1	BCDP	C4 K4Y20
4	END	

(2) The following program outputs 32-bit data from D0 to D1 to Y40 through Y67.



7-element display unit

[Ladder Mode]



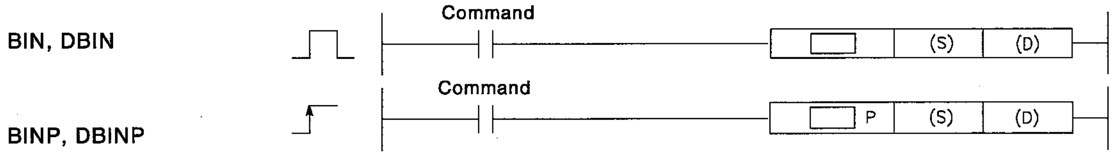
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	D/	D0 K10000 D2
6	DBCDP	D2 K6Y50
9	BCD	D4 K4Y40
12	END	

6.3.2 Conversion from BCD 4-digit and 8-digit data to BIN data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
(S)	o							o	—
(D)	—							—	—

[Instruction Symbol] [Execution Condition] □ indicates BIN(P)/DBIN(P)



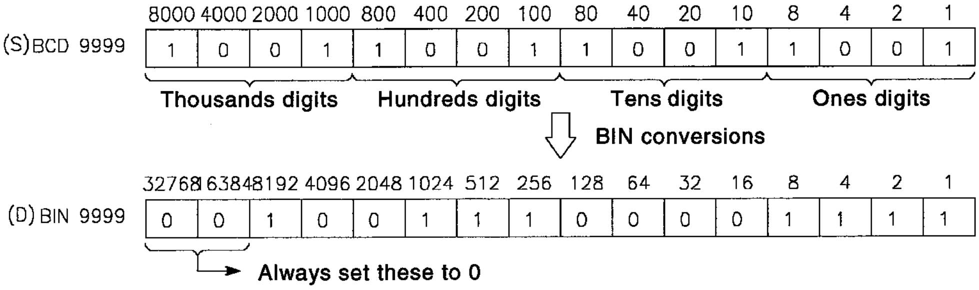
Set Data

Set Data	Meaning	Data Type
(S)	BCD data or first number of device storing BCD data	BCD 4/8 digits
(D)	First number of device that will store BIN data	BIN 16/32 bits

Functions

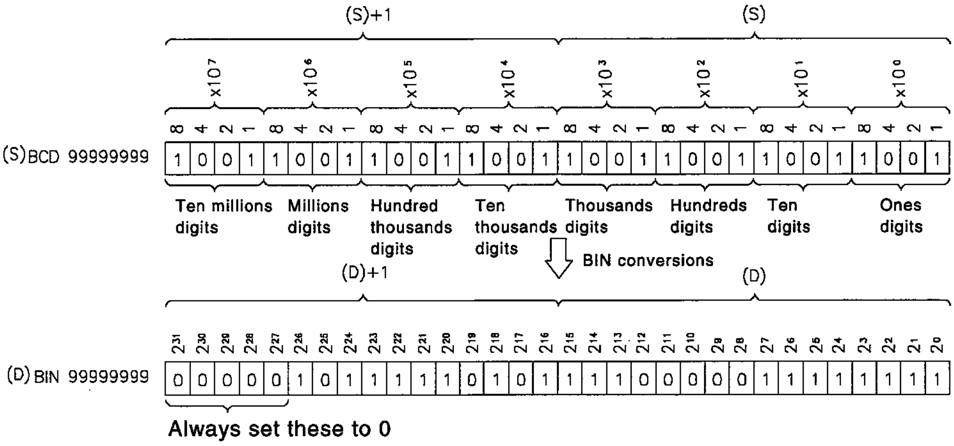
BIN

Converts BCD data (0 to 9999) at device designated by (S) to BIN data, and stores at the device designated by (D).



DBIN

Converts BCD data (0 to 99999999) at device designated by (S) to BIN data, and stores at the device designated by (D).



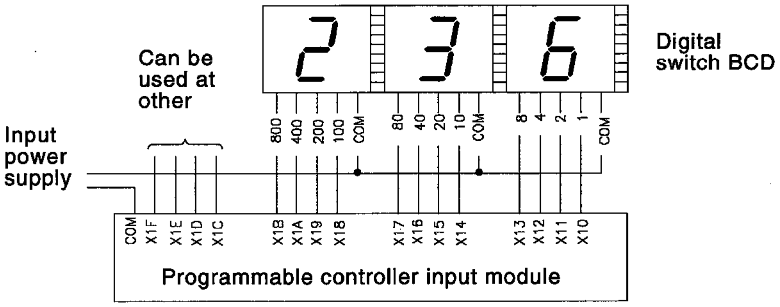
Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

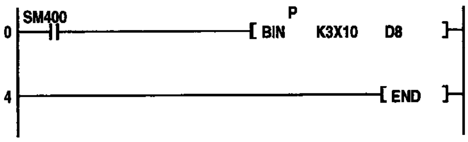
- The values in the individual digits designated by (S) are not in the range of 0 to 9.

Program Example

(1) The following program converts the BCD data at X10 through X1B to BIN when X8 is ON, and stores it at D8.



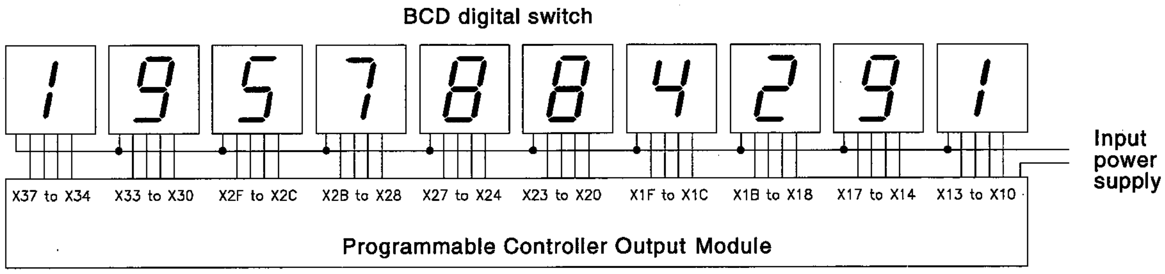
[Ladder Mode]



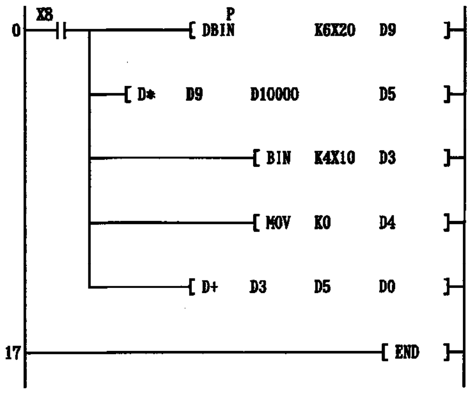
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	BINP	X3X10 D8
4	END	

(2) The following program converts the BCD data at X10 through X37 to BIN when X8 is ON, and stores it at D0 and D1.



[Ladder Mode]



[List Mode]

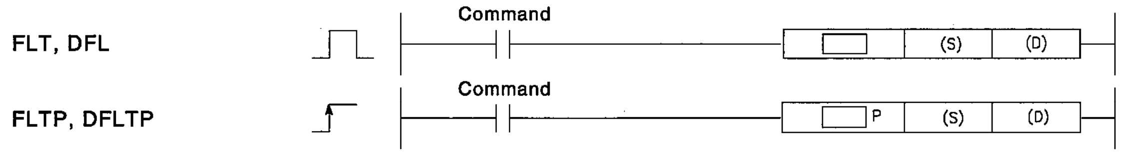
Steps	Instruction	Device
0	LD	X8
1	DBINP	K6X20 D9
4	D*	D9 D10000 D5
8	BIN	K4X10 D3
11	MOV	K0 D4
13	D+	D3 D5 D0
17	END	

If the data set at X10 to X37 is a BCD value which exceeds 2147483647, the value at D0 and D1 will be a negative value, because it exceeds the range of numerical values that can be handled by a 32-bit device.

6.3.3 Conversion from BIN 16 and 32-bit data to floating decimal point

Set Data	Usable Devices								
	Internal Devices (System, Word)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		o	o		o		—
(D)	—	o		—	o		—		—

[Instruction Symbol] [Execution Condition] □ indicates FLT(P)/DFLT(P)



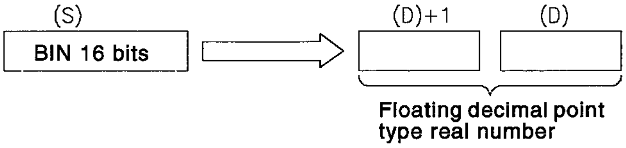
Set Data

Set Data	Meaning	Data Type
(S)	First device number where integer data for the purpose of conversion to floating decimal point data is being stored	BIN 16/32 bits
(D)	First device number that will store converted floating decimal point data	Real number

Functions

FLT

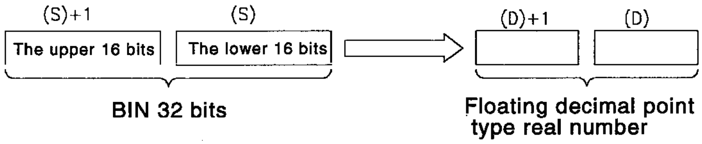
(1) Converts 16-bit BIN data designated by (S) to floating decimal point real number, and stores at device number designated by (D).



(2) BIN values between -32768 and 32767 can be designated by (S).

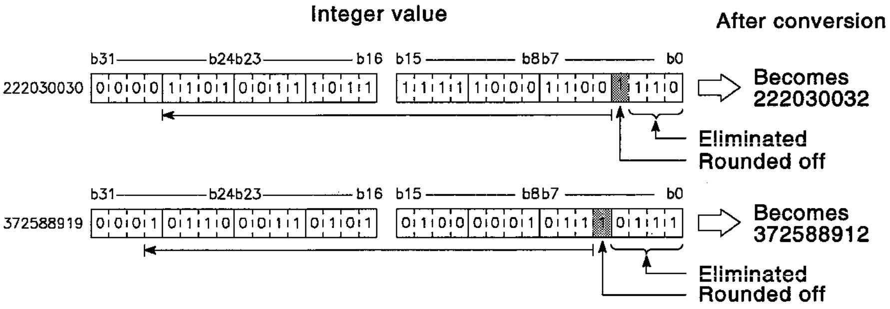
DFLT

(1) Converts 32-bit BIN data designated by (S) to floating decimal point real number, and stores at device number designated by (D).



(2) BIN values between -2147483648 and 2147483647 can be designated by (S)+1 and (S).

(3) Due to the fact that floating decimal point type real numbers are processed by simple 32-bit processing, the number of significant digits is 24 bits if the display is binary and approximately 7 digits if the display is decimal.
For this reason, if the integer exceeds the range of -16777216 to 16777215 (24-bit BIN value), errors can be generated in the conversion value.
The conversion results round off at the 25th bit from the highest bit of the integer value, and eliminate everything from the 26th bit and beyond.



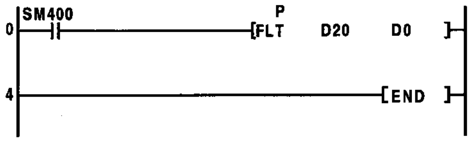
Operation Errors

(1) There are no errors associated with the FLT(P) or DFLT(P) instructions.

Program Example

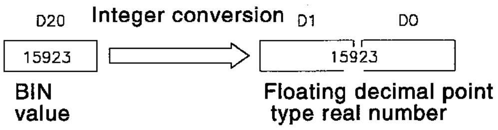
(1) The following program converts the BIN 16-bit data at D20 to a floating decimal point type real number and stores the result at D0 and D1.

[Ladder Mode]



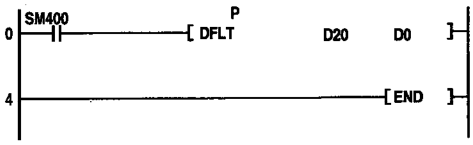
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	FLTP	D20 D0
4	END	



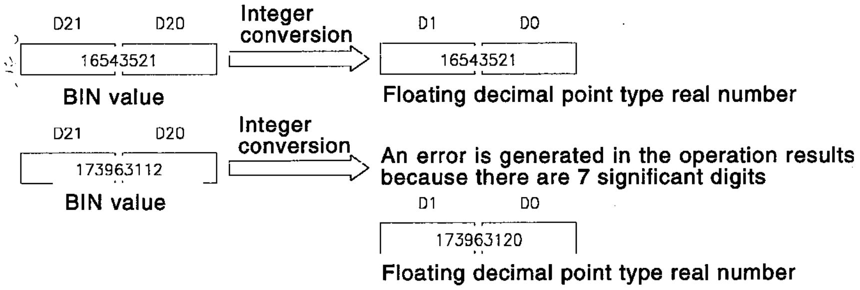
(2) The following program converts the BIN 32-bit data at D20 and D21 to a floating decimal point type real number, and stores the result at D0 and D1.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DINTP	D20 D0
4	END	



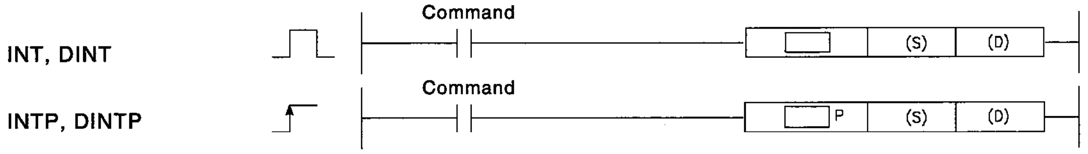
An error is generated in the operation results because there are 7 significant digits

6.3.4 Conversion from floating decimal point data to BIN 16- and 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	o	o		o	o		o	—	—

[Instruction Symbol] [Execution Condition]

□ indicates INT(P)/DINT(P)



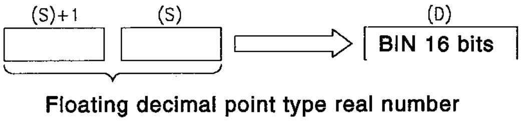
Set Data

Set Data	Meaning	Data Type
(S)	First device number storing floating decimal point data that will be converted to BIN value	Real number
(D)	First device number to store BIN value after conversion	BIN 16/32 bits

Functions

INT

(1) Converts the floating decimal point real number designated at (S) into BIN 16-bit data and stores it at the device number designated at (D).



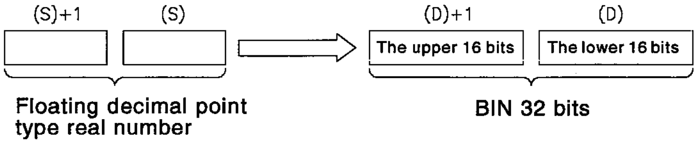
(2) The range of floating decimal point type real numbers that can be designated at (S)+1 or (S) is from -32768 to 32768.

(3) Stores integer values stored at (D) as BIN 16-bit values.

(4) After conversion, the first digit after the decimal point of the real number is rounded off.

DINT

(1) Converts floating decimal point type real number designated by (S) to BIN 32-bit data, and stores the result at the device number designated by (D).



(2) The range of floating decimal point type real numbers that can be designated at (S)+1 or (S) is from -2147483648 to 2147483647.

(3) The integer value stored at (D)+1 and (D) is stored as BIN 32 bits.

(4) After conversion, the first digit after the decimal point of the real number is rounded off.

Operation Errors

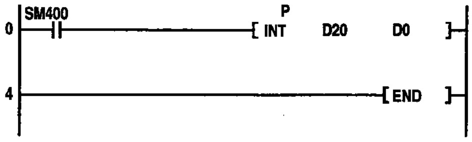
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The floating decimal point type data designated by (S) when the INT instruction was used was outside the -31768 to 32767 range.
- The floating decimal point type data designated by (S) when the DINT instruction was used was outside the -2147483648 to 2147483647 range.

Program Example

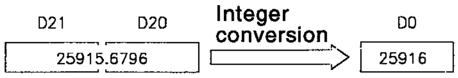
(1) The following program converts the floating decimal point type real number at D20 and D21 to BIN 16-bit data, and stores the result at D0.

[Ladder Mode]

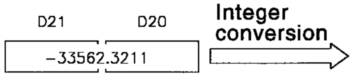


[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	INTP	D20
		D0
4	END	



Floating decimal point type real number BIN value

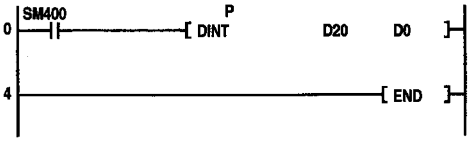


Because the set data is less than -32768, an operation error is returned

Floating decimal point type real number

(2) The following program converts the floating decimal point type real number at D20 and D21 to BIN 32-bit data and stores the result at D0 and D1.

[Ladder Mode]

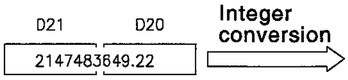


[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DINTP	D20 D0
4	END	

Floating decimal point type real number

BIN value



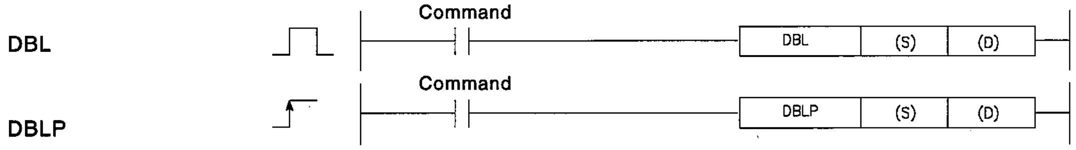
Because the set data is larger than 2147483647, an operation error is returned.

Floating decimal point type real number

6.3.5 Conversion from BIN 16-bit to BIN 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]

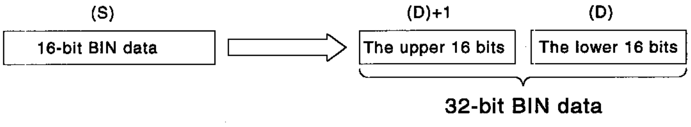


Set Data

Set Data	Meaning	Data Type
(S)	First number of device where BIN 16-bit data is stored	BIN 16 bits
(D)	First number of device where BIN 32-bit data is stored after conversion	BIN 32 bits

Functions

Converts BIN 16-bit data at device designated by (S) to BIN 32-bit data with sign, and stores the result at a device designated by (D).



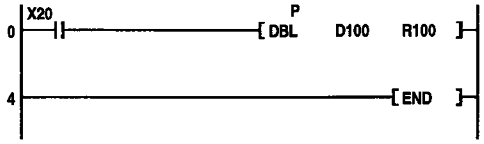
Operation Errors

(1) There are no errors associated with the DBL(P) instruction.

Program Example

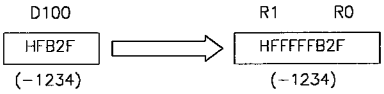
(1) The following program converts the BIN 16-bit data stored at D100 to BIN 32-bit data when X20 is ON, and stores at R100 and R101.

[Ladder Mode]



[List Mode]

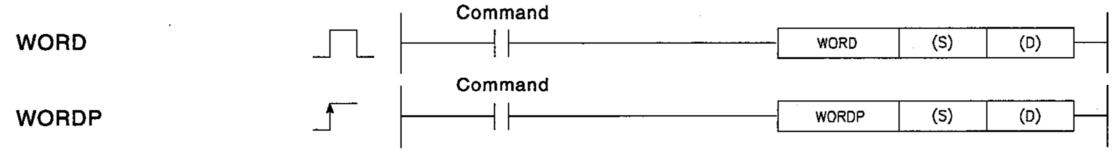
Steps	Instruction	Device
0	LD	X20
1	DBLP	D100 R100
4	END	



6.3.6 Conversion from BIN 32-bit to BIN 16-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]

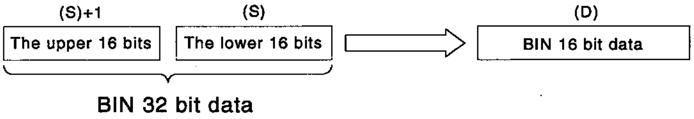


Set Data

Set Data	Meaning	Data Type
(S)	First number of device where BIN 32-bit data is stored	BIN 32 bits
(D)	First number of device where BIN 16-bit data will be stored after conversion	BIN 16 bits

Functions

Converts BIN 32-bit data at device designated by (S) to BIN 16-bit data with sign, and stores the result at a device designated by (D).



Operation Errors

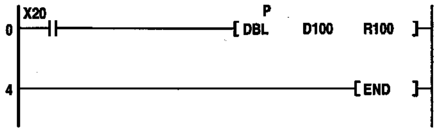
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The contents of the data designated by (S)+1 and (S) are outside the -32768 and 32767 range.(Error code: 4100)

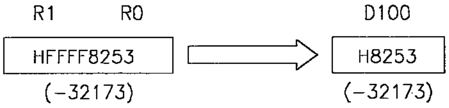
Program Example

(1) The following program converts the BIN 32-bit data at R100 and R101 to BIN 16-bit data when X20 is ON, and stores it at D100.

[Ladder Mode] [List Mode]



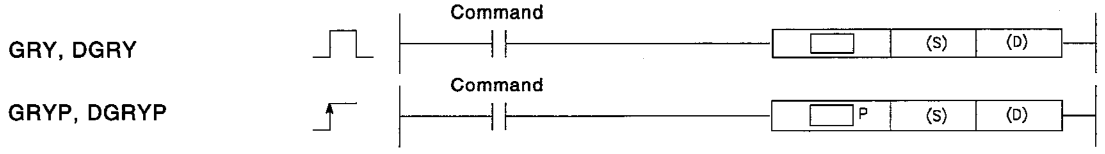
Steps	Instruction	Device
0	LD	X20
1	DBLP	D100 R100
4	END	



6.3.7 Conversion from BIN 16 and 32-bit data to Gray code

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	—							—	—

[Instruction Symbol] [Execution Condition] □ indicates GRY(P)/DGRY(P)



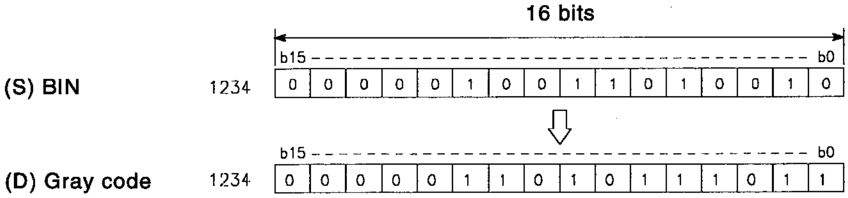
Set Data

Set Data	Meaning	Data Type
(S)	BIN data, or first number of the device where BIN data is stored	BIN 16/32 bits
(D)	First number of device to store Gray code after conversion	Gray code

Functions

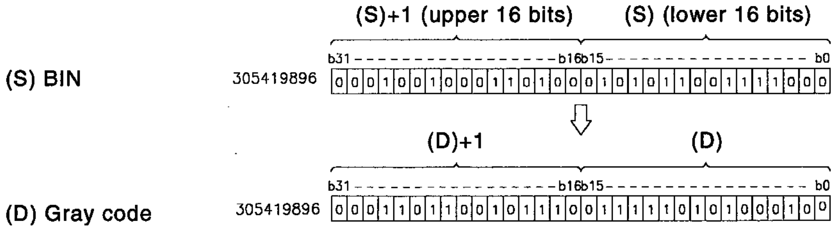
GRY

Converts BIN 16-bit data at the device designated by (S) to Gray code, and stores result at device designated by (D).



DGRY

Converts BIN 32-bit data at the device designated by (S) to Gray code, and stores result at device designated by (D).



Operation Errors

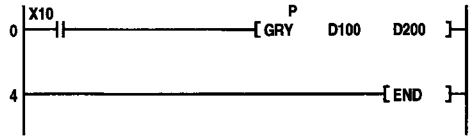
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data at (S) is a negative number.

Program Example

(1) The following program converts the BIN data at D100 to Gray code when X10 is ON, and stores result at D200.

[Ladder Mode]

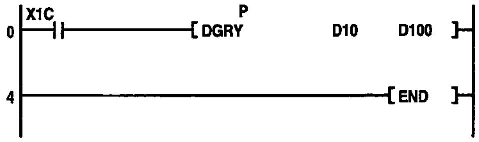


[List Mode]

Steps	Instruction	Device
0	LD	X10
1	GRYP	D100 D200
4	END	

(2) The following program converts the BIN data at D10 and D11 to Gray code when X1C is ON, and stores it at D100 and D101.

[Ladder Mode]



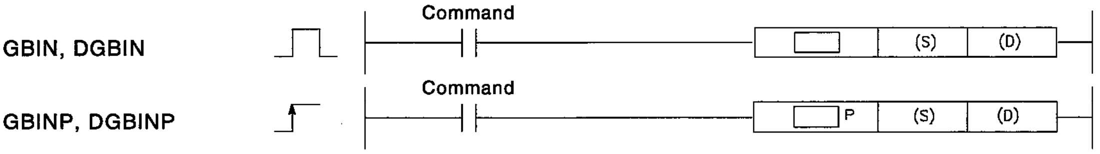
[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	DGRYP	D10 D100
4	END	

6.3.8 Conversion of Gray code to BIN 16 and 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	○							○	—
(D)	○							—	—

[Instruction Symbol] [Execution Condition] □ indicates GBIN(P)/DGBIN(P)



Set Data

Set Data	Meaning	Data Type
(S)	Gray code data or the first number of device where Gray code data is being stored	Gray code
(D)	First number of the device to store BIN data after conversion	BIN 16/32 bits

Functions

GBIN

Converts Gray code data at device designated by (S) to BIN 16-bit data and stores at device designated by (D)

16 Bits

		b15															b0
(S) Gray code	1234	0	0	0	0	0	1	1	0	1	0	1	1	1	0	1	1
(D) BIN	1234	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0

DGBIN

Converts Gray code data at device designated by (S) to BIN 32-bit data and stores at device designated by (D)

(S)+1 (upper 16 bits) (S) (lower 16 bits)

		b31															b16	b15															b0
(S) Gray code	305419896	0	0	0	1	1	0	1	1	0	0	1	0	1	1	1	0	0	1	1	1	1	1	0	1	0	1	0	0	0	1	0	0

(D)+1 (D)

		b31															b16	b15															b0
(D) BIN	305419896	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0

Operation Errors

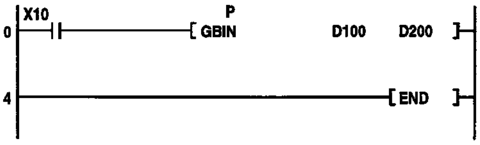
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Data at (S) when GBIN instruction was issued is outside the 0 to 32767 range.
- Data at (S) when DGBIN instruction was issued is outside the 0 to 2147483647 range.

Program Example

(1) The following program converts the Gray code data at D100 when X10 is ON to BIN data, and stores the result at D200.

[Ladder Mode]

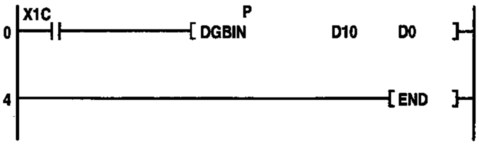


[List Mode]

Steps	Instruction	Device
0	LD	X10
1	GBINP	D100 D200
4	END	

(2) The following program converts the Gray code data at D10 and D11 to BIN data when X1C is ON, and stores the result at D0 and D1.

[Ladder Mode]



[List Mode]

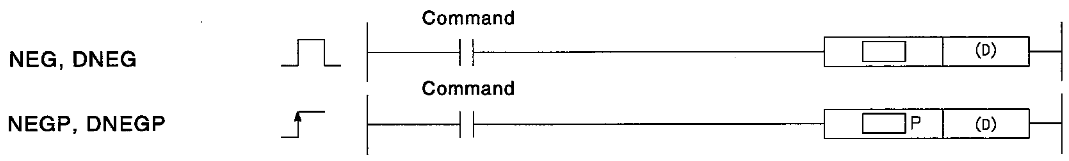
Steps	Instruction	Device
0	LD	X1C
1	DGBINP	D10 D0
4	END	

6.3.9 Complement of 2 of BIN 16- and 32-bit data (sign reversal)

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	o							—	

[Instruction Symbol] [Execution Condition]

□ indicates NEG(P)/DNEG(P)



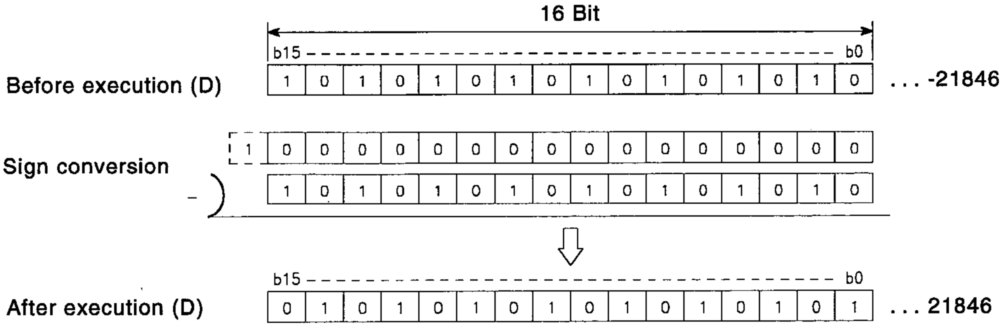
Set Data

Set Data	Meaning	Data Type
(D)	First number of device storing data for the complement of 2 operation	BIN 16/32 bits

Functions

NEG

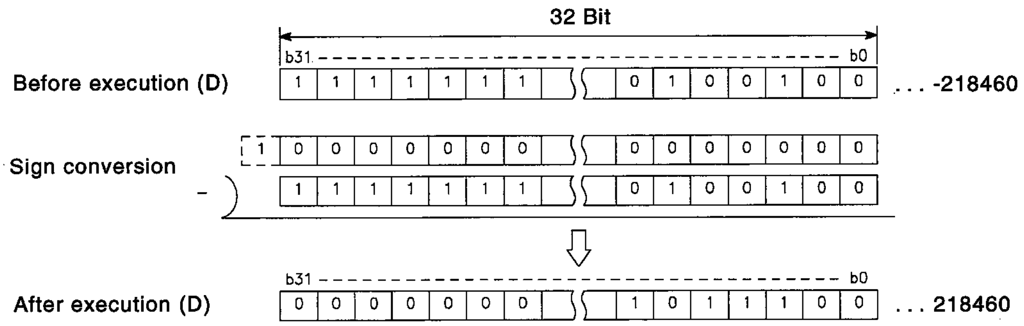
(1) Reverses the sign of the 16-bit device designated by (D) and stores at the device designated by (D)



(2) Used when reversing positive and negative signs

DNEG

(1) Reverses the sign of the 32-bit device designated by (D) and stores at the device designated by (D)



(2) Used when reversing positive and negative signs

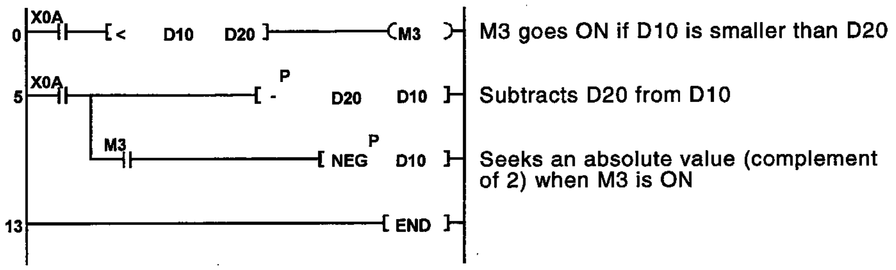
Operation Errors

(1) There are no operation errors associated with the NEG(P) or DNEG(P) instructions.

Program Example

(1) The following program calculates a total for the data at D10 through D20 when XA goes ON, and seeks an absolute value if the result is negative.

[Ladder Mode]



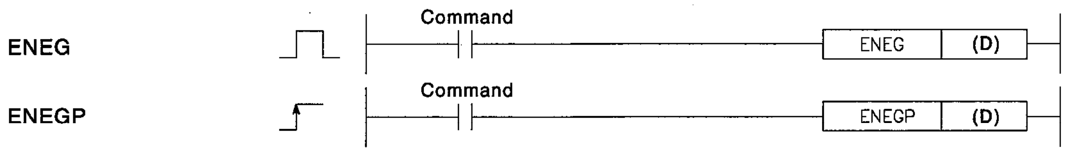
[List Mode]

Steps	Instruction	Device
0	LD	X0A
1	AND<	D10
		D20
4	OUT	M3
5	LD	X0A
6	-P	D20
		D16
10	AND	M3
11	NEGP	D10
13	END	

6.3.10 Sign reversal for floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	—	o		—	o		—	—	

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(D)	First number of device storing floating decimal point data for which sign will be inverted	Real number

Functions

(1) Reverses the sign of the floating decimal point type real number data designated by (D), and stores at the device designated by (D).

(2) Used when reversing positive and negative signs

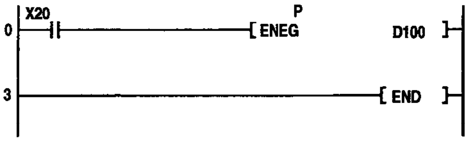
Operation Errors

(1) There are no errors associated with the ENEG(P) instruction.

Program Example

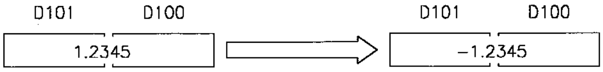
(1) The following program inverts the sign of the floating decimal point type real number data at D100 and D101 when X20 goes ON, and stores result at D100 and D101.

[Ladder Mode]



[List Mode]

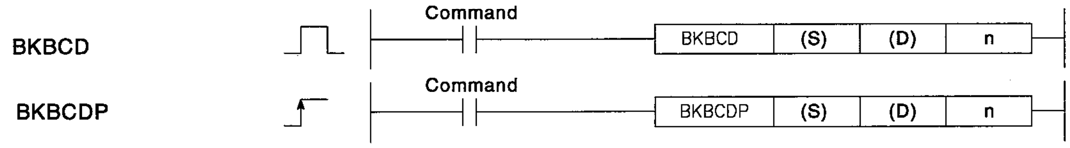
Steps	Instruction	Device
0	LD	X20
1	ENEGP	D100
3	END	



6.3.11 Conversion from block BIN 16-bit data to BCD 4-digit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—					—
(D)	—	o		—					—
n	o	o		o					—

[Instruction Symbol] [Execution Condition]

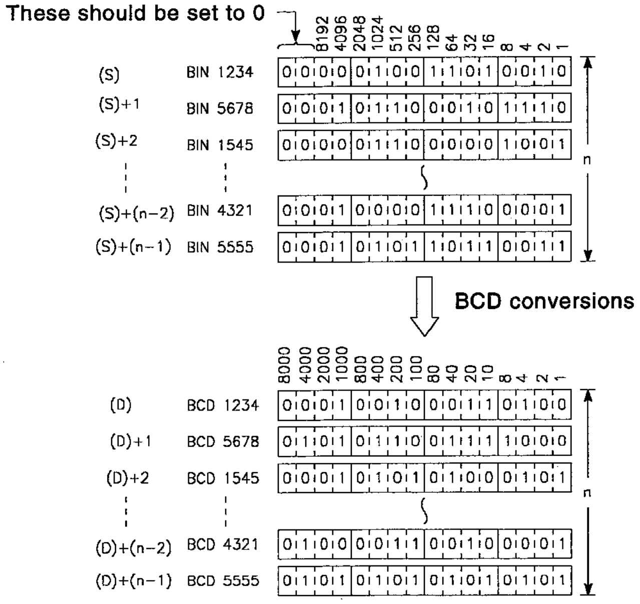


Set Data

Set Data	Meaning	Data Type
(S)	First number of device storing BIN data	BIN 16 bits
(D)	First number of device which will store BCD data after conversion	
n	Number of data blocks converted	

Functions

(1) Converts BIN data (0 to 9999) n-points from device designated by (S) to BCD, and stores result following the device designated by (D).



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

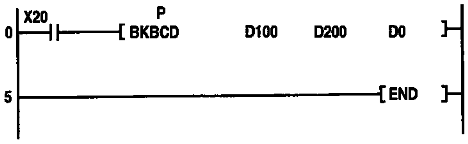
- The range n-points from the device at (S) or (D) exceeds the relevant device (Error code: 4101)
- The data n points from the device designated by (S) is outside the 0 to 9999 range. (Error code: 4100)
- The (S) and (D) devices overlap. (Error code: 4101)

(2) See Section 3.6 for information regarding errors during index qualification.

Program Example

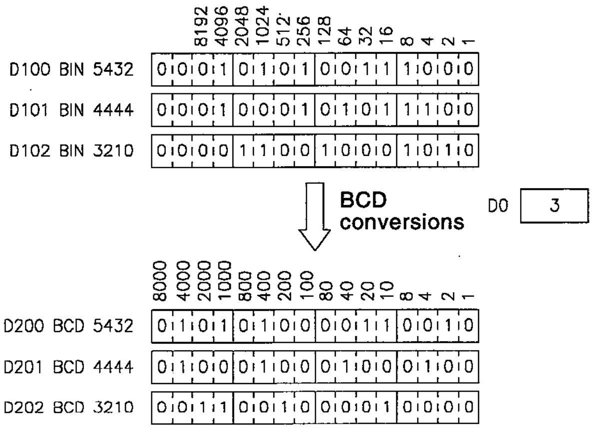
(1) The following program converts BIN 16-bit data the number of points from D100 corresponding to the value stored at D0 to BCD when X20 goes ON, and stores the result following D200.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	BKBCDP	D100 D200 D0
5	END	



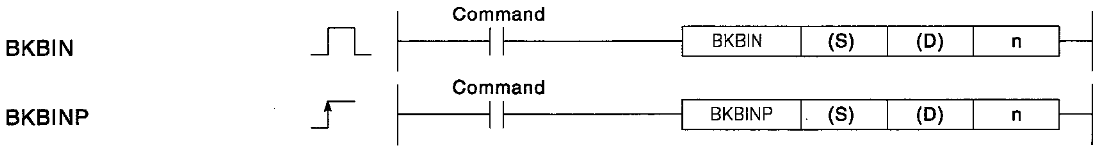
	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
D100 BIN 5432	0	0	0	1	0	1	0	1	0	0	1	1	1	0	0	0
D101 BIN 4444	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0
D102 BIN 3210	0	0	0	0	1	1	0	0	1	0	0	0	1	0	1	0

	8000	4000	2000	1000	800	400	200	100	80	40	20	10	8	4	2	1
D200 BCD 5432	0	1	0	1	0	1	0	0	0	0	1	1	0	0	1	0
D201 BCD 4444	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
D202 BCD 3210	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

6.3.12 Conversion from block BCD 4-digit data to block BIN 16-bit data

<table>
<tr><th rowspan="3">Set Data</th><th colspan="9">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System, User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th rowspan="2">Constant K, H</th><th rowspan="2">Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th></tr>
<tr><td>(S)</td><td>—</td><td colspan="2">o</td><td colspan="5">—</td><td>—</td></tr>
<tr><td>(D)</td><td>—</td><td colspan="2">o</td><td colspan="5">—</td><td>—</td></tr>
<tr><td>n</td><td>o</td><td colspan="2">o</td><td colspan="5">o</td><td>—</td></tr>
</table>

[Instruction Symbol] [Execution Condition]

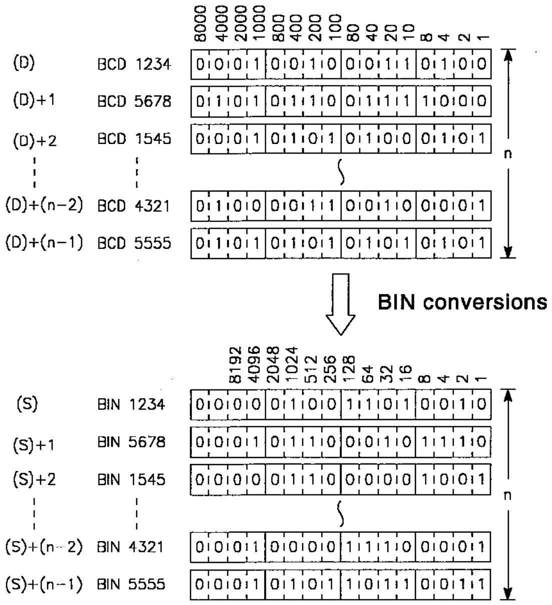


Set Data

Set Data	Meaning	Data Type
(S)	First number of device storing BCD data	BIN 16 bits
(D)	First number of the device to store BIN data after conversion	
n	Number of data blocks converted	

Functions

(1) Converts BCD data (0 to 9999) n-points from device designated by (S) to BIN, and stores result following the device designated by (D).



Operation Errors

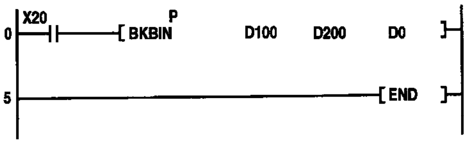
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range n-points from the (S) or (D) device exceeds the relevant device.
- The data n-points at the (S) device is outside the 0 to 9999 range.
- The (S) and (D) devices overlap.

Program Example

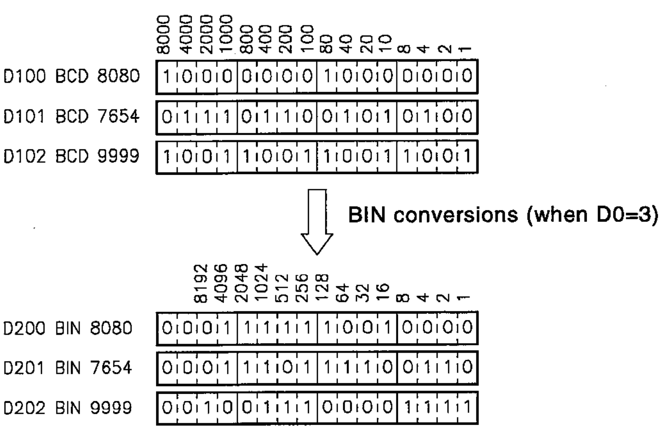
(1) The following program converts BCD data the number of points from D100 corresponding to the value stored at D0 to BIN data when X20 goes ON, and stores the result from D200 onward.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	BKBINP	D100 D200 D0
5	END	

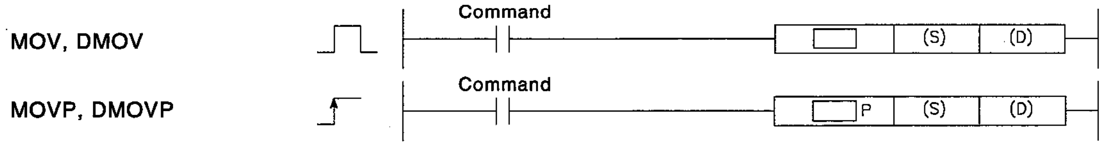


6.4 Data Transfer Instructions

6.4.1 16-bit and 32-bit data transfers

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition] □ indicates MOV/DMOV



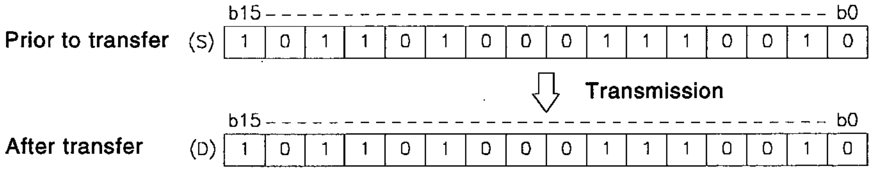
Set Data

Set Data	Meaning	Data Type
(S)	Source data or first number of device storing data	BIN 16/32 bits
(D)	Number of destination device	

Functions

MOV

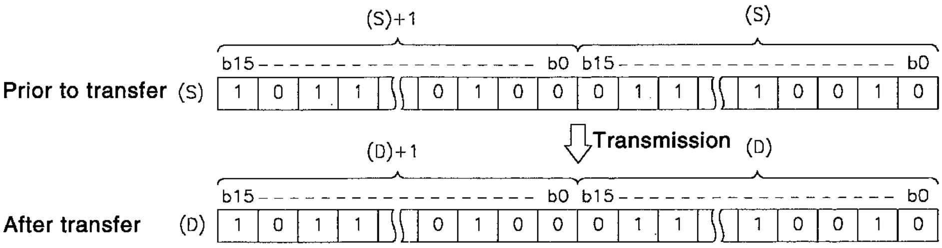
(1) Transfers the 16-bit data from the device designated by (S) to the device designated by (D).



	b15															b0
Prior to transfer (S)	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	0
After transfer (D)	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	0

DMOV

(2) Transfers 32-bit data at the device designated by (S) to the device designated by (D).



	(S)+1 b15											(S) b0
Prior to transfer (S)	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	0
After transfer (D)	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	0

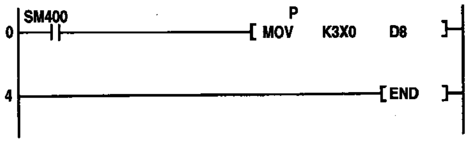
Operation Errors

(1) There are no operation errors associated with the MOV(P) or DMOV(P) instructions.

Program Example

(1) The following program stores input data from X0 to XB at D8.

[Ladder Mode]

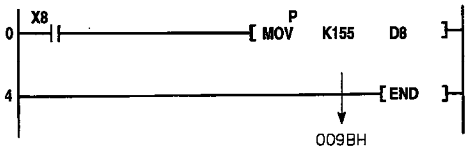


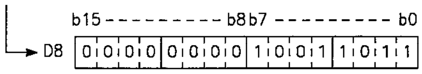
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	MOVP	K3X0
		D8
4	END	

(2) The following program stores the constant K155 at D8 when X8 goes ON.

[Ladder Mode]



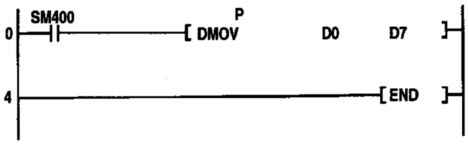


[List Mode]

Steps	Instruction	Device
0	LD	X8
1	MOVP	K155
		D8
4	END	

(3) The following program stores the data from D0 and D1 at D7 and D8.

[Ladder Mode]

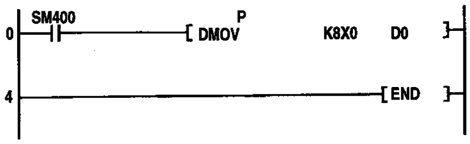


[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DMOVP	D0
		D7
4	END	

(4) The following program stores the data from X0 to X1F at D0 and D1.

[Ladder Mode]



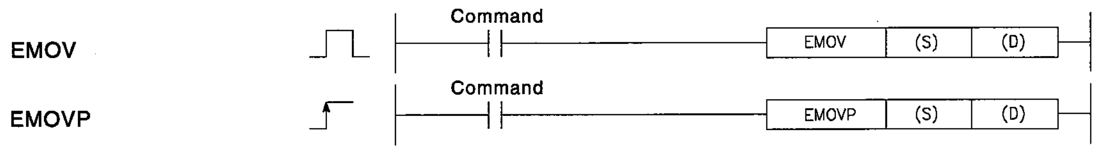
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DMOVP	K8K0
		D0
4	END	

6.4.2 Floating decimal point data transfers

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

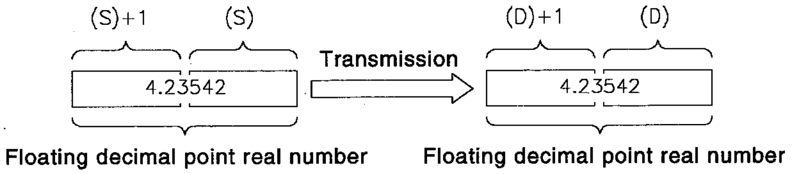


Set Data

Set Data	Meaning	Data Type
(S)	Transfer data, or number of device storing transfer data	Real number
(D)	Number of device to store transferred data	

Functions

(1) Transfers floating decimal point type real number data being stored at the device designated by (S) to a device designated by (D).



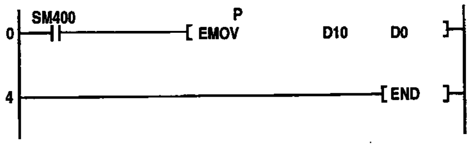
Operation Errors

(1) There are no operation errors associated with the EMOV(P) instruction.

Program Example

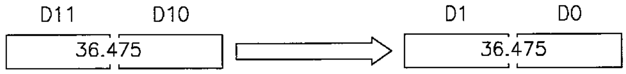
(1) The following program stores the real numbers at D10 and D11 at D0 and D1.

[Ladder Mode]



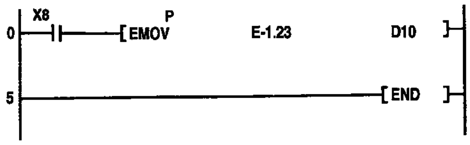
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	EMOVP	D10
		D0
4	END	



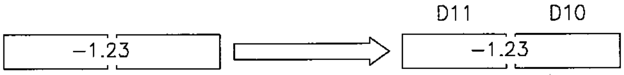
(2) The following program stores the real number -1.23 at D10 and D11 when X8 is ON.

[Ladder Mode]



[List Mode]

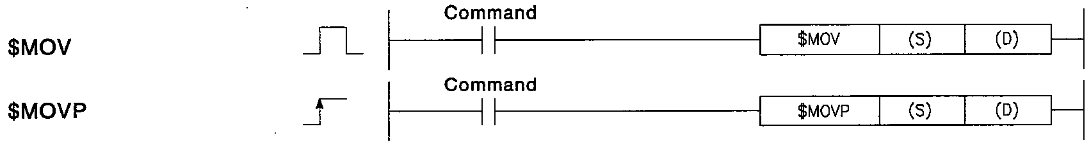
Steps	Instruction	Device
0	LD	X8
1	EMOVP	E-1.23
		D10
5	END	



6.4.3 Character string transfers

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—
(D)	—	o		—				—	—

[Instruction Symbol] [Execution Condition]

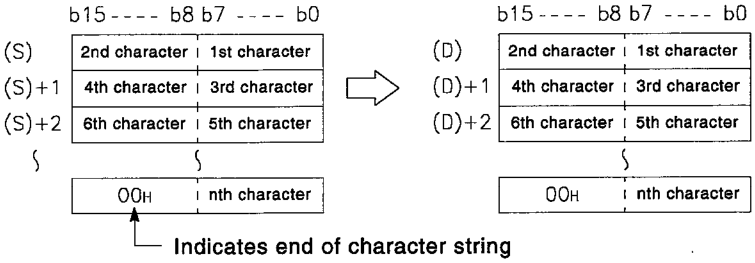


Set Data

Set Data	Meaning	Data Type
(S)	Character string to transfer, or the first number of the device storing the character string	Character string
(D)	First number of device to store transferred character string	

Functions

(1) Transfers character string data stored from device number designated by (S) from device number designated by (D) onward.
A character string transfer involves the transfer of data from the device number designated by (S) to the device number storing the "00_H" code in one operation.



(2) Processing will be performed without error even in cases where the range for the devices storing the character data to be transferred ((S) to (S)+n) overlaps with the range of the devices which will store the character string data after it has been transferred ((D) to (D)+n). The following occurs when the character string data that had been stored from D10 to D13 is transferred to D11 to D14:

	b15 ---- b8	b7 ---- b0
D10	32H (2)	31H (1)
D11	34H (4)	33H (3)
D12	36H (6)	35H (5)
D13	00H	
D14		

⇨

	b15 ---- b8	b7 ---- b0
D10	32H (2)	31H (1)
D11	32H (2)	31H (1)
D12	34H (4)	33H (3)
D13	36H (6)	35H (5)
D14	00H	

... Remains as the character string it was prior to transfer

(3) If the "00H" code is being stored at lower bytes of (S)+n, "00H" will be stored at both the higher bytes and the lower bytes of (D)+n.

	b15 ---- b8	b7 ---- b0
(S)	42H (B)	41H (A)
(S)+1	44H (D)	43H (C)
(S)+2	45H (E)	00H

Higher byte is not transferred

⇨

	b15 ---- b8	b7 ---- b0
(D)	42H (B)	41H (A)
(D)+1	44H (D)	43H (C)
(D)+2	00H	00H

The "00H" code is automatically stored at the higher byte.

Operation Errors

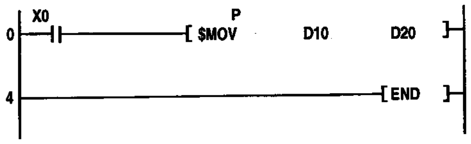
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- There is no "00H" code stored between the device number designated by (S) and the relevant device. (Error code: 4101)
- It is not possible to store the entire designated character string in the number of points from the device designated by (D) to the final device number cited. (Error code: 4101)

Program Example

(1) The following program transfers the character string data being stored from D10 to D12 to D20 through D22 when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	$MOVP	D10 D20
4	END	

	b15 b8	b7 b0
D10	4DH (M)	2AH (*)
D11	45H (E)	45H (E)
D12	00H	

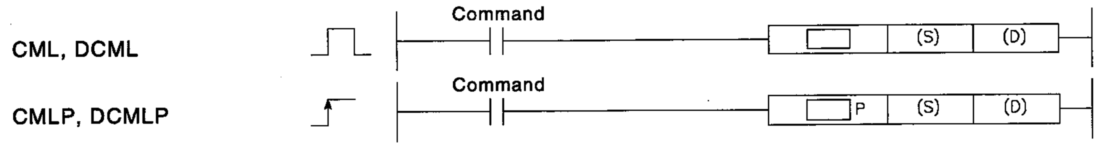
⇨

	b15 b8	b7 b0
D20	4DH (M)	2AH (*)
D21	45H (E)	45H (E)
D22	00H	

6.4.4 16-bit and 32-bit negation transfers

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition] □ indicates CML/DCML



Set Data

Set Data	Meaning	Data Type
(S)	Data to be inverted, or number of device storing this data	BIN 16/32 bits
(D)	Number of device that will store results of inversion	

Functions

CML

(1) Inverts 16-bit data designated by (S) bit by bit, and transfers the result to the device designated by (D)

	b15															b0
Before execution (S)	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	0

⇩ Inversion

	b15															b0
After execution (D)	0	1	0	0	1	0	1	1	1	0	0	0	1	1	0	1

DCML

(1) Inverts 32-bit data designated by (S) bit by bit, and transfers the result to the device designated by (D)

	(S)+1											(S)						
	b15								b0	b15								b0
Before execution (S)	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	0

⇩ Inversion

	(D)+1											(D)						
	b15								b0	b15								b0
After execution (D)	0	1	0	0	1	0	1	1	1	0	0	0	1	1	0	1

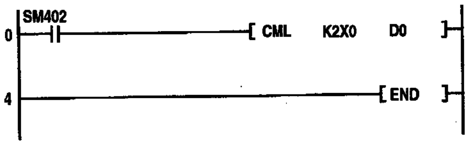
Operation Errors

(1) There are no operation errors associated with the CML(P) or DCML(P) instructions.

Program Example

(1) The following program inverts the data from X0 to X7, and transfers result to D0.

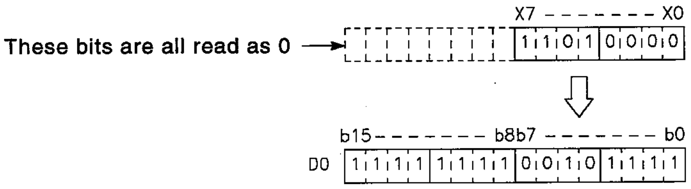
[Ladder Mode]



[List Mode]

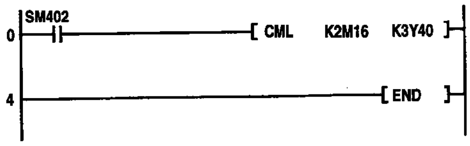
Steps	Instruction	Device
0	LD	SM402
1	CML	K2X0 D0
4	END	

When the number of bits at (S) is less than the number of bits at (D)



(2) The following program inverts the data at M16 through M23, and transfers the result to Y40 to Y47.

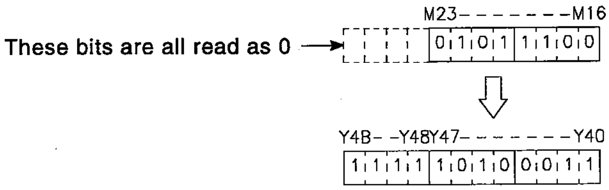
[Ladder Mode]



[List Mode]

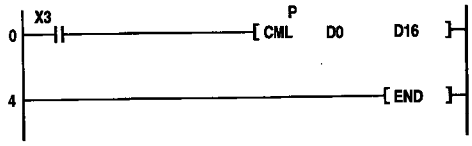
Steps	Instruction	Device
0	LD	SM402
1	CML	K2M16 K3Y40
4	END	

When the number of bits at (S) is less than the number of bits at (D)



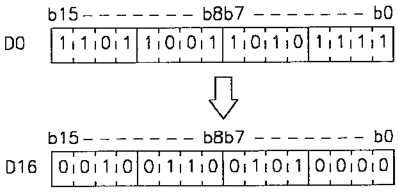
(3) The following program inverts the data at D0 when X3 is ON, and stores the result at D16.

[Ladder Mode]



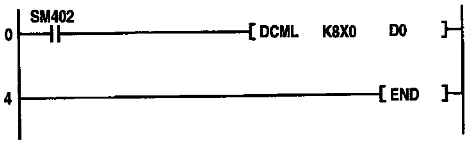
[List Mode]

Steps	Instruction	Device
0	LD	X3
1	CMLP	D0 D16
4	END	



(4) The following program inverts the data at X0 to X1F, and transfers results to D0 and D1.

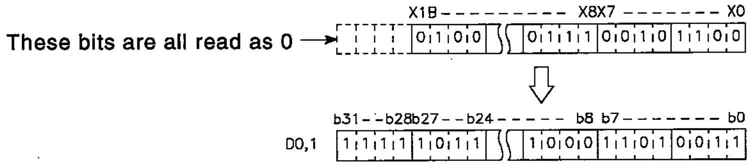
[Ladder Mode]



[List Mode]

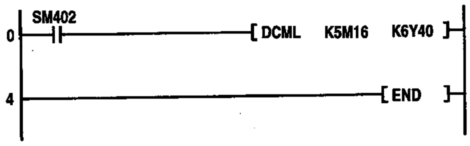
Steps	Instruction	Device
0	LD	SM402
1	DCML	K8X0 D0
4	END	

When the number of bits at (S) is less than the number of bits at (D)



(5) The following program inverts the data at M16 through M35, and transfers it to Y40 to Y53.

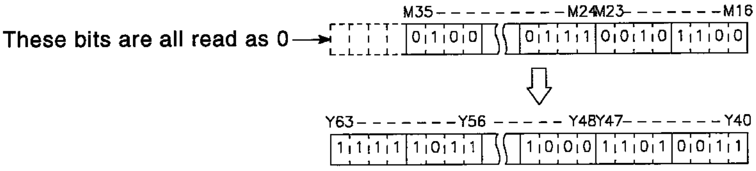
[Ladder Mode]



[List Mode]

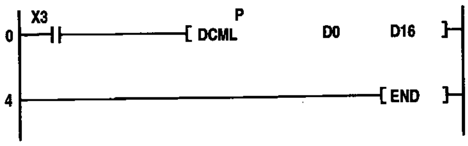
Steps	Instruction	Device
0	LD	SM402
1	DCML	K5M16 K6Y40
4	END	

When the number of bits at (S) is less than the number of bits at (D)



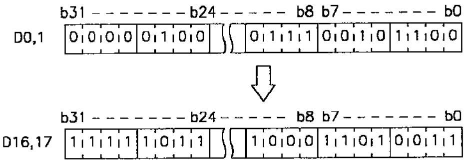
(6) Inverts the data at D0 and D1 when X3 is ON, and stores the result at D16 and D17.

[Ladder Mode]



[List Mode]

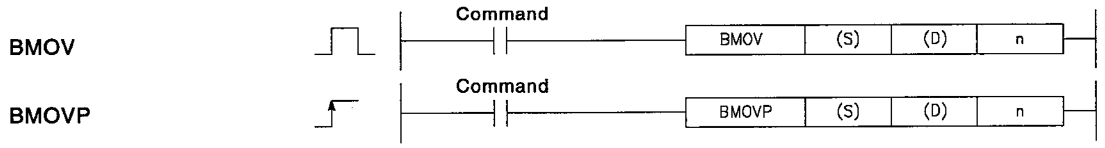
Steps	Instruction	Device
0	LD	X3
1	DCMLP	D0 D16
4	END	



6.4.5 Block 16-bit data transfers

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o						—		—
(D)	o						—		—
n	o						o		—

[Instruction Symbol] [Execution Condition]

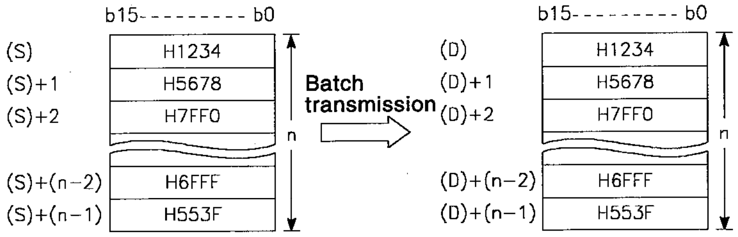


Set Data

Set Data	Meaning	Data Type
(S)	First number of device storing data to be transferred	BIN 16 bits
(D)	First number of destination device	
n	Number of transfers (If special direct device (L□/G□) is used: 1 to 6144)	

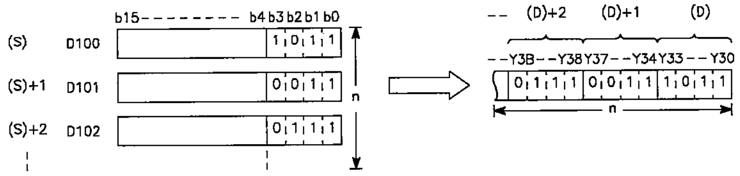
Functions

(1) Transfers in batch 16-bit data n-points from the device designated by (S) to location n-points from the device designated by (D).



(2) Transfers can be accomplished even in cases where there is an overlap between the source and destination device.
In the case of transmission to the smaller device number, transmission is from (S); for transmission to the larger device number, transmission is from (S) + (n-1).

(3) When (S) is a word device and (D) is a bit device, the object for the word device will be the number of bits designated by the bit device digit designation.
If K1Y30 has been designated by (D), the lower four bits of the word device designated by (S) will become the object.



(4) If bit device has been designated for (S) and (D), then (S) and (D) should always have the same number of digits.

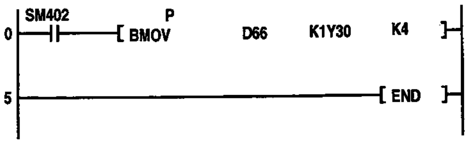
Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.
- The device range n-points from (S) or (D) exceeds the relevant device. (Error code: 4101)
- The number of transfers exceeds 6144 when a special direct device is used. (Error code: 4101)

Program Example

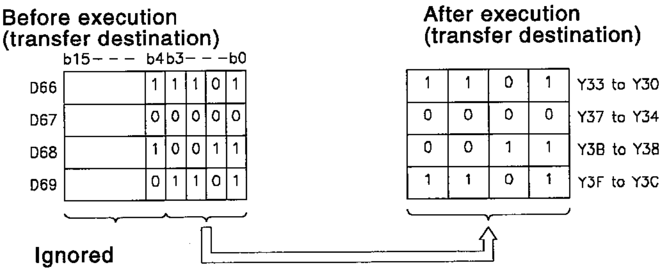
(1) The following program outputs the lower 4 bits of data at D66 to D69 to Y30 through Y3F in 4-point units.

[Ladder Mode]



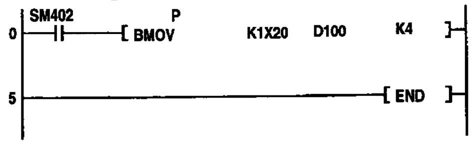
[List Mode]

Steps	Instruction	Device
0	LD	SM402
1	BMOVP	D66 K1Y30 K4
5	END	



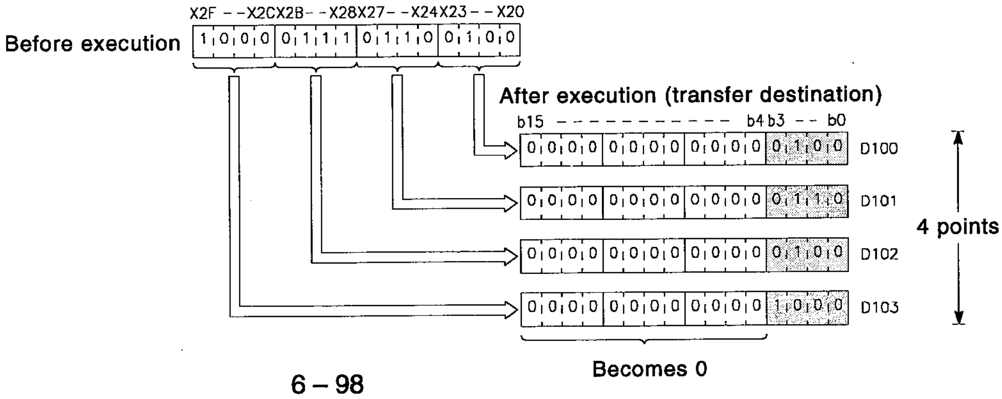
(2) The following program outputs the data at X20 to X2F to D100 through D103 in 4-point units.

[Ladder Mode]



[List Mode]

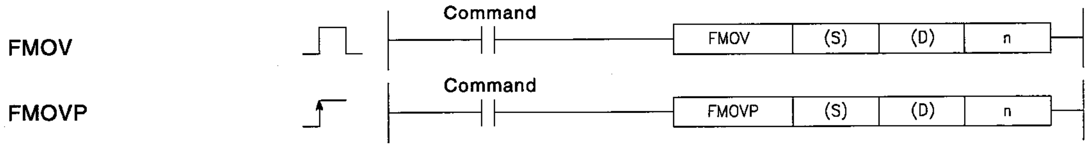
Steps	Instruction	Device
0	LD	SM402
1	BMOVP	K1X20 D100 K4
5	END	



6.4.6 Identical 16-bit data block transfers

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o						o		—
(D)	o						—		—
n	o						o		—

[Instruction Symbol]. [Execution Condition]

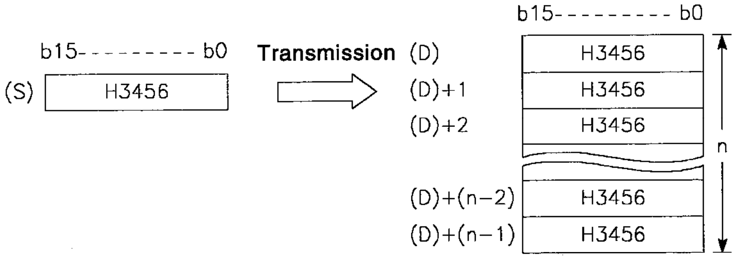


Set Data

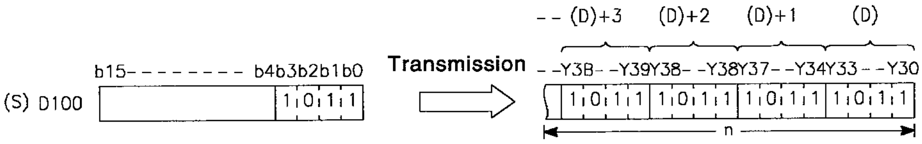
Set Data	Meaning	Data Type
(S)	Data to transfer, or first number of device storing data to transfer	BIN 16 bits
(D)	First number of destination device	
n	Number of transfers (If special direct device (U□/G□) is used: 1 to 6144)	

Functions

(1) Transfers 16-bit data from device designated by (S) to location n-points from device designated by (D).



(2) In cases where (S) designates a word device and (D) a bit device, the number of bits designated by digit designation for the bit device will be the object bits for the word device.
If K1Y30 has been designated by (D), the object bits for the word device designated by (S) will be the lower 4 bits.



(3) If bit device has been designated for (S) and (D), then (S) and (D) should always have the same number of digits.

Operation Errors

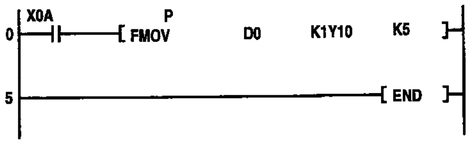
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The device range n-points from (D) exceeds the device range. (Error code: 4101)
- The number of transfers exceeds 6144 when a special direct device is used. (Error code: 4101)

Program Example

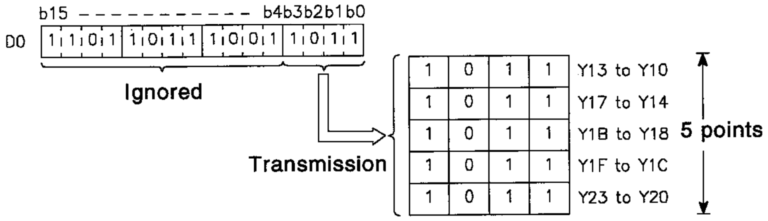
(1) The following program outputs the lower 4 bits of D0 when XA goes ON to Y10 to Y23 in 4-bit units.

[Ladder Mode]



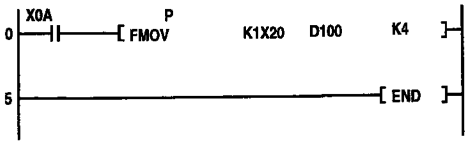
[List Mode]

Steps	Instruction	Device
0	LD	X0A
1	FMOVP	D0 K1Y10 K5
5	END	



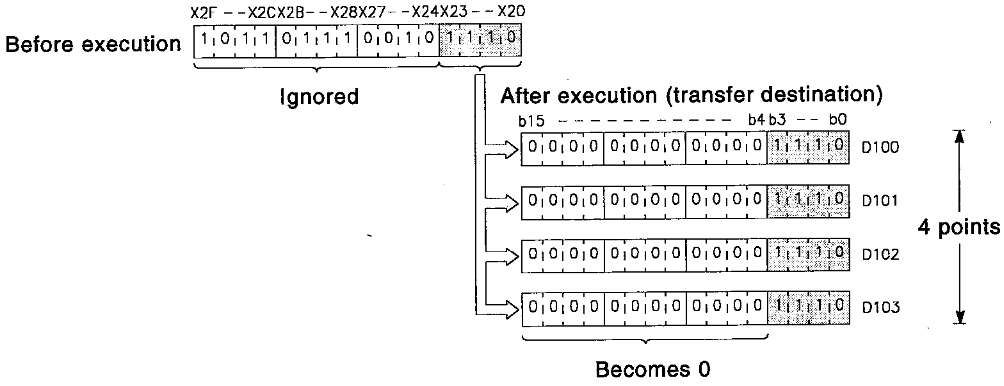
(2) The following program outputs the data at X20 through X23 to D100 through D103 when XA goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0A
1	FMOVP	K1X20 D100 K4
5	END	



6.4.7 16-bit and 32-bit data exchanges

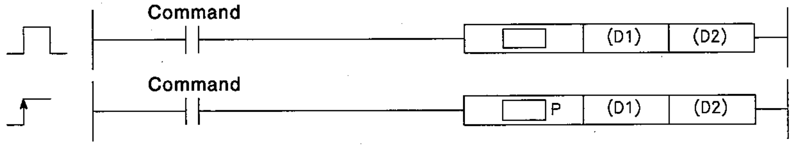
Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J⬚\⬚		Special Function Module U⬚\G⬚	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D1)	o							—	
(D2)	o							—	

[Instruction Symbol] [Execution Condition]

☐ indicates XCH/DXCH

XCH, DXCH

XCHP, DXCHP



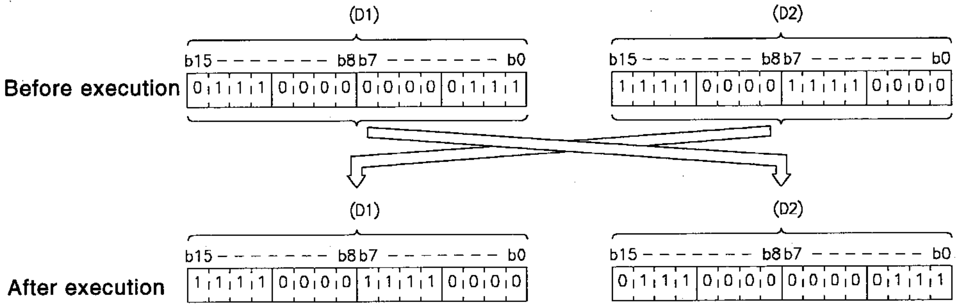
Set Data

Set Data	Meaning	Data Type
(D1)	First number of device storing data to be exchanged	BIN 16/32 bits
(D2)		

Functions

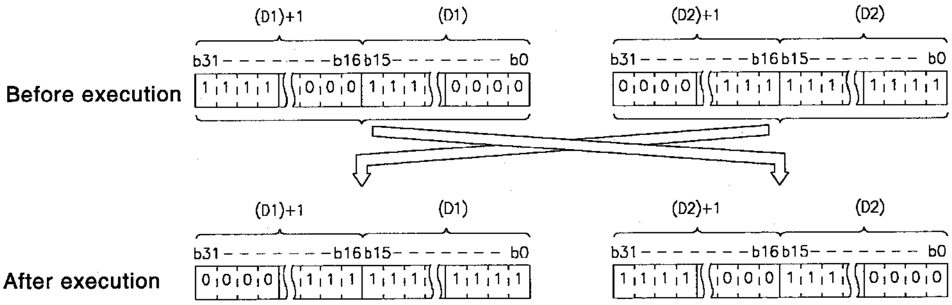
XCH

(1) Conducts 16-bit data exchange between (D1) and (D2).



DXCH

(1) Conducts 32-bit data exchange between (D1)+1, (D1) and (D2)+1, (D2).



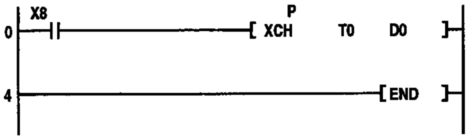
Operation Errors

(1) There are no errors associated with the XCH(P) and DXCH(P) instructions.

Program Example

(1) The following program exchanges the present value of T0 with the contents of D0 when X8 goes ON.

[Ladder Mode]

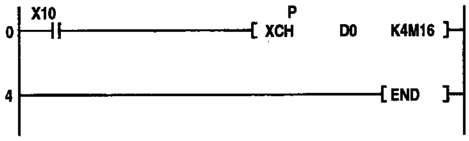


[List Mode]

Steps	Instruction	Device
0	LD	X8
1	XCHP	T0
		D0
4	END	

(2) The following program exchanges the contents of D0 with the data from M16 through M31 when X10 goes ON.

[Ladder Mode]

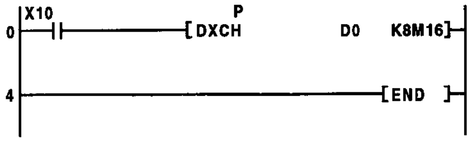


[List Mode]

Steps	Instruction	Device
0	LD	X10
1	XCHP	D0
		K4M16
4	END	

(3) The following program exchanges the contents of D0 and D1 with the data at M16 through M47 when X10 goes ON.

[Ladder Mode]

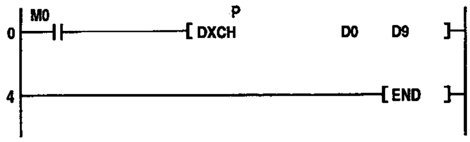


[List Mode]

Steps	Instruction	Device
0	LD	X10
1	DXCHP	D0
		K8M16
4	END	

(4) The following program exchanges the contents of D0 and D1 with those of D9 and D10 when M0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	M0
1	DXCHP	D0
		D9
4	END	

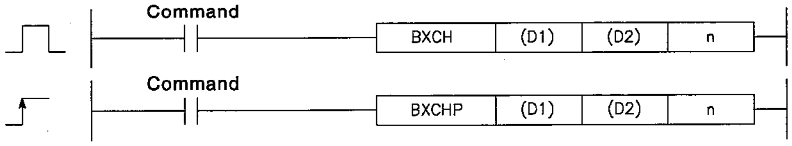
6.4.8 Block 16-bit data exchanges

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D1)	—	o		—					—
(D2)	—	o		—					—
n	o	o		o					—

[Instruction Symbol] [Execution Condition]

BXCH

BXCHP

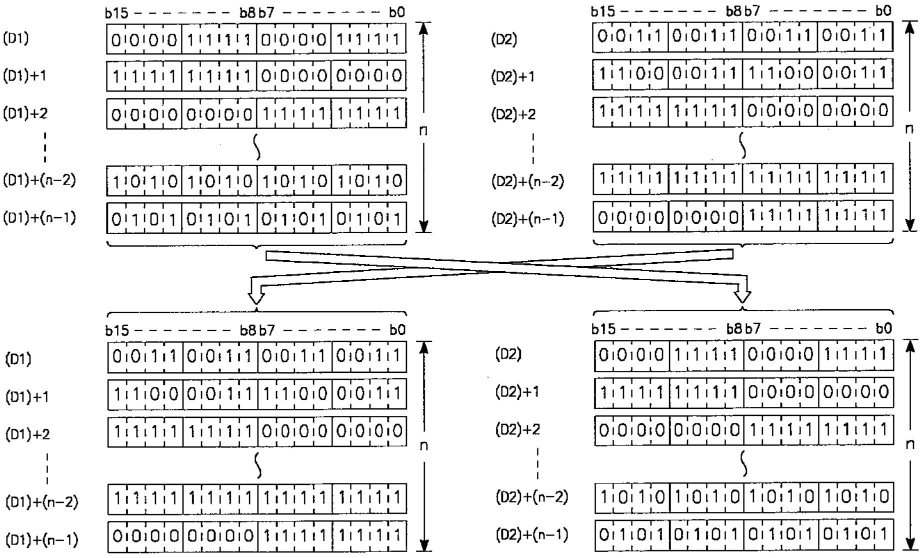


Set Data

Set Data	Meaning	Data Type
(D1)	First number of device storing data to be exchanged	BIN 16 bits
(D2)		
n	Number of exchanges	

Functions

(1) Exchanges 16-bit data n-points from device designated by (D1) and 16-bit data n-points from device designated by (D2).



Operation Errors

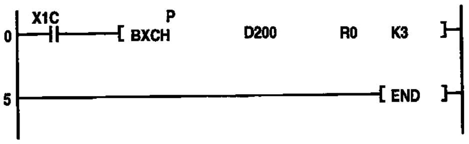
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range n-points from the (D1) or (D2) devices exceeds relevant device. (Error code: 4101)
- (D1) and (D2) devices are overlapping. (Error code: 4101)

Program Example

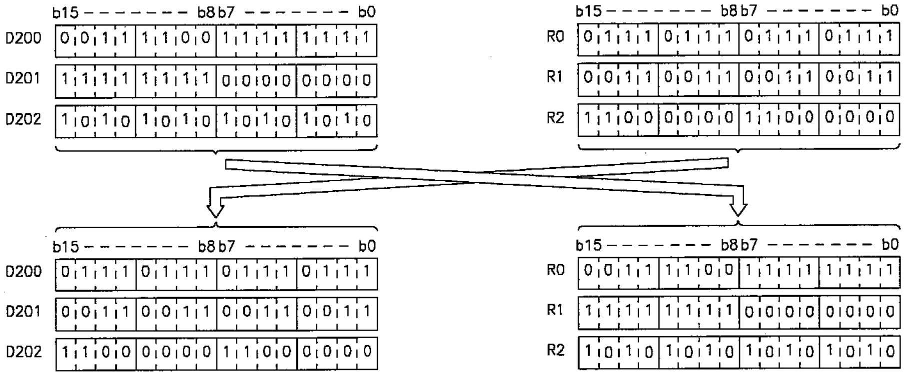
(1) The following program exchanges 16-bit data for 3 points from D200 for 16-bit data for 3 points from R0 when X1C goes ON.

[Ladder Mode]



[List Mode]

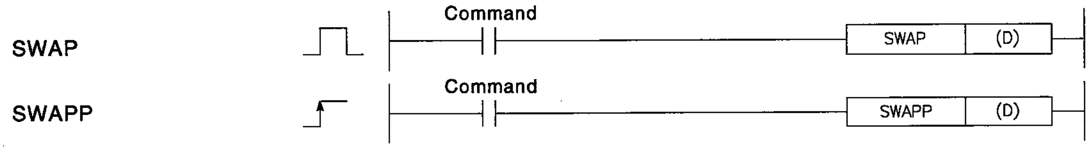
Steps	Instruction	Device
0	LD	X1C
1	BXCHP	D200 R0 K3
5	END	



6.4.9 Upper and lower byte exchanges

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	o							—	

[Instruction Symbol] [Execution Condition]

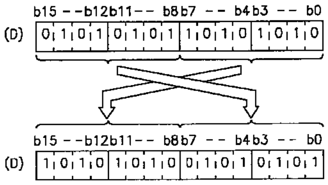


Set Data

Set Data	Meaning	Data Type
(D)	First number of device where data is stored	BIN 16 bits

Functions

(1) Exchanges the higher and lower 8 bits of the device designated by (D).



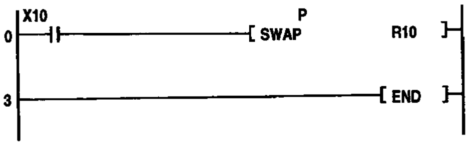
Operation Errors

(1) There are no operation errors associated with the SWAP(P) instruction.

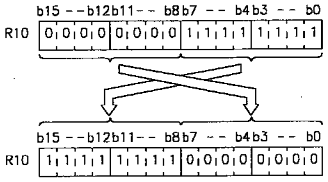
Program Example

(1) The following program exchanges the higher 8 bits and lower 8 bits of R10 when X10 goes ON.

[Ladder Mode] [List Mode]



Steps	Instruction	Device
0	LD	X10
1	SWAPP	R10
3	END	

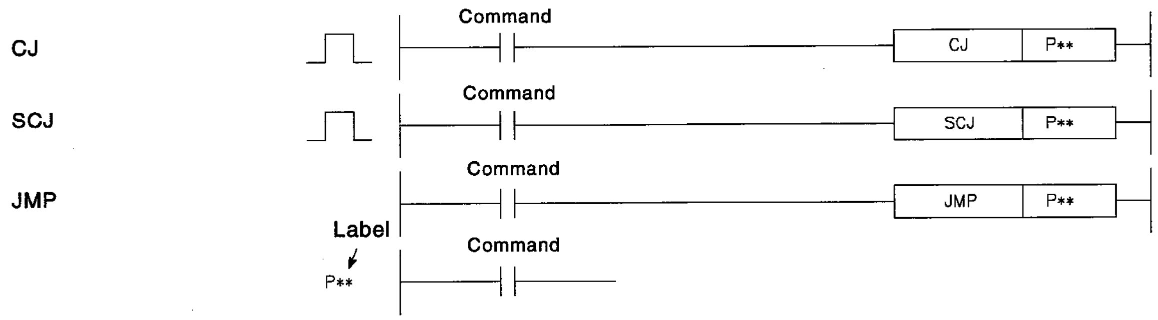


6.5 Program Branch Instruction

6.5.1 Pointer branch instructions

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				P
P	—								o

[Instruction Symbol] [Execution Condition]



Set Data

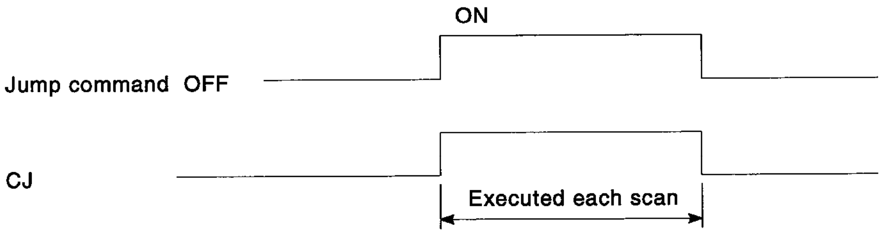
Set Data	Meaning	Data Type
P**	Pointer number of jump destination	Device name

Functions

CJ

(1) Executes program of designated pointer number within the same program file when jump command is ON.

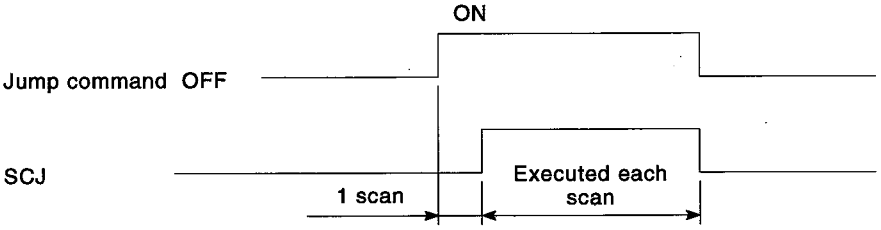
(2) Executes next step in program when jump command is OFF.



SCJ

(1) Executes program of designated pointer number within the same program file from next scan when jump command goes from OFF to ON.

(2) Executes next step in program when jump command is OFF or when it goes from ON to OFF.



JMP

(1) Unconditionally executes program of designated pointer number within the same program file.

POINTS

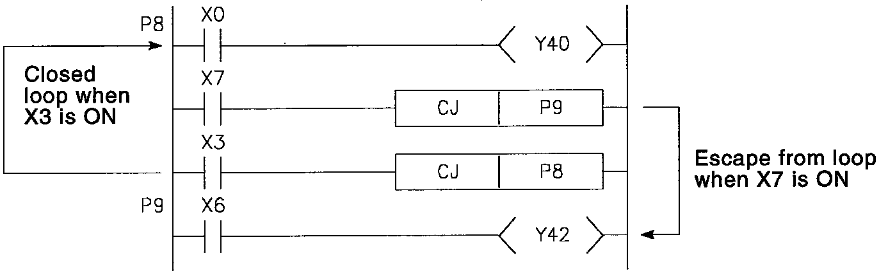
Note the following points when using the jump instruction.

(1) After the timer coil has gone ON, accurate measurements cannot be made if there is an attempt to jump the timer of a coil that has been turned ON using the CJ, SCJ or JMP instructions.

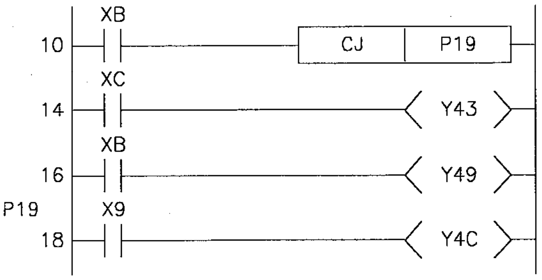
(2) Scan time is shortened if the CJ, SCJ or JMP instruction is used to force a jump to the OUT instruction.

(3) Scan time is shortened if the CJ, SCJ or JMP instruction is used to force a jump to the rear.

(4) The CJ, SCJ, and JMP instructions can be used to jump to a step prior to the step currently being executed.
However, it is necessary to consider methods to get out of the loop so that the watchdog timer does not time out in the process.

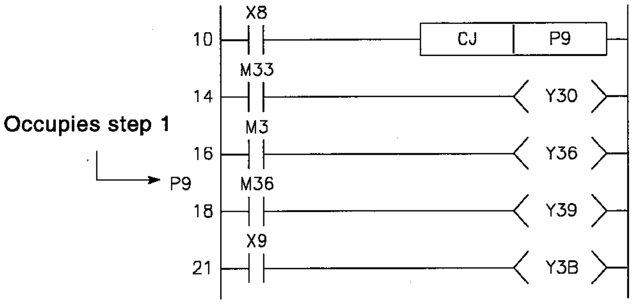


(5) The device to which a jump has been made with CJ, SCJ or JMP does not change.



Jumps to label P19 when XB goes ON.
Y43 and Y49 do not change regardless of whether XC is ON or OFF during execution of CJ instruction.

(6) The label (P*) occupies step 1.



(7) Jump instructions can be used only for pointer numbers within the same program file.

(8) If a jump is made to a pointer number inside the skip range during a skip operation, program execution will be taken up following the pointer number of the jump destination.

Operation Errors

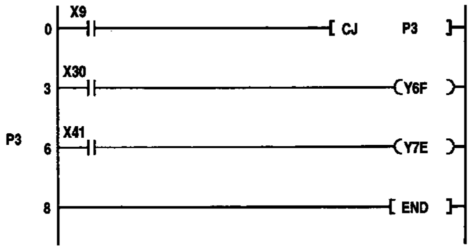
(1) In the following cases an operation is returned, the error flag (SM0) goes ON, and the error code is stored at SD0.

- The pointer number designated does not come prior to the END instruction. (Error code: 4210)
- A pointer number which is not in use as a label in the same program has been designated. (Error code: 4210)
- A common pointer has been designated (Error code: 4210)

Program Example

(1) The following program jumps to P3 when X9 goes ON.

[Ladder Mode]

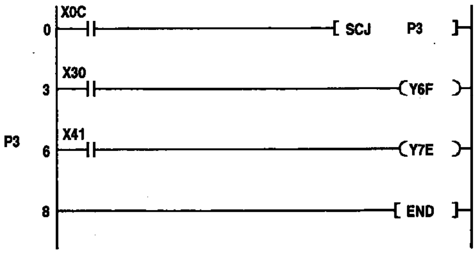


[List Mode]

Steps	Instruction	Device
0	LD	X9
1	CJ	P3
3	LD	X30
4	OUT	Y6F
5		P3
6	LD	X41
7	OUT	Y7E
8	END	

(2) The following program jumps to P3 from the next scan after XC goes ON.

[Ladder Mode]



[List Mode]

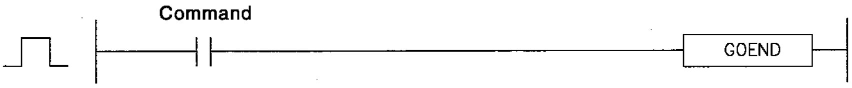
Steps	Instruction	Device
0	LD	X0C
1	SCJ	P3
3	LD	X30
4	OUT	Y6F
5		P3
6	LD	X41
7	OUT	Y7E
8	END	

6.5.2 Jump to END

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]

GOEND



Functions

(1) Jumps to FEND or END instruction in the same program file.

Operation Errors

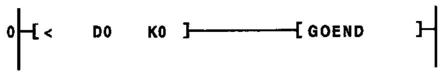
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- A GOEND instruction has been executed after the execution of a CALL, FCALL, ECALL, or EFCALL instruction, and prior to the execution of the RET instruction. (Error code: 4211)
- A GOEND instruction has been executed after the execution of a FOR instruction, and prior to the execution of the NEXT instruction. (Error code: 4200)
- A GOEND instruction has been executed during an interrupt program but prior to the execution of the IRET instruction. (Error code: 4221)
- A GOEND instruction was executed between the CHKCIR and CHKEND instruction block. (Error code: 4230)
- A GOEND instruction was executed between the IX and IXEND instruction block. (Error code: 4231)

Program Example

(1) The following program jumps to the END instruction if D0 holds a negative number.

[Ladder Mode]



[List Mode]

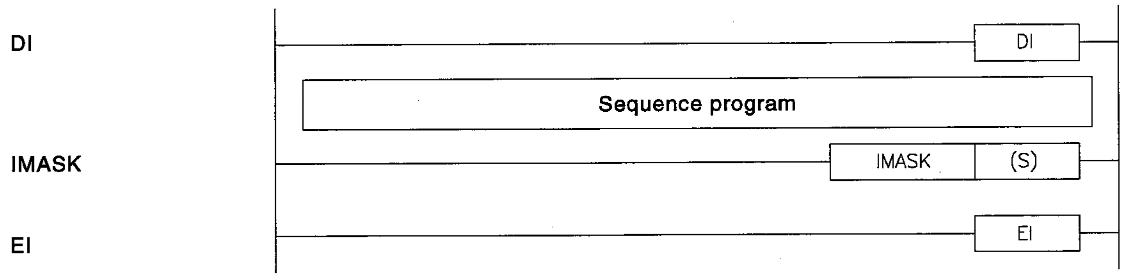
Steps	Instruction	Device
0	LD<	D0 K0
3	GOEND	

6.6 Program Execution Control Instructions

6.6.1 Interrupt disable/enable instructions, interrupt program mask

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—		

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	Interrupt mask data or initial number of device where interrupt mask data is being stored	BIN 16 bits

Functions

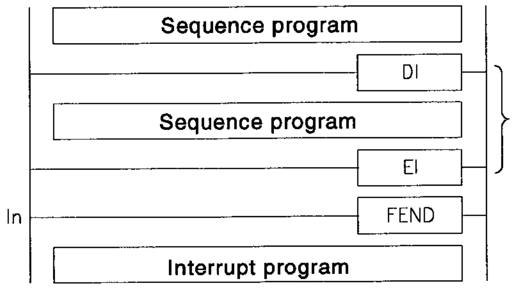
DI

(1) Disables the execution of an interrupt program until the EI instruction has been executed, even if a start cause for the interrupt program occurs.

(2) A DI state is entered when power is turned ON or when the system has been reset by use of the RUN/STOP switch.

EI

The EI instruction is used to clear the interrupt disable state resulting from the execution of the DI instruction, and to create a state in which the interrupt program designated by the interrupt pointer number certified by the IMASK instruction can be executed.



Even though an interrupt condition might be generated between the DI and EI instructions, the interrupt program will be held until the entire cycle from DI to EI has been processed.

IMASK

(1) Enables or disables the execution of the interrupt program marked by the designated interrupt pointer by use of the bit pattern in the three points from the device designated by (S).

- 1 (ON) .... Interrupt program execution enabled
- 0 (OFF) ... Interrupt program execution disabled

(2) The interrupt pointer numbers corresponding to the individual bits are as shown below:

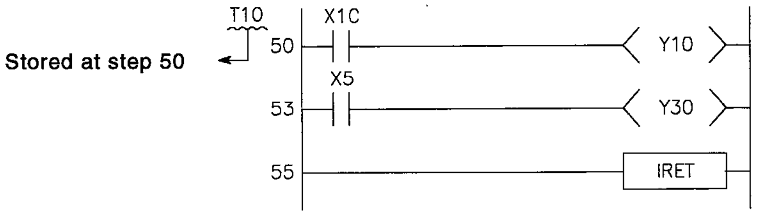
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(S)	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
(S)+1	I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
(S)+2	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

(3) When the power is turned ON, or when the CPU has been reset by using the RUN/STOP key switch, interrupt programs from I0 to I31 are in the execution enabled state, and interrupt programs from I32 to I47 are in the execution disabled state.

(4) The statuses of the (S), (S)+1, and (S)+2 devices are stored from SD715 to SD717 (the IMASK instruction mask pattern storage area).

POINTS

(1) An interrupt pointer occupies 1 step.



(2) See the QnACPU Programming Manual (Fundamentals) for information on interrupt conditions.

(3) The DI state (interrupt disabled) is active during the execution of an interrupt program.
Do not insert EI instructions in interrupt programs to attempt the execution of multiple interrupts, with interrupt programs running inside interrupt programs.

(4) If there are EI and DI instructions within a master control, these instructions will be executed regardless of the execution/non-execution status of the MC instruction.

Operation Errors

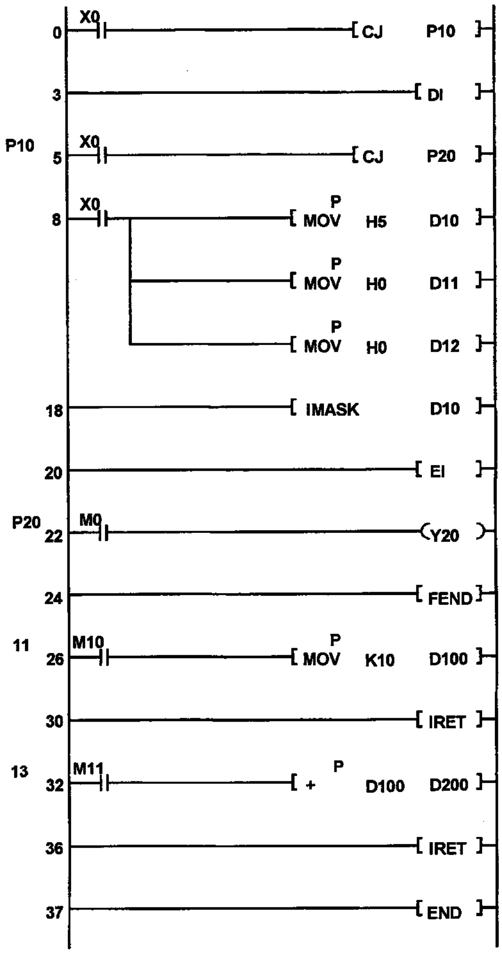
(1) There are no operation errors associated with the DI and EI instructions.

(2) There are no operation errors associated with the IMASK instruction.

Program Example

(1) The following program creates an execution enabled state for the interrupt program marked by the interrupt pointer number when X0 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	CJ	P10
3	DI	
4		P10
5	LD	X0
6	CJ	P20
8	LD	X0
9	MOVP	H5 D10
12	MOVP	H0 D11
15	MOVP	H0 D12
18	IMASK	D10
20	EI	
21		P20
22	LD	M0
23	OUT	Y20
24	FEND	
25		I1
26	LD	M10
27	MOVP	K10 D100
30	IRET	
31		I3
32	LD	M11
33	+P	D100 D200
36	IRET	
37	END	

6.6.2 Recovery from interrupt programs

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]

IRET

Functions

(1) Indicates the completion of interrupt program processing.

(2) Returns to sequence program processing following the execution of the IRET instruction.

Operation Errors

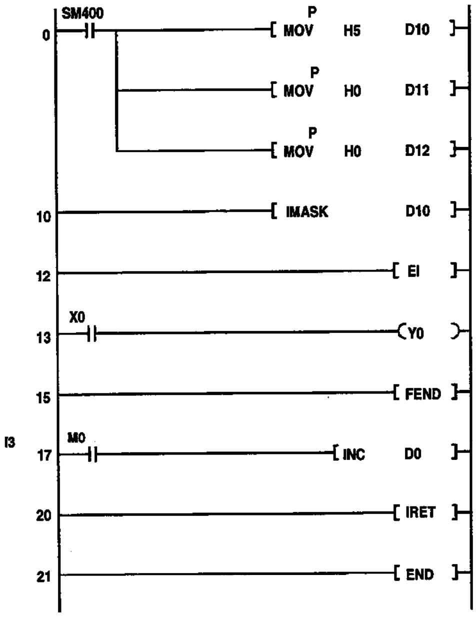
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- There is no pointer corresponding to the interrupt number. (Error code: 4220)
- The IRET instruction has been issued prior to the execution of the interrupt program.(Error code: 4223)
- An END, FEND, GOEND, or STOP instruction as been executed after the generation of an interrupt and prior to the execution of the IRET instruction.(Error code: 4221)

Program Example

(1) The following program adds 1 to D0 if M0 is ON when the number 3 interrupt is generated.

[Ladder Mode]



[List Mode]

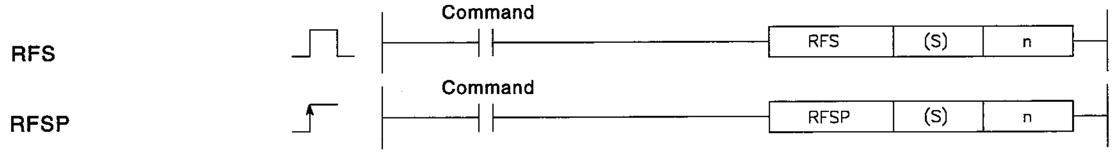
Steps	Instruction	Device
0	LD	SM400
1	MOVP	H5
		D10
4	MOVP	H0
		D11
7	MOVP	H0
		D12
10	IMASK	D10
12	EI	
13	LD	X0
14	OUT	Y0
15	FEND	
16		I3
17	LD	M0
18	INC	D0
20	IRET	
21	END	

6.7 I/O Refresh Instructions

6.7.1 I/O refresh

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o (only X, Y)	—							—
n	o	o							—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	The initial device number of the device that will conduct refresh operation	Bit
n	Number of points to be refreshed	BIN 16 bits

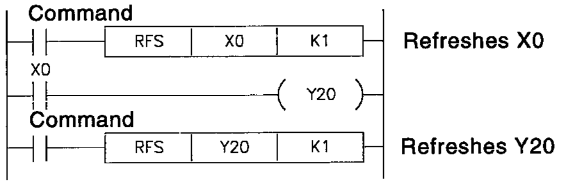
Functions

(1) Refreshes only the device being scanned during a scan, and functions to fetch input from external sources or to output data to an output module.

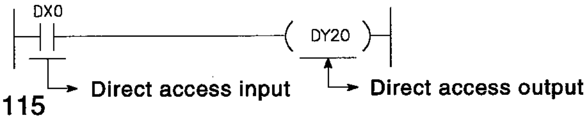
(2) Fetching of input from or sending output to an external source is conducted in batch only after the execution of an END instruction, so it is not possible to output a pulse signal to an outside source during the execution of a scan.
When a refresh operation is conducted, inputs (X) or outputs (Y) of the device numbers relevant to the program being executed are forcibly refreshed, so it is possible to output a pulse signal to an external source during a scan.

(3) Use direct access inputs (DX) or direct access outputs (DY) to refresh inputs (X) or outputs (Y) in 1-point units.

[Program based on the RFS instruction]



[Program based on DX and DY] ⇩



Operation Errors

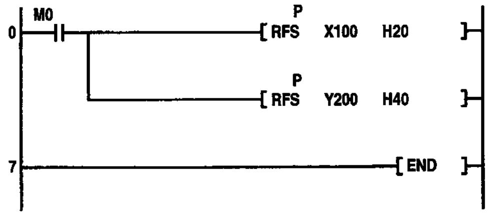
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range n points from the device designated by (S) exceeds the proximate I/O range.

Program Example

(1) The following program refreshes X100 through X11F and Y200 through Y23F when M0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	M0
1	RFSP	X100 H20
4	RFSP	Y200 H40
7	END	

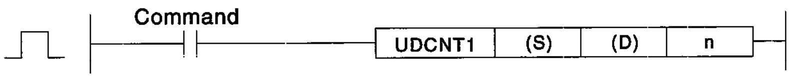
6.8 Other Convenient Instructions

6.8.1 Count 1-phase input up or down

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o (Only X)	—	—						—
(D)	—	o (Only C)	—						—
n	o	o	o						—

[Instruction Symbol] [Execution Condition]

UDCNT1



Set Data

Set Data	Meaning	Data Type
(S)	(S)+0: Input number for count input	Bit
	(S)+1: For setting count up or down • OFF : Count up (add numbers when counting) • ON : Count down (subtract numbers when counting)	
(D)	Number of counter that will perform count on UDCNT1 instruction	Word
n	Set value	BIN16

Functions

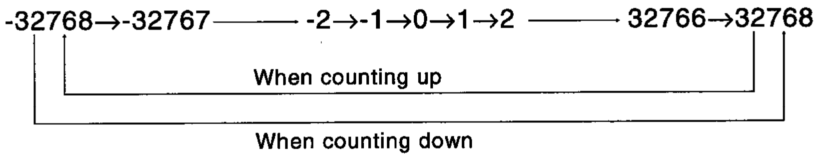
(1) When the input designated at (S) goes from OFF to ON, the present value of the counter designated at (D) will be updated.

(2) The direction of the count is determined by the ON/OFF status of the input designated by (S)+1.

- OFF: Count up (counts by adding to the present value)
- ON: Count down (counts by subtracting from the present value)

(3) Count processing is conducted as described below:

- When the count is going up, the counter contact designated at (D) goes ON when the present value becomes identical with the setting value designated by n.
  However, the present value count will continue even when the contact of the counter designated at (D) goes ON.
  (See Program Example (1))
- When the count is going down, the counter for the contact designated at (D) goes OFF when the present value reaches the set value minus 1. (See Program Example (1))
- The counter designated at (D) is a ring counter.
  If it is counting up when the present value is 32767, the present value will become -32768.
  Further, if it is counting down when the present value is -32768, the present value will become 32767.
  The count processing performed on the present value is as shown below:



(4) Count processing based on the UDCNT1 instruction starts the count when the count command goes from OFF to ON, and suspends the count when it goes from ON to OFF.
When the count command goes from OFF to ON once again, the count is restarted from the value in effect when it was suspended.

(5) The RST instruction clears the present value of the counter designated at (D) and turns the contact OFF.

POINTS

(1) Count processing for counters using the UDCNT1 instruction is conducted during QnACPU interrupts (5 ms). For this reason, only pulses with ON and OFF times of over 5 ms can be counted.

(2) The setting values cannot be changed during a count based on the UDCNT1 instruction (when the input designated at (S) is ON). To change the setting values, first turn the input designated at (S) OFF.

(3) Counters which have been designated by the UDCNT1 instruction cannot be used by other instructions. If they are used by other instructions, they will not be capable of returning an accurate count.

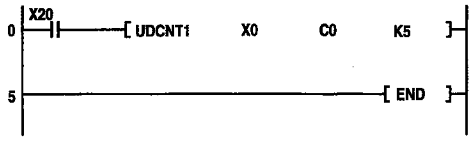
Operation Errors

(1) There are no operation errors associated with the UDCNT1 instruction.

Program Example

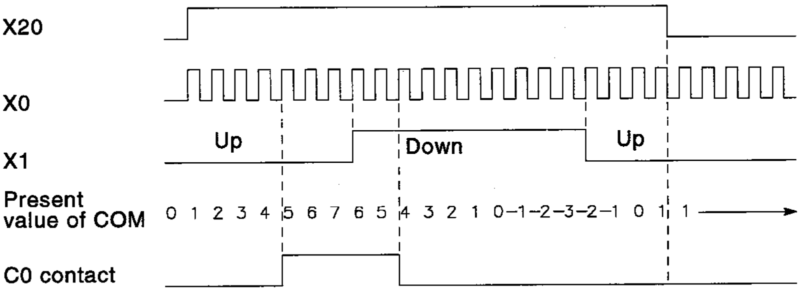
(1) This program uses C0 (up and down counter) to count the number of times X0 goes from off to ON after X20 has gone ON.

[Ladder Mode]



[List Mode]

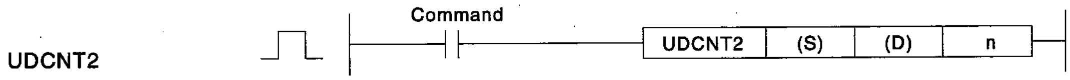
Steps	Instruction	Device
0	LD	X20
1	UDCNT1	X0 C0 K5
5	END	



6.8.2 Counter 2-phase input up or down

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o (only X)	—	—						—
(D)	—	o (only C)	—						—
n	o	o	o						—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	Input number for count input: (S)+0 (A phase pulse) Input number for count input: (S)+1 (B phase pulse)	Bit
(D)	Number of counter that will perform count on UDCNT2 instruction	Word
n	Set value	BIN16

Functions

(1) The present value of the counter designated by (D) is updated depending on the status of the input designated by (S) (A phase pulse) and the status of the input designated by (S)+1 (B phase pulse).(2)

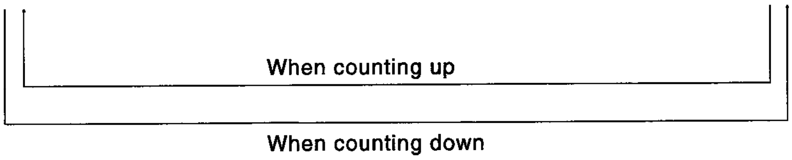
(2) Direction of the count is determined in the following manner:

- When (S) is ON, if (S)+1 goes from OFF to ON, count up operation is performed (values are added to the present value of the counter).
- When (S) is ON, if (S)+1 goes from ON to OFF, count down operation is performed (values are subtracted from the present value of the counter).
- No count operation is performed if (S) is OFF.

(3) Count processing is conducted as described below:

- When the count is going up, the counter contact designated at (D) goes ON when the present value becomes identical with the setting value designated by n.
  However, the present value count will continue even when the contact of the counter designated at (D) goes ON.
  (See Program Example (1))
- When the count is going down, the counter for the contact designated at (D) goes OFF when the present value reaches the setting value minus 1. (See Program Example (1))
- The counter designated at (D) is a ring counter.
  If it is counting up when the present value is 32767, the present value will become -32768.
  Further, if it is counting down when the present value is -32768, the present value will become 32767.
  The count processing performed on the present value is as shown below:

-32768→-32767 —— -2→-1→0→1→2 ———— 32766→32768



(4) Count processing conducted according to the UDCNT2 instruction begins when the count command goes from OFF to ON, and is suspended when it goes from ON to OFF.
When the count command goes from OFF to ON once again, the count is restarted from the value in effect when it was suspended.

(5) The RST instruction clears the present value of the counter designated at (D) and turns the contact OFF.

POINTS

(1) Count processing conducted for counters using the UDCNT2 instruction is conducted via QnACPU interrupts (5 ms).
For this reason, only pulses with ON and OFF times of over 5 ms can be counted.

(2) The set value cannot be changed while a count operation performed according to the UDCNT2 instruction is being executed (while the input designated by (S) is ON).
To change the set value, first turn the input designated at (S) OFF.

(3) Counters designated by the UDCNT2 instruction cannot be used by any other instruction.
If they are used by other instructions, they will not be capable of returning an accurate count.

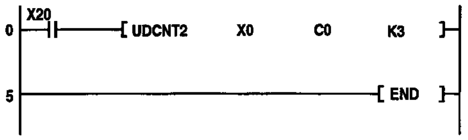
Operation Errors

(1) There are no operation errors associated with the UDCNT2 instruction.

Program Example

(1) The following program performs a count operation as instructed by C0 (count up or down) on the status of X0 and X1 after X20 has gone ON.

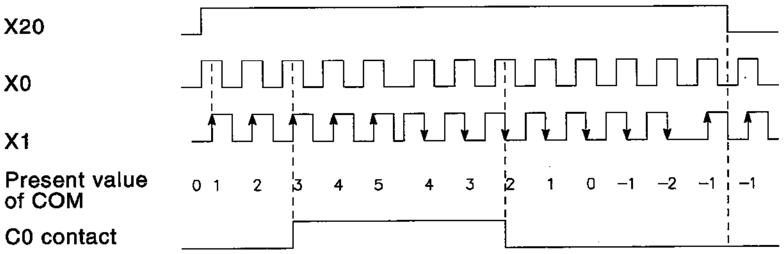
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	M20
1	UDCNT2	X0
		C0
		K3
5	END	

[Operation]



6.8.3 Teaching timer

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	—	○		—					—
n	—	○		○					—

[Instruction Symbol] [Execution Condition]

TTMR

Set data

Set Data	Meaning	Data Type
(D)	(D)+0: Storage device for measurement value	BIN 16 bits
	(D)+1: For QnACPU system use	
n	Measurement value multiplier	

Functions

(1) The time that the measurement command is on is measured in units of seconds, then multiplied by the multiplier designated by n and the product is stored at the device designated by (D).

(2) When the measurement command goes from OFF to ON, the device designated by (D) or (D)+1 is cleared.

(3) The multipliers that can be designated by n are as shown below:

(S)	Multiplier
0	1
1	10
2	100

POINTS

(1) Time measurements are conducted when the TTMR instruction is executed.
Using the JMP or similar instruction to jump the TTMR instruction will make it impossible to get an accurate measurement.

(2) Do not change the multiplier designated by n while the TTMR instruction is being executed.
Changing this multiplier will result in an inaccurate value being returned.

(3) The TTMR instruction can also be used in low speed type programs.

(4) The device designated by (D)+1 is used by the QnACPU system, so users should not change its value.
If users do change this value, the value stored in the device designated by (D) will no longer be accurate.

Operation Errors

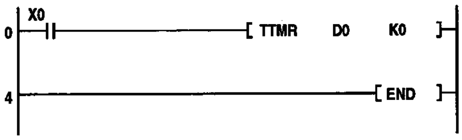
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The multiplier designated by n is outside the range of from 0 to 2. (Error code: 4100)

Program example

(1) The following program stores the amount of time that X0 is ON at D0.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	TTMR	D0
		K0
4	END	

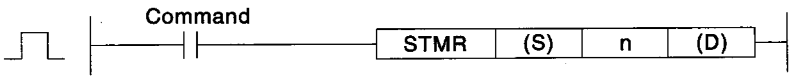
6.8.4 Special function timer

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	—	Δ *	—	—					—
n	o	—	—	—					—
(D)	—	o	o	o					—

*: Can be used only by timer (T) data

[Instruction Symbol] [Execution Condition]

STMR



Set Data

Set Data	Meaning	Data Type
(S)	Time number	Word
n	Set value	BIN 16 bits
(D)	(D)+0: Off delay timer output (D)+1: One shot timer output after OFF (D)+2: One shot timer output after ON (D)+3: ON delay timer output	Bit

Functions

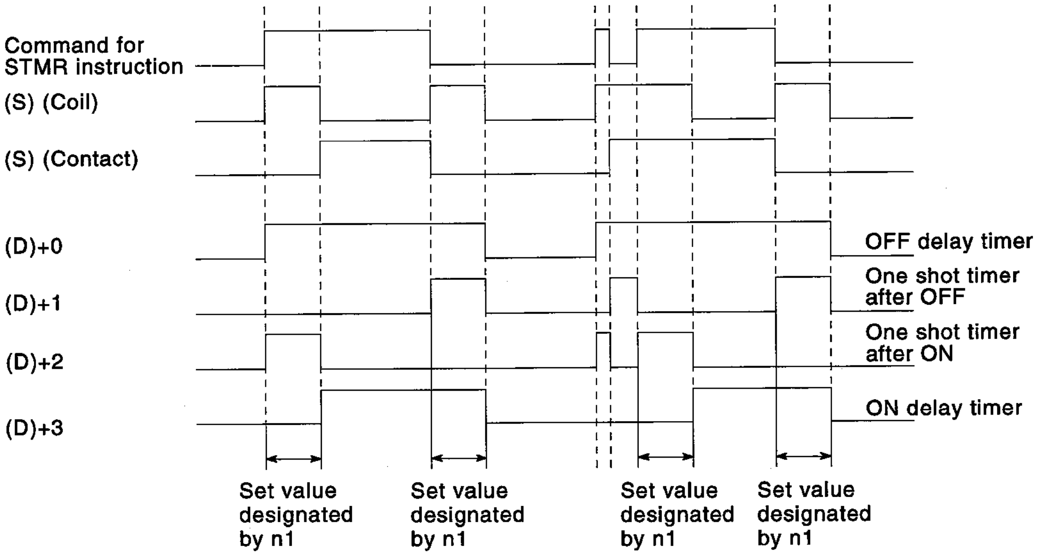
(1) The STMR instruction uses the 4 points from the device designated by (D) to perform four types of timer output.

- OFF delay timer output ((D)+0)
  Goes ON at the leading edge of the command for the STMR instruction, and, after the trailing edge of the command, goes OFF when the amount of time designated by n has passed.
- One shot timer output after OFF ((D)+1)
  Goes ON at the trailing edge of the command for the STMR instruction, and goes OFF when the amount of time designated by n has passed.
- One shot timer output after ON ((D)+2)
  Goes ON at the leading edge of the command for the STMR instruction, and goes OFF either when the amount of time designated by n has passed, or when the command for the STMR instruction goes OFF.
- ON delay timer output ((D)+3)
  Goes ON at the trailing edge of the timer coil, and after the trailing edge of the command for the SRMR instruction, goes OFF when the amount of time designated by n has passed.

(2) The timer coil designated by (S) goes ON at the leading edge of the command for the STMR instruction, and begins the measurement of the present value.

- The timer coil measures to the point where the value reaches the set value designated by n, then enters a time up state and goes OFF.
- If the command for the SRMR instruction goes OFF before the timer coil reaches the time up state, it will remain ON.
  Timer measurement is suspended at this time.
  When the STRM instruction command goes ON once again, the present value will be cleared to 0 and measurement will begin once again.

(3) The timer contact goes ON at the leading edge of the command for the STMR instruction, and after the trailing edge is reached, the timer coil goes OFF at the trailing edge of the STMR instruction command. The timer contact is used by the QnACPU system, and cannot be used by the user.



(4) Measurement of the present value of the timer designated by the STMR instruction is conducted during the execution of the STMR instruction.
If the STMR instruction is jumped with the JMP or similar instruction, it will not be possible to get accurate measurement.

(5) Measurement unit for the timer designated by (D) is identical to the low speed timer.

(6) A value between 1 and 32767 can be set for n.

(7) The timer designated by (S) cannot be used by the OUT instruction. If the STMR instruction and the OUT instruction use the same timer number, accurate operation will not be conducted.

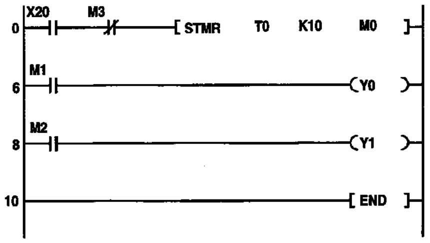
Operation Errors

(1) There are no errors associated with the STMR instruction.

Program Example

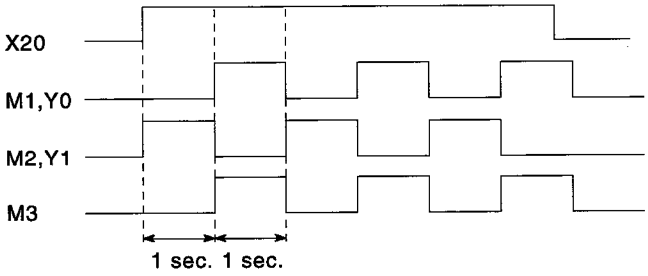
(1) The following program turns Y0 and Y1 ON and OFF once each second (flicker) when X20 is ON. (Uses the 100ms timer)

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	ANI	M3
2	STMR	T0 K10 M0
6	LD	M1
7	OUT	Y0
8	LD	M2
9	OUT	Y1
10	END	

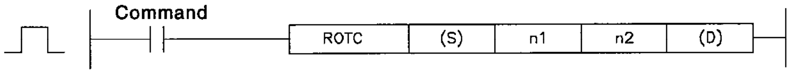


6.8.5 Rotary table near path rotation control

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—					—
n1	o	o		o					—
n2	o	o		o					—
(D)	o	—		—					—

[Instruction Symbol] [Execution Condition]

ROTC



Set Data

Set Data	Meaning	Data Type
(S)	(S)+0: Measures table rpm (for system use)	BIN 16 bits
	(S)+1: Call station number	
	(S)+2: Call item number	
n1	Number of divisions on table (from 2 to 32767)	
n2	Number of low-speed sections (value from 0 to n1)	
(D)	(D)+0: A phase input signal	Bit
	(D)+1: B phase input signal	
	(D)+2: 0 point detection input signal	
	(D)+3: High speed forward rotation output signal (for system use)	
	(D)+4: Low speed forward rotation output signal (for system use)	
	(D)+5: Stop output signal (for system use)	
	(D)+6: High speed reverse rotation output signal (for system use)	
	(D)+7: Low speed reverse rotation output signal (for system use)	

Functions

(1) This control functions to enable near path rotation of the rotary table to the position of the station number designated by (S)+1 in order to remove or deposit an item whose number has been designated by (S)+2 on a rotary table with equal divisions of the value designated by n1.

(2) The item number and station number are controlled as items allocated by counterclockwise rotation.

(3) The system uses (S)+0 as a counter to instruct it as to what item is at which number counting from station number 0.
Do not rewrite the sequence program data.
Accurate controls will not be possible in cases where users have rewritten the data.

(4) The value of n2 should be less than the number of table divisions that were designated by n1.

(5) (D)+0 and (D)+1 are A and B phase input signals that are used to detect whether the direction of the rotary table rotation is forward or reverse.
The direction of rotation is judged by whether the B phase pulse is at its leading or trailing edge when the A phase pulse is ON:

- When the B phase is at the leading edge: Forward rotation (clockwise rotation)
- When the B phase is at the trailing edge: Reverse rotation (counterclockwise rotation)

(6) (D)+2 is the 0 point detection output signal that goes ON when item number 0 has arrived at the No. 0 station.
When the device designated by (D)+2 goes ON while the ROTC instruction is being executed, (S)+0 is cleared.
It is best to perform this clear operation first, then to begin near path rotation with the ROTC instruction.

(7) The data from (D)+3 to (D)+7 consists of output signals needed to control the table's operation.
The output signal of one of the devices from (D)+3 to (D)+7 will go ON in response to the execution results of the ROTC instruction.

(8) If operation results immediately prior to the ROTC instruction are OFF, all signals from (D)+3 to (D)+7 will be OFF without near path rotation controls having been performed.

(9) The ROTC instruction can be used only one time in all programs where it is executed.
Attempts to use it more than one time will result in inaccurate operations.

Operation Errors

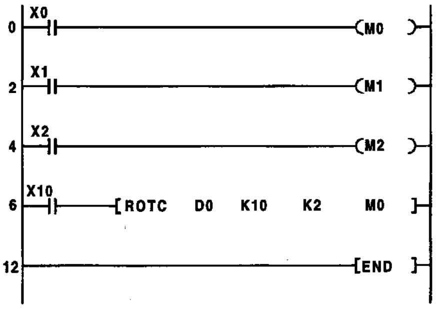
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The value of (S)+0 to +2, or the value of n2 is greater than n1.
(Error code: 4100)

Program Example

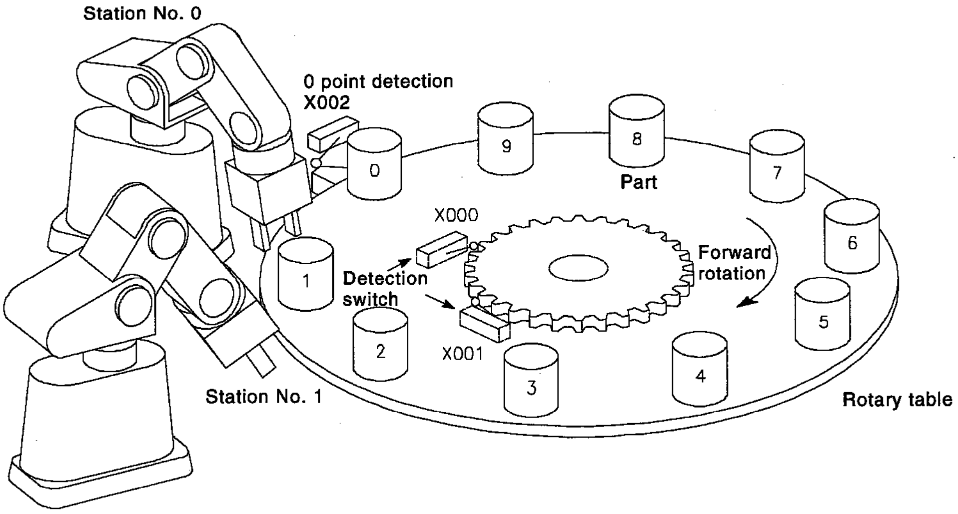
(1) The following program deposits the item at section D2 on a 10-division rotary table at the station at section D1, and the two sections ahead and behind this determine the rotation direction and control speed of the motor when the table is being rotated at low speed.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	OUT	M0
2	LD	X1
3	OUT	M1
4	LD	X2
5	OUT	M2
6	LD	X10
7	ROTC	D0
		K10
		K2
		M0
12	END	

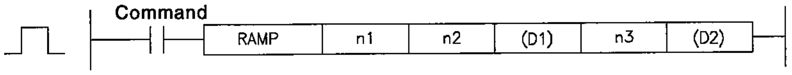


6.8.6 Ramp signal

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n1	o	o						o	—
n2	o	o						o	—
(D1)	o	o						—	—
n3	o	o						o	—
(D2)	o	—						—	—

[Instruction Symbol] [Execution Condition]

RAMP



Set Data

Set Data	Meaning	Data Type
n1	Initial value	BIN 16 bits
n2	Final value	
(D1)	D1+0: Present value	
	D1+1: Number of times executed (for system use)	
n3	Number of times moved	
(D2)	D2+0: Completion device	Bit
	D2+1: Selected bit where data is to be saved at completion.	

Functions

(1) Gradually changes the contents of the device designated by D1+0 from the value designated by n1 to the value designated by n2.

(2) For n3, designate the number of scans required to move data from n1 to n2.

(3) The system uses D1+1 to store the number of times the instruction has been executed.

(4) When the move is completed to the final value, the completion device designated by D2+0 will go ON.
The ON/OFF status of the completion device and the contents of D1+0 are determined by the ON/OFF status of the device designated by D2+1.

- When D2+1 is OFF, D2+0 will go OFF at the next scan, and the RAMP instruction will begin a new move operation from the value currently at D1+0.
- When D2+1 is ON, D2+0 will remain ON, and the contents of D1+0 will not change.
  When the command goes OFF, the D2+0 device will go OFF.

(5) When the command is turned OFF during the execution of this instruction, the contents of D1+0 will not change following this.
When the command goes ON again, the RAMP instruction will begin a new move from the present value at D1+0.

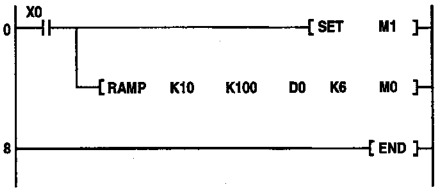
Operation Errors

(1) There are no operation errors associated with the RAMP instruction.

Program Example

(1) The following program changes the contents of D0 from 10 to 100 in a total of 6 scans, and saves the contents of D0 when the move has been completed.

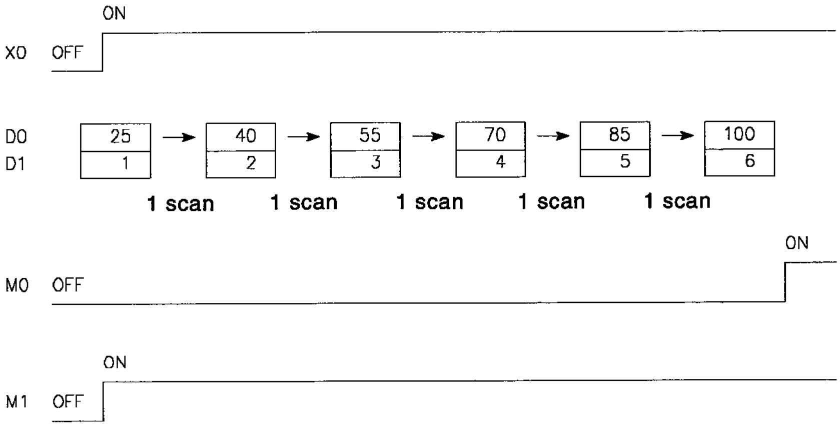
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	SET	M1
2	RAMP	K10
		K100
		D0
		K6
		M0
8	END	

[Operation]



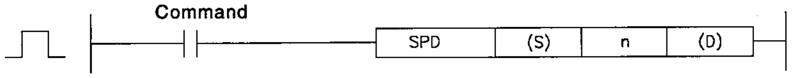
6.8.7 Pulse density measurement

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	Δ *	—		o					—
n	o	o		—					—
(D)	—	o		o					—

*: Only input (X) can be used

[Instruction Symbol] [Execution Condition]

SPD

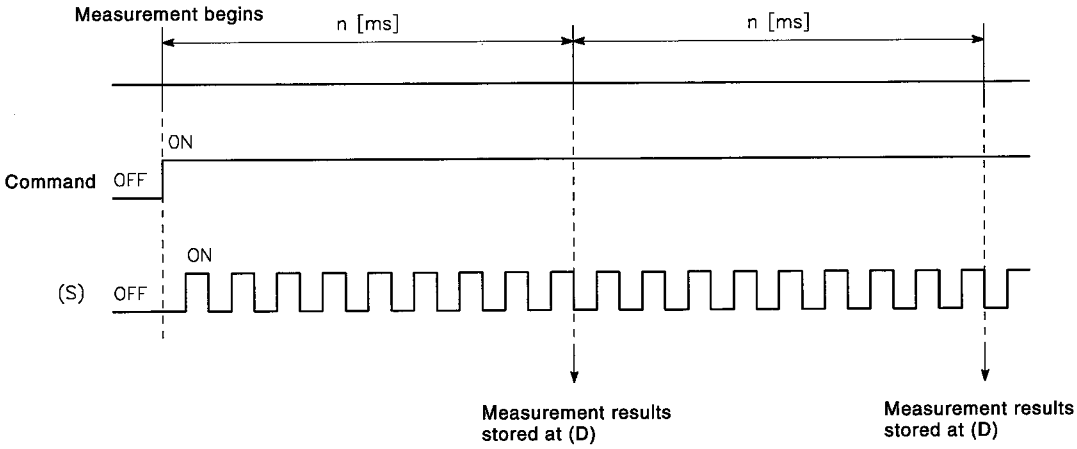


Set Data

Set Data	Meaning	Data Type
(S)	Pulse input	Bit
n	Measurement time (unit: ms)	BIN 16 bits
(D)	Initial number of device which stores measurement results	

Functions

(1) Input from the device designated by (S) is counted for just the amount of time designated by n1, and results of the count are stored in the device designated by D1.



(2) When measurement directed by the SPD instruction has been completed, measurement is done again from 0.
To suspend measurement directed by the SPD instruction, turn the command OFF.

(3) The SPD instruction registers the data from the designated device in the QnACPU work area, and the actual count operation is conducted during a 5 ms system interrupt.
For this reason, there are limits on the number of times the instruction can be used.
Anything exceeding these use limits will not be processed.

Operation Errors

(1) There are no operation errors associated with the SPD instruction.

Program Example

(1) The following program measures the pulses input to X0 for a period of 500 ms when X10 goes ON, and stores the result at D0.

[Ladder Mode]

[List Mode]

Steps	Instruction	Device
0	LD	X10
1	SPD	X0 K500 D0
5	END	

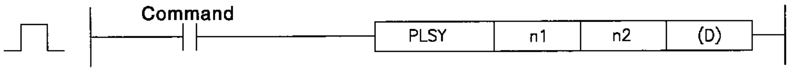
6.8.8 Fixed cycle pulse output

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
n1	o	o							—
n2	o	o							—
(D)	Δ *	—							—

*: Only output (Y) can be used

[Instruction Symbol] [Execution Condition]

PLSY



Set Data

Set Data	Meaning	Data Type
n1	Number of device where frequency is set	BIN 16 bits
n2	Device No. of device that sets the number of outputs	
(D)	Number of device where pulse output is conducted	Bit

Functions

(1) Outputs a pulse at a frequency designated by n1 the number of times designated by n2, to the output module with the output signal (Y) designated by D1.

(2) Frequencies between 1 Hz and 100 Hz can be designated by n1.
If n1 has been set at 0, a continuous PLSY will be output.

(3) The number of outputs that can be designated by n2 is between 1 and 32767.

(4) Only an output number corresponding to the output module can designated for pulse output at (D).

(5) Pulse output commences with the command leading edge of the PLSY instruction.
Do not turn the command of the PLSY instruction OFF during pulse output.
Pulse output is suspended when the PLSY instruction command goes OFF.

(6) The PLSY instruction registers the designated device data in the QnACPU work area, and the actual output operation is processed during 5 ms system interrupts.
For this reason, the PLSY instruction can be used only once in the entire program executed by the QnACPU.

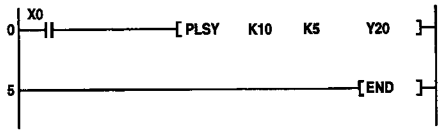
Operation Errors

(1) There are no operation errors associated with the PLSY instruction.

Program Example

(1) The following program outputs a 10 Hz pulse 5 times to Y20 when X0 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	PLSY	K10
		K5
		Y20
5	END	

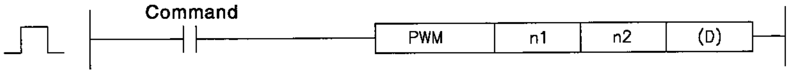
6.8.9 Pulse width modulation

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
n1	o	o							—
n2	o	o							—
(D)	Δ *	—							—

*: Only output (Y) can be used

[Instruction Symbol] [Execution Condition]

PWM

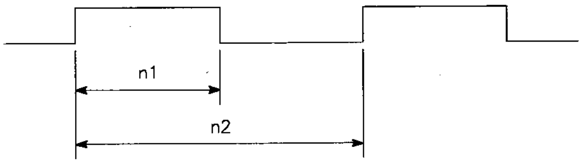


Set Data

Set Data	Meaning	Data Type
n1	Number of device where ON time is set	BIN 16 bits
n2	Number of device where cycle is set	
(D)	Number of device which will perform pulse output	Bit

Functions

(1) Outputs the pulse of the cycle set by n2, for the amount of time ON designated by n1, to the output module designated by (D).



(2) The ON time set by n2 can be between 1 ms and 32767 ms.

(3) The pulse width of n1 can be set at between 1 ms and 32767 ms. However, the value set by n1 must be lower than the value designated by n2.

(4) The PWM instruction registers the data from the designated device in the QnACPU work area, and the actual output operation is conducted during a 5 ms system interrupt.
For this reason, the PWM instruction can be used only once by the entire program executed by the QnACPU.

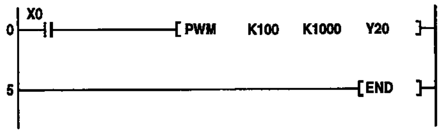
Operation Errors

(1) There are no operation errors associated with the PWM instruction.

Program Example

(1) The following program outputs a 100 ms pulse once each second to Y20 when X0 is ON.

[Ladder Mode]



[List Mode]

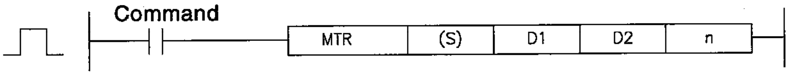
Steps	Instruction	Device
0	LD	X0
1	PWM	K100 K1000 Y20
5	END	

6.8.10 Matrix input

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	o	—							—
D1	o	—							—
D2	o	—							—
n	o	o							—

[Instruction Symbol] [Execution Condition]

MTR



Set Data

Set Data	Meaning	Data Type
(S)	Initial input device	Bit
D1	Initial output device	
D2	Initial number of device that will store matrix input data	
n	Number of input rows	BIN 16 bits

Functions

(1) Successively reads the input from 16 points starting from the input number designated by (S), multiplied by n-rows, then stores the data fetched in this operation from the device designated by D2 onward.

(2) One row (16 points) can be fetched in 1 scan.

(3) Fetching from the first to the nth row is progressively repeated.

(4) The first through the 16th points store the first row of data and the next 16 points store the second row of data at the devices following the device designated by D2.
For this reason, the space of 16xn-points from the device designated by D2 are occupied by the MTR instruction.

(5) D1 is the output needed to select the row which will be fetched, and the system automatically turns it ON and OFF.
It uses the n-points from the device designated by D1.

(6) Only device numbers divisible by 16 can be designated for (S), D1 and D2.

(7) The value for n can be designated between 2 and 8.

Operation Errors

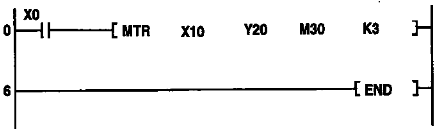
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The device number designated by (S), D1, or D2 is not divisible by 16. (Error No.: 4101)
- The device designated by (S) is outside the actual input range. (Error No.: 4101)
- The device designated by D1 is outside the actual output range. (Error No.: 4101)
- The space 16 x n-points following the device designated by D2 exceeds the relevant device range. (Error No.: 4101)
- The value for n2 is not between 2 and 8. (Error No. 4100)

Program Example

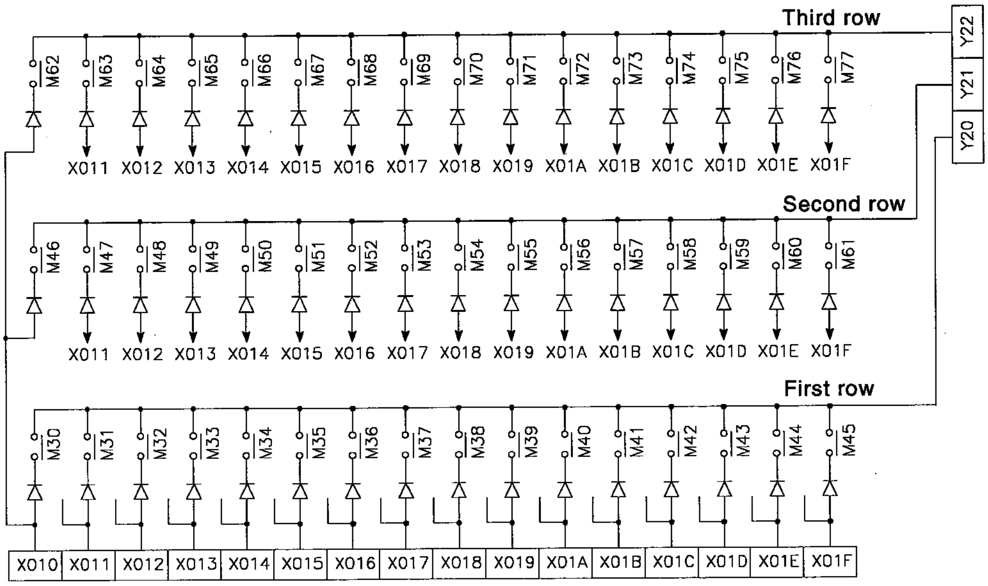
(1) The following program fetches 16 points x 3 rows starting from X10 when X0 is ON, and stores it from M30 onward.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MTR	X10 Y20 M30 K3
6	END	



POINTS

(1) Note that the MTR instruction directly operates on actual input and cutput.

(2) An MTR instruction execution interval must be longer than the total of response time of input and output modules.
If the set interval is shorter than the value indicated above, an input cannot be read correctly.
If the scan time in a sequence program is short, select the constant scan and set the scan time longer than the total of response time.

7. APPLICATION INSTRUCTIONS

The following application instructions are available:

Instruction	Meaning	Reference Section
Logical operation instructions	Logical operations such as logical sum, logical product, etc.	Section 7.1
Rotation instruction	Rotation of designated data	Section 7.2
Shift instruction	Shift of designated data	Section 7.3
Bit processing instructions	Sets and resets bit data; bit extraction	Section 7.4
Data processing instructions	Data processing including data searches, decoding, encoding, and the like	Section 7.5
Structured program instructions	Repeated operation, subroutine program calls, index qualification in ladder units	Section 7.6
Table operation instruction	Read/Write of FIFO table	Section 7.7
Buffer memory access instruction	Read/Write from buffer memory of special functions modules	Section 7.8
Display instructions	Output character codes to external devices, displays on display units	Section 7.9
Debugging and failure diagnoses instructions	Check, status latch, sampling trace, program trace	Section 7.10
Character string processing instructions	Character string (ASCII code data) processing	Section 7.11
Special function instructions	BCD type real number processing; floating decimal point type real number processing	Section 7.12
Data control instructions	Output value controls based on input data range checks	Section 7.13
File register switching instructions	Sets file registers; switches block numbers	Section 7.14
Clock instructions	Read/write clock data	Section 7.15
Peripheral device instructions	Display of peripheral device messages; key input	Section 7.16
Program instructions	Instructions to switch program execution conditions	Section 7.17
Other instructions	Instructions that do not fit in the above categories, such as watchdog timer reset instructions and timing clock instructions	Section 7.18

7.1 Logical Operation Instructions

(1) Logical operation instructions include logical sum, logical product, and the like, which are processed in 1-bit units as discussed below.

Category	Processing Details	Formula for Operation	Example		
			A	B	Y
Logical product (AND)	Becomes 1 only when both input A and input B are 1; otherwise, is 0	$Y=A \cdot B$	0	0	0
			0	1	0
			1	0	0
			1	1	1
Logical sum (OR)	Becomes 0 only when both input A and input B are 0; otherwise, is 1	$Y=A+B$	0	0	0
			0	1	1
			1	0	1
			1	1	1
Exclusive OR (XOR)	Becomes 0 if input A and input B are equal; otherwise, is 1	$Y=\overline{A} \cdot B+A \cdot \overline{B}$	0	0	0
			0	1	1
			1	0	1
			1	1	0
NON exclusive logical sum (XNR)	Becomes 1 if input A and input B are equal; otherwise, is 0	$Y=(\overline{A}+B)(A+\overline{B})$	0	0	1
			0	1	0
			1	0	0
			1	1	1

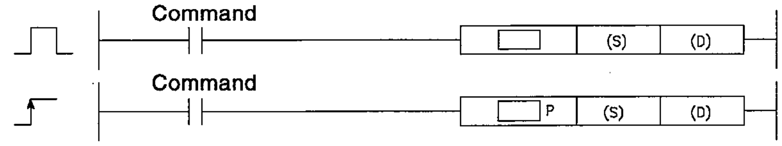
7.1.1 Logical products with 16-bit and 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]   □ indicate WAND/DAND

WAND, DAND

WANDP, DANDP



Set Data

Set Data	Meaning	Data Type
(S)	Data from which logical product will be determined, or initial number of devices storing such data	BIN 16/32 bits
(D)		

WAND

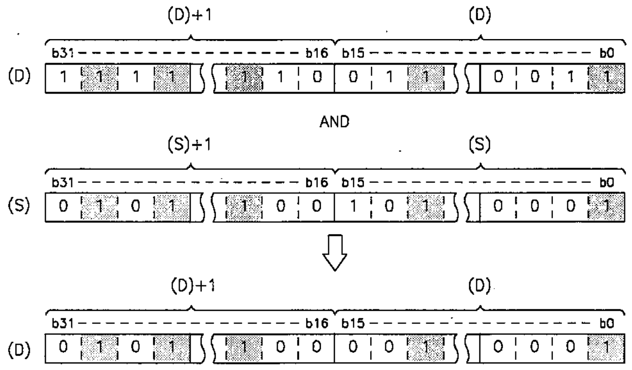
(1) A logical product operation is conducted for each bit of the 16-bit data of the device designated at (D) and the 16-bit data of the device designated at (S), and the results are stored in the device designated at (D).

	b15							b8	b7							b0
(D)	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
								AND								
(S)	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								⇩								
(D)	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0

(2) In the case of bit devices, digits above the number designated are processed as 0 in the operation.

DAND

(1) Conducts a logical product operation on each bit of the 32-bit data for the device designated by (D) and the 32-bit data for the device designated by (S), and stores the results at the device designated by (D).



(2) In the case of bit devices, digits above the number designated are processed as 0 in the operation. [See Program Example (2)]

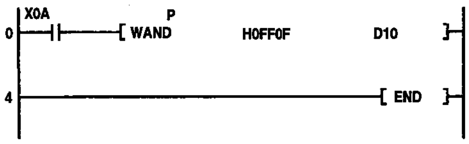
Operation Errors

(1) There are no operation errors associated with the WAND(P) or DAND(P) instructions.

Program Example

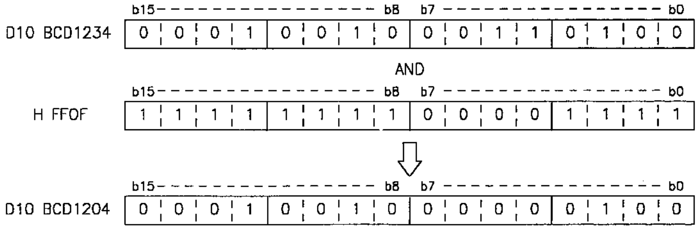
(1) The following program masks the digit in the 10s place of the 4-digit BCD value at D10 (second digit from the end) to 0 when XA is ON.

[Ladder Mode]



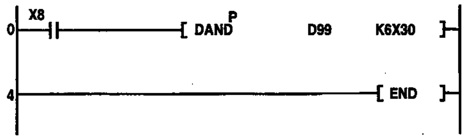
[List Mode]

Steps	Instruction	Device
0	LD	X0A
1	WANDP	H0FF0F D10
4	END	



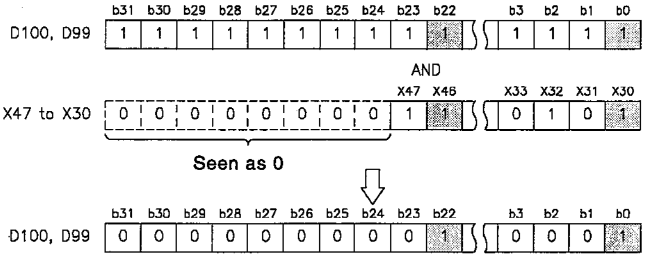
(2) The following program performs a logical product operation on the data at D99 and D100, and the 24-bit data between X30 and X47 when X8 is ON, and stores the results at D99 and D100.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X8
1	DANDP	D99 K6X30
4	END	



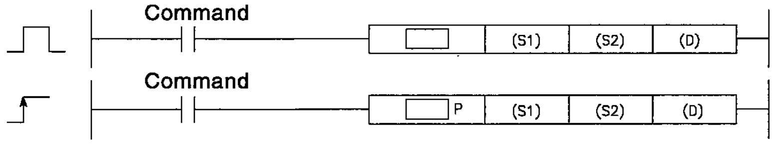
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
(S1)	o							o	—
(S2)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]

□ indicates WAND/DAND

WAND, DAND

WANDP, DANDP

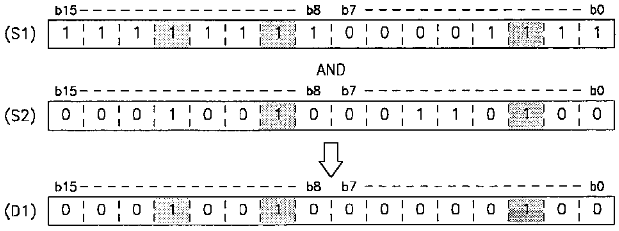


Set Data

Set Data	Meaning	Data Type
(S1) (S2)	Data from which logical product will be determined, or initial number of devices storing such data	BIN 16/32 bits
(D)	Initial number of devices where logical product operation results will be stored	

WAND

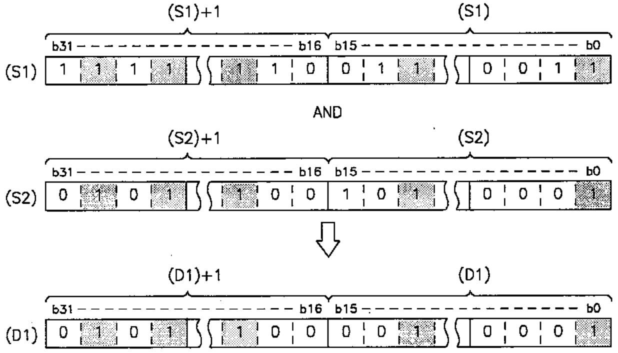
(1) Conducts a logical product operation on each bit of the 16-bit data of the device designated by (S1) and the 16-bit data of the device designated by (S2), and stores the results at the device designated by (D1).



(2) In the case of bit devices, digits above the number designated are processed as 0 in the operation. [See Program Examples (1) and (2)]

DAND

(1) Conducts a logical product operation on each bit of the 32-bit data of the device designated by (S1) and the 32-bit data of the device designated by (S2), and stores the results at the device designated by (D1).



(2) In the case of bit devices, digits above the number designated are processed as 0 in the operation. [See Program Example (3)]

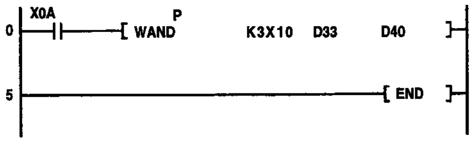
Operation Errors

(1) There are no operation errors associated with the WAND(P) or DAND(P) instructions.

Program Example

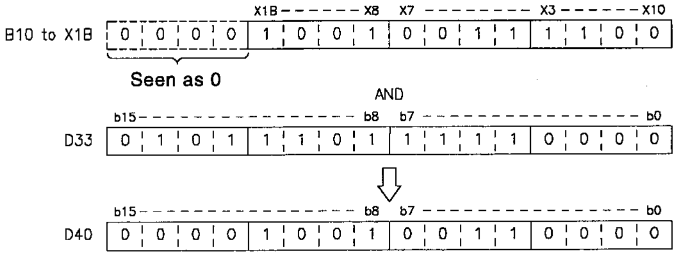
(1) The following program performs a logical product operation on the data from X10 through X1B and the data at D33 when XA is ON, and stores the results at D40.

[Ladder Mode]



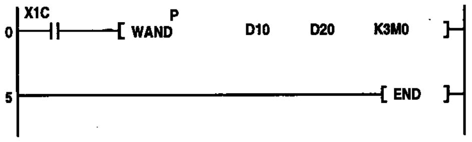
[List Mode]

Steps	Instruction	Device
0	LD	X0A
1	WANDP	K3X10 D33 D40
5	END	



(2) The following program performs a logical product operation on the data at D10 and at D20 when X1C is ON, and stores the results from M0 through M11.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	WANDP	D10 D20 K3M0
5	END	

	b15 – – – – b12	b11 – – – – b8	b7 – – – – b4	b3 – – – – b0
D10	0 0 0 0	1 0 1 0	1 0 1 0	1 0 1 0

AND

	b15 – – – – b12	b11 – – – – b8	b7 – – – – b4	b3 – – – – b0
D20	0 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1

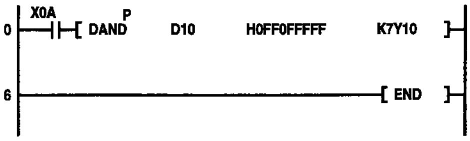
⇩

M15 – – – – – – M12	M11 – – – – – – M8	M7 – – – – – – M4	M3 – – – – – – M0
1 1 1 1	1 0 1 0	1 0 1 0	1 0 1 0

Does not change

(3) The following program masks the digit in the hundred-thousands place of the 8-digit BCD value at D10 and D11 (sixth digit from the end) to 0 when XA is ON, and outputs the results to from Y10 to Y2B.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0A
1	DANDP	D10 H0FF0FFFFF K7Y10
6	END	

b31 – – – – – – – – – – – – – – – – b16 b15 – – – – – – – – – – – – – – – – b0

D10, D11(BCD12345678)	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0

AND

b31 – – – – – – – – – – – – – – – – b16 b15 – – – – – – – – – – – – – – – – b0

H FFOFFFF	1 1 1 1	1 1 1 1	0 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1

⇩

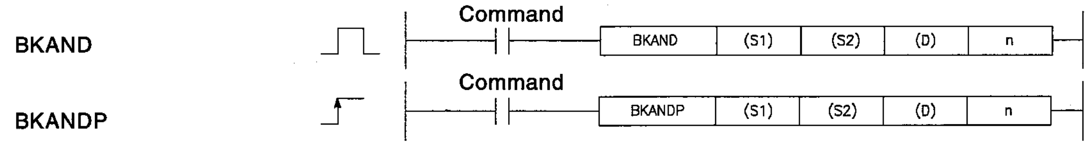
	Y2F – – Y2C	Y2B – – Y28	Y27 – – Y24	Y23 – – Y20	Y1F – – Y1C	Y1B – – Y18	Y17 – – Y14	Y13 – – Y10
Y2B to Y10	1 1 1 0	0 0 1 0	0 0 0 0	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0

Does not change

7.1.2 Block logical products

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—				—	—
(S2)	—	o		—				o	—
(D)	—	o		—				—	—
n	o	o		o				o	—

[Instruction Symbol] [Execution Condition]

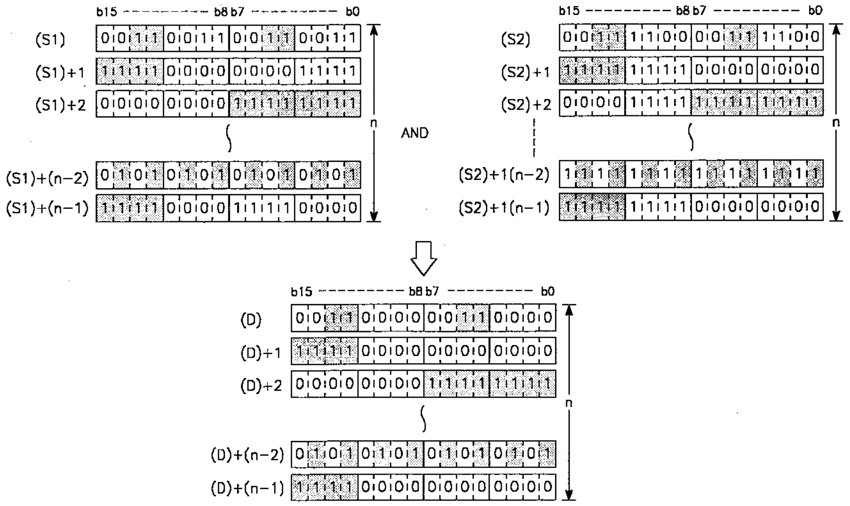


Set Data

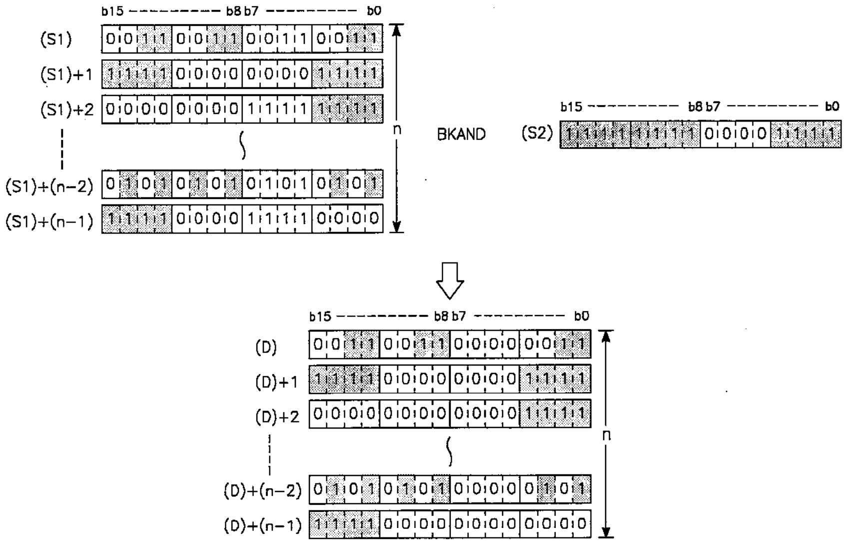
Set Data	Meaning	Data Type
(S1)	First number of the devices where the data subjected to the logical product operation is stored	BIN 16 bits
(S2)	First number of the logical operation data or the devices where the logical operation data is stored	
(D)	First number of the devices where the operation results are stored	
n	Number of operation data bits	

Functions

(1) Performs a logical product operation on the data located in the n-points from the device designated by (S1), and the data located in the n-points from the device designated by (S2), and stores the results at the device designated by (D).



(2) The constant designated by (S2) can be between -32768 and 32767 (BIN 16 bits).



Operation Errors

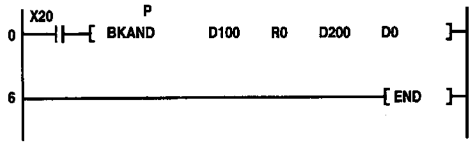
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The n-bit range from the (S1), (S2), or (D) device exceeds the range of that device. (Error code: 4101)

Program Example

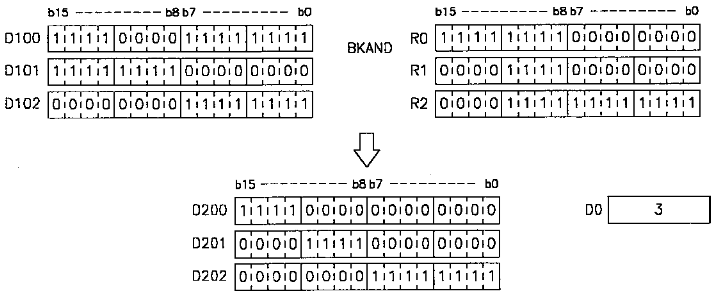
(1) The following program performs a logical product operation on the number of points of data corresponding to the values stored in D100 and D0, and the number of points of data corresponding to the values stored in R0 and D0, and stores the results from D200 onwards, when X20 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	BKANDP	D100 R0 D200 D0
6	END	

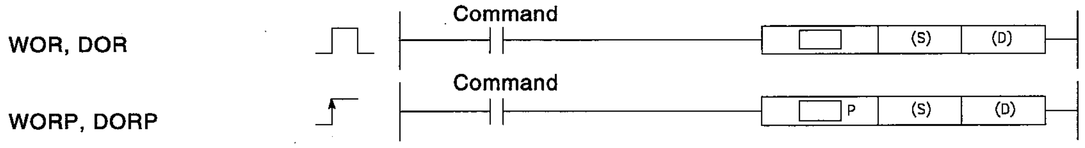


7.1.3 Logical sums of 16-bit and 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	○							○	—
(D)	○							—	—

[Instruction Symbol] [Execution Condition]

☐ indicates WOR/DOR

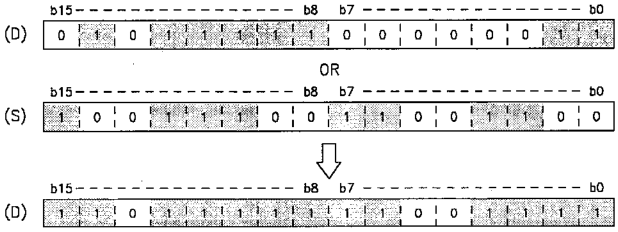


Set Data

Set Data	Meaning	Data Type
(S)	The data on which a logical sum operation will be performed, or the initial number of the devices storing this data	BIN 16/32 bits
(D)		

WOR

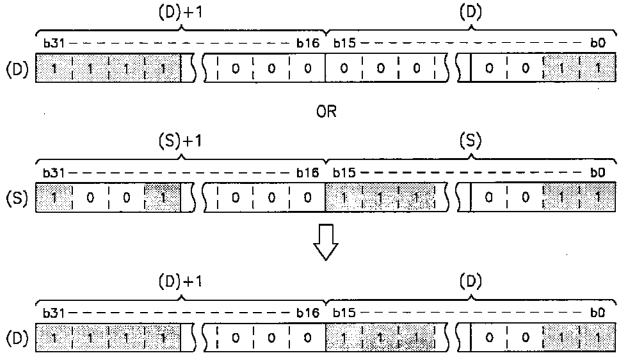
(1) Conducts a logical sum operation on each bit of the 16-bit data of the device designated by (D) and the 16-bit data of the device designated by (S), and stores the results at the device designated by (D).



(2) In the case of bit devices, digits above the number designated are processed as 0 in the operation.

DOR

(1) Conducts a logical sum operation on each bit of the 32-bit data of the device designated by (D) and the 32-bit data of the device designated by (S), and stores the results at the device designated by (D).



(2) In the case of bit devices, digits above the number designated are processed as 0 in the operation.

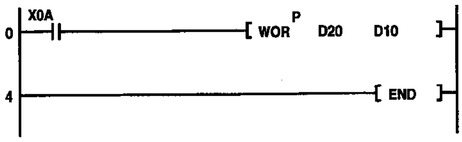
Operation Errors

(1) There are no operation errors associated with the WOR(P) or DOR(P) instructions.

Program Example

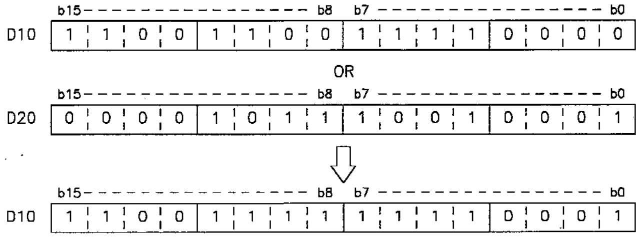
(1) The following program performs a logical sum operation on the data at D10 and D20 when XA goes ON, and stores the results at D10.

[Ladder Mode]



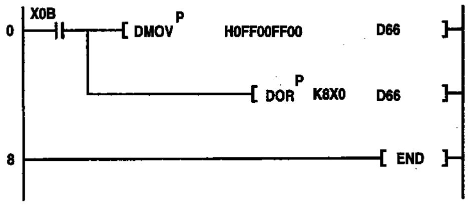
[List Mode]

Steps	Instruction	Device
0	LD	X0A
1	WORP	D20 D10
4	END	



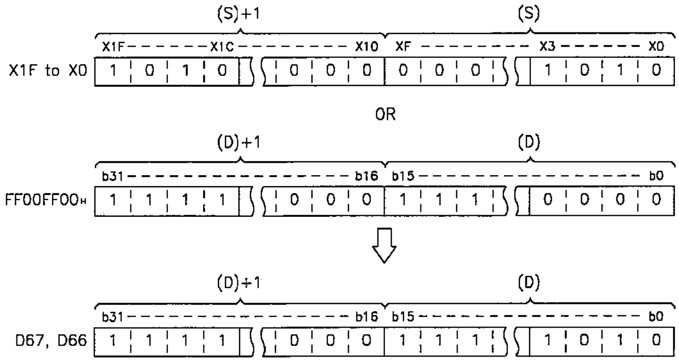
(2) The following program performs a logical sum operation on the 32-bit data from X0 to X1F, and on the hexadecimal value FF00FF00H when XB goes ON, and stores the results at D66 and D67.

[Ladder Mode]



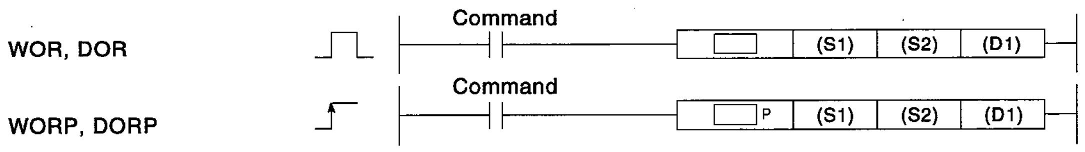
[List Mode]

Steps	Instruction	Device
0	LD	X0B
1	DMOVP	H0FF00FF00 D66
5	DORP	K8X0 D66
8	END	



Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
(S1)	o							o	—
(S2)	o							o	—
(D1)	o							—	—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S1) (S2)	The data on which a logical sum operation will be performed, or the initial number of the devices storing this data	BIN 16/32 bits
(D1)	Initial number of devices that will store the results of the logical sum operation	

WOR

(1) Conducts a logical sum operation on each bit of the 16-bit data of the device designated by (S1) and the 16-bit data of the device designated by (S2), and stores the results at the device designated by (D1).

	b15							b8	b7							b0
(S1)	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0

OR

	b15							b8	b7							b0
(S2)	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	1

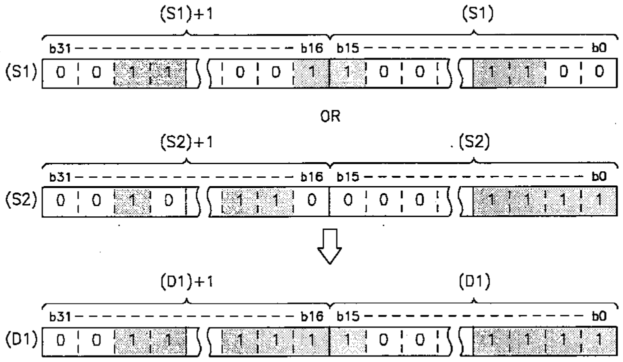
⇩

	b15							b8	b7							b0
(D1)	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1	1

(2) In the case of bit devices, digits above the number designated are processed as 0 in the operation. [See Program Example (1)]

DOR

(1) Conducts a logical sum operation on each bit of the 32-bit data of the device designated by (S1) and the 32-bit data of the device designated by (S2), and stores the results at the device designated by (D1).



(2) In the case of bit devices, digits above the number designated are processed as 0 in the operation. [See Program Example (2)]

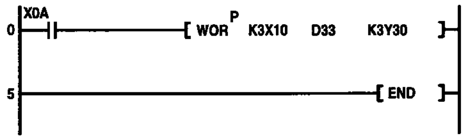
Operation Errors

(1) There are no operation errors associated with the WOR(P) or DOR(P) instructions.

Program Example

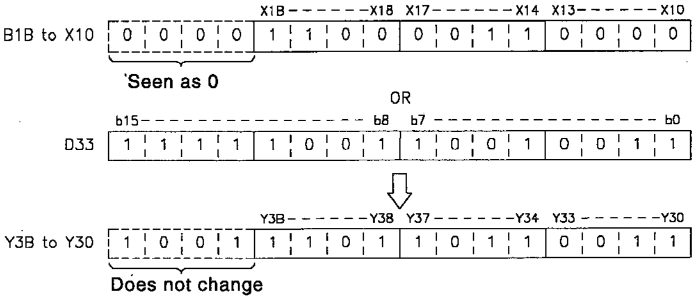
(1) The following program performs a logical sum operation on the data from X10 to X1B, and the data at D33, and stores the result at Y30 through Y3B when XA is ON.

[Ladder Mode]



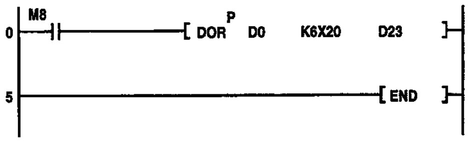
[List Mode]

Steps	Instruction	Device
0	LD	X0A
1	WORP	K3X10 D33
5	END	K3Y30



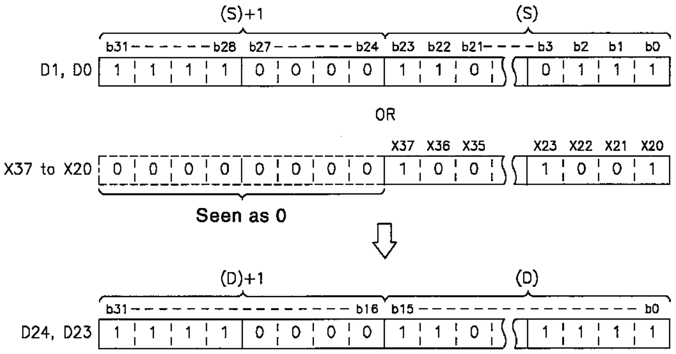
(2) The following program performs a logical sum operation on the 32-bit data at D0 and D1, and the 24-bit data from X20 to X37, and stores the results at D23 and D24 when M8 is ON.

[Ladder Mode]



[List Mode]

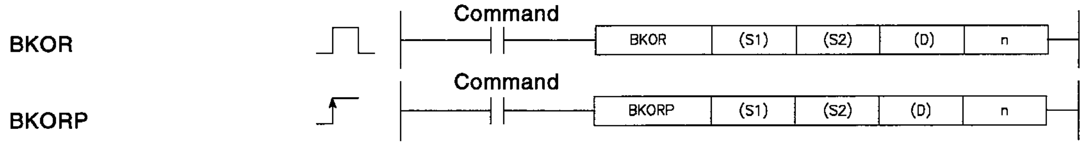
Steps	Instruction	Device
0	LD	M8
1	DORP	D0
		K6X20
5	END	D23



7.1.4 Block logical sum operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J⬚\⬚		Special Function Module U⬚\G⬚	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—				—	—
(S2)	—	o		—				o	—
(D)	—	o		—				—	—
n	o	o		o				o	—

[Instruction Symbol] [Execution Condition]

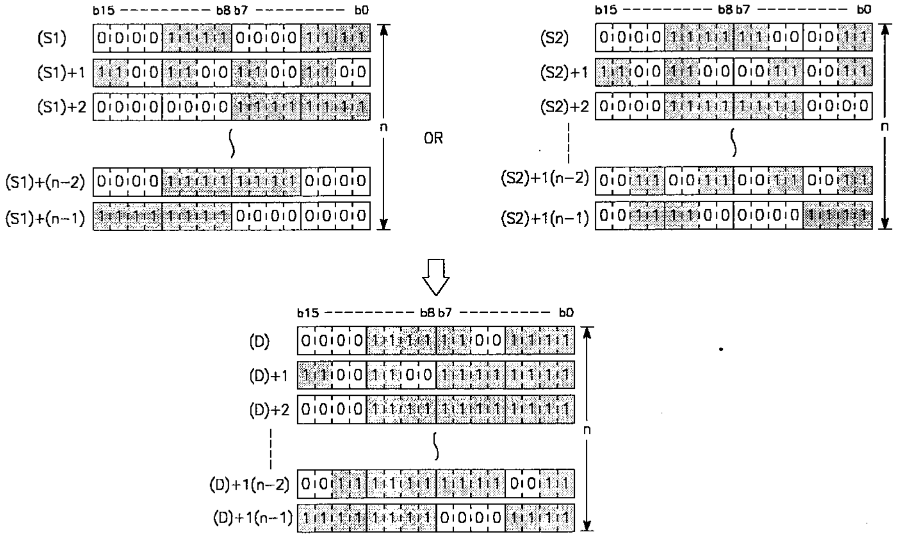


Set Data

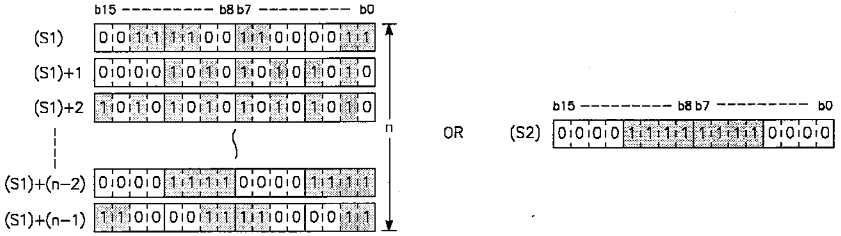
Set Data	Meaning	Data Type
(S1)	First number of the devices where the data subjected to the logical product operation is stored	BIN 16 bits
(S2)	First number of the logical operation data or the devices where the logical operation data is stored	
(D)	First number of the devices where the operation results are stored	
n	Number of operation data bits	

Functions

(1) Performs a logical sum operation on the data located in the n-points from the device designated by (S1), and the data located in the n-points from the device designated by (S2), and stores the results at the device designated by (D).



(2) The constant designated by (S2) can be between -32768 and 32767 (BIN 16 bits).

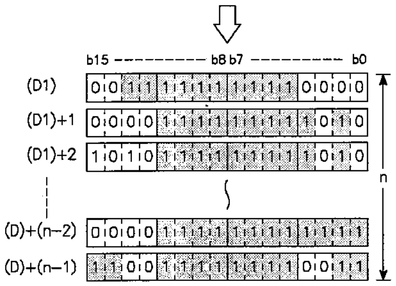


b15 --------- b8 b7 --------- b0

(S1)	0 0 1 1	1 1 0 0	1 1 0 0	0 0 1 1
(S1)+1	0 0 0 0	1 0 1 0	1 0 1 0	1 0 1 0
(S1)+2	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0
(S1)+(n−2)	0 0 0 0	1 1 1 1	0 0 0 0	1 1 1 1
(S1)+(n−1)	1 1 0 0	0 0 1 1	1 1 0 0	0 0 1 1

b15 --------- b8 b7 --------- b0

(S2)	0 0 0 0	1 1 1 1	1 1 1 1	0 0 0 0



b15 --------- b8 b7 --------- b0

(D1)	0 0 1 1	1 1 1 1	1 1 1 1	0 0 0 0
(D1)+1	0 0 0 0	1 1 1 1	1 1 1 1	1 0 1 0
(D1)+2	1 0 1 0	1 1 1 1	1 1 1 1	1 0 1 0
(D)+(n−2)	0 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1
(D)+(n−1)	1 1 0 0	1 1 1 1	1 1 1 1	0 0 1 1

Operation Errors

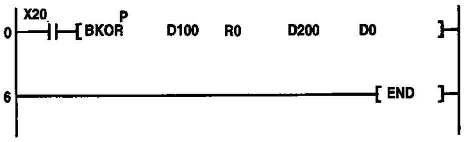
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The n-bit range from the (S1), (S2), or (D) device exceeds the range of that device. (Error code: 4101)

Program Example

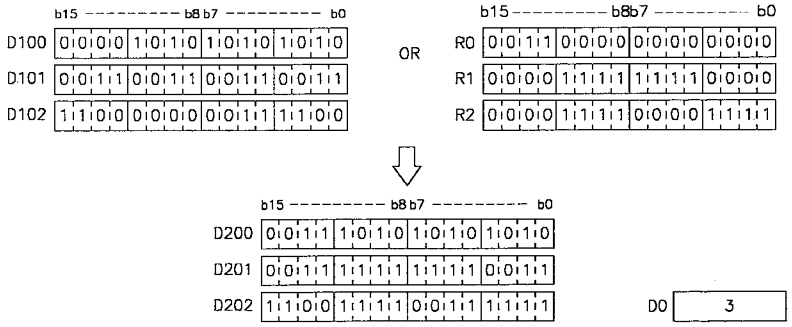
(1) The following program conducts a logical operation on the number of points of data corresponding to the value stored in D100 and D0, and the number of points of data corresponding to the data stored in R0 and D0 when X20 goes ON, and stores the results from D200 onwards.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	BKORP	D100 R0 D200 D0
6	END	



b15 --------- b8 b7 --------- b0

D100	0 0 0 0	1 0 1 0	1 0 1 0	1 0 1 0
D101	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1
D102	1 1 0 0	0 0 0 0	0 0 1 1	1 1 0 0

OR

b15 --------- b8b7 --------- b0

R0	0 0 1 1	0 0 0 0	0 0 0 0	0 0 0 0
R1	0 0 0 0	1 1 1 1	1 1 1 1	0 0 0 0
R2	0 0 0 0	1 1 1 1	0 0 0 0	1 1 1 1

b15 --------- b8 b7 --------- b0

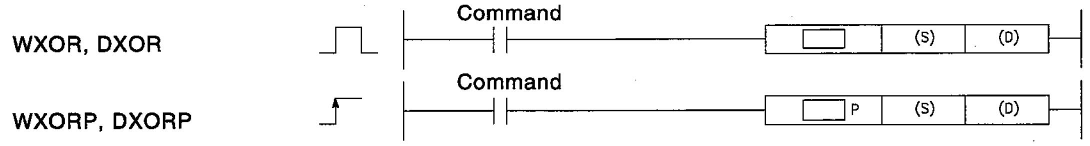
D200	0 0 1 1	1 0 1 0	1 0 1 0	1 0 1 0
D201	0 0 1 1	1 1 1 1	1 1 1 1	0 0 1 1
D202	1 1 0 0	1 1 1 1	0 0 1 1	1 1 1 1

D0	3

7.1.5 16-bit and 32-bit exclusive OR operations

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	○							○	—
(D)	○							—	—

[Instruction Symbol] [Execution Condition] □ indicates WXOR/DXOR

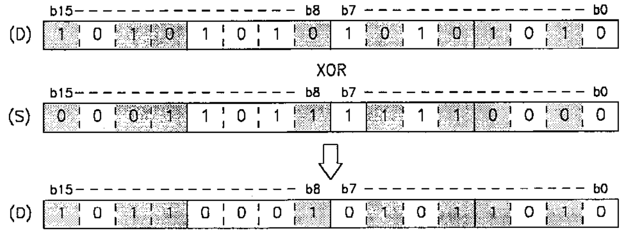


Set Data

Set Data	Meaning	Data Type
(S)	Data on which exclusive OR operation will be performed, or initial number of devices storing such data	BIN 16/32 bits
(D)		

WXOR

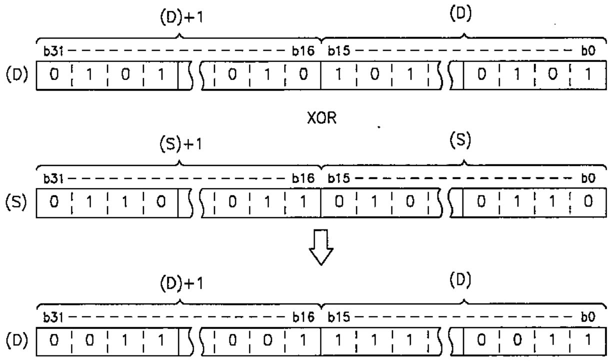
(1) Conducts an exclusive OR operation on each bit of the 16-bit data of the device designated by (D) and the 16-bit data of the device designated by (S), and stores the results at the device designated by (D).



(2) Everything beyond the digit designation of a bit device is treated as 0.

DXOR

(1) Conducts an exclusive OR operation on the 32-bit data designated by (D) and the 32-bit data designated by (S), and stores the results at the device designated by (D).



(2) Everything beyond the digit designation of a bit device is treated as 0.

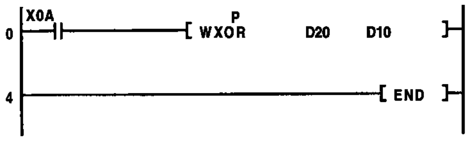
Operation Errors

(1) If index qualification has not been conducted, there are no operation errors associated with the WXOR(P) or DXOR(P) instructions.

Program Example

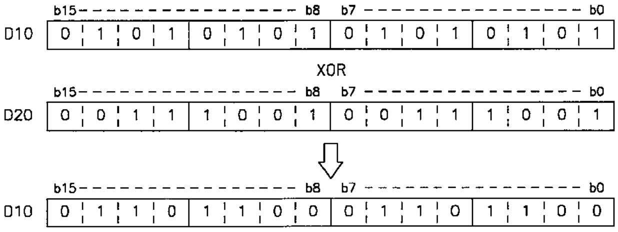
(1) The following program performs an exclusive OR operation on the data at D10 and D20 when XA is ON, and stores the result at D10.

[Ladder Mode]



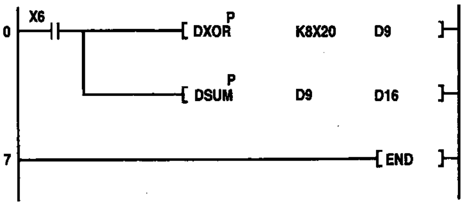
[List Mode]

Steps	Instruction	Device
0	LD	X0A
1	WXORP	D20 D10
4	END	



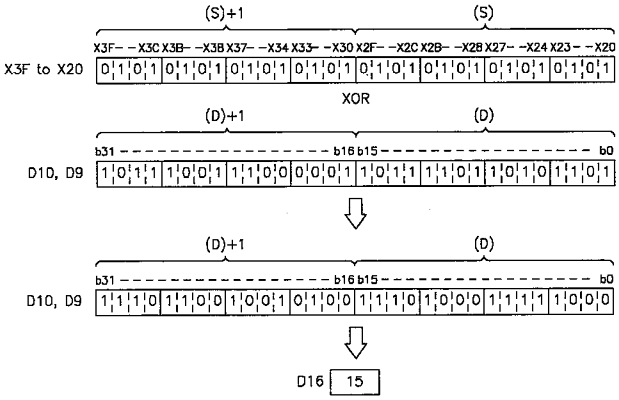
(2) The following program compares the bit pattern of the 32-bit data from X20 through X3F with the bit pattern of the data at D9 and D10 when X6 is ON, and stores the number of differing bits at D16.

[Ladder Mode]



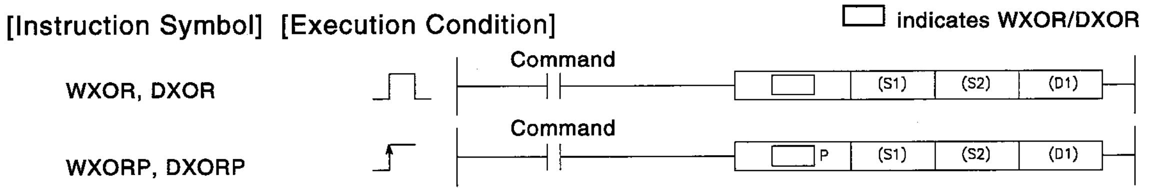
[List Mode]

Steps	Instruction	Device
0	LD	X6
1	DXORP	K8X20 D9
4	DSUMP	D9 D16
7	END	



Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o							o	—
(S2)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]

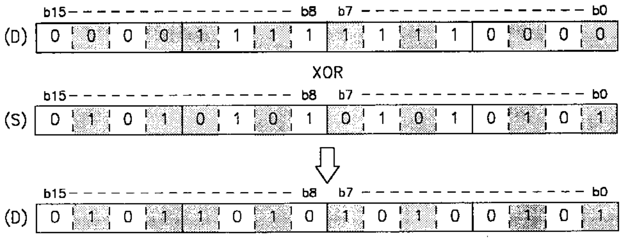


Set Data

Set Data	Meaning	Data Type
(S1)	Data on which exclusive OR operation will be performed, or initial number of devices storing such data	BIN 16/32 bits
(S2)		
(D)	Initial number of the devices that will store results of the exclusive OR operation	

WXOR

(1) Conducts an exclusive OR operation on each bit of the 16-bit data of the device designated by (S1) and the 16-bit data of the device designated by (S2), and stores the results at the device designated by (D1).



	b15							b8	b7							b0
(D)	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0

XOR

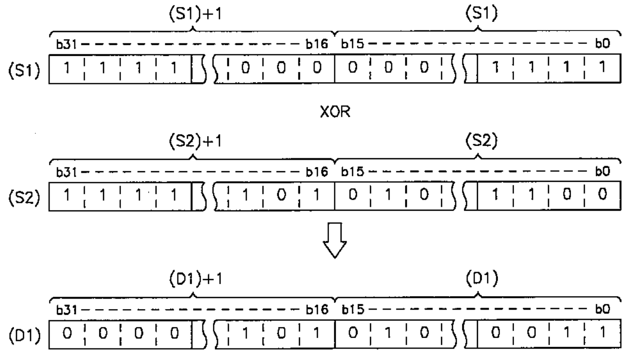
	b15							b8	b7							b0
(S)	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

	b15							b8	b7							b0
(D)	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1

(2) Everything beyond the digit designation of a bit device is treated as 0. [See Program Example (2)]

DXOR

(1) Conducts an exclusive OR operation on the 32-bit data designated by (S1) and the 32-bit data designated by (S2), and stores the results at the device designated by (D1).



(2) Everything beyond the digit designation of a bit device is treated as 0.

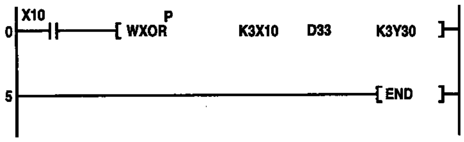
Operation Errors

(1) There are no operation errors associated with the WXOR(P) or DXOR(P) instructions.

Program Example

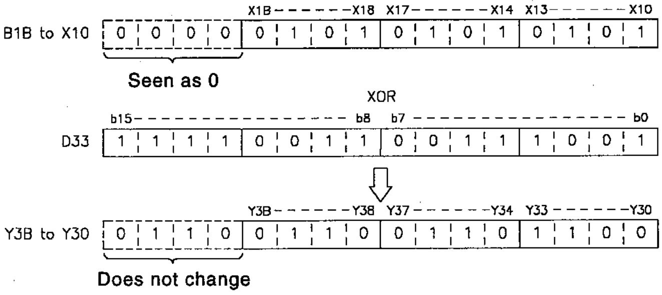
(1) The following program conducts an exclusive OR operation on the data from X10 to X1B and the data at D33 when X10 is ON, and outputs the result to from Y30 to Y3B.

[Ladder Mode]



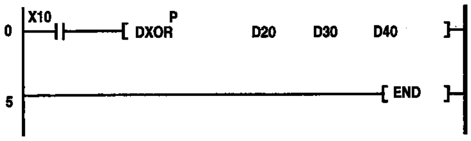
[List Mode]

Steps	Instruction	Device
0	LD	X10
1	WXORP	K3X10 D33 K3Y30
5	END	



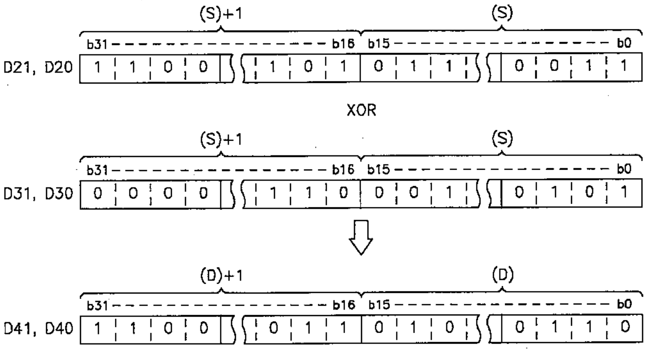
(2) The following program conducts an exclusive OR operation on the data at D20 and D21, and the data at D30 and D31 when X10 goes ON, and stores the results at D40 and D41.

[Ladder Mode]



[List Mode]

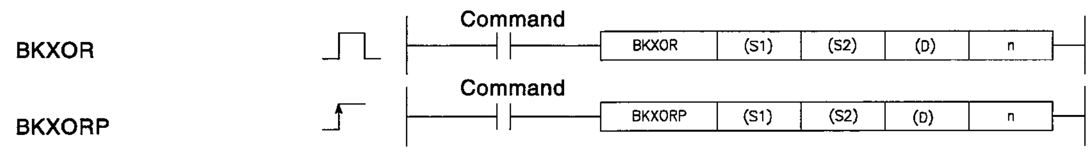
Steps	Instruction	Device
0	LD	X10
1	DXORP	D20 D30 D40
5	END	



7.1.6 Block exclusive OR operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—				—	—
(S2)	—	o		—				o	—
(D)	—	o		—				—	—
n	o	o		o				o	—

[Instruction Symbol] [Execution Condition]

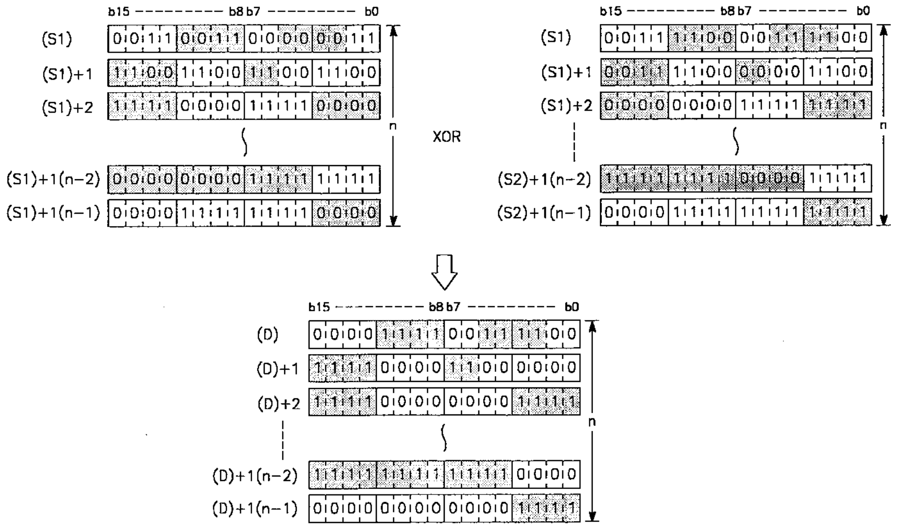


Set Data

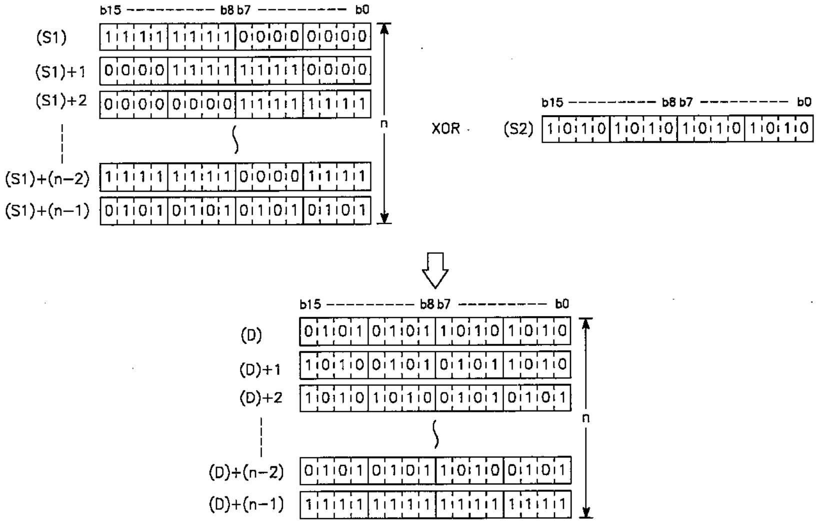
Set Data	Meaning	Data Type
(S1)	First number of the devices where the data subjected to the logical product operation is stored	BIN 16 bits
(S2)	First number of the logical operation data or the devices where the logical operation data is stored	
(D)	First number of the devices where the operation results are stored	
n	Number of operation data bits	

Functions

(1) Performs an exclusive OR operation on the data located in the n-points from the device designated by (S1), and the data located in the n-points from the device designated by (S2), and stores the results at the device designated by (D).



(2) The constant designated by (S2) can be between -32768 and 32767 (BIN 16 bits).



Operation Errors

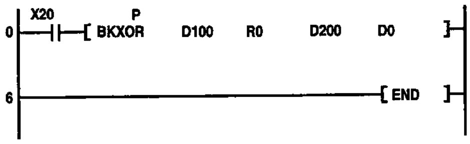
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The n-bit range from the (S1), (S2), or (D) device exceeds the range of that device. (Error code: 4101)

Program Example

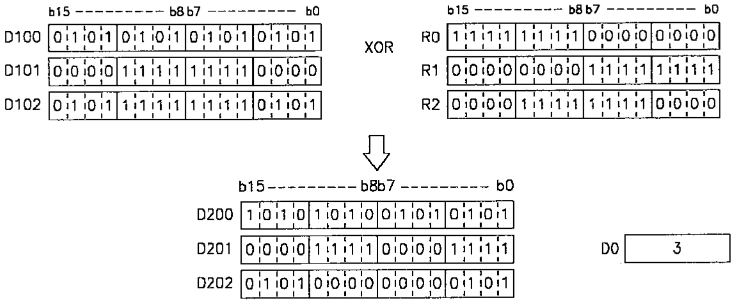
(1) The following program performs a non-exclusive logical sum operation when X20 goes ON on the values of the data in the number of points stored in D100 and D0, and the values of the data in the number of points stored in R0 and D0, and stores the results of this operation at devices starting from D200 onwards.

[Ladder Mode]



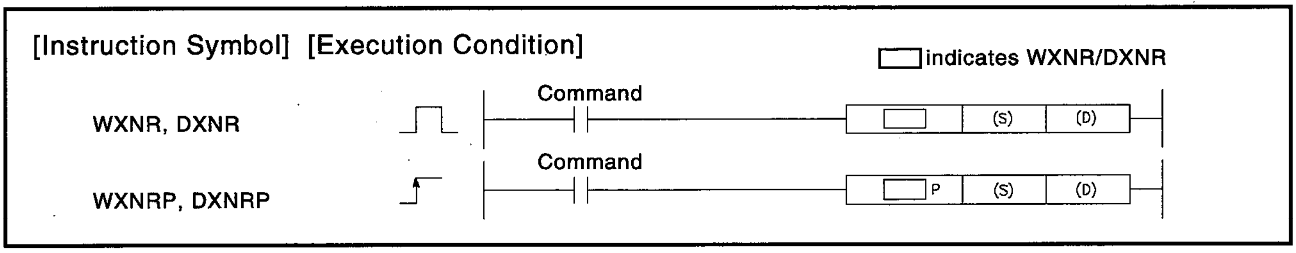
[List Mode]

Steps	Instruction	Device
0	LD	X20
1	BKXORP	D100 R0 D200 D0
6	END	



7.1.7 16-bit and 32-bit data non-exclusive logical sum operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

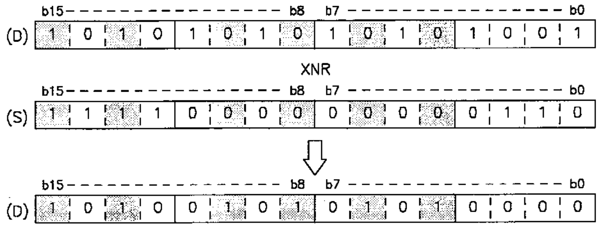


Set Data

Set Data	Meaning	Data Type
(S)	Data on which non-exclusive logical sum operation will be performed, or initial number of devices where such data is being stored	BIN 16/ 32 bits
(D)		

WXNR

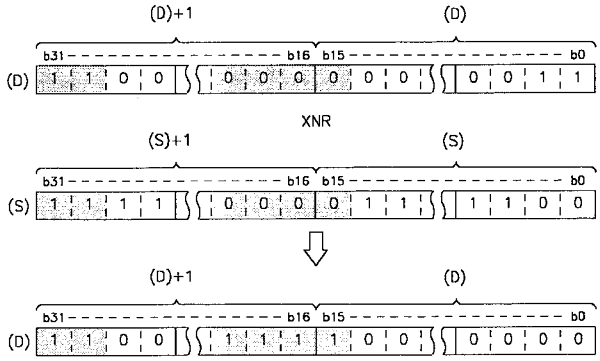
(1) Conducts a non-exclusive logical sum operation on the 16-bit data of the device designated by (D) and the 16-bit data of the device designated by (S), and stores the results at the device designated by (D).



(2) Everything beyond the digit designation of a bit device is treated as 0.

DXNR

(1) Conducts a non-exclusive logical sum operation on the 32-bit data of the device designated by (D) and the 32-bit data of the device designated by (S), and stores the results at the device designated by (D).



(2) Everything beyond the digit designation of a bit device is treated as 0.

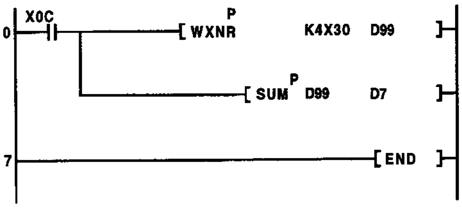
Operation Errors

(1) There are no operation errors associated with the WXNR(P) or DXNR(P) instructions.

Program Example

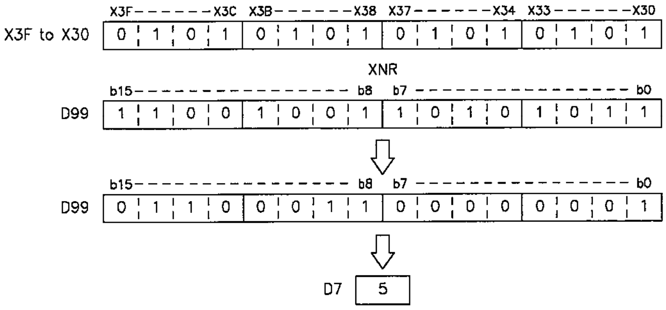
(1) The following program compares the bit patterns of the 16-bit data located from X30 through X3F with the bit patterns of the 16-bit data at D99 when XC is ON, and stores the number of identical patterns at D7.

[Ladder Mode]



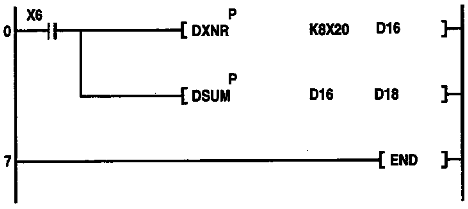
[List Mode]

Steps	Instruction	Device
0	LD	X0C
1	WXNRP	K4X30 D99
4	SUMP	D99 D7
7	END	



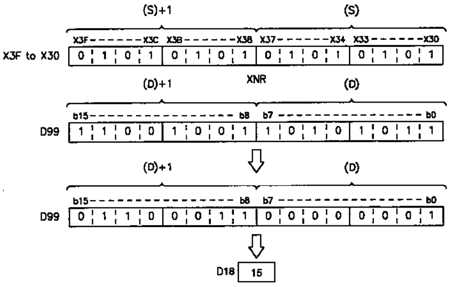
(2) The following program compares the bit patterns of the 32-bit data located from X20 through X3F with the bit patterns of the data at D16 and D17 when X6 is ON, and stores the number of identical bit patterns at D18.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X6
1	DXNRP	K8X20 D16
4	DSUMP	D16 D18
7	END	

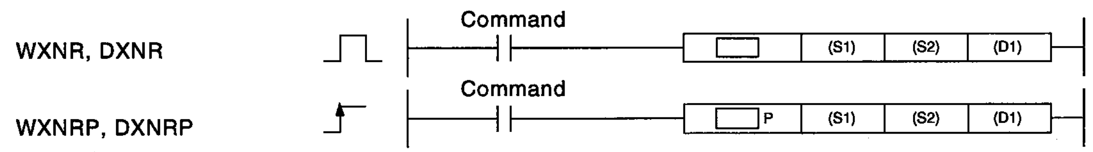


REMARK

* : See Section 7.5.2 for more information on the SUMP/DSUM instructions.

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o							o	—
(S2)	o							o	—
(D1)	o							—	—

[Instruction Symbol] [Execution Condition]

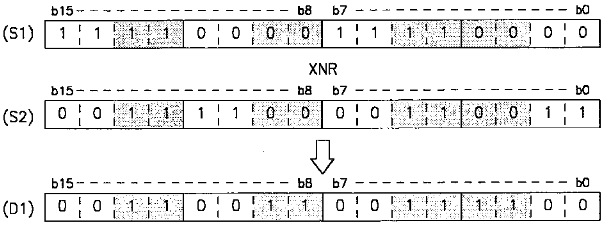


Set Data

Set Data	Meaning	Data Type
(S1)	Data on which non-exclusive logical sum operation will be performed, or initial number of devices where such data is being stored	BIN 16/32 bits
(S2)		
(D1)	Initial number of devices that will store results of the non-exclusive logical sum operation	

WXNR

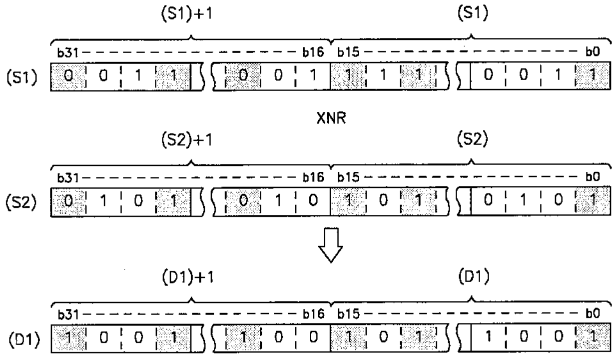
(1) Conducts a non-exclusive logical sum operation on the 16-bit data of the device designated by (S1) and the 16-bit data of the device designated by (S2), and stores the results at the device designated by (D1).



(2) Everything beyond the digit designation of a bit device is treated as 0.

DXNR

(1) Conducts a non-exclusive logical sum operation on the 32-bit data of the device designated by (S1) and the 32-bit data of the device designated by (S2), and stores the results at the device designated by (D1).



(2) Everything beyond the digit designation of a bit device is treated as 0.

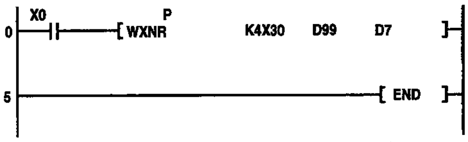
Operation Errors

(1) There are no operation errors associated with the WXNR(P) or DXNR(P) instructions.

Program Example

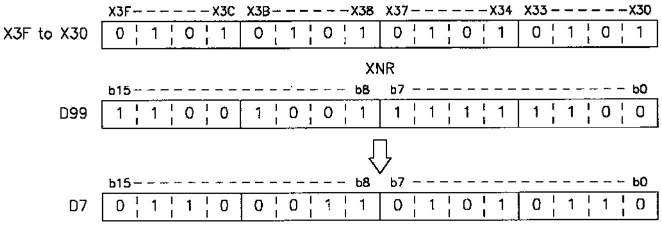
(1) The following program performs a non-exclusive logical sum operation on the 16-bit data from X30 through X3F and the data at D99 when X0 is ON, and stores the results at D7.

[Ladder Mode]



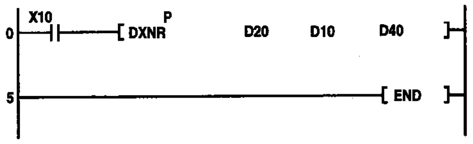
[List Mode]

Steps	Instruction	Device
0	LD	X0
1	WXNRP	K4X30 D99 D7
5	END	



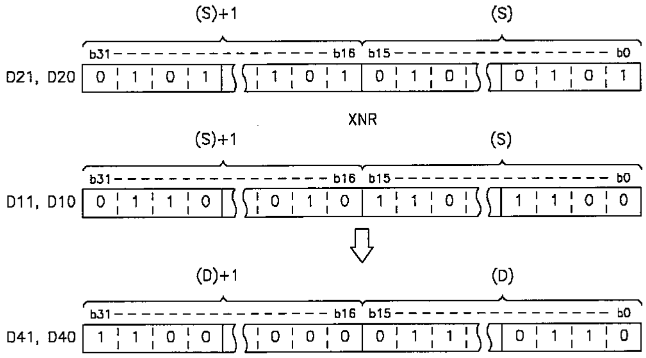
(2) The following program performs a non-exclusive logical sum operation on the 32-bit data at D20 and D21 and the data at D10 and D11 when X10 is ON, and stores the result at D40 and D41.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X10
1	DXNRP	D20 D10 D40
5	END	



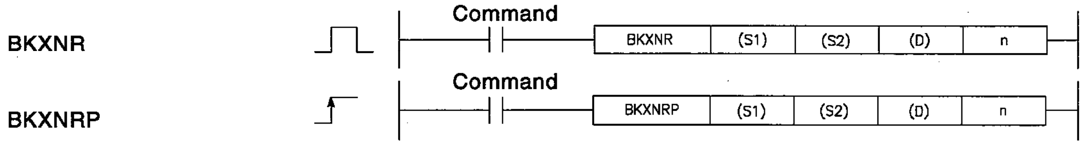
REMARK

* : See Section 7.5.2 for information on the SUM/DSUMP instructions.

7.1.8 Block non-exclusive logical sum operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—				—	—
(S2)	—	o		—				o	—
(D)	—	o		—				—	—
n	o	o		o				o	—

[Instruction Symbol] [Execution Condition]

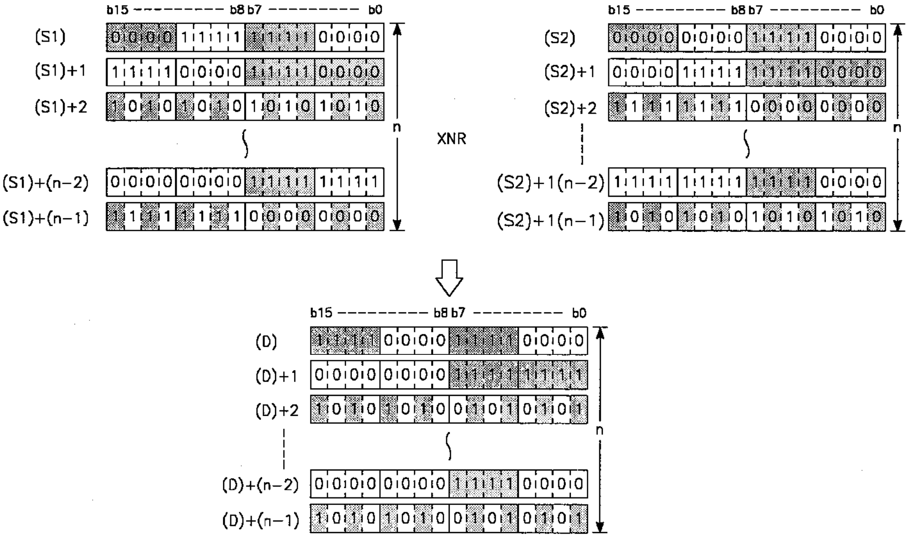


Set Data

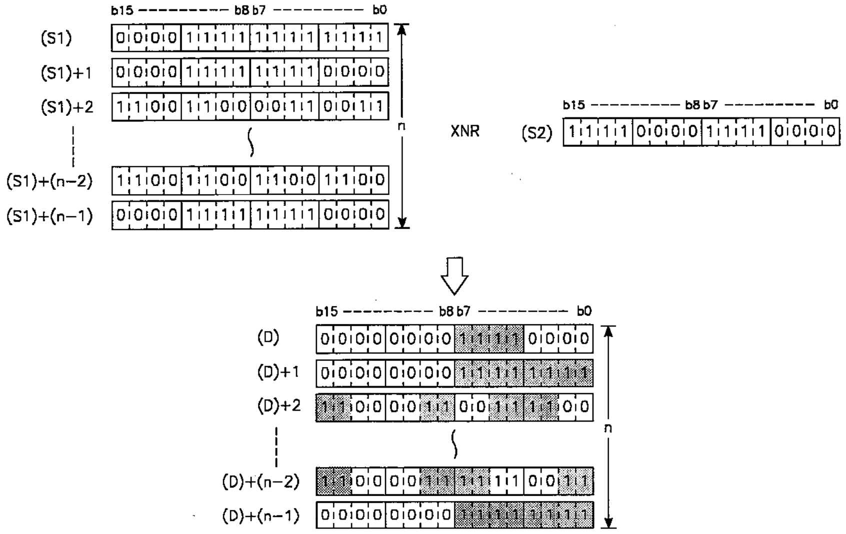
Set Data	Meaning	Data Type
(S1)	First number of the devices where the data subjected to the logical product operation is stored	BIN 16 bits
(S2)	First number of the logical operation data or the devices where the logical operation data is stored	
(D)	First number of the devices where the operation results are stored	
n	Number of operation data bits	

Functions

(1) Performs a non-exclusive logical sum operation on the data located in the n-points from the device designated by (S1), and the data located in the n-points from the device designated by (S2), and stores the results at the device designated by (D).



(2) The constant designated by (S2) can be between -32768 and 32767 (BIN data).



(S1) 0000 1111 1111 1111
(S1)+1 0000 1111 1111 0000
(S1)+2 1100 1100 0011 0011
(S1)+(n−2) 1100 1100 1100 1100
(S1)+(n−1) 0000 1111 1111 0000

(S2) 1111 0000 1111 0000

(D) 0000 0000 1111 0000
(D)+1 0000 0000 1111 1111
(D)+2 1100 0011 0011 1100
(D)+(n−2) 1100 0011 1111 0011
(D)+(n−1) 0000 0000 1111 1111

Operation Errors

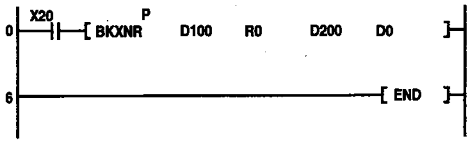
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The n-bit range from the (S1), (S2), or (D) device exceeds the range of that device. (Error code: 4101)

Program Example

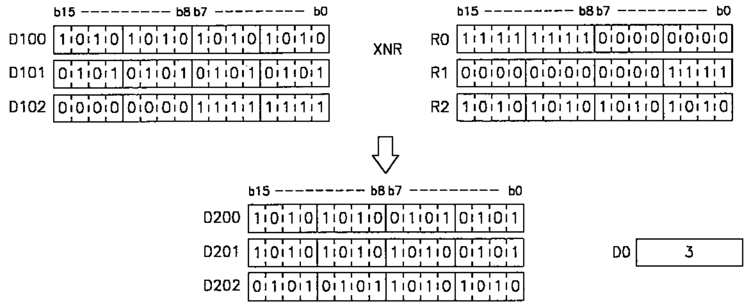
(1) The following program performs a non-exclusive logical sum operation on the number of data corresponding to the value stored in D100 and D0 and the number of points of data corresponding to the value stored at in R0 and D0 when X20 is ON, and stores the result from D200 onwards.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	BKXNRP	D100 R0 D200 D0
6	END	



D100 1010 1010 1010 1010
D101 0101 0101 0101 0101
D102 0000 0000 1111 1111

R0 1111 1111 0000 0000
R1 0000 0000 0000 1111
R2 1010 1010 1010 1010

D200 1010 1010 0101 0101
D201 1010 1010 1010 0101
D202 0101 0101 1010 1010

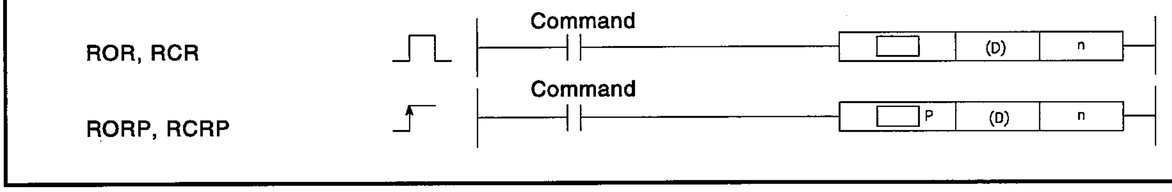
D0 3

7.2 Rotation Instruction

7.2.1 Right rotation of 16-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	o							—	—
n	o							o	—

[Instruction Symbol] [Execution Condition]



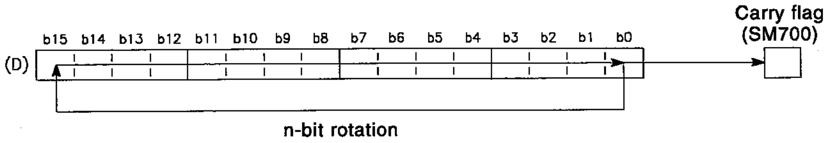
Set Data

Set Data	Meaning	Data Type
(D)	Initial number of devices to perform rotation	BIN 16 bits
n	Number of rotations (from 0 to 15)	

Functions

ROR

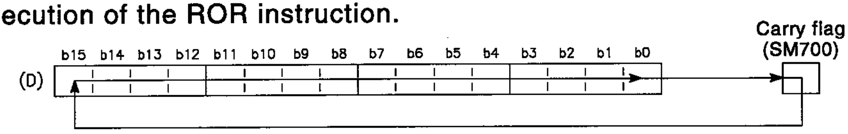
(1) Rotates 16-bit data of the device designated by (D), not including the carry flag, n-bits to the right.
The carry flag is ON or OFF depending on the status prior to the execution of the ROR instruction.



(2) When a bit device has been designated at (D), rotation is conducted by the data in the specified number of digits.
The number of bits by which rotation will actually be performed in such cases is the remainder of the quotient of "n" divided by the specified number of bits.
For example, when n=16 and the specified number of bits is 12 bits, the quotient of 16 divided by 12 is 1 with a remainder of 4, and thus there will be a 4-bit rotation.

RCR

(1) Rotates 16-bit data of the device designated by (D), including carry flag, n bits to the right.
The carry flag is ON or OFF depending on the status prior to the execution of the ROR instruction.



n-bit rotation

(2) When a bit device has been designated at (D), rotation is conducted with the data of the specified number of digits.
The number of bits by which rotation will actually be performed in such cases is the remainder of the quotient of "n" divided by the specified number of bits.
For example, when n=16 and the specified number of bits is 12 bits, the quotient of 16 divided by 12 is 1 with a remainder of 4, and thus there will be a 4-bit rotation.

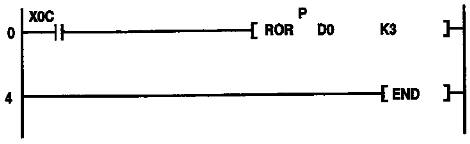
Operation Errors

(1) There are no operation errors associated with the ROR(P) or RCR(P) instructions.

Program Example

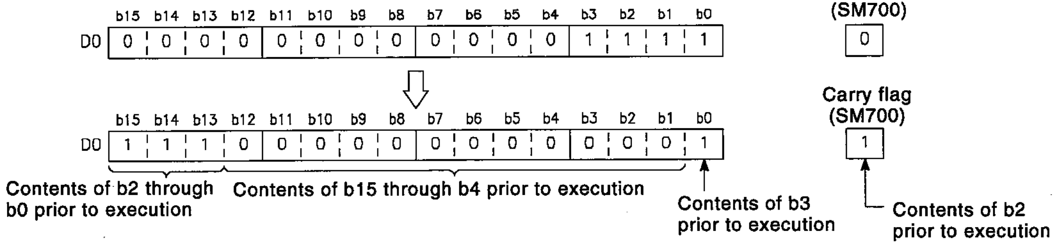
(1) The following program rotates the contents of D0, though not including the carry flag, 3 bits to the right when XC is ON.

[Ladder Mode]



[List Mode]

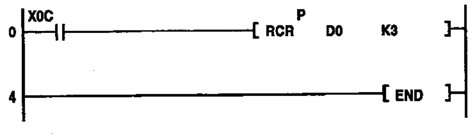
Steps	Instruction	Device
0	LD	X0C
1	RORP	D0 K3
4	END	



	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	Carry flag (SM700)
D0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
D0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

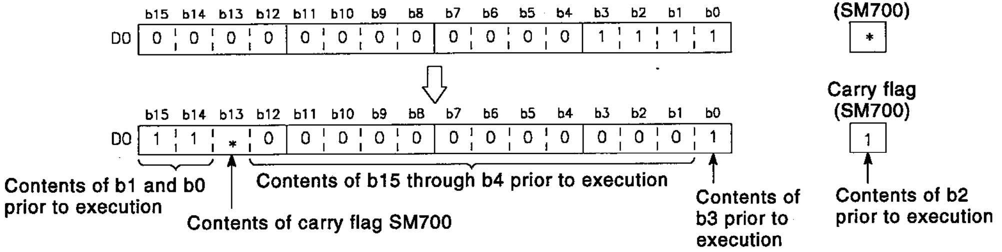
(2) The following program rotates the contents of D0, including the carry flag, 3 bits to the right when XC is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0C
1	RCRP	D0 K3
4	END	

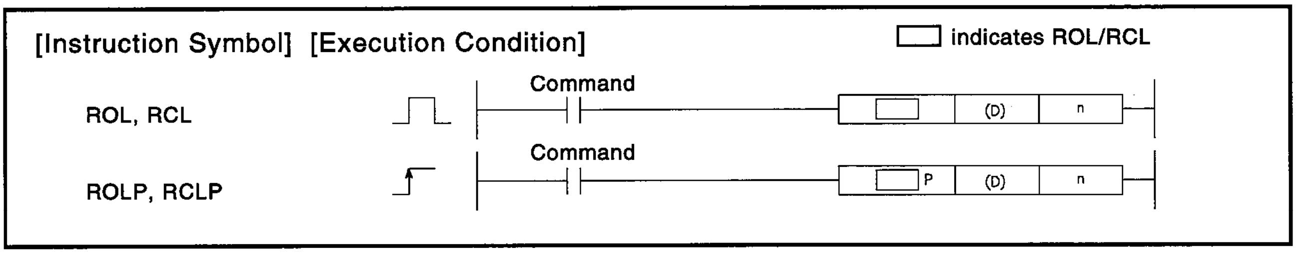


	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	Carry flag (SM700)
D0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	*
D0	1	1	*	0	0	0	0	0	0	0	0	0	0	0	0	1	1

* The carry flag goes ON or OFF depending on its status prior to the execution of RCR.

7.2.2 Left rotation of 16-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	o							—	—
n	o							o	—



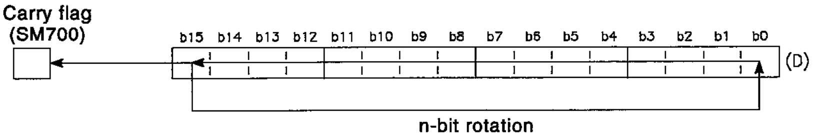
Set Data

Set Data	Meaning	Data Type
(D)	Initial number of devices to perform rotation	BIN 16 bits
n	Number of rotations (from 0 to 15)	

Functions

ROL

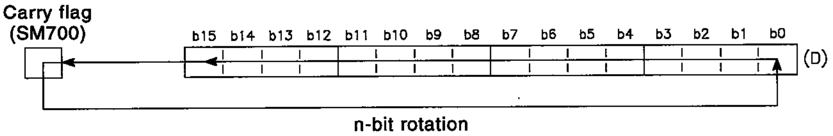
(1) Rotates the 16-bit data of the device designated at (D), not including the carry flag, n-bits to the left.
The carry flag goes ON or OFF depending on its status prior to the execution of ROL instruction.



(2) When a bit device has been designated at (D), rotation is conducted by the data in the specified number of digits.
The number of bits by which rotation will actually be performed in such cases is the remainder of the quotient of "n" divided by the specified number of bits.
For example, when n=16 and the specified number of bits is 12 bits, the quotient of 16 divided by 12 is 1 with a remainder of 4, and thus there will be a 4-bit rotation.

RCL

(1) Rotates the 16-bit data of the device designated by (D), including the carry flag, n bits to the left.
The carry flag goes ON or OFF depending on its status prior to the execution of RCL instruction.



(2) When a bit device has been designated at (D), rotation is conducted with the data of the specified number of digits.
The number of bits by which rotation will actually be performed in such cases is the remainder of the quotient of "n" divided by the specified number of bits.
For example, when n=16 and the specified number of bits is 12 bits, the quotient of 16 divided by 12 is 1 with a remainder of 4, and thus there will be a 4-bit rotation.

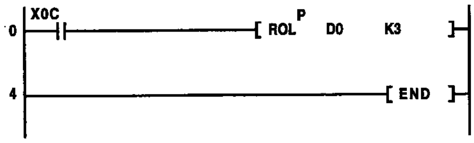
Operation Errors

(1) There are no operation errors associated with the ROL(P) or RCL(P) instructions.

Program Example

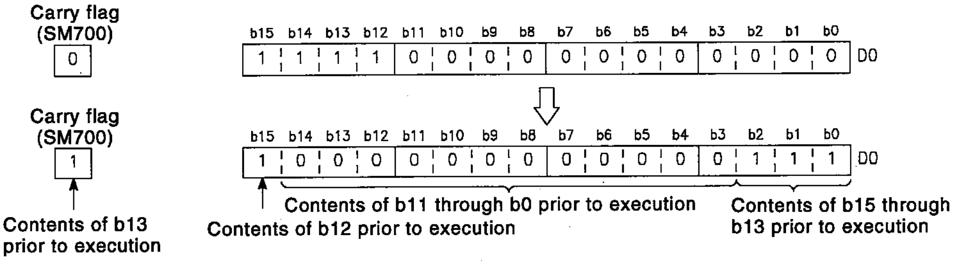
(1) The following program rotates the contents of D0, not including the carry flag, 3 bits to the left when XC is ON.

[Ladder Mode]



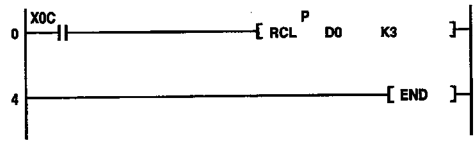
[List Mode]

Steps	Instruction	Device
0	LD	X0C
1	ROLP	D0 K3
4	END	



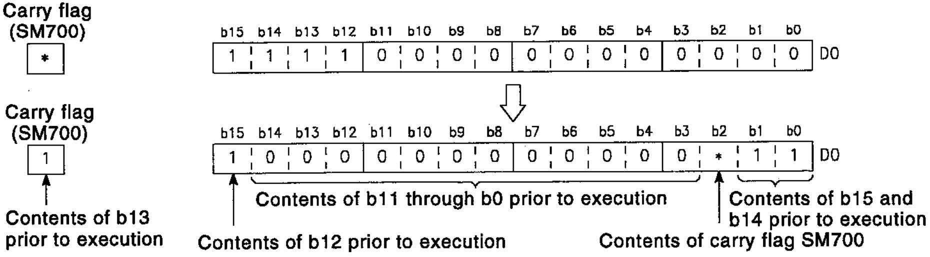
(2) The following program rotates the contents of D0, including the carry flag, 3 bits to the left when XC is ON.

[Ladder Mode]



[List Mode]

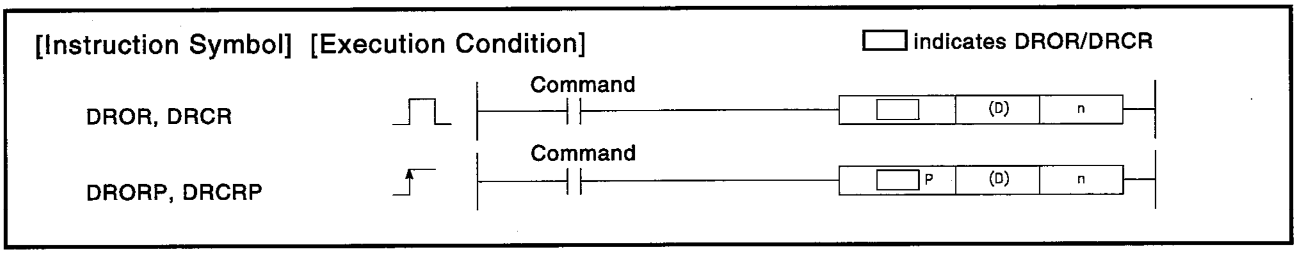
Steps	Instruction	Device
0	LD	X0C
1	RCLP	D0 K3
4	END	



* The carry flag goes ON or OFF depending on its status prior to the execution of RCR.

7.2.3 Right rotation of 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	o							—	—
n	o							o	—



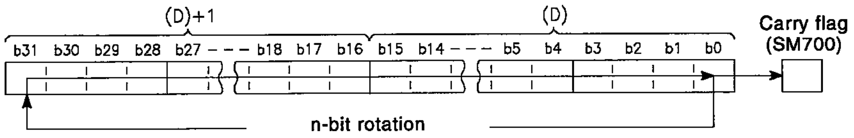
Set Data

Set Data	Meaning	Data Type
(D)	First device number of devices to perform rotation	BIN 32 bits
n	Number of rotations (from 0 to 15)	BIN 16 bits

Functions

DROR

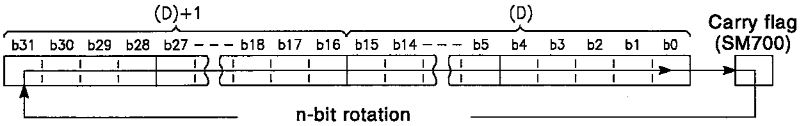
(1) The 32-bit data of the device designated at (D), not including the carry flag, is rotated n-bits to the right.
The carry flag goes ON or OFF depending on its status prior to the execution of the DROR instruction.



(2) When a bit device has been designated at (D), rotation is conducted by the data in the specified number of digits.
The number of bits by which rotation will actually be performed in such cases is the remainder of the quotient of "n" divided by the specified number of bits.
For example, if n=31 and the number of bits designated by digit designation was 24 bits, there would be 7 bits rotated, because the formula would be 31 divided by 24, which equals 1 with a remainder of 7.

DRCR

(1) Rotates 32-bit data, including carry flag, at device designated by (D) n bits to the right.
The carry flag goes ON or OFF depending on its status prior to the execution of the DRCR instruction.



(2) When a bit device has been designated at (D), rotation is conducted by the data in the specified number of digits.
The number of bits by which rotation will actually be performed in such cases is the remainder of the quotient of "n" divided by the specified number of bits.
For example, if n=31 and the number of bits designated by digit designation was 24 bits, there would be 7 bits rotated, because the formula would be 31 divided by 24, which equals 1 with a remainder of 7.

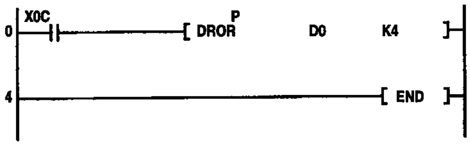
Operation Errors

(1) There are no operation errors associated with the DROR(P) or DRCR(P) instructions.

Program Example

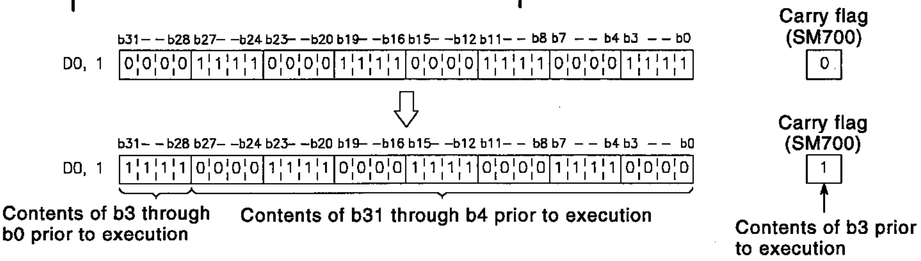
(1) The following program rotates the contents of D0 and D1, not including the carry flag, 4 bits to the right when XC is ON.

[Ladder Mode]



[List Mode]

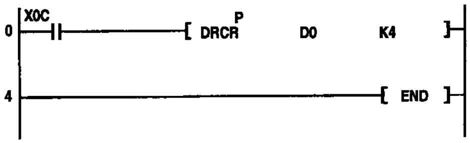
Steps	Instruction	Device
0	LD	X0C
1	DRORP	D0 K4
4	END	



Contents of b3 through b0 prior to execution
Contents of b31 through b4 prior to execution
Contents of b3 prior to execution

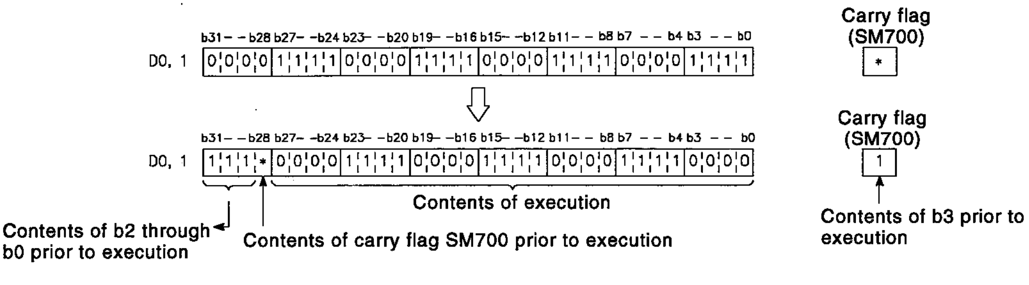
(2) The following program rotates the contents of D0 and D1, including the carry flag, 4 bits to the right when XC is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0C
1	DRCRP	D0 K4
4	END	



Contents of execution
Contents of b2 through b0 prior to execution
Contents of carry flag SM700 prior to execution
Contents of b3 prior to execution

7.2.4 Left rotation of 32-bit data

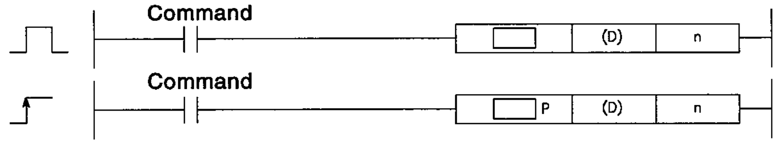
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(D)	o							—	—
n	o							o	—

[Instruction Symbol] [Execution Condition]

□ indicates DROL/DRCL

DROL, DRCL

DROLP, DRCLP



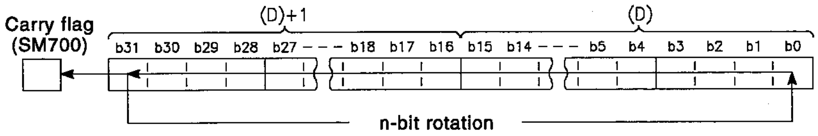
Set Data

Set Data	Meaning	Data Type
(D)	First device number of devices to perform rotation	BIN 32 bits
n	Number of rotations (0 through 31)	BIN 16 bits

Functions

DROL

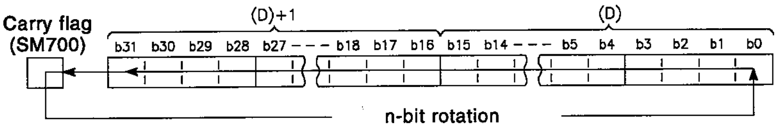
(1) The 32-bit data of the device designated at (D), not including the carry flag, is rotated n-bits to the left.
The carry flag goes ON or OFF depending on its status prior to the execution of the DROL instruction.



(2) When a bit device has been designated at (D), rotation is conducted by the data in the specified number of digits.
The number of bits by which rotation will actually be performed in such cases is the remainder of the quotient of "n" divided by the specified number of bits.
For example, if n=31 and the number of bits designated by digit designation was 24 bits, there would be 7 bits rotated, because the formula would be 31 divided by 24, which equals 1 with a remainder of 7.

DRCL

(1) Rotates 32-bit data, including carry flag, at device designated by (D) n bits to the left.
The carry flag goes ON or OFF depending on its status prior to the execution of the DRCL instruction.



(2) When a bit device has been designated at (D), rotation is conducted by the data in the specified number of digits.
The number of bits by which rotation will actually be performed in such cases is the remainder of the quotient of "n" divided by the specified number of bits.
For example, if n=31 and the number of bits designated by digit designation was 24 bits, there would be 7 bits rotated, because the formula would be 31 divided by 24, which equals 1 with a remainder of 7.

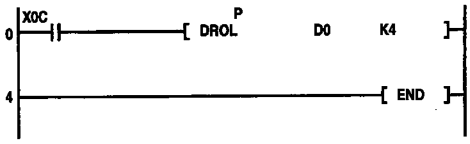
Operation Errors

(1) There are no operation errors associated with the DROL(P) or DRCL(P) instructions.

Program Example

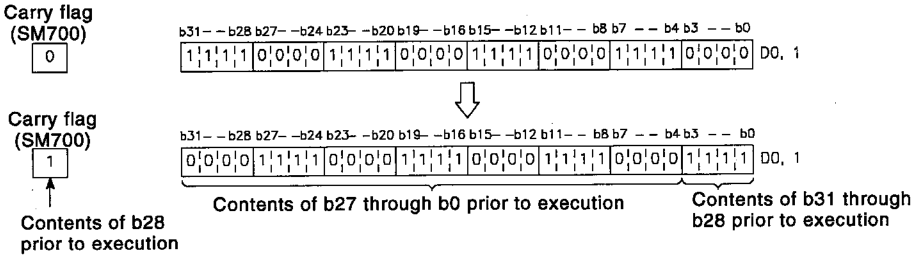
(1) The following program rotates the contents of D0 and D1, not including the carry flag, 4 bits to the left when XC is ON.

[Ladder Mode]



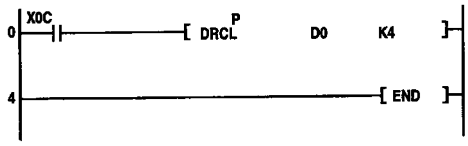
[List Mode]

Steps	Instruction	Device
0	LD	X0C
1	DROLP	D0 K4
4	END	



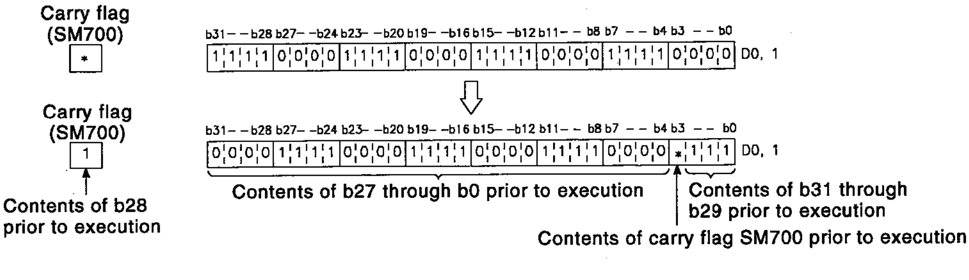
(2) The following program rotates the contents of D0 and D1, including the carry flag, 4 bits to the left when XC is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0C
1	DRCLP	D0 K4
4	END	



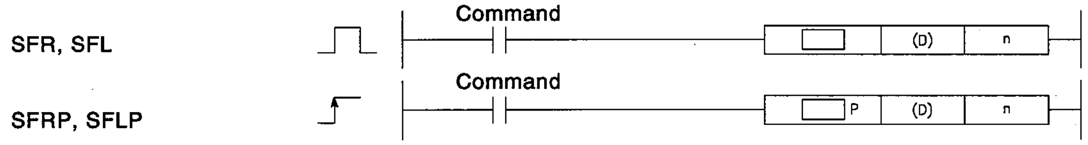
7.3 Shift Instruction

7.3.1 n-bit shift to right or left of 16-bit data

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	o							—	—
n	o							o	—

[Instruction Symbol] [Execution Condition]

□ indicates SFR/SFL



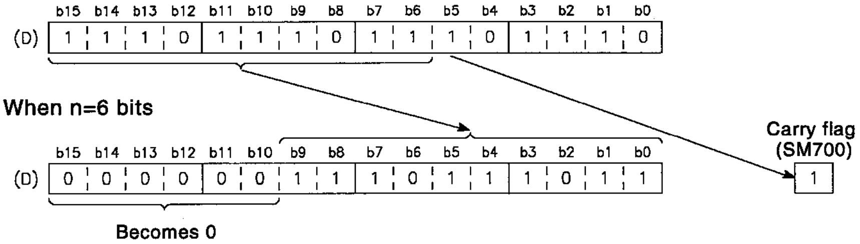
Set Data

Set Data	Meaning	Data Type
(D)	First device number of devices where shift data is being stored	BIN 16 bits
n	Number of shifts (0 to 15)	

Functions

SFR

(1) Causes a shift to the right by n bits of the 16-bit data from the device designated at (D).

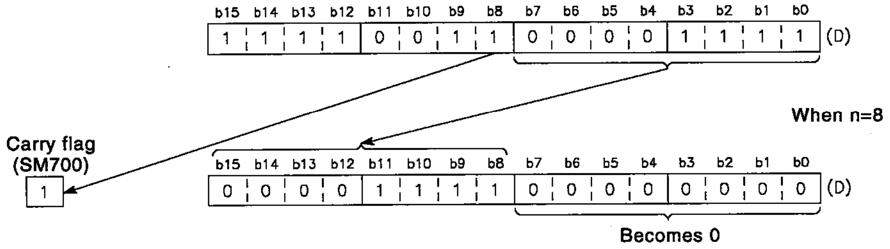


(2) A total of n bits from the upper level become 0.

(3) A shift by the data of the specified digit is conducted when a bit device has been designated at (D). [See Program Example (1)]

SFL

(1) Shifts 16-bit data at device designated by (D) n-bits to the left.



(2) The bits starting at n-bits from lowest bit become 0.

(3) A shift by the data of the specified digit is conducted when a bit device has been designated at (D). [See Program Example (2)]

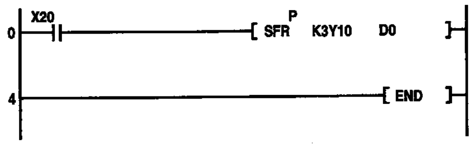
Operation Errors

(1) There are no operation errors associated with the SFR(P) or SFL(P) instructions.

Program Example

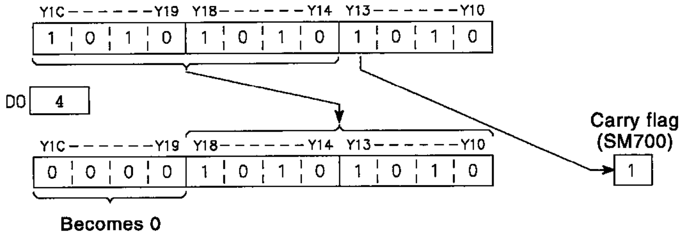
(1) The following program shifts the contents of Y10 through Y1C to the right by the number of bits designated by D0 when X20 goes ON.

[Ladder Mode]



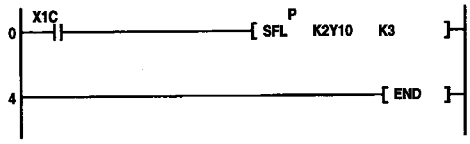
[List Mode]

Steps	Instruction	Device
0	LD	X20
1	SFRP	K3Y10 D0
4	END	



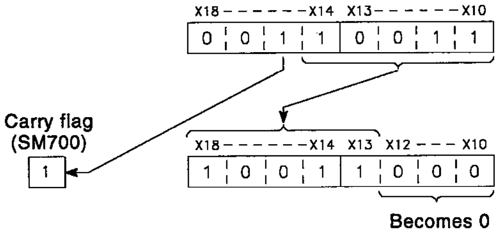
(2) The following program shifts the contents of X10 through X18 3 bits to the left when X1C is ON.

[Ladder Mode]



[List Mode]

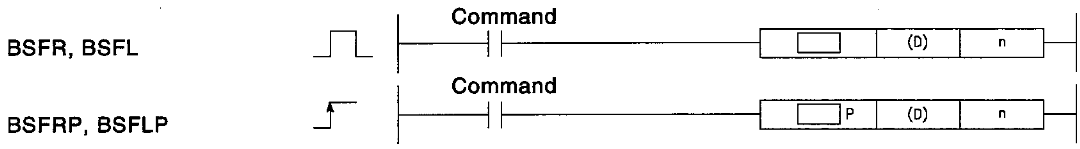
Steps	Instruction	Device
0	LD	X1C
1	SFLP	K2Y10 K3
4	END	



7.3.2 1-bit shift to right or left of n-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				U
(D)	o	—							—
n	—	o							—

[Instruction Symbol] [Execution Condition] □ indicates BSFR/BSFL



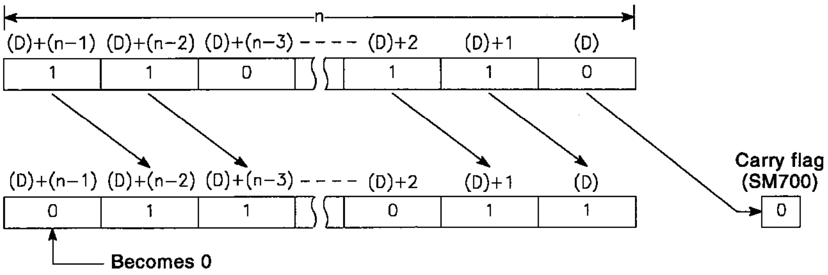
Set Data

Set Data	Meaning	Data Type
(D)	First device number of devices to shift	Bit
n	Number of devices where shift will be conducted	BIN 16 bits

Functions

BSFR

(1) Shifts data n points from device designated by (D) 1 bit to the right.

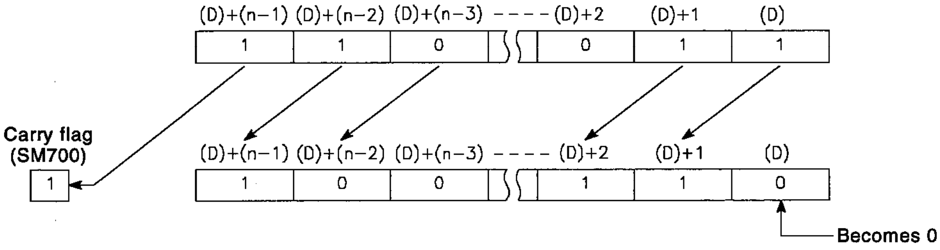


(2) Device designated by (D)+(n-1) becomes 0.

Functions

BSFL

(1) Shifts data n points from device designated by (D) 1 bit to the left.



(2) The device designated by (D) becomes 0.

Operation Errors

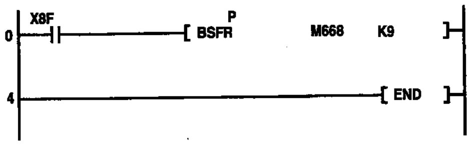
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range of the device n-points from the (D) device exceeds the relevant device. (Error code: 4101)

Program Example

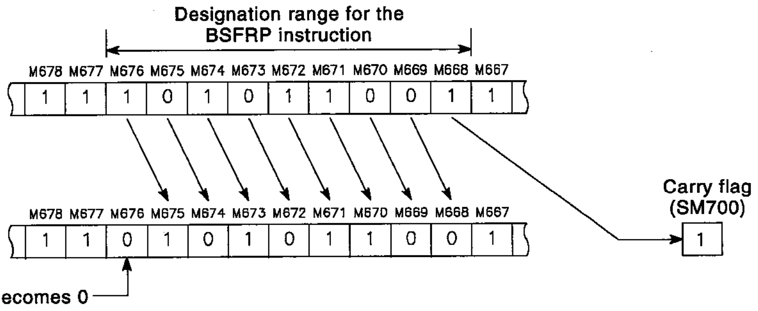
(1) The following program shifts the data at M668 through M676 to the right when X8F is ON.

[Ladder Mode]



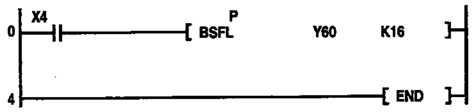
[List Mode]

Steps	Instruction	Device
0	LD	X8F
1	BSFRP	M668 K9
4	END	



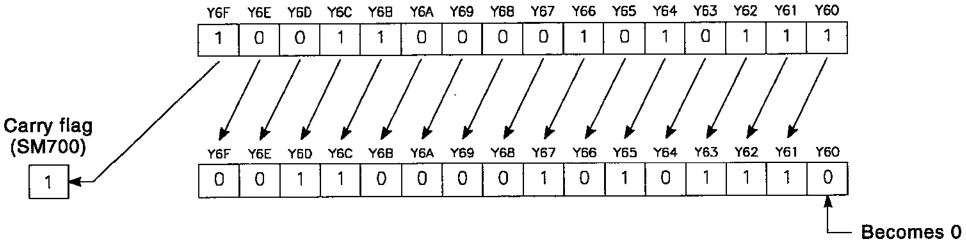
(2) The following program shifts the output from Y60 through Y6F to the left when X4 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X4
1	BSFLP	Y60 K16
4	END	



7.3.3 1-word shift to right or left of n-word data

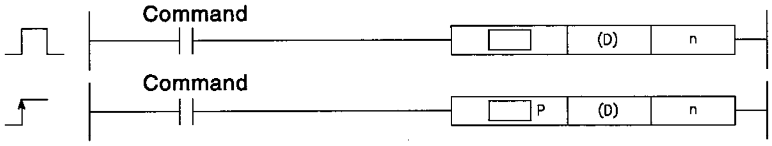
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(D)	—	o		—					—
n	o	o		o					—

[Instruction Symbol] [Execution Condition]

□ indicates DSFR/DSFL

DSFR, DSFL

DSFRP, DSFLP



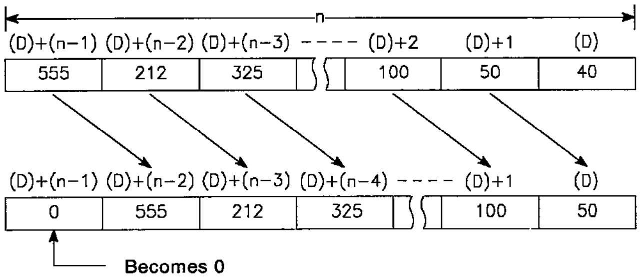
Set Data

Set Data	Meaning	Data Type
(D)	First device number of devices to shift	BIN 16 bits
n	Number of devices where shift will be conducted	BIN 16 bits

Functions

DSFR

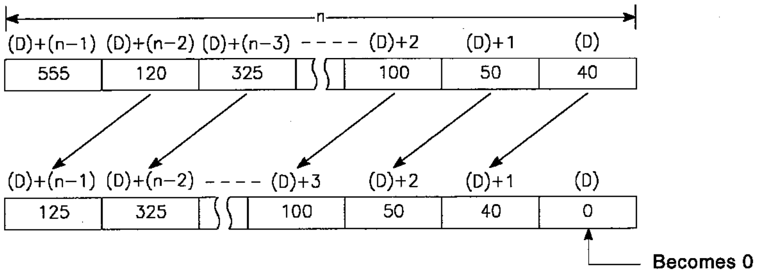
(1) Shifts data n points from device designated by (D) 1-word to the right.



(2) Device designated by (D)+(n-1) becomes 0.

DSFL

(1) Shifts data n points from device designated by (D) 1-word to the left.



(2) The device designated by (D) becomes 0.

Operation Errors

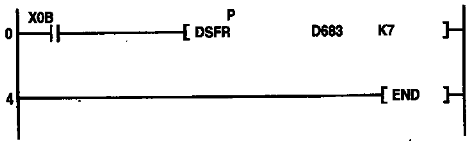
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range of the device n-points from the (D) device exceeds the relevant device. (Error code: 4101)

Program Example

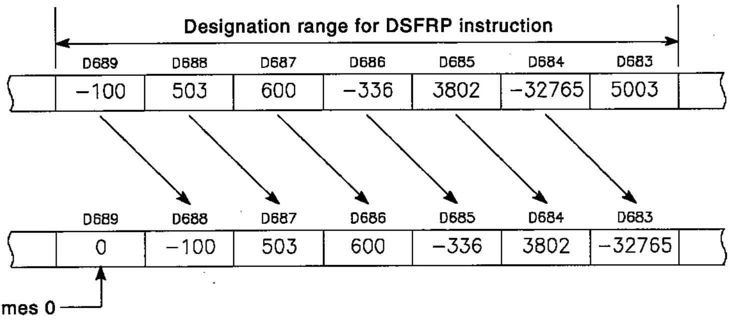
(1) The following program shifts the contents of D683 through D689 to the right when XB is ON.

[Ladder Mode]



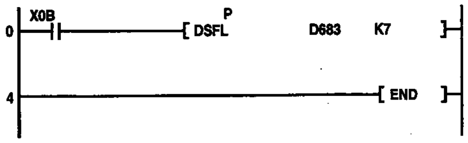
[List Mode]

Steps	Instruction	Device
0	LD	X0B
1	DSFRP	D683 K7
4	END	



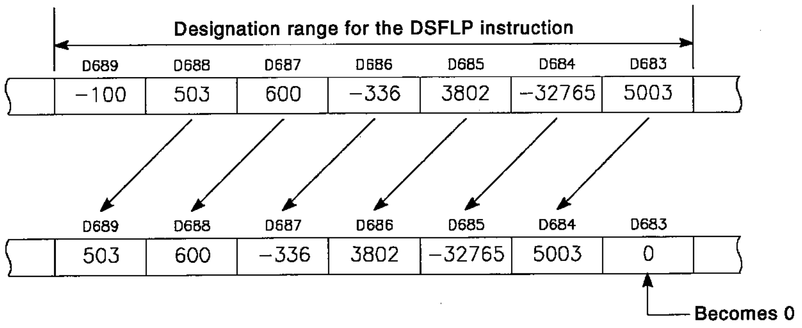
(2) The following program shifts the contents of D683 through D689 to the left when XB is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0B
1	DSFLP	D683 K7
4	END	

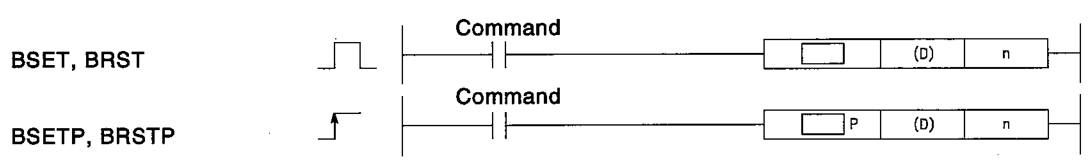


7.4 Bit Processing Instructions

7.4.1 Bit set and reset for word devices

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J⁚⁚\⁚⁚		Special Function Module U⁚⁚\G⁚⁚	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(D)	o							—	—
n	o							o	—

[Instruction Symbol] [Execution Condition]



Set Data

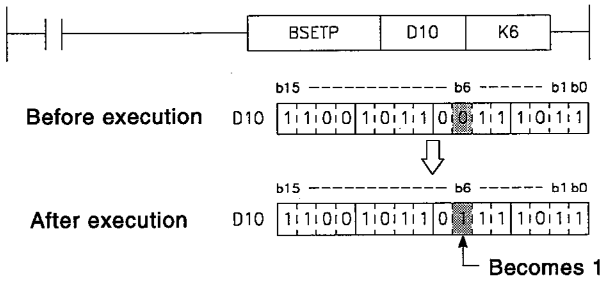
Set Data	Meaning	Data Type
(D)	Device number where bit set or reset will be conducted	BIN 16 bits
n	Bit number where bit set or reset will be conducted	

Functions

BSET

(1) Sets (sets "1" at) the nth bit in the word device designated at (D).

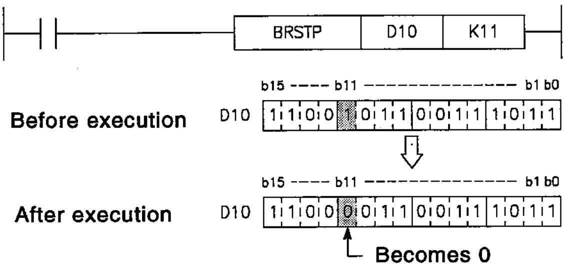
(2) The value for n can be set between 0 and 15.
If this value exceeds 15, the operation will be performed with the lower 4 bits of data.



BRST

(1) Resets the nth bit of a word device designated by (D) to 0.

(2) The value for n can be set between 0 and 15.
If this value exceeds 15, the operation will be performed with the lower 4 bits of data.



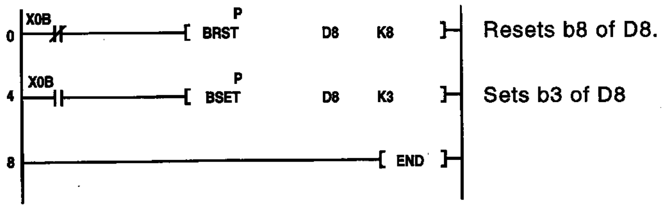
Operation Errors

(1) There are no operation errors associated with the BSET(P) or BRST(P) instructions.

Program Example

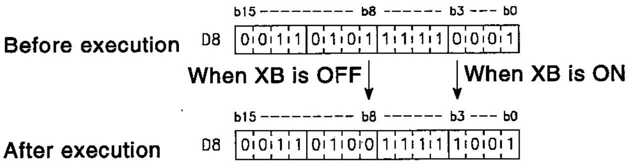
(1) The following program resets the 8th bit of D8 (b8) to 0 when XB is OFF, and sets the 3rd bit of D8 (b3) to 1 when XB is ON.

[Ladder Mode]



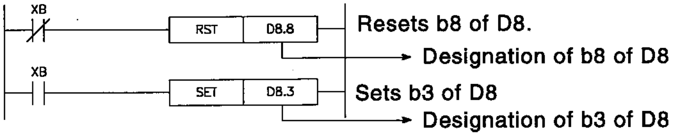
[List Mode]

Steps	Instruction	Device
0	LDI	X0B
1	BRSTP	D8
		K8
4	LD	X0B
5	BSETP	D8
		K3
8	END	



REMARK

Bit set or reset of word devices can also be conducted by bit designation of word devices. (See the QnACPU Programming Manual (Fundamentals) for information on bit designation of word devices.)
The processing of program example (1) would be conducted as shown below if bit designation of a word device had been used:

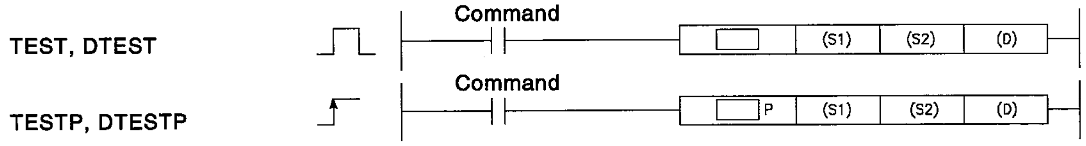


7.4.2 Bit tests

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o						o	—	—
(S2)	o						o	o	—
(D)	o						—	—	—

[Instruction Symbol] [Execution Condition]

□ indicates TSET/DTEST



Set Data

Set Data	Meaning	Data Type
(S1)	Number of device storing extracted bit data	Word
(S2)	Location of extracted bit data	Word
(D)	Number of bit device storing extracted bit data	Bit

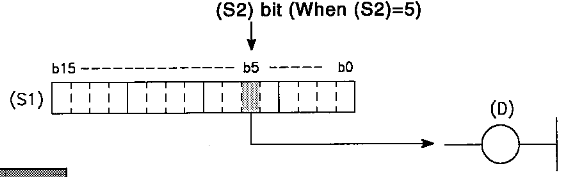
Functions

TEST

(1) Fetches bit data at the location designated by (S2) within the word device designated by (S1), and writes it to the bit device designated by (D).

(2) The bit device designated by (D) is OFF when the relevant bit is "0" and ON when it is "1".

(3) The position designated by (S2) indicates the position of an individual bit in a 1-word data block (0 to 15).

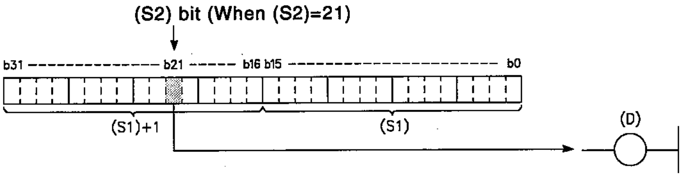


DTEST

(1) Fetches bit data at the location designated within the 2-word device designated by (S1), or (S1)+1, and writes it to the bit device designated by (D).

(2) The bit device designated by (D) is OFF when the relevant bit is "0" and ON when it is "1".

(3) The position designated by (S2) indicates the position of an individual bit in a 2-word data block (0 to 31).



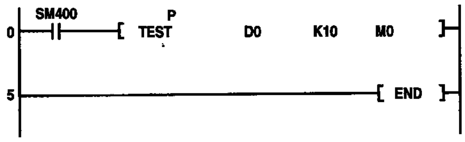
Operation Errors

(1) There are no operation errors associated with the TEST(P) or DTEST(P) instructions.

Program Example

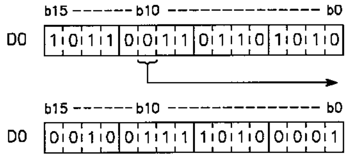
(1) The following program turns M0 ON or OFF based on the status of the 10th bit in the 1-word data block (D0).

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	TESTP	D0
		K10
5	END	M0

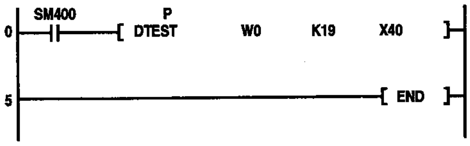


M0 turned OFF because b10 is "0"

M0 turned ON because b10 is "1"

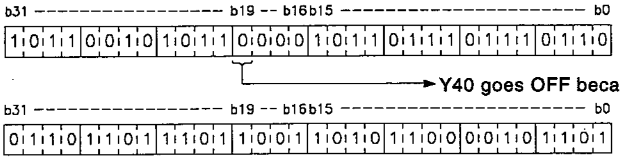
(2) The following program turns Y40 ON or OFF, depending on the status of the 19th bit of the 2-word data of W0 and W1.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DTESTP	W0
		K19
5	END	X40



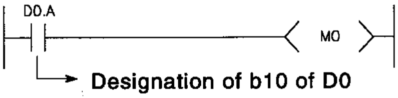
Y40 goes OFF because b19 is "0"

Y40 goes ON because b19 is "1"

REMARK

Programs using the TEST instruction can be rewritten as programs using bit designation of word devices.

If the program in example (1) were changed to use bit designation of a word device, it would appear as follows:

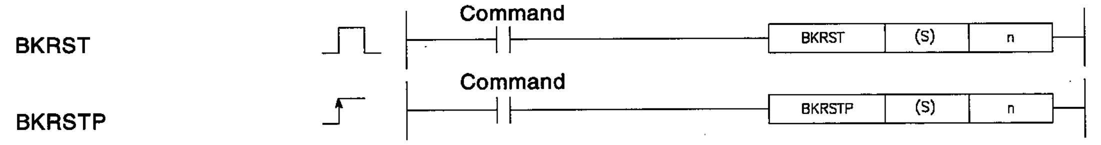


M0 goes ON or OFF depending on the ON/OFF status of b10 (D0.A) of D0.

7.4.3 Batch reset of bit devices

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o			—					—
n	o			o					—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices to reset	Bit
n	Number of devices which will be reset	BIN 16 bits

Functions

(1) Resets bit device n-points from the bit device designated by (S).

Device	Status
Annunciator (F)	· Turns device n-points from annunciator (F) number designated by (S) OFF. · Clears data at SD64 through SD79 of annunciator number turned OFF and compresses remaining data forward. · The number of annunciators stored from SD64 through SD79 is stored at SD63.
Timer (T) Counter (C)	Sets the current value n-points from timer (T) or counter (C) designated by (S) to 0, and turns coil contact OFF.
Bit devices other than the above	Turns OFF coil or contact n-points from the device designated by (S).

(2) If the designated device is OFF, the device status will not change.

Operation Errors

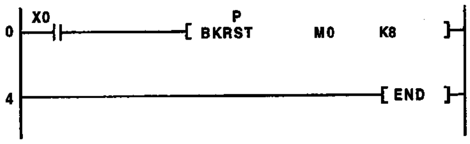
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Range n-points from (S) device exceeds relevant device.
(Error code:4101)

Program Example

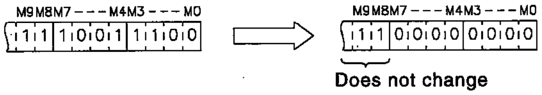
(1) The following program turns OFF devices from M0 through M7 when X0 goes ON.

[Ladder Mode]



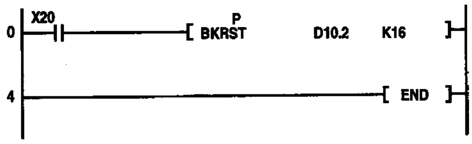
[List Mode]

Steps	Instruction	Device
0	LD	X0
1	BKRSTP	M0
		K8
4	END	



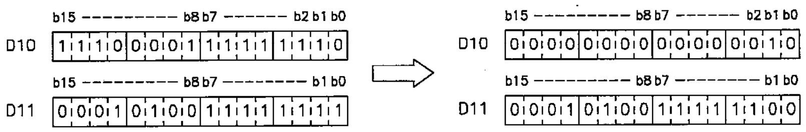
(2) The following program sets data from 2nd bit (b2) of D10 to 1st bit (b1) of D11 to 0 when X20 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	BKRSTP	D10.2
		K16
4	END	



7.5 Data Processing Instructions

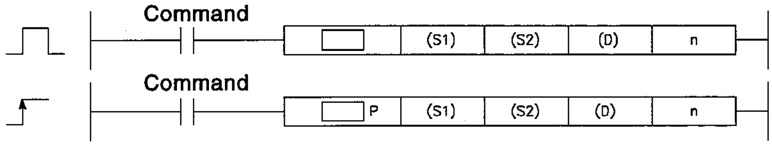
7.5.1 16-bit and 32-bit data searches

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
(S1)	o	o		o	o			o	—
(S2)	—	o		—	—			—	—
(D)	—	o		—	o			—	—
n	o	o		o	o			o	—

[Instruction Symbol] [Execution Condition]

SER, DSER

SERP, DSERP



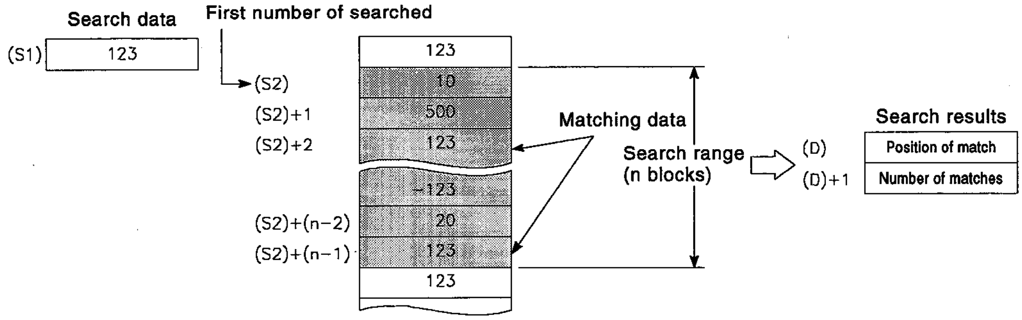
Set Data

Set Data	Meaning	Data Type
(S1)	Search data or the first device number of the devices where search data is being stored	Word
(S2)	Data which is object of search, or first number of the device where such data is being stored	
(D)	First device number of devices which will store search results	
n	Number of searches	

Functions

SER

(1) Takes the 16-bit data of the device designated at (S1) as the entry code to search for n number of blocks from the 16-bit data from the device designated at (S2).
The number of matches with the entry code is stored at the device designated by (D)+1, and the relative value of the number of points that the device where the first match was found is from (S2) is stored at the device designated by (D).



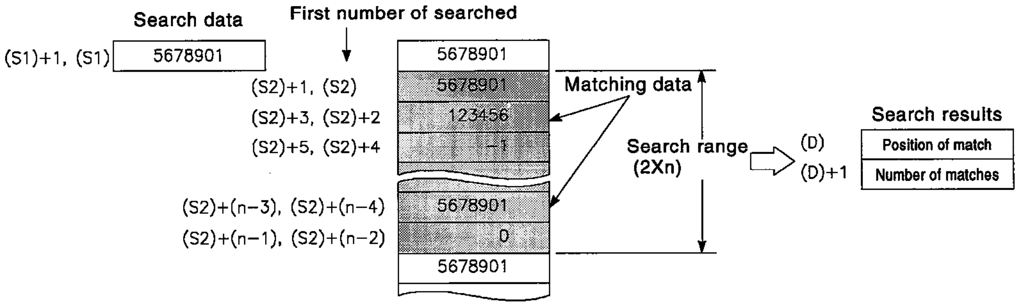
(2) No processing is conducted if n is 0 or a negative value.

(3) If no matches are found in the search, the devices designated at (D) and (D)+1 become "0".

(4) When the searched data is sorted in ascending order, processing speed can be increased by using a search with the binary search method when SM702* is ON.
If the searched data is not being sorted in ascending order, accurate search results cannot be obtained if SM702 is turned ON.

DSER

(1) Takes the 32-bit data of the device designated by (S1)+1, (S1) as the entry code for the search, and searches through n-points in 32-bit units from the device designated by (S2) (2 x n-points).
The number of matches with the entry code is stored at the device designated by (D)+1, and the relative value of the number of points that the device where the first match was found is from (S2) is stored at the device designated by (D).



(2) No processing is conducted if n is 0 or a negative value.

(3) If no matches are found in the search, the devices designated at (D) and (D)+1 become "0".

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The location n-points from the (S2) device exceeds the designated device range. (Error code: 4101)

(2) See Section 3.6 for information on errors during index qualification.

REMARK

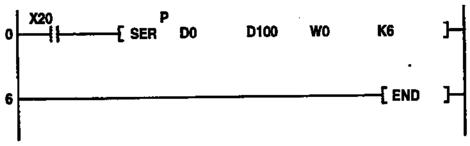
* : SM702 is the special relay for setting the search method.

- SM702 OFF: Consecutive search
  (Method whereby the data subject to the search is compared with the data being searched for, starting from the head of the data.)
- SM702 ON: Binary search method
  (Method whereby data at the designated location in the data subject to the search is compared with the data being searched for, to determine which side of this location the object of the search exists; the search range is then progressively narrowed by repeating this step, until the object of the search is found.)

Program Example

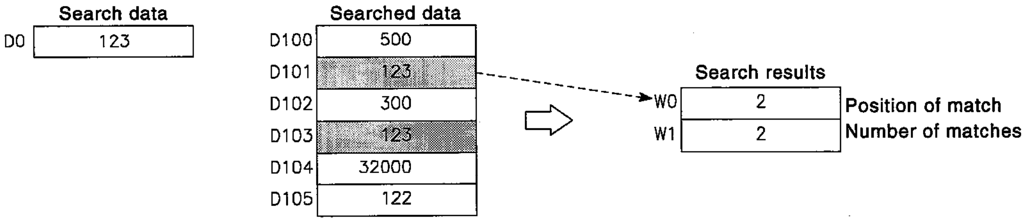
(1) The following program searches D100 through D105 for the contents of D0 when X20 is ON, and stores the search results at W0 and W1.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	SERP	D0 D100 W0 K6
6	END	



Search data

D0	123

Searched data

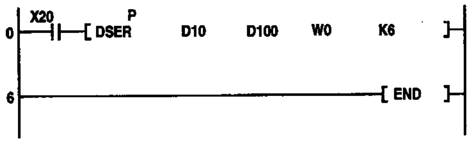
D100	500
D101	123
D102	300
D103	123
D104	32000
D105	122

Search results

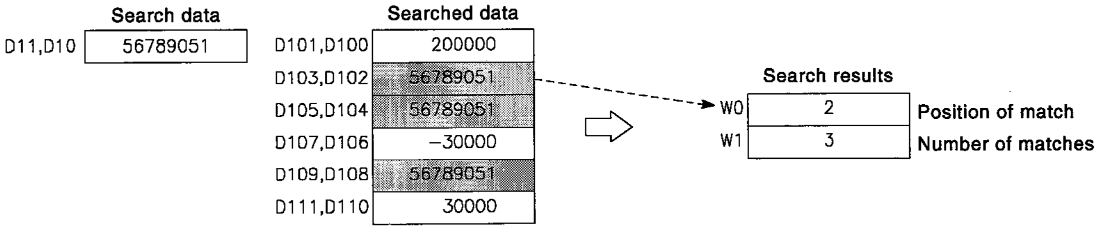
W0	2	Position of match
W1	2	Number of matches

(2) The following program searches D100 through D111 for the contents of D11and D10 when X20 is ON, and stores the search results at W0 and W1.

[Ladder Mode]



Steps	Instruction	Device
0	LD	X20
1	DSERP	D10 D100 W0 K6
6	END	



Search data

D11,D10	56789051

Searched data

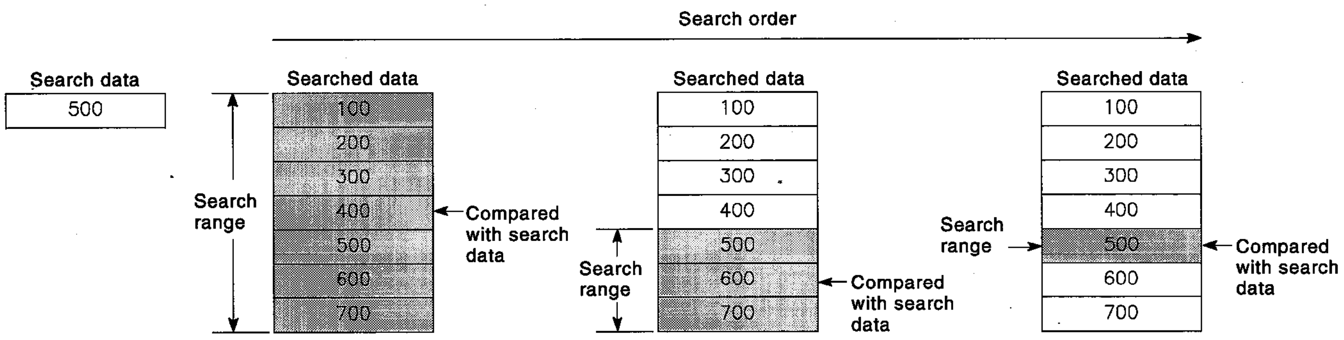
D101,D100	200000
D103,D102	56789051
D105,D104	56789051
D107,D106	−30000
D109,D108	56789051
D111,D110	30000

Search results

W0	2	Position of match
W1	3	Number of matches

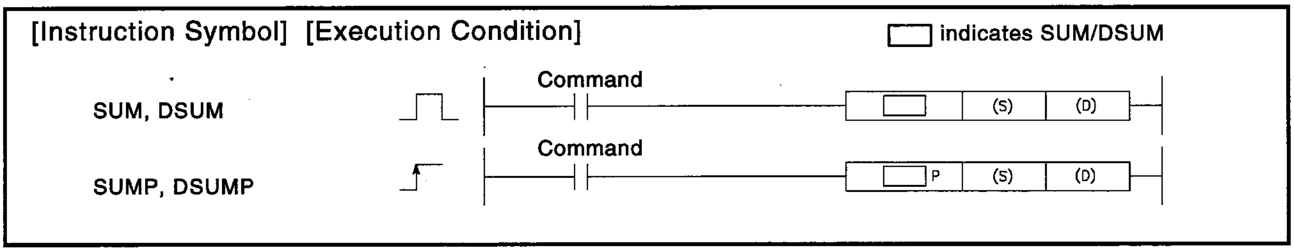
REMARK

The binary search method compares the data within the searched data with the search data, then repeats the comparison, with interim data that meets the proper conditions, to find the search data.



7.5.2 16-bit and 32 bit data checks

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—



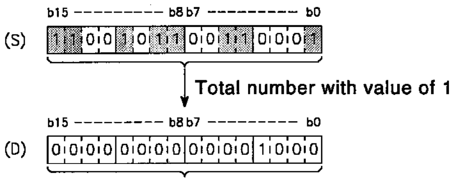
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices that will count total number of bits that are at 1	BIN 16/ 32 bits
(D)	First device number of devices that will store total number of bits	

Functions

SUM

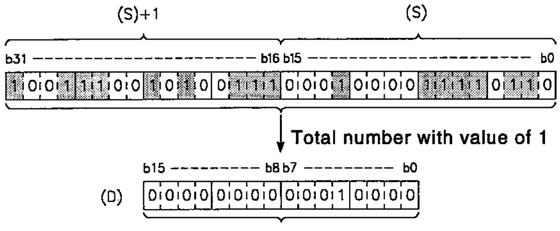
From the 16-bit data in the device designated by (S), stores the total number of bits that are at 1, in the device designated by (D).



Total number with value of 1 stored as BIN
(There are 8 instances in the example)

DSUM

From the 32-bit data in the device designated by (S), stores the total number of bits that are at 1, in the device designated by (D).



Total number with value of 1 stored as BIN
(There are 16 instances in the example)

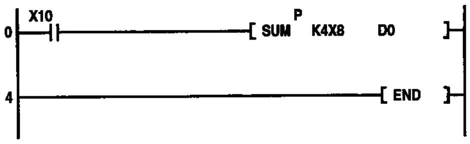
Operation Errors

(1) There are no operation errors associated with the SUM(P) or DSUM(P) instructions.

Program Example

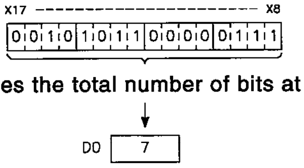
(1) The following program stores the number of bits from X8 through X17 which are ON when X10 is ON at D0.

[Ladder Mode]



[List Mode]

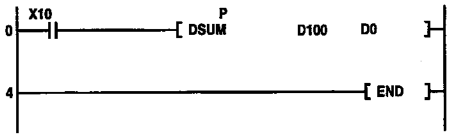
Steps	Instruction	Device
0	LD	X10
1	SUMP	K4X8 D0
4	END	



Stores the total number of bits at 1 in D0.

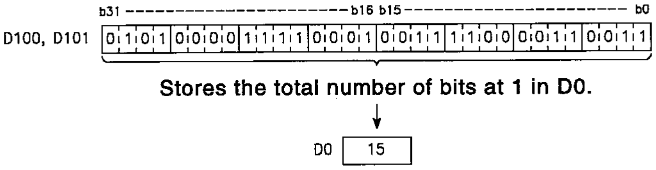
(2) The following program stores the number of bits from D100 and D101 which are ON when X10 is ON at D0.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X10
1	DSUMP	D100 D0
4	END	

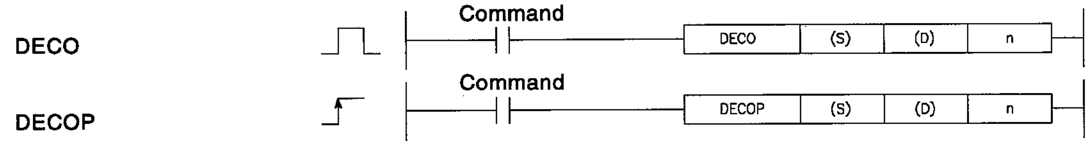


Stores the total number of bits at 1 in D0.

7.5.3 Decoding from 8 to 256 bits

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o			o				o	—
(D)	o			—				—	—
n	o			o				o	—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	Number of decoded data or device where decoded data is stored	BIN 16 bits
(D)	First device number of devices where decoding results are stored	Device name
n	Valid bit length (1 to 8)	BIN 16 bits

Functions

DECO

(1) Decodes the lower n-bits of the device designated by (S), and stores the results at a location 2^n-bits from the device designated by (D).

(2) The value of n can be designated between 1 and 8

(3) No processing is conducted if n=0, and there are no changes in the bits 2^n from the device designated at (D).

(4) Bit devices are treated as 1 bit, and word devices as 16 bits.

Operation Errors

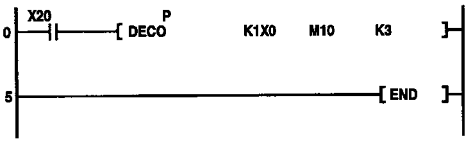
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The value of n is not in the 0 to 8 range. (Error code: 4100)
- The range 2^n bits from (D) exceeds the range of the relevant device. (Error code: 4101)

Program Example

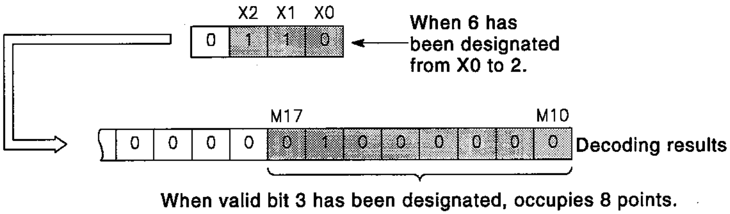
(1) The following program decodes the 3 bits from X0 and stores the results at M10 when X20 is ON.

[Ladder Mode]



[List Mode]

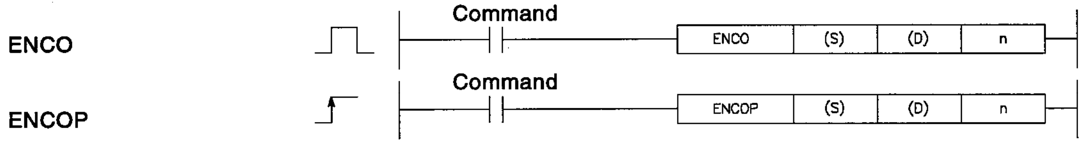
Steps	Instruction	Device
0	LD	X20
1	DECOP	K1X0 M10 K3
5	END	



7.5.4 Encoding from 256 to 8 bits

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o			—				o	—
(D)	o			o				—	—
n	o			o				o	—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	Encoded data, or first device number of devices where encoded data is being stored	BIN 16 bits
(D)	Number of device where encoding results will be stored	
n	Valid bit length (1 to 8)	

Functions

ENCO

(1) Encodes data 2^n bits from (S) and stores at (D).

(2) Valid settings for n are from 0 to 8.

(3) If n=0, there will be no operation, and the contents of (D) will not change.

(4) Bit devices are treated as 1 bit, and word devices as 16 bits.

(5) If more than 1 bit is at 1, processing will be conducted at the upper bit location.

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The value of n is not in the 0 to 8 range. (Error code: 4100)
- The value 2^n bits from (S) is outside the 0 to 8 range. (Error code: 4101)
- All data 2^n bits from (S) is "0". (Error code: 4100)

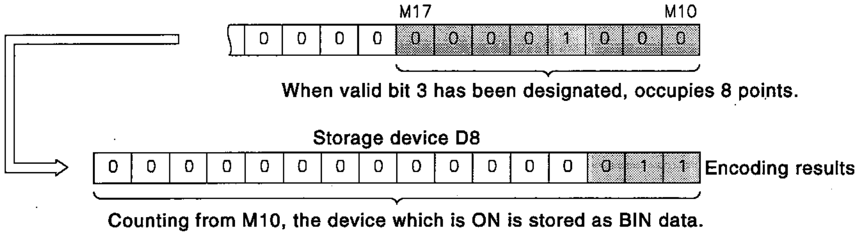
Program Example

(1) The following program encodes the 3 bits from M10 when X20 is ON, and stores the results at D8.

[Ladder Mode]

[List Mode]

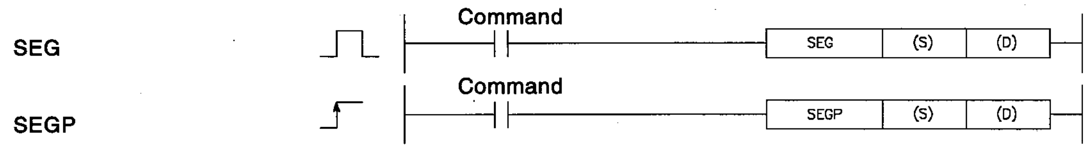
Steps	Instruction	Device
0	LD	X20
1	ENCOP	M10 D8 K3
5	END	



7.5.5 7-segment decode

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	Decoded data or first device number of devices where decoded data is being stored	BIN 16 bits
(D)	First device number of devices where decoding results are stored	

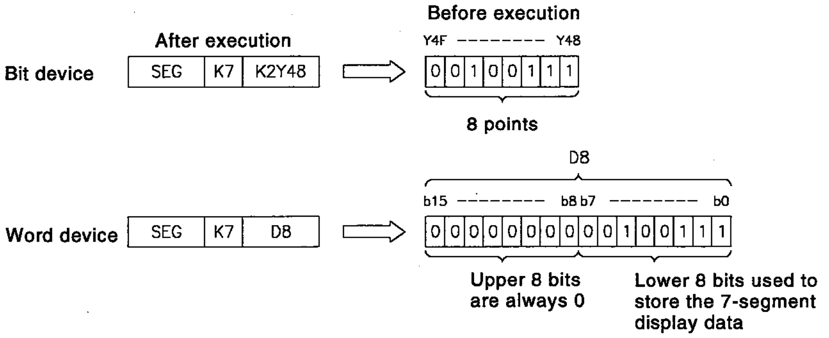
Functions

SUM

(1) Decodes the data from 0 to F designated by the lower 4 bits of (S) to 7-segment display data, and stores at (D).

(2) If (D) is a bit device, indicates the first number of the devices storing the 7-segment display data; if it is a word device, indicates the number of the device that is storing the data.

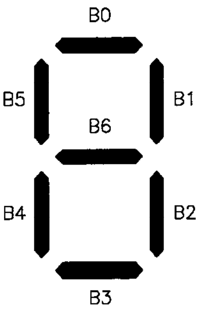
(3) Storage is done as follows for bit devices and word devices:



Operation Errors

(1) There are no operation errors associated with the SEG(P) instruction.

7-segment decode display

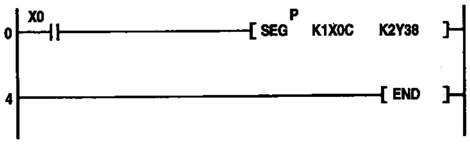
(S)		Configuration of 7 Segments	(D)								Display Data
Hexadecimal	Bit Pattern		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	b
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	d
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

Program Example

First number of bit device
Lowest bit of word device

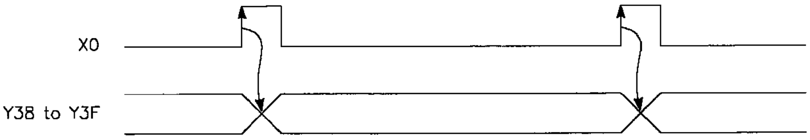
(1) The following program converts the data from XC to XF when X0 is ON to 7-segment display data and outputs it to Y38 through Y3F.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	SEGP	K1X0C K2Y38
4	END	



* The data from Y38 to Y3F will not change until the next data is output.

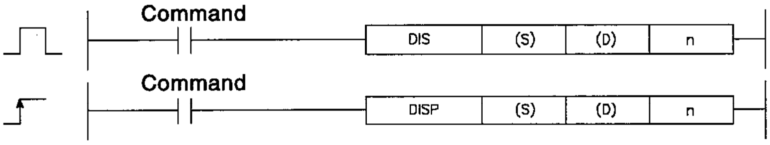
7.5.6 4-bit groupings of 16-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		o					—
(D)	—	o		—					—
n	o	o		o					—

[Instruction Symbol] [Execution Condition]

DIS

DISP



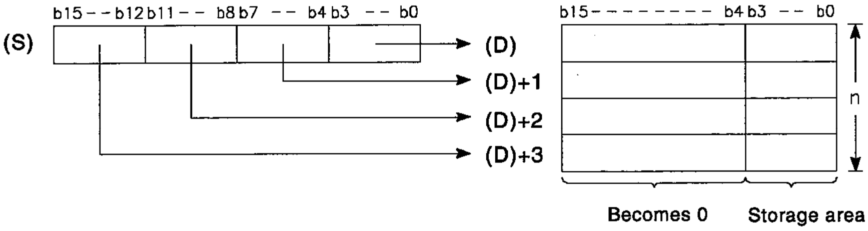
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where data to be dissociated is stored	BIN 16 bits
(D)	First device number of devices where dissociated data is stored	
n	Number of dissociations (1 through 4) : No processing if 0	

Functions

DIS

(1) Stores the lower n-digits (1 digit is 4 bits) of the 16-bit data designated by (S) at the lower 4 bits n-points from the device designated by (D).



(2) The upper 12 bits n-points from the device designated by (D) become 0.

(3) The value of n can be designated between 1 and 4.

(4) If n=0, there will be no operation, and the contents n-points from (D) will not change.

Operation Errors

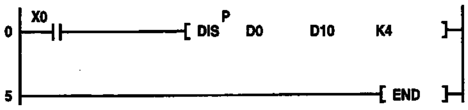
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range n-points from (D) exceeds the relevant device. (Error code: 4101)
- The value of n is outside the 0 to 4 range. (Error code: 4100)

Program Example

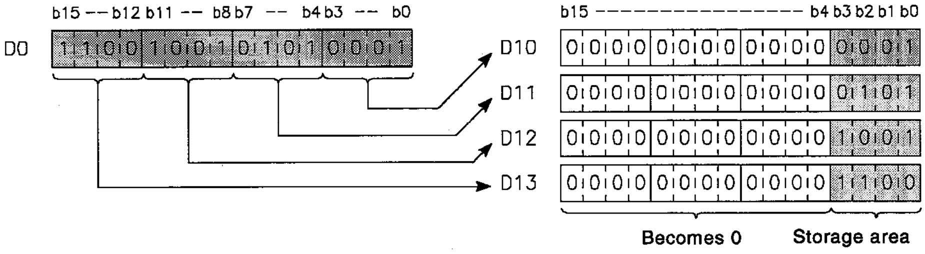
(1) The following program dissociates the 16-bit data from D0 into 4-bit groups, and stores from D10 to 13 when X0 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	DISP	D0
		D10
		K4
5	END	



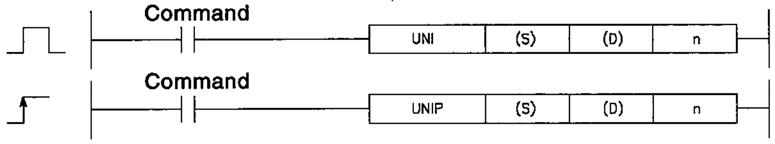
7.5.7 4-bit linking of 16-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		—				—	—
(D)	—	o		o				—	—
n	o	o		o				o	—

[Instruction Symbol] [Execution Condition]

UNI

UNIP



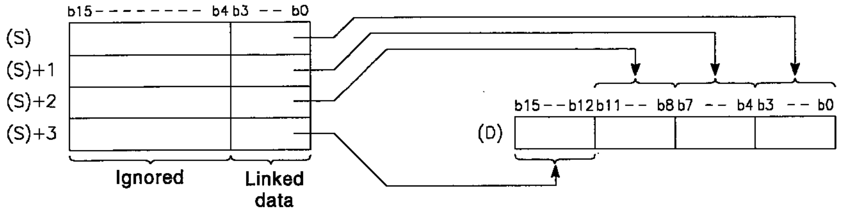
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where data to be linked is being stored	BIN 16 bits
(D)	First device number of devices where linked data will be stored	
n	Number of links (1 through 4) : No processing if 0	

Functions

UNI

(1) Links lower 4 bits of 16-bit data n-points from device designated by (S) to 16-bit device designated by (D).



(2) The bits of the upper (4-n) digits of the device designated by (D) become 0.

(3) The value of n can be designated between 1 and 4.

(4) If n=0, there will be no operation, and the contents of device (D) will not change.

Operation Errors

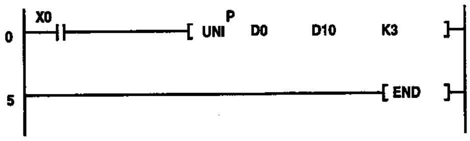
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range n-points from (S) exceeds the relevant device. (Error code: 4101)
- The value of n is outside the 0 to 4 range. (Error code: 4100)

Program Example

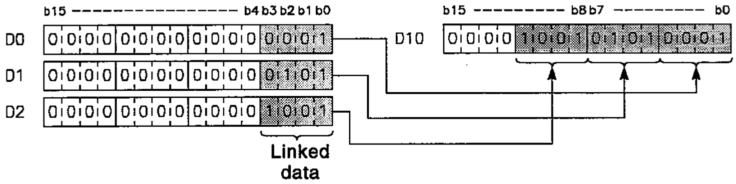
(1) The following program links the lower 4 bits of D0 through D2 when X0 is ON, and stores them at D10.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	UNIP	D0
		D10
5	END	K3



7.5.8 Dissociation or linking of random data

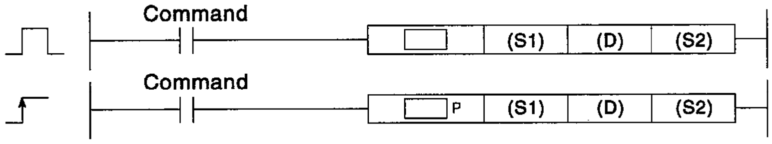
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—					
(D)	—	o		—					
(S2)	—	o		—					

[Instruction Symbol] [Execution Condition]

□ indicates NDIS/NUNI

NDIS, NUNI

NDISP, NUNIP



Set Data

Set Data	Meaning	Data Type
(S1)	First device number of devices storing data to be dissociated or linked	BIN 16 bits
(D)	First device number of devices to store data which will be dissociated or linked	
(S2)	First device number of devices to store the unit by which dissociation or linking will be done	

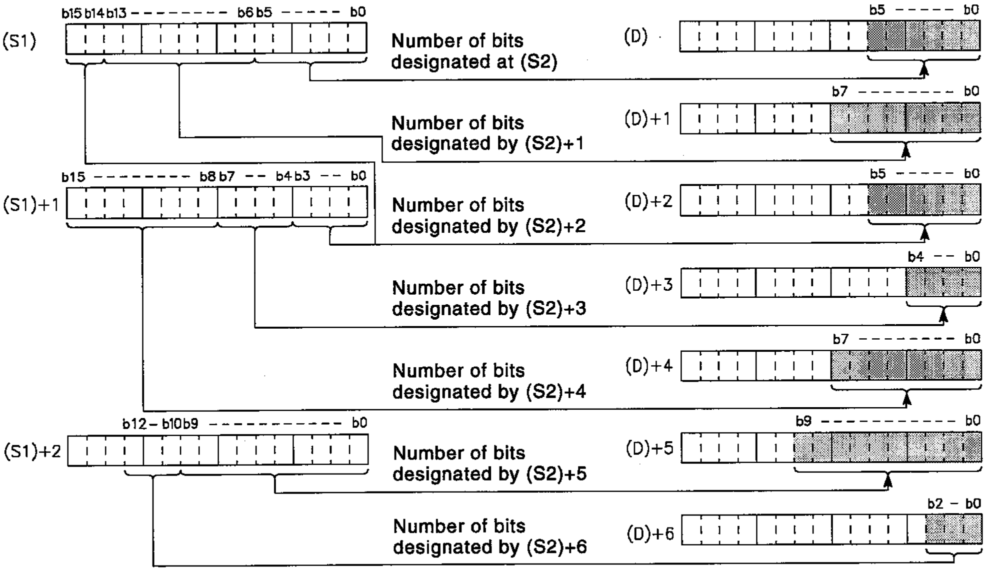
Functions

NDIS

(1) Separates data stored in device numbers starting from that designated at (S1) into the number of individual bits designated at (S2), and stores this data in device numbers starting from that designated at (D).

Specification of number of separated bits

(S2)	6
(S2)+1	8
(S2)+2	6
(S2)+3	4
(S2)+4	8
(S2)+5	10
(S2)+6	3
(S2)+7	0 ← Setting completed

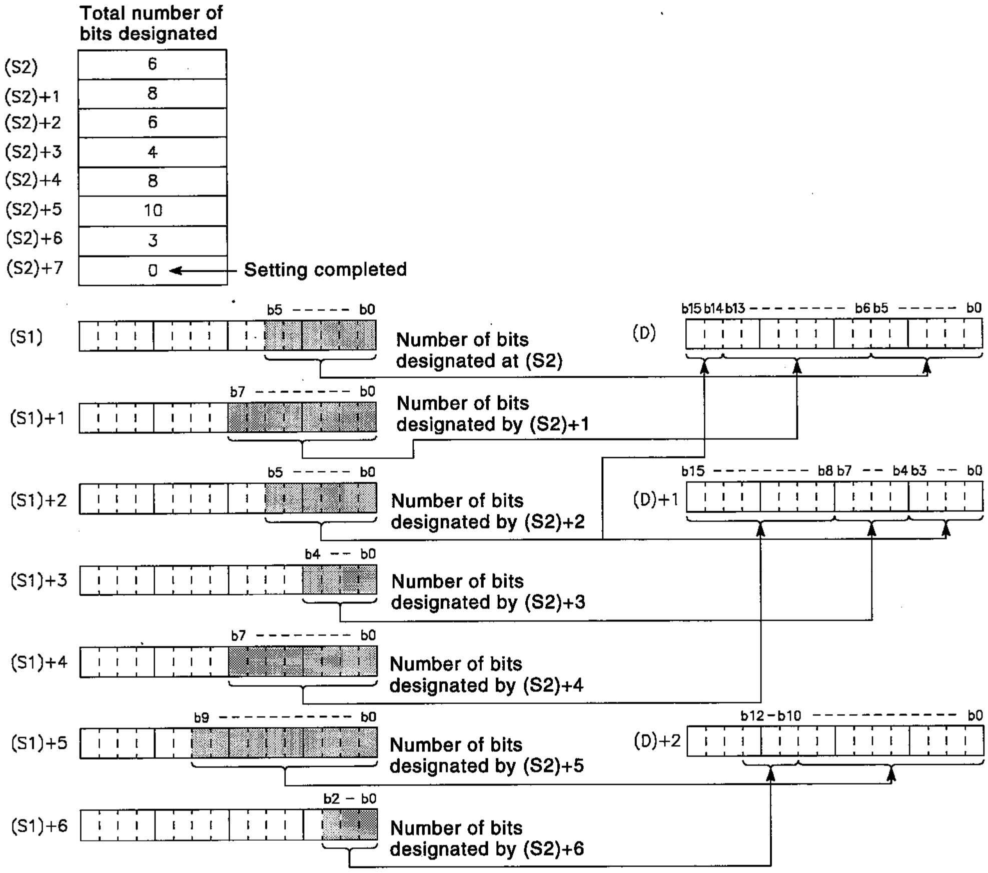


(2) The number of dissociated bits designated at (S2) can be designated within a range of from 1 to 16 bits.

(3) Bits from the device number designated at (S2) to the device number where "0" is stored are processed as dissociated bits.

NUNI

(1) Links individual bits of data stored following the device number designated by (S1) in the number of bits specified by (S2), and stores them following the device number designated by (D).



(2) The number of bits to be linked as designated by (S2) can be within a range of from 1 to 16.

(3) Processing will be performed on the number of bits to be linked from the device number designated by (S2) to the device number storing "0".

Operation Errors

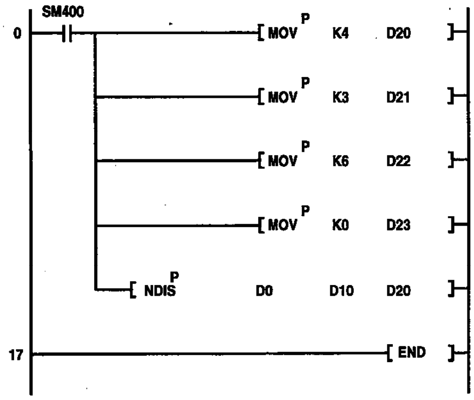
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The number of bits to be dissociated or linked as specified by (S2), or the device use range specified by (S1) or (D) exceeds the final device number of their respective devices. (Error code: 4101)
- The number of bits for dissociation or linking specified by (S2) has not been set within the range of from 1 to 16 bits. (Error code: 4100)

Program Example

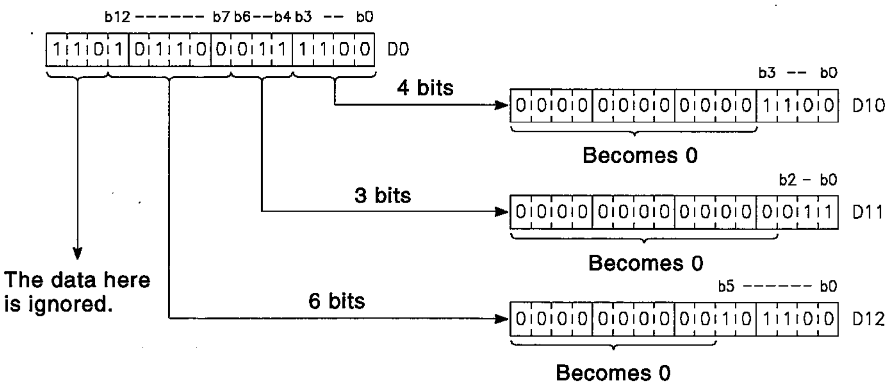
(1) The following program separates data of 4, 3, and 6 bits respectively from the lower bits of D0, and stores them from D10 to D12.

[Ladder Mode]



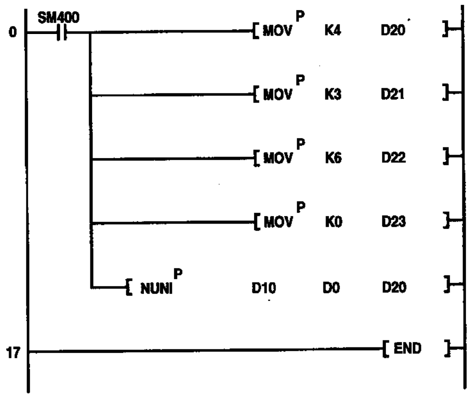
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	MOVP	K4 D20
4	MOVP	K3 D21
7	MOVP	K6 D22
10	MOVP	K0 D23
13	NDISP	D0 D10 D20
17	END	



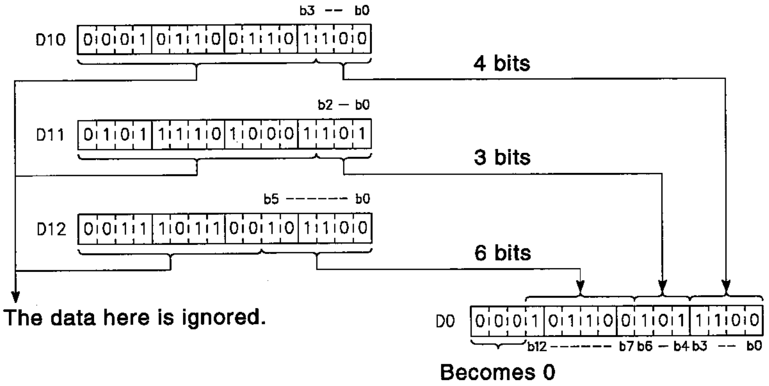
(2) The following program links the lower 4 bits of data from D10, the lower 3 bits of data from D11, and the lower 6 bits of data from D12, and stores at D0.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	MOVP	K4 D20
4	MOVP	K3 D21
7	MOVP	K6 D22
10	MOVP	K0 D23
13	NUNIP	D10 D0 D20
17	END	

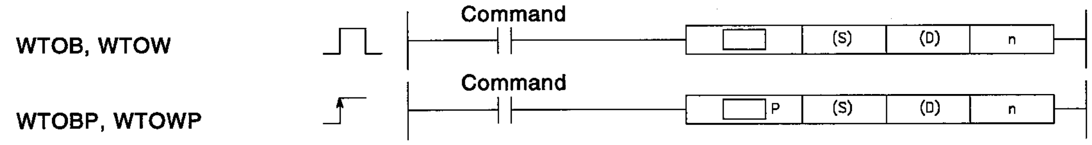


7.5.9 Data dissociation and linking in byte units

<table>
<tr><th rowspan="3">Set Data</th><th colspan="9">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System, User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th rowspan="2">Constant K, H</th><th rowspan="2">Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th></tr>
<tr><td>(S)</td><td>—</td><td colspan="2">o</td><td colspan="5">—</td><td>—</td></tr>
<tr><td>(D)</td><td>—</td><td colspan="2">o</td><td colspan="5">—</td><td>—</td></tr>
<tr><td>n</td><td>o</td><td colspan="2">o</td><td colspan="5">o</td><td>—</td></tr>
</table>

[Instruction Symbol] [Execution Condition]

□ indicates WTOB/WTOW



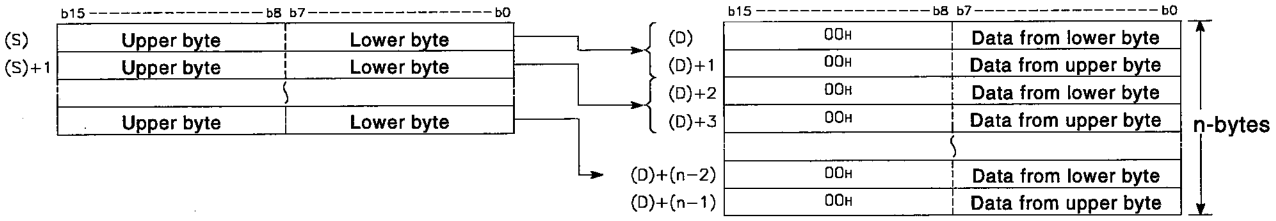
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where the data to be dissociated or linked in byte units is being stored	BIN 16 bits
(D)	First device number of devices where the results of the dissociation or linking in byte units will be stored	
n	Number of bytes to be dissociated or linked	

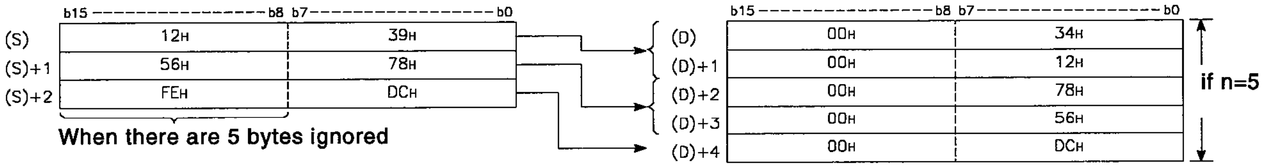
Functions

WTOB

(1) Dissociates n-bytes of the 16-bit data stored following the device number designated by (S), and stores them following the device designated by (D).



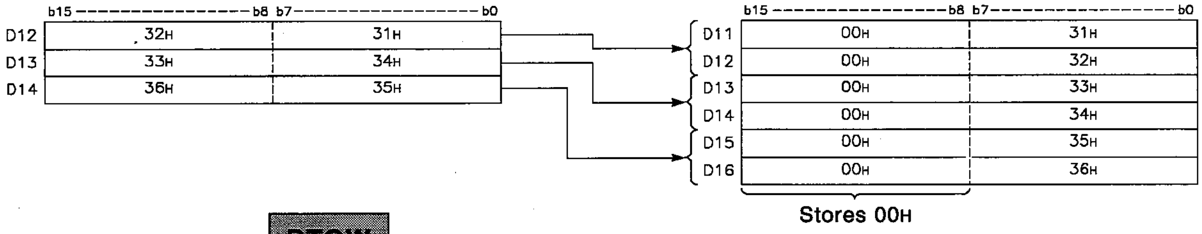
For example, if n=5, data through the lower 8 bits of (S) to ((S)+2) would be stored from ((D) through (D)+4).



(2) Setting the number of bytes with n automatically determines the range of the 16-bit data designated by (S) and the range of the devices to store the byte data designated by (D).

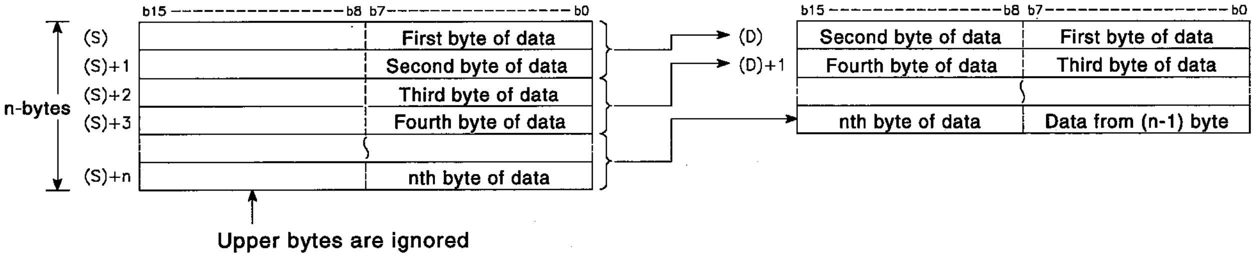
(3) No processing will be conducted when the number of bytes designated by n is "0".

(4) The "00H" code will automatically be stored at the upper 8 bits of the byte storage device designated by (D).

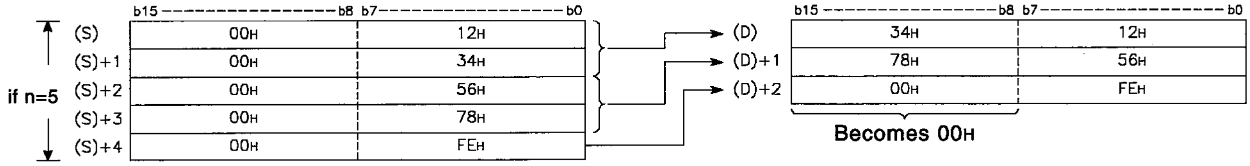


BTOW

(1) Links in word units the lower 8 bits of the 16-bit data n-words following the device number designated by (S), and stores following the device number designated by (D).
The upper 8 bytes of the data n-words following the device number designated by (S) are ignored.
Further, if n is an odd number, 0 is stored at the upper 8 bits of the device where the nth byte of data is stored.



For example, if n=5, the lower 8 bits of data from (S) to ((S)+4) are linked and stored at (D) to ((D)+2).

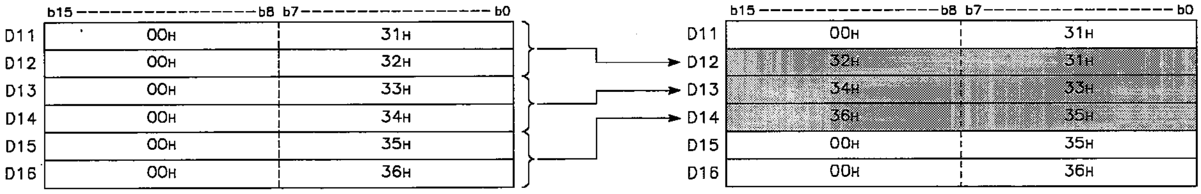


(2) Setting the number of bytes with n automatically determines the range of the byte data designated by (S) and the range of the devices to store the byte data designated by (D).

(3) No processing will be conducted when the number of bytes designated by n is "0".

(4) The upper 8 bits of the byte storage device designated by (S) are ignored, and the lower 8 bits are used.

(5) Accurate processing will be conducted even in cases where there is an overlap between the device range ((S) to (S)+n) that will store the data to be linked to and the device range ((D) to (D)+n) that will store the data to be linked.



	b15 — b8	b7 — b0
D11	00H	31H
D12	00H	32H
D13	00H	33H
D14	00H	34H
D15	00H	35H
D16	00H	36H

	b15 — b8	b7 — b0
D11	00H	31H
D12	32H	31H
D13	34H	33H
D14	36H	35H
D15	00H	35H
D16	00H	36H

For example, the following will take place in a case where the lower 8 bits of D11 through D16 are to be stored at D12 through D14:

Operation Errors

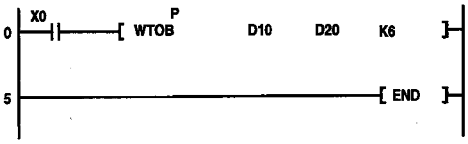
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range of the number of bytes designated by n following the device number designated by (S) exceeds the relevant device range. (Error code: 4101)
- The range of the number of bytes designated by n following the device number designated by (D) exceeds the relevant device range. (Error code: 4101)

Program Example

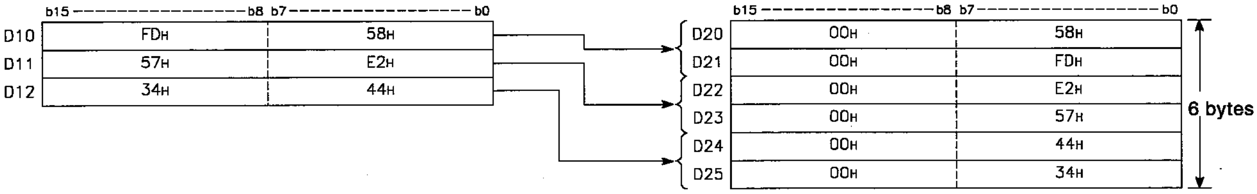
(1) The following program dissociates the data at D10 through D12 in byte units and stores from D20 through D25 when X0 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	WTOBP	D10 D20 K6
5	END	

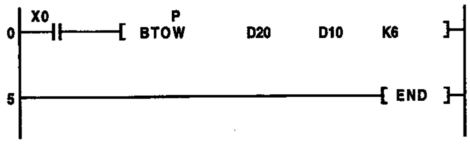


	b15 — b8	b7 — b0
D10	FDH	58H
D11	57H	E2H
D12	34H	44H

	b15 — b8	b7 — b0
D20	00H	58H
D21	00H	FDH
D22	00H	E2H
D23	00H	57H
D24	00H	44H
D25	00H	34H

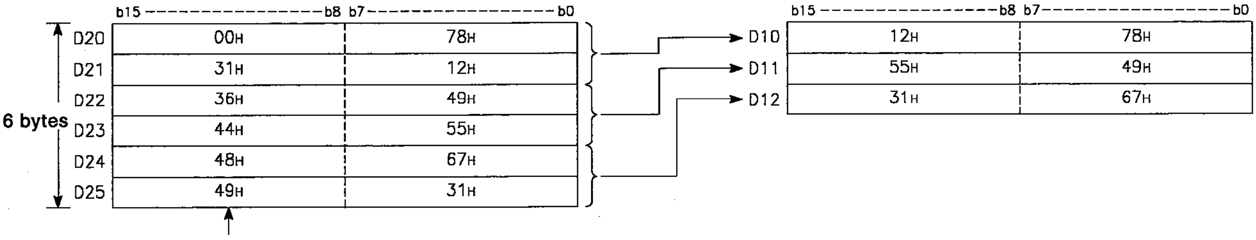
(2) The following program links the lower 8 bits of data from D20 through D25 and stores the result at D10 through D12 when X0 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	BTOWP	D20 D10 K6
5	END	



	b15 — b8	b7 — b0
D20	00H	78H
D21	31H	12H
D22	36H	49H
D23	44H	55H
D24	48H	67H
D25	49H	31H

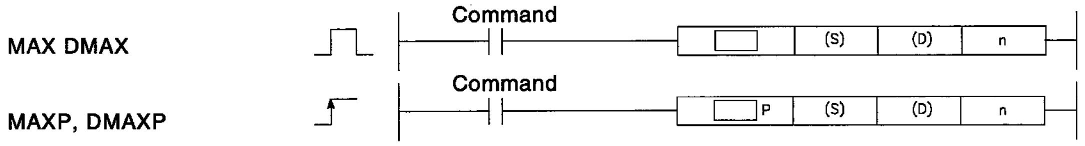
	b15 — b8	b7 — b0
D10	12H	78H
D11	55H	49H
D12	31H	67H

Upper bytes are ignored

7.5.10 Maximum value search for 16- and 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—					—
(D)	—	o		—					—
n	o	o		o					—

[Instruction Symbol] [Execution Condition] □ indicates MAX/DMAX



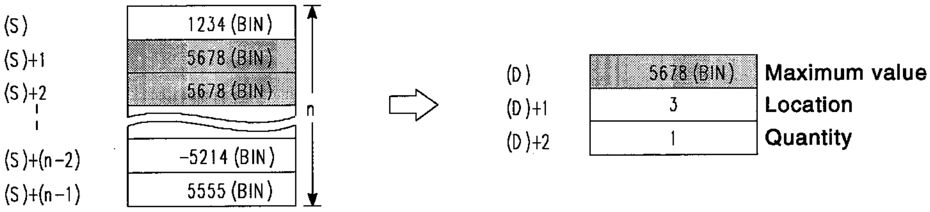
Set Data

Set Data	Meaning	Data Type
(S)	First number of devices for maximum value search	BIN 16/32 bits
(D)	First number of devices where results of maximum value search are stored	
n	Number of data blocks searched	BIN 16 bits

Functions

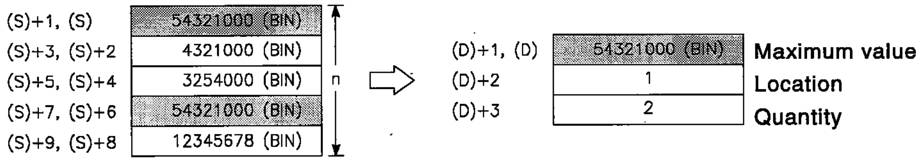
MAX

(1) Searches for the maximum value from the n blocks of 16-bit data from the device designated at (S), and stores this value in the device designated at (D).
Searches from the device designated at (S) to the device number where the first maximum value discovered is stored, then stores how many points this device number is from the (S) device in (D)+1, and stores the number of maximum values in (D)+2.



DMAX

(2) Searches for the maximum value from the n blocks of 32-bit data from the device designated by (S), and stores this value in the device designated by (D) and (D)+1.
Searches from the device designated by (S) to the device number where the first maximum value discovered is stored, then stores how many points this device number is from the (S) device in (D)+2, and stores the number of maximum values in (D)+3.



(S)+1, (S)	54321000 (BIN)
(S)+3, (S)+2	4321000 (BIN)
(S)+5, (S)+4	3254000 (BIN)
(S)+7, (S)+6	54321000 (BIN)
(S)+9, (S)+8	12345678 (BIN)

(D)+1, (D)	54321000 (BIN)	Maximum value
(D)+2	1	Location
(D)+3	2	Quantity

Operation Errors

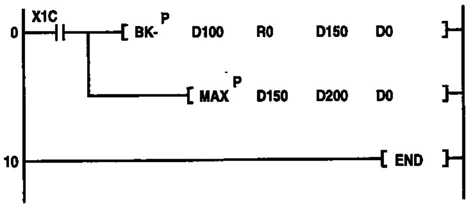
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range n points from the (S) device exceeds the relevant device. (Error code: 4101)

Program Example

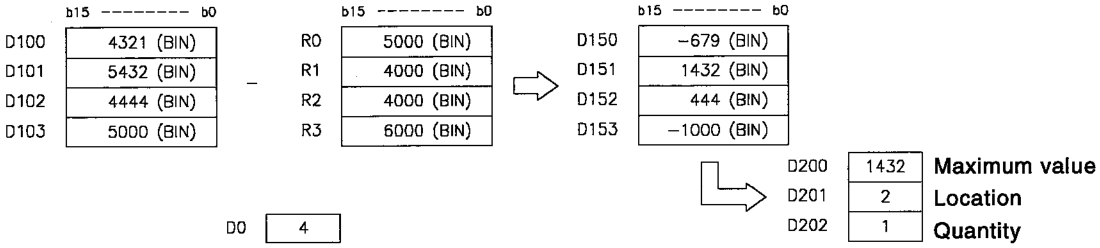
(1) The following program example when X1C goes ON subtracts the data in the number of devices designated by the value stored in D0, starting from R0, from the data in the number of devices designated by the value stored in D0, starting from D100, searches for the maximum value from the result of this calculation, and stores it from D200 to D202.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	BK-P	D100 R0 D150 D0
6	MAXP	D150 D200 D0
10	END	



	b15 -------- b0
D100	4321 (BIN)
D101	5432 (BIN)
D102	4444 (BIN)
D103	5000 (BIN)

	b15 -------- b0
R0	5000 (BIN)
R1	4000 (BIN)
R2	4000 (BIN)
R3	6000 (BIN)

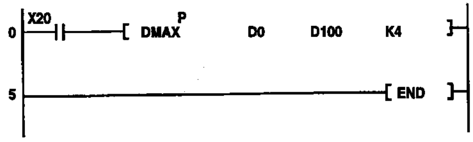
	b15 -------- b0
D150	−679 (BIN)
D151	1432 (BIN)
D152	444 (BIN)
D153	−1000 (BIN)

D200	1432	Maximum value
D201	2	Location
D202	1	Quantity

D0 4

(2) The following program searches for the maximum value from the 32-bit data at D0 through D7, and stores it at D100 through D103 when X20 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	DMAXP	D0 D100 K4
5	END	

D1, D0	3786213 (BIN)
D3, D2	-3235 (BIN)
D5, D4	8744740 (BIN)
D7, D5	7141821 (BIN)

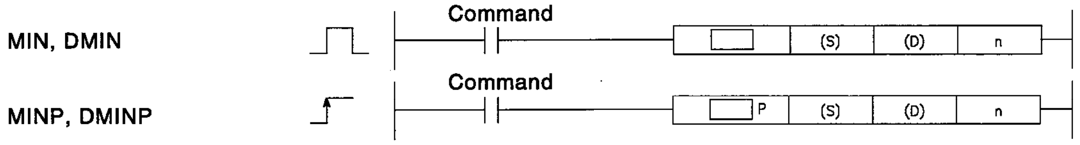
⇨

D100, 101	8744740
D102	2
D103	2

7.5.11 Minimum value search for 16 and 32 bits data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—					—
(D)	—	o		—					—
n	o	o		o					—

[Instruction Symbol] [Execution Condition] □ indicates MIN/DMIN



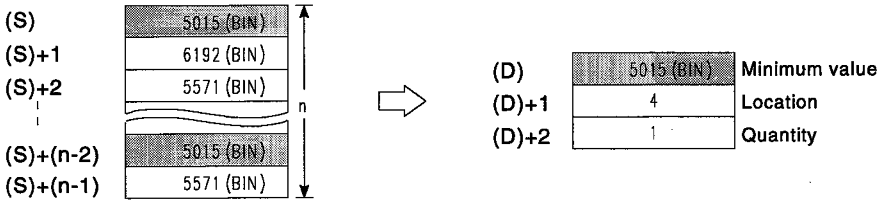
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices subject to search for minimum value	BIN 16/32 bits
(D)	First device number of devices that will store the results of the minimum value search	
n	Number of data blocks searched	BIN 16 bits

Functions

MIN

(1) Searches for the minimum value from the n blocks of 16-bit BIN data from the device designated by (S), and stores this value in the device designated by (D).
Searches from the device designated by (S), and the device number where the first minimum value discovered is stored then stores the x-number from (S) in (D)+1, and stores the number of minimum values in (D)+2.



DMIN

(2) Searches for the minimum value in the n blocks of 32-bit BIN data from the device designated by (S), and stores this value in the device designated by (D) and (D)+1.
Searches from the device designated by (S) to the device number where the first minimum value discovered is stored, then stores how many points this device number is from the (S) device in (D)+2, and stores the number of minimum values in (D)+3.

(S)	22342001 (BIN)
(S)+2	37282010 (BIN)
(S)+4	22342001 (BIN)
(S)+6	59872019 (BIN)

⇨

(D), (D)+1	22342001 (BIN)	Minimum value
(D)+2	1	Location
(D)+3	2	Quantity

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

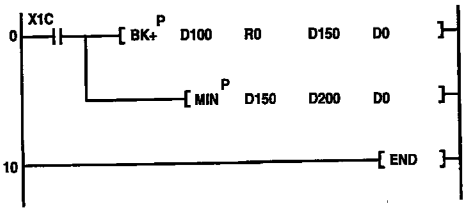
- The range n points from the (S) device exceeds the relevant device. (Error code: 4101)

(2) See Section 3.6 for information regarding errors during index qualification.

Program Example

(1) The following program, when X1C goes ON, adds the number of points of data designated by the value stored in D0, starting from D100, to the number of points of data designated by D0 and starting from R0, searches the result for the minimum value, and stores it from D200 through D202.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	BK+P	D100 R0 D150 D0
6	MINP	D150 D200 D0
10	END	

	b15 -------- b0
D100	5542 (BIN)
D101	5857 (BIN)
D102	4590 (BIN)
D103	4450 (BIN)

+

	b15 -------- b0
R0	5500 (BIN)
R1	4000 (BIN)
R2	4500 (BIN)
R3	6000 (BIN)

⇨

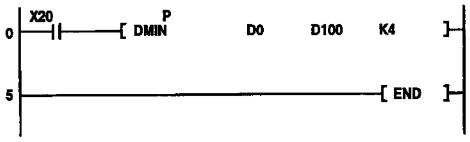
	b15 -------- b0
D150	11042 (BIN)
D151	9857 (BIN)
D152	9090 (BIN)
D153	10450 (BIN)

D200	9090	Minimum value
D201	3	Location
D202	1	Quantity

D0 4

(2) The following program, when X20 goes ON, searches for the minimum value from the 32-bit data contained from D0 through D7, and stores it from D100 through D103.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	DMINP	D0 D100 K4
5	END	

D1, D0	57020175 (BIN)
D3, D2	2070166 (BIN)
D5, D4	3596045 (BIN)
D7, D6	−69386 (BIN)

⇨

D101, 100	−69386
D102	1
D103	2

7.5.12 BIN 16 and 32 bits data sort operations

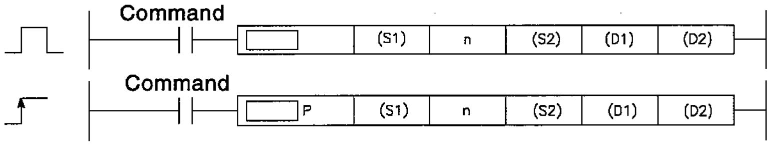
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—					—
n	o	o		—					—
(S2)	o	o		o					—
(D1)	o	—		o					—
(D2)	—	o		—					—

[Instruction Symbol] [Execution Condition]

□ indicates SORT/DSORT

SORT, DSORT

SORTP, DSORTP



Set Data

Set Data	Meaning	Data Type
(S1)	First device number of table which will do the sort	BIN 16/32 bits
n	Number of data blocks to sort	BIN 16 bits
(S2)	Number of data blocks for comparison in 1 sort execution	BIN 16 bits
(D1)	Number of bit device to be forced ON when sort has been completed	Bit
(D2)	Device for use by system	BIN 16 bits

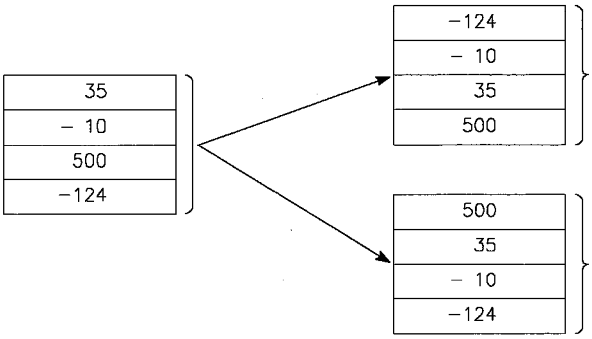
Functions

SORT

(1) Sorts (rearranges data) BIN 16-bit data n points from (S1) in ascending or descending order.

(2) Sort order is designated by the ON/OFF status of SM703:

- When SM703 is OFF : Ascending order sort
- When SM703 is ON : Descending order sort



(3) Several scans are required for sorts performed by the SORT instruction.
The number of scans executed until completion is the value obtained by dividing the maximum number of times executed until the completion of the sort by the number of data blocks compared at one execution designated by (S2). (Decimal fractions are rounded up.)
When the value of (S2) is increased, the number of scans until completion of the sort is reduced, but the amount of time per scan is lengthened.

(4) The maximum number of executions until completion of the sort should be calculated according to the following equation:
Maximum number of executions until completion
= (n)∗(n-1)÷2÷(S2) [times executed]

For example, if n=10 and (S2)=1, this would require 10∗(10-1)÷2÷1=45 scans for completion.
If S2 were to equal 2 in the above example, then the number of scans until completion of the sort would be 45÷2=22.5, becoming 23 because the decimal fraction is rounded up.

(5) The device designated by (D1) (the completion device) is turned OFF by the execution of the SORT instruction, and ON when the sort is completed.
Because the device designated by (D1) is maintained in the ON state after the completion of the sort, the user must turn it OFF if required.

(6) The 2 points from the device designated by (D2) are used by the system during the execution of the SORT instruction.
These 2 points from the device designated by (D2) should therefore not be used by the user.

(7) If the value of n is changed during the execution of the SORT instruction, the sort will be conducted in accordance with the number of sort data blocks after the change.

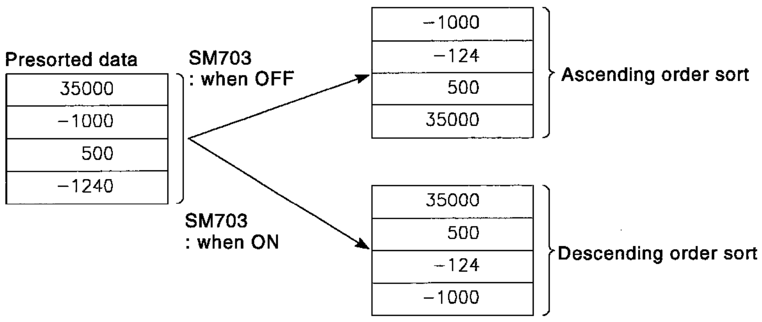
(8) If the command is turned OFF during the execution of a sort, the sort operation will be suspended.
It will be resumed from the beginning when the command is turned back ON.

DSORT

(1) Sorts (rearranges data) BIN 32-bit data n points from (S1) in ascending or descending order.

(2) Sort order is designated by the ON/OFF status of SM703:

- When SM703 is OFF : Ascending order sort
- When SM703 is ON : Descending order sort



(3) Several scans are required for sorts performed by the DSORT instruction.
The number of scans executed until completion is the value obtained by dividing the maximum number of times executed until the completion of the sort by the number of data blocks compared at one execution designated by (S2). (Decimal fractions are rounded up.)
When the value of (S2) is increased, the number of scans until completion of the sort is reduced, but the amount of time per scan is lengthened.

(4) The maximum number of executions until completion of the sort should be calculated according to the following equation:
Maximum number of executions until completion
= (n)∗(n-1)÷2÷(S2) [times executed]

For example, if n=10 and (S2)=1, this would require 10∗(10-1)÷2÷1=45 scans for completion.
If S2 were to equal 2 in the above example, then the number of scans until completion of the sort would be 45÷2=22.5, becoming 23 because the decimal fraction is rounded up.

(5) The device designated by (D1) (the completion device) is turned OFF by the execution of the SORT instruction, and ON when the sort is completed.
Because the device designated by (D1) is maintained in the ON state after the completion of the sort, the user must turn it OFF if required.

(6) The 2 points from the device designated by (D2) are used by the system during the execution of a DSORT instruction.
These 2 points from the device designated by (D2) should therefore not be used by the user.

(7) If the value of n is changed during the execution of the SORT instruction, the sort will be conducted in accordance with the number of sort data blocks after the change.

(8) If the command is turned OFF during the execution of a sort, the sort operation will be suspended.
It will be resumed from the beginning when the command is turned back ON.

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range n/2∗n points from (S1) exceeds the relevant device range. (Error code: 4101)
- (S2) is 0 or is a negative value (Error code: 4101)
- The value of (D2)+0 is greater than the value of n (Error code: 4101)
- The value of (D2)+1 is greater than the value of (D2)+0 (Error code: 4101)

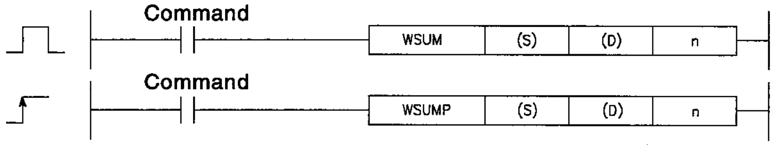
7.5.13 Calculation of totals for 16-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				—	—
(D)	o	o		o				—	—
n	o	o		o				o	—

[Instruction Symbol] [Execution Condition]

WSUM

WSUMP



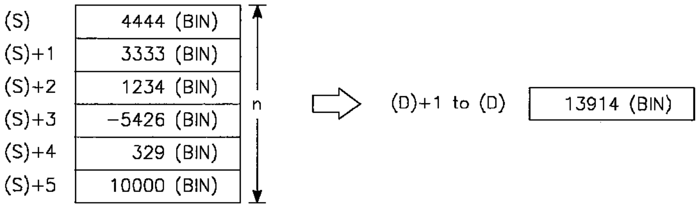
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where data for calculating total value is stored	BIN 16 bits
(D)	First device number of devices storing total value	BIN 32 bits
n	Number of data blocks	BIN 16 bits

Functions

WSUM

(1) Adds all 16-bit BIN data for n blocks from the device designated at (S), and stores it in the device designated at (D).



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

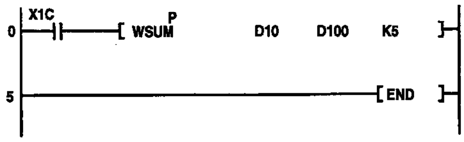
- The range n points from the (S) device exceeds the relevant device. (Error code: 4101)

Program Example

WSUM

(1) The following program when X1C goes ON adds the 16-bit BIN data from D10 to D14, and stores it in D100 and D101.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	WSUMP	D10 D100
5	END	K5

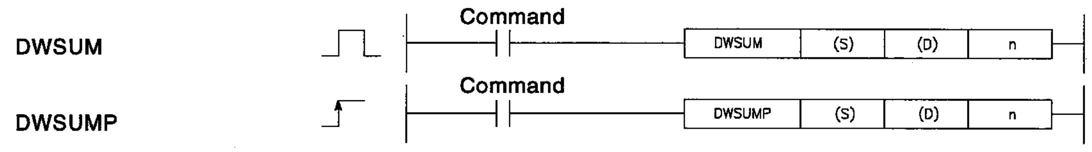
D10	4500 (BIN)
D11	2500 (BIN)
D12	−3276 (BIN)
D13	6780 (BIN)
D14	4444 (BIN)

⇨ D101, D100 | 14948 (BIN) |

7.5.14 Calculation of totals for 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				—	—
(D)	o	o		—				—	—
n	o	o		o				o	—

[Instruction Symbol] [Execution Condition]



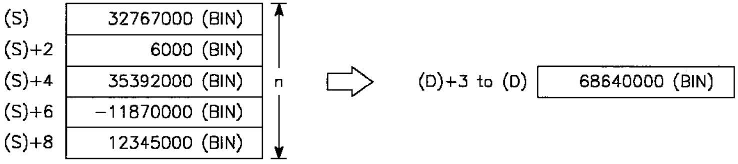
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where data for calculating the total value is stored	BIN 32 bits
(D)	First device number of devices storing total value	BIN 64 bits
n	Number of data blocks	BIN 16 bits

Functions

DWSUM

(1) Adds all 32-bit BIN data n-points from the device designated by (S), and stores the result at the device designated by (D).



Operation Errors

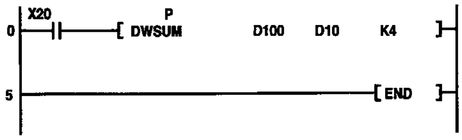
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range n points from the (S) device exceeds the relevant device. (Error code: 4101)

Program Example

(1) The following program adds the 32-bit BIN data at D100 through D107, and stores the result at D10 and D13 when X20 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	DWSUMP	D100
		D10
5	END	K4

D101, D100	11245600 (BIN)
D103, D102	27543200 (BIN)
D105, D104	558800 (BIN)
D107, D106	−15675000 (BIN)

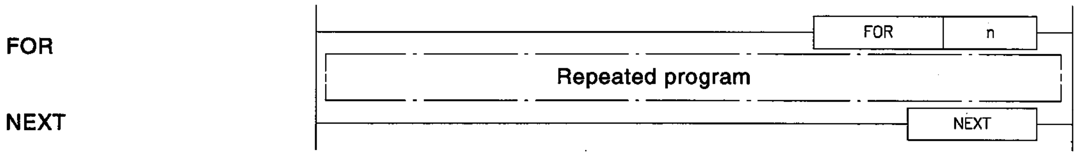
⇨ D13 to D10 | 23672600 (BIN)

7.6 Structured Program Instructions

7.6.1 FOR to NEXT instruction loop

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n	o								—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
n	Number of repetitions of the FOR to NEXT loop (from 1 to 32767).	BIN 16 bits

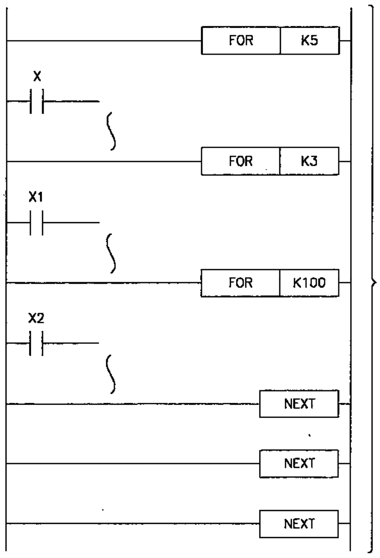
Functions

(1) When the processing in the FOR to NEXT loop is executed n-times without conditions, the step following the NEXT instruction will be executed.

(2) The value of n can be designated at between 1 and 32767. If it is designated at a value of from -32768 to 0, it will be executed as though n=1.

(3) If you do not desire to execute the processing called for within the FOR to NEXT loop, use the CJ or SCJ instruction to jump past it.

(4) FOR instructions can be nested up to 16 deep.



FOR instructions can be nested up to 16 deep.

Operation Errors

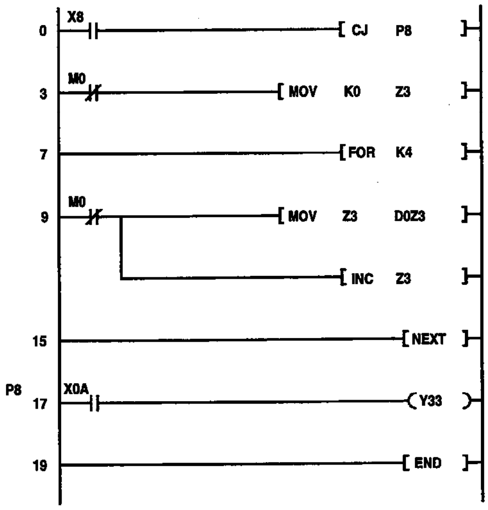
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- An END (FEND) instruction is executed after the execution of a FOR instruction, but before the execution of a NEXT instruction. (Error code: 4200)
- A NEXT instruction is executed prior to the execution of a FOR instruction. (Error code: 4201)
- A STOP instruction has been inserted within the FOR to NEXT loop. (Error code: 4200)
- The 17th FOR instruction is encountered when FOR instructions have been nested. (Error code: 4202)

Program Example

(1) The following program executes the FOR to NEXT loop when X8 is OFF, and does not execute it when X8 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X8
1	CJ	P8
3	LDI	M0
4	MOV	K0
		Z3
7	FOR	K4
9	LDI	M0
10	MOV	Z3
		D0Z3
13	INC	Z3
15	NEXT	
16		P8
17	LD	X0A
18	OUT	Y33
19	END	

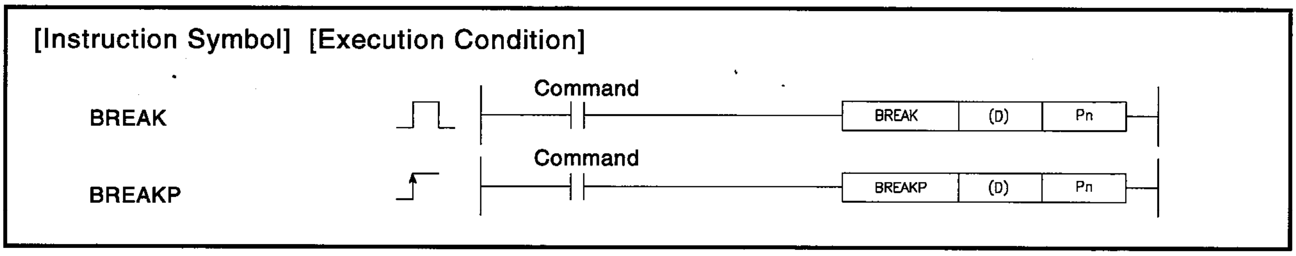
REMARKS

1) To force an end to the repetitious execution of the FOR to NEXT loop during the execution of the loop, insert a BREAK instruction.
See Section 7.6.2 for details concerning the use of the BREAK instruction.

2) Use the EGP/EGF instructions to conduct index qualification for the FOR to NEXT contact.

7.6.2 Forced end of FOR to NEXT instruction loop

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				P
(D)	o							—	—
Pn	o							—	o

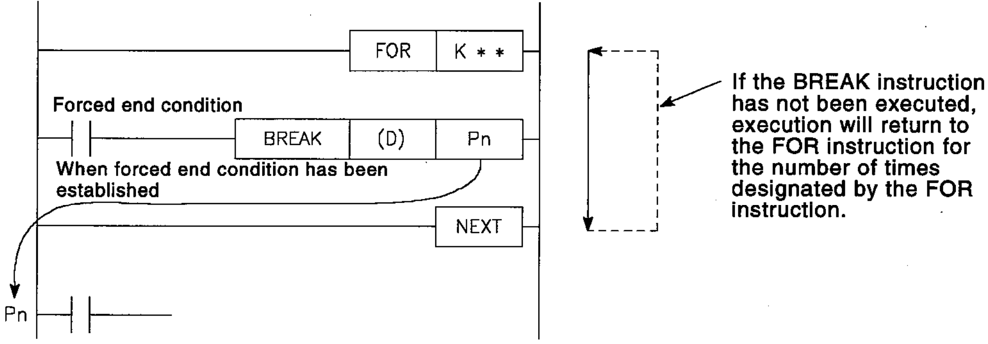


Set Data

Set Data	Meaning	Data Type
(D)	Number of device that will store the number of repetitions remaining	BIN 16 bits
Pn	Number of branch destination pointer when the end of processing repetitions.	Device name (pointer)

Functions

(1) Forces the end of repetition processing based on the FOR to NEXT instruction loop, and shifts to the execution of the pointer designated by Pn.



(2) The number of repetitions remaining at the point that the FOR to NEXT loop was brought to a forced end is stored at (D). However, the number of times when the BREAK instruction was executed is also included in the number of repetitions remaining.

(3) The BREAK instruction can be used only during the execution of a FOR to NEXT instruction loop.

(4) The BREAK instruction can be used only when there is only one level of nesting. If an end is forced when there are multiple nesting levels, execute the same number of BREAK instructions as there are nesting levels.

Operation Errors

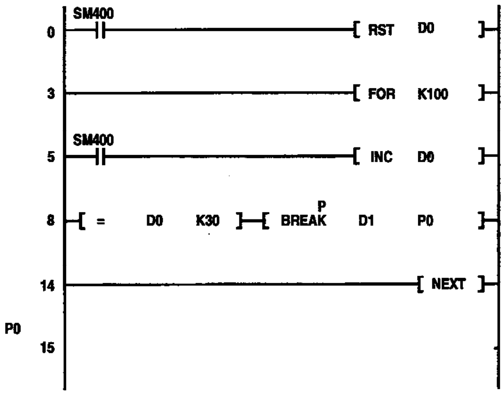
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The BREAK instruction is used in a case other than with the FOR to NEXT instruction loop. (Error code: 4203)
- The jump destination for the pointer designated by Pn does not exist. (Error code: 4210)

Program Example

(1) The following program forces the FOR to NEXT loop to end when the value of D2 reaches 30 (when the FOR to NEXT loop has been executed 30 times).

[Ladder Mode]



[List Mode]

Step	Instruction	Device
0	LD	SM400
1	RST	D0
3	FOR	K100
5	LD	SM400
6	INC	D0
8	LD=	D0 K30
11	BREAKP	D1 P0
14	NEXT	
15		P0

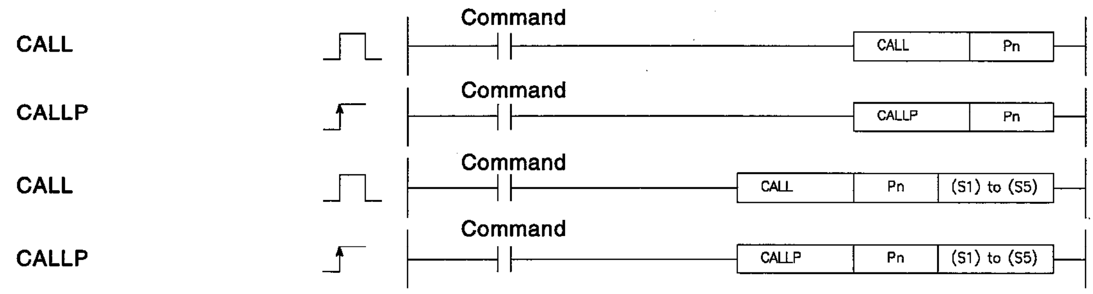
REMARK

* : The value 71 is stored at D1 when the BREAK instruction is executed.

7.6.3 Sub-routine program calls

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
P	—	o		—					
(S1) to (S5)	—	o		—					

[Instruction Symbol] [Execution Condition]



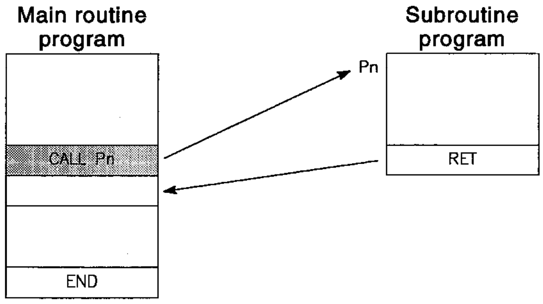
Set Data

Set Data	Meaning	Data Type
Pn	First pointer number of sub-routine program	Device name
(S1) to (S5)	Device number that passes to sub-routine program as an argument	Bit BIN 16 bits BIN 32 bits

Functions

(1) When the CALL(P) instruction is executed, executes the sub-routine program of the program specified by Pn.

[The CALL(P) instruction can execute sub-routine programs specified by a pointer within the same program file and sub-routine programs specified by a common pointer.]



(2) When function devices (FX, FY, FD) are used by a sub-routine program, specify a device with (S1) through (S5) corresponding to the function device.
The contents of the devices specified by (S1) through (S5) are as indicated below.

(a) Prior to execution of the sub-routine program, bit data is transmitted to FX, and word data is transmitted to FD.

(b) After the execution of the sub-routine program, the contents of FY and FD are transmitted to the corresponding devices.

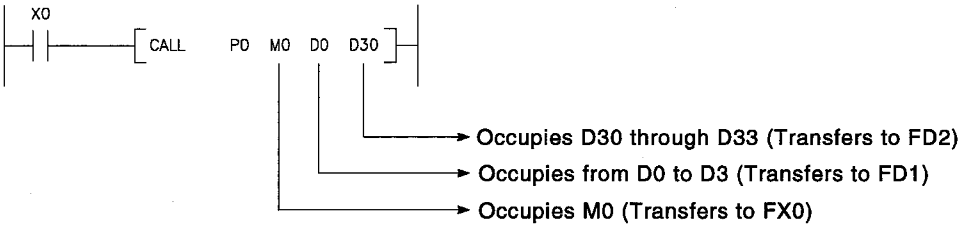
(c) The processing units for the function devices are as follows:

- FX, FY : Bits
- FD : 4-word units

The size of the data to be dealt with will differ depending on the device specified in the argument.

Function Devices	Device Used	Data Size	Remarks
· FX · FY	· Bit device	1 point	
	· When bit designation has been made for word device	1 bit	
· FD	· When digit designation of a bit device is used · Index register · Normal number	2 words	The upper 2 words of FD become 0
	· Word device	4 words	

[Main routine program]



Occupies D30 through D33 (Transfers to FD2)
Occupies from D0 to D3 (Transfers to FD1)
Occupies M0 (Transfers to FX0)

(3) (S1) through (S5) can be used with the CALL(P) instruction.

(4) The number of function devices to be used by a sub-routine program must be identical to the number of arguments in the CALL(P) instruction.
Also, the function device and the type of CALL(P) argument used should be identical.

(5) Device numbers specified by the CALL(P) instruction should not overlap.
If they do overlap, it will not be possible to obtain accurate calculations.

(6) The word device (4 words) which is used in an argument of a CALL instruction should not be used in a sub-routine program.
If used, it will not be possible to obtain accurate calculations.

(7) Up to 16 nesting levels are possible with the CALL(P) instruction.
However, this 16 levels is the total number of levels in the CALL(P), FCALL(P), ECALL(P), and EFCALL(P) instructions.

(8) Devices which are turned ON within sub-routine programs will be latched even if the sub-routine program is not executed.
Devices which are turned ON during the execution of a sub-routine program can be turned OFF by the execution of the FCALL(P) instruction.

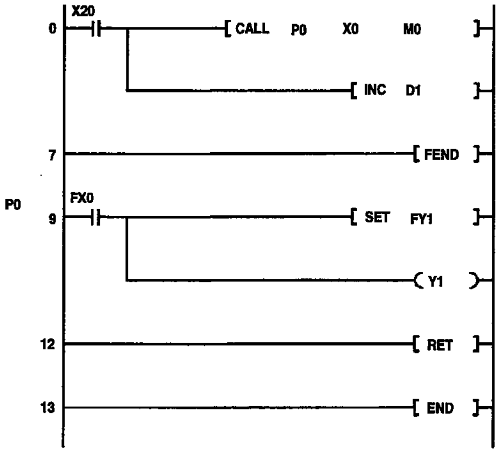
Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Following the execution of the CALL(P) instruction, an END, FEND, GOEND, or STOP instruction is executed before the execution of the RET instruction. (Error code: 4211)
- An RET instruction is executed prior to the execution of the CALL(P) instruction. (Error code: 4212)
- A 17th nesting level is executed. (Error code: 4213)
- There is no sub-routine program for the pointer specified in the CALL(P) instruction. (Error code: 4210)
- A function device (FX, FY, or FD) is specified in the argument. (Error code: 4101)

Program Example

(1) The following program executes a sub-routine program with argument when X20 goes ON.



Step	Instruction	Device
0	LD	X20
1	CALL	P0
		X0
		M0
5	INC	D1
7	FEND	
8		P0
9	LD	FX0
10	SET	FY1
11	OUT	Y1
12	RET	
13	END	

7.6.4 Return from sub-routine programs

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
—	—								

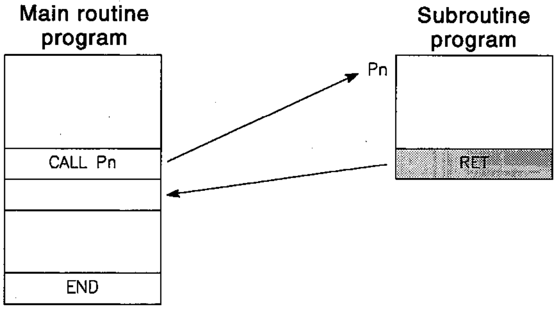
[Instruction Symbol] [Execution Condition]

RET

Functions

(1) Indicates end of sub-routine program

(2) When the RET instruction is executed, returns to the step following the CALL(P), FCALL(P), ECALL(P), EFCALL(P) instruction which called the sub-routine program.



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Following the execution of the CALL(P) instruction, an END, FEND, GOEND, or STOP instruction is executed before the execution of the RET instruction. (Error code: 4211)
- An RET instruction is executed prior to the execution of the CALL(P) instruction. (Error code: 4212)

7.6.5 Sub-routine program output OFF calls

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
P	—	o		—					
(S1) to (S5)	—	o		—					

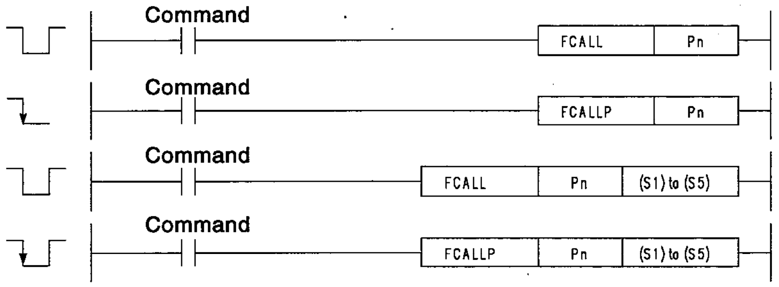
[Instruction Symbol] [Execution Condition]

FCALL

FCALLP

FCALL

FCALLP



Set Data

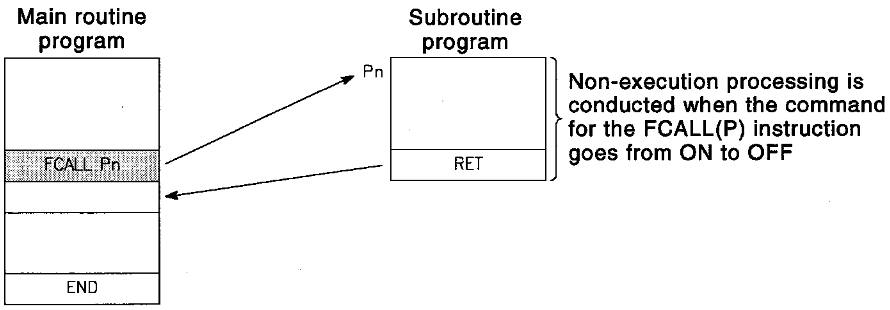
Set Data	Meaning	Data Type
Pn	First pointer number of sub-routine program	Device name
(S1) to (S5)	Device number that passes to sub-routine program as an argument	Bit BIN 16 bits BIN 32 bits

Functions

(1) When the FCALL(P) command goes from ON to OFF, non-execution processing will be conducted on the sub-routine program of the pointer designated by Pn.

[The FCALL(P) instruction can execute sub-routine programs designated by a pointer within the same program file, and sub-routine programs designated by common pointers.]

(a) Non-execution processing is identical to the processing that is conducted when the condition contacts for the individual coil instructions are in the OFF state.

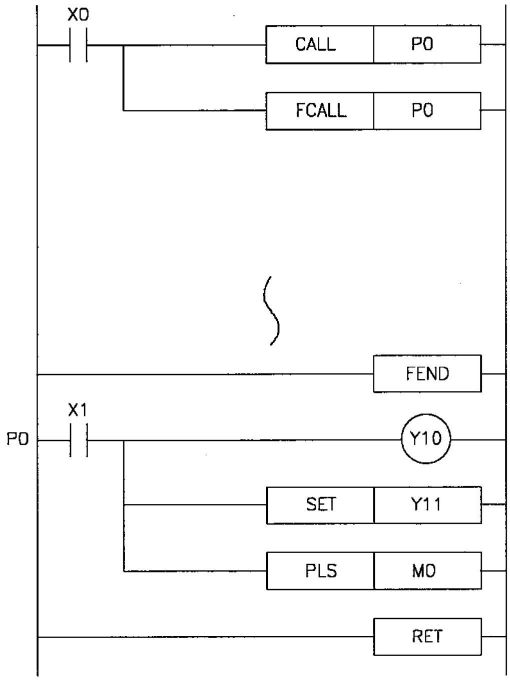


(b) The operation results for the individual coil instructions following non-execution processing will be as follows, regardless of the ON/OFF status of the individual contacts:

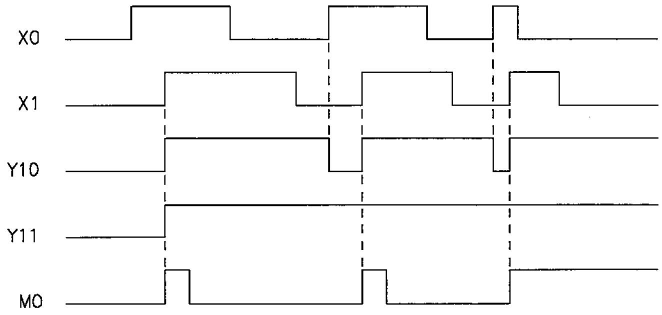
- OUT instruction.................................... Forced OFF
- SET instruction
- RST instruction
- SFT instruction ...... Maintains status
- Basic instructions
- Application instructions
- PLS instruction ...... Processing identical to when condition contacts are OFF
- Pulse creation instructions (□P)
- Present value of low speed and high speed timers 0
- Present value of retentive timer ....Preserves
- Present value of counter

(2) The FCALL(P) instruction is used in conjunction with the CALL(P) instruction.

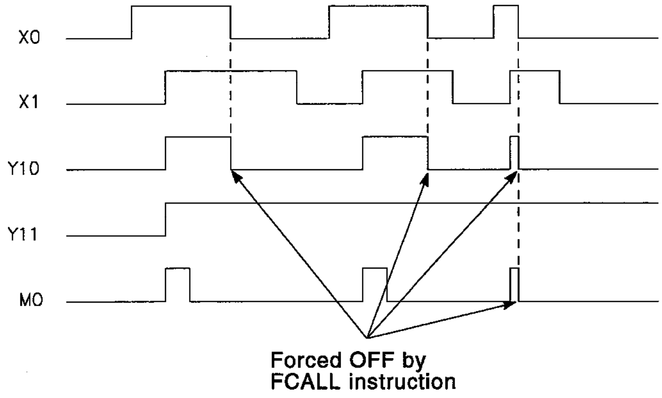
(3) If the FCALL(P) instruction is not executed (cases where only the CALL instruction is utilized), each individual coil instruction output status will be preserved when the sub-routine program instruction goes OFF because the sub-routine programs will not have been executed.
When the FCALL(P) instruction is executed, the OUT instruction and PLS instruction (including □ P instructions) can be forced OFF to perform sub-routine program non-execution processing.



When FCALL instruction is not used



When FCALL instruction is used



(4) When function devices (FX, FY, FD) are used by a sub-routine program, specify devices with (S1) through (S5) corresponding to the function devices.
The function devices which can be designated with (S1) through (S5) have the roles described below:

(a) Prior to execution of the sub-routine program, bit data is transmitted to FX, and word data is transmitted to FD.

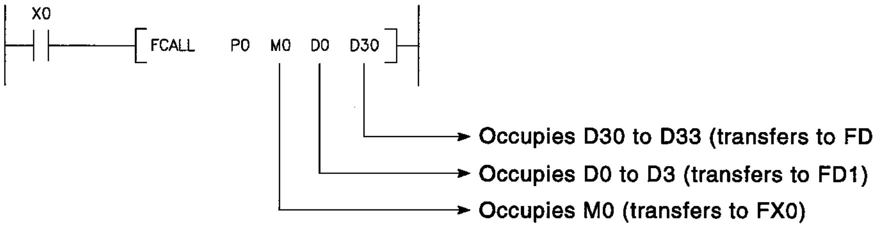
(b) After the execution of the sub-routine program, the contents of FY and FD are transmitted to the corresponding devices.

(c) The processing units for the function devices are as follows:

- FX, FY : Bits
- FD : 4-word units
  The size of the data to be dealt with will differ depending on the device specified in the argument.

Function Devices	**Device Used**	**Data Size**	**Remarks**
· FX · FY	· Bit device	1 point	
	· When bit designation has been made for word device	1 bits	
· FD	· When digit designation of a bit device is used · Index register · Normal number	2 words	The upper 2 words of FD become 0
	· Word device	4 words	

[Main routine program]



Occupies D30 to D33 (transfers to FD2)
Occupies D0 to D3 (transfers to FD1)
Occupies M0 (transfers to FX0)

(5) The FCALL(P) instruction can use from (S1) through (S5).

(6) The number of function devices to be used by a sub-routine program must be identical to the number of arguments in the FCALL(P) instruction.
The function devices and types of FCALL(P) arguments should also be identical.

(7) None of the device numbers designated in FCALL(P) instruction arguments should be duplicated.
If they do overlap, it will not be possible to obtain accurate calculations.

(8) Up to 16 nesting levels are possible with the FCALL(P) instruction. However, this 16 levels is the total number of levels in the CALL(P), FCALL(P), ECALL(P), and EFCALL(P) instructions.

Operation Errors

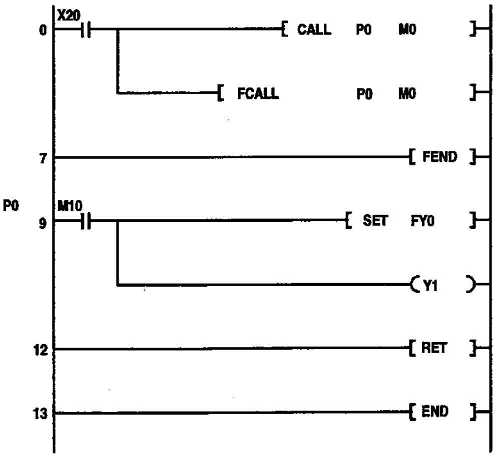
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- An END, FEND, GOEND, or STOP instruction is executed prior to the execution of the RET instruction, but after the execution of the FCALL(P) instruction. (Error code: 4211)
- An RET instruction is executed prior to the execution of the FCALL(P) instruction. (Error code: 4212)
- A 17th nesting level is executed. (Error code: 4213)
- The sub-routine program of the pointer designated by the FCALL(P) instruction does not exist. (Error code: 4210)
- An annunciator (F) has been designated in an argument. (Error code: 4101)

Program Example

(1) The following program executes a sub-routine program with argument when X20 is ON, and forces non-execution processing when X20 goes from ON to OFF.

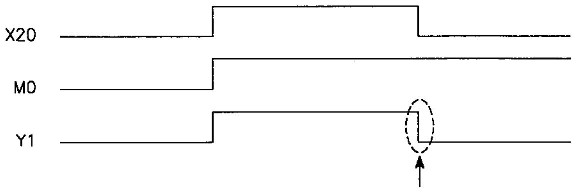
[Ladder Mode]



[List Mode]

Step	Instruction	Device
0	LD	X20
1	CALL	P0
		M0
4	FCALL	P0
		M0
7	FEND	
8		P0
9	LD	M10
10	SET	FY0
11	OUT	Y1
12	RET	
13	END	

[Operation]

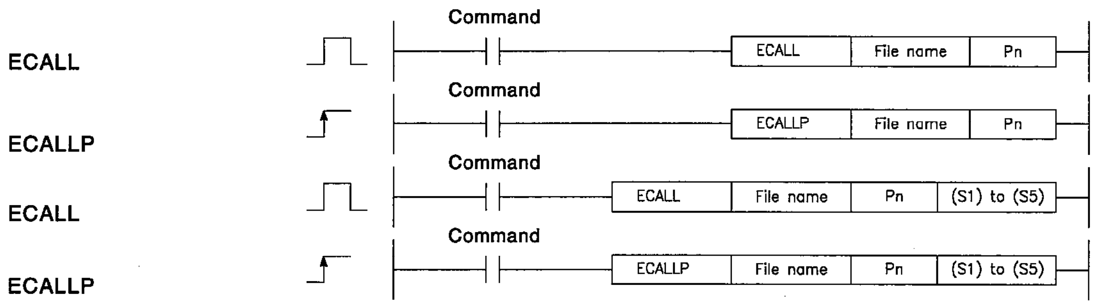


Forced OFF by FCALL instruction

7.6.6 Sub-routine calls between program files

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
P	—	o		—					
(S1) to (S5)	—	o		—					

[Instruction Symbol] [Execution Condition]

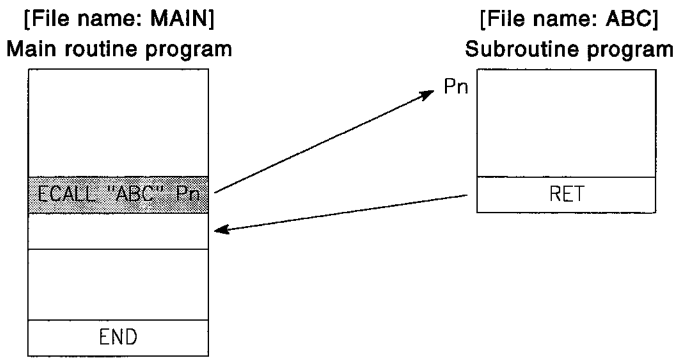


Set Data

Set Data	Meaning	Data Type
File name	Name of program file to be called	Character string
Pn	First pointer number of sub-routine program	Device name
(S1) to (S5)	Device number that passes to sub-routine program as an argument	Bit BIN 16 bits BIN 32 bits

Functions

(1) Executes the sub-routine program of the pointer designated by Pn in the designated program file name when the ECALL(P) instruction is executed.
The ECALL(P) instruction can be used to call a sub-routine program that uses a local pointer from a different program file.



(2) A file name can designate only the names of program files stored in internal memory (drive 0).

(3) Extensions are not needed when designating file names.

(4) When function devices (FX, FY, FD) are used by a sub-routine program, specify a device with (S1) through (S5) corresponding to the function device.
The contents of the devices specified by (S1) through (S5) are as indicated below.

(a) Prior to execution of the sub-routine program, bit data is transmitted to FX, and word data is transmitted to FD.

(b) After the execution of the sub-routine program, the contents of FY and FD are transmitted to the corresponding devices.

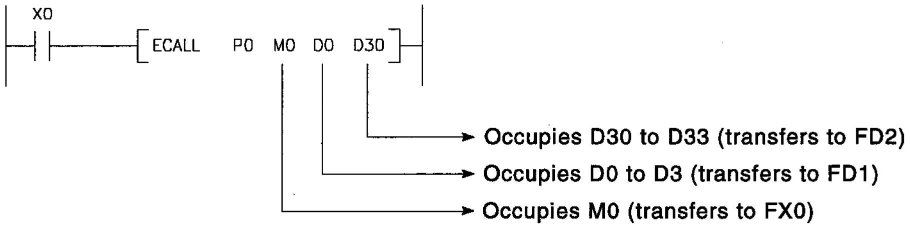
(c) The processing units for the function devices are as follows:

- FX, FY : Bits
- FD : 4-word units

The size of the data to be dealt with will differ depending on the device specified in the argument.

Function Device	Device Used	Data Size	Remarks
FX	Bit device	1 point	
FY	When bit designation has been made for word device	1 bits	
FD	When digit designation of a bit device is used Index register Normal number	2 words	The upper 2 words of FD become 0
	Word device	4 words	

[Main routine program]



Occupies D30 to D33 (transfers to FD2)
Occupies D0 to D3 (transfers to FD1)
Occupies M0 (transfers to FX0)

(5) From (S1) through (S5) can be used by the ECALL instruction.

(6) The number of function devices used by sub-routine programs must be identical to the number of arguments used by the ECALL(P) instruction.
Further, the function devices should be identical to the types of arguments used by the ECALL(P) instruction.

(7) The numbers of the devices designated by the arguments in the ECALL(P) instruction should not overlap.
If they do overlap, it will not be possible to obtain accurate calculations.

(8) Up to 16 levels of nesting can be used with the ECALL(P) instruction. However, this 16 levels is the total number of levels in the CALL(P), FCALL(P), ECALL(P), and EFCALL(P) instructions.

(9) Devices which are turned ON within sub-routine programs will be latched even if the sub-routine program is not executed.
Devices turned ON during the execution of a sub-routine program can be turned OFF by the EFCALL(P) instruction.

Operation Errors

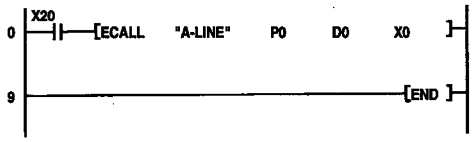
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0 if index qualification has not been performed.

- An END, FEND, GOEND, or STOP instruction is executed prior to the execution of the RET instruction, but after the execution of the ECALL(P) instruction. (Error code: 4211)
- An RET instruction is executed prior to the execution of the ECALL(P) instruction. (Error code: 4212)
- A 17th nesting level is executed. (Error code: 4213)
- The sub-routine program of the pointer designated by the ECALL(P) instruction does not exist. (Error code: 4210)
- A function device (FX, FY, or FD) is specified in the argument. (Error code: 4101)
- The designated file does not exist. (Error code: 4210)
- The designated file cannot be executed. (Error code: 2411)

Program Example

(1) The following program executes the P0 A-LINE.QPG when X20 is ON.

[Ladder Mode]



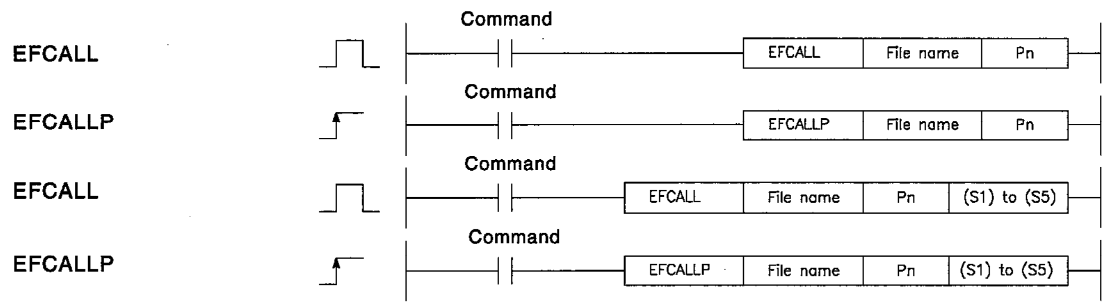
[List Mode]

Steps	Instruction	Device
0	LD	X20
1	ECALL	"A-LINE" P0 D0 X0
9	END	

7.6.7 Sub-routine output OFF calls between program files

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
P	—	o		—					
(S1) to (S5)	—	o		—					

[Instruction Symbol] [Execution Condition]



Set Data

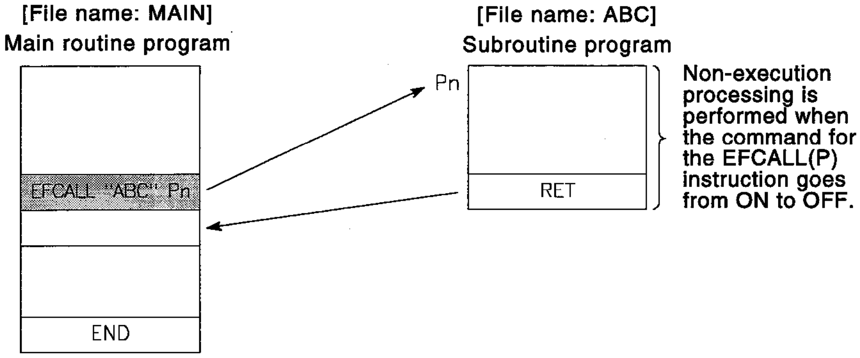
Set Data	Meaning	Data Type
File name	Name of program file to be called	Character string
Pn	First pointer number of sub-routine program	Device name
(S1) to (S5)	Device number that passes to sub-routine program as an argument	Bit BIN 16 bits BIN 32 bits

Functions

(1) Non-execution processing is performed on the sub-routine program of the pointer designated by Pn when the EFCALL(P) command goes from ON to OFF.

[The EFCALL(P) can also be used to call a sub-routine program that uses a local pointer from a different program file.]

(a) Non-execution processing is identical to the processing that is conducted when the condition contacts for the individual coil instructions are in the OFF state.



(b) The operation results for the individual coil instructions following non-execution processing will be as follows, regardless of the ON/OFF status of the individual contacts:

- OUT instruction .... Forced OFF
- SET instruction
- RST instruction
- SFT instruction
- Basic instructions
- Application instructions

.... Maintains status

- PLS instruction
- Pulse creation instructions (□P)

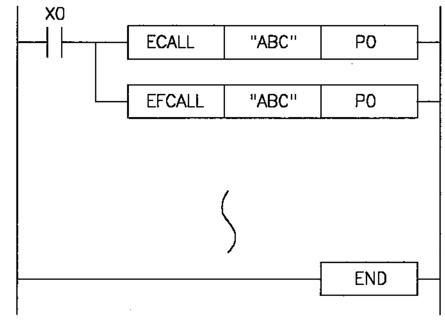
.... Processing identical to when condition contacts are OFF

- Present value of low speed and high speed timers 0
- Present value of retentive timer
- Present value of counter

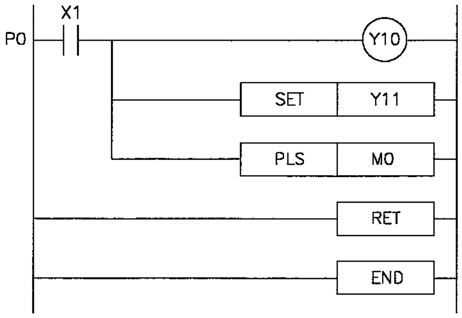
.... Preserves

(2) The EFCALL(P) instruction is used in combination with the ECALL(P) instruction.

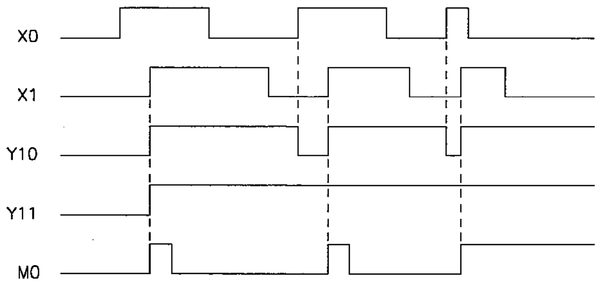
(3) If the EFCALL(P) instruction is not executed (only the ECALL(P) instruction is used), the sub-routine program will not be executed when the sub-routine program command goes OFF, so the output conditions of the individual coil instructions will be preserved.
When the EFCALL(P) instruction is executed, non-execution processing is performed for the sub-routine program, so it is possible to force the OUT instruction and the PLS instruction (including □P instructions) OFF.



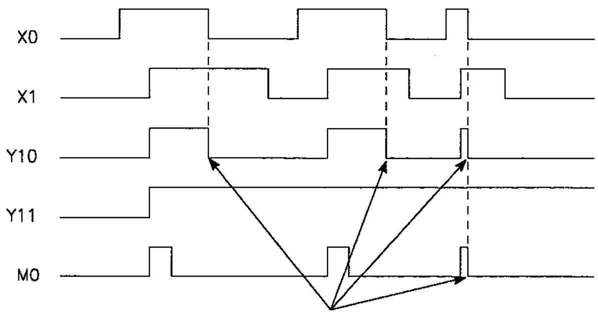
[File name: ABC]



When EFCALL instruction is not used



When EFCALL instruction is used



Forced OFF with the EFCALL instruction

(4) When function devices (FX, FY, FD) are used by a sub-routine program, specify a device with (S1) through (S5) corresponding to the function device.

(a) Prior to execution of the sub-routine program, bit data is transmitted to FX, and word data is transmitted to FD.

(b) After the execution of the sub-routine program, the contents of FY and FD are transmitted to the corresponding devices.

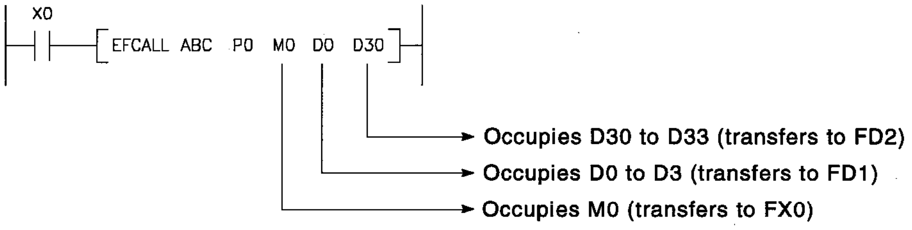
(c) The processing units for the function devices are as follows:

- FX, FY : Bits
- FD : 4-word units

The size of the data to be dealt with will differ depending on the device specified in the argument.

Function Device	Device Used	Data Size	Remarks
FX	Bit device	1 point	
FY	When bit designation has been made for word device	1 bits	
FD	When digit designation of a bit device is used Index register Normal number	2 words	The upper 2 words of FD become 0
	Word device	4 words	

[Main routine program]



(5) (S1) through (S5) can be used with the EFCALL(P) instruction.

(6) The number of function devices used by sub-routine programs must be identical to the number of arguments used by the EFCALL(P) instruction.
Further, the function devices should be identical to the types of arguments used by the EFCALL(P) instruction.

(7) The numbers of the devices designated by the arguments in the EFCALL(P) instruction should not overlap.
If they do overlap, it will not be possible to obtain accurate calculations.

(8) Up to 16 levels of nesting can be used with the EFCALL(P) instruction. However, this 16 levels is the total number of levels in the CALL(P), FCALL(P), ECALL(P), and EFCALL(P) instructions.

(9) Devices which are turned ON within sub-routine programs will be latched even if the sub-routine program is not executed.
Devices turned ON during the execution of a sub-routine program can be turned OFF by the EFCALL(P) instruction.

Operation Errors

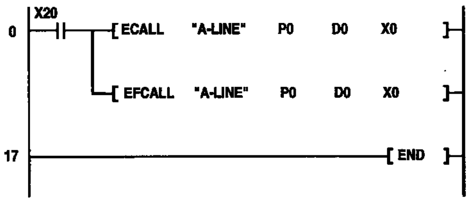
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0 if index qualification has not been performed.

- An END, FEND, GOEND, or STOP instruction is executed prior to the execution of the RET instruction, but after the execution of the EFCALL(P) instruction. (Error code: 4211)
- An RET instruction is executed prior to the execution of the EFCALL(P) instruction. (Error code: 4212)
- A 17th nesting level is executed. (Error code: 4213)
- The sub-routine program of the pointer designated by the EFCALL(P) instruction does not exist. (Error code: 4210)
- An annunciator (F) has been designated in an argument. (Error code: 4101)
- The designated file does not exist. (Error code: 4210)
- The designated file cannot be executed. (Error code: 2411)

Program Example

(1) The following program executes a sub-routine program with argument when X0 is ON, and forces non-execution processing when X20 goes from ON to OFF.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	ECALL	"A-LINE" P0 D0 X0
9	EFCALL	"A-LINE" P0 D0 X0
17	END	

7.6.8 Refresh instruction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]

COM

Functions

Processing for the COM instruction differs depending on the ON/OFF status of special relay SM775.

- When SM775 is OFF : Link refresh and general data processing are carried out.
- When SM775 is ON : Carries out only general data processing

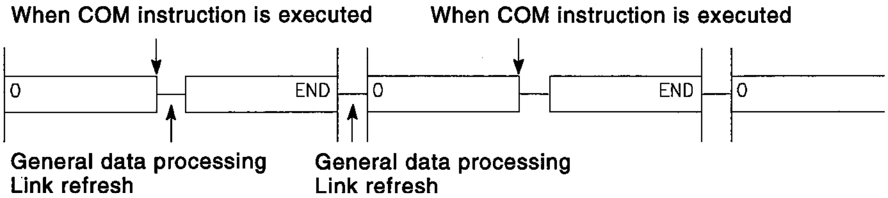
When SM775 is OFF

(1) The COM instruction is used in the following cases.

(a) To speed up communications processing with remote I/O stations

(b) When it is desirable to stabilize data exchanges with another station that has different scan times during the execution of a data link

(2) At the point when the COM instruction is executed, the QnACPU temporarily suspends the processing of the sequence program and performs the general data processing done at the time of END processing, and link refresh processing also.



(3) The COM instruction can be inserted into sequence programs as often as desired.
However, the scan time of the sequence program will be lengthened by the time taken for general data processing or link refresh operations.

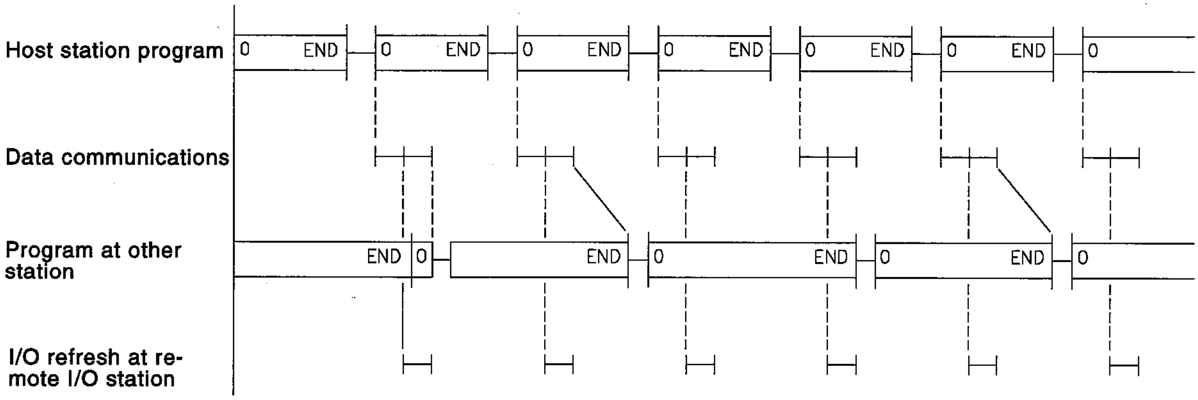
REMARK

The following processing is conducted as a part of general data processing:

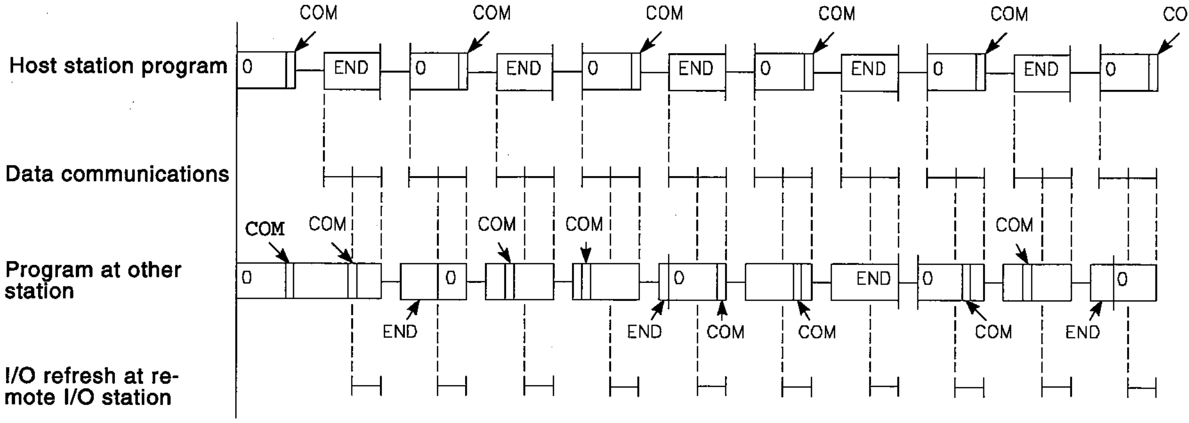
- Exchange of signals between the QnACPU and peripheral devices
- Monitor processing of other stations
- Read processing by the serial communications module of the buffer memory of another special function module

(4) Data communications using the COM instruction

(a) Example of data communications when COM instruction is not used



(b) Example of data communications when COM instruction has been used



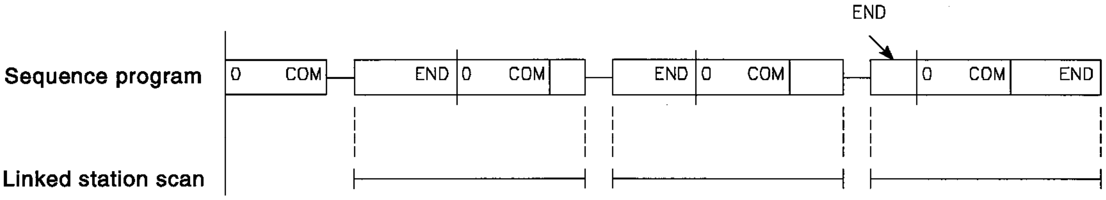
1) When the COM instruction is used at the host station, it is possible to increase the number of data communication repetitions with the remote I/O station unconditionally, as shown in (b) above, and thus to speed up data communications.

2) In cases where the remote station sequence program scan time is longer than the scan time of the host station, using the COM instruction at the remote station will enable the avoidance of timing difficulties in which data cannot be input, as shown in (a) above.

3) When the COM instruction has been used at the other station, a link refresh will be performed each time that station receives a command from the host station.

• Between step 0 and COM instruction • Between COM instruction and COM instruction • Between COM instruction and END instruction	Link refresh can be performed once each time between these events

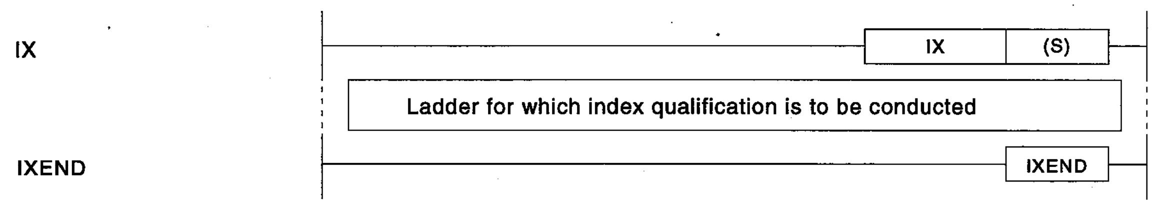
(5) If the scan time from the linked station is longer than the sequence program scan time at the host station, designating the COM instruction at the host station will not increase the speed of data communications.



7.6.9 Index qualification of entire ladder

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—					

[Instruction Symbol] [Execution Condition]



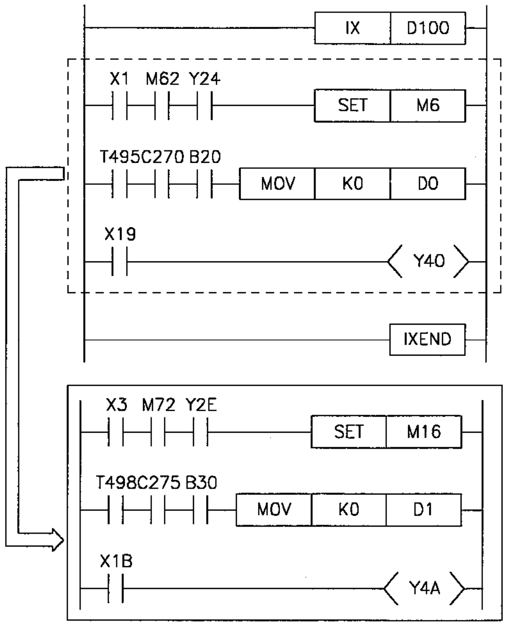
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where index qualification data is being stored	BIN 16 bits

Functions

(1) Conducts index qualification on device numbers for all devices in the ladder from the IX instruction to the IXEND instruction.

(2) Index qualification of device numbers is accomplished by presetting qualification values for each device, which causes these values to be added to all devices used in the ladder between the IX and IXEND instruction, and program processing is conducted according to the added to device numbers.



Qualification value

D100	8	T
D101	5	C
D102	2	X
D103	10	Y
D104	10	M
D105	20	L
D106	16	B
D107	20	V
D108	1	D

- The value "2" is added to X1 and X19, so that they are processed as X3 and X1B respectively
- The value "10" (A_H) is added to Y24 and Y40, so they are processed as Y2E and Y4A respectively
- The value "10" is added to M6 and M62, so they are processed as M16 and M72 respectively
- The value "16" (10_H) is added to B20, so it is processed as B30
- The value "8" is added to T495, so it is processed as T498
- The value "5" is added to C270, so it is processed as C275
- The value "1" is added to D0, so it is processed as D1

(3) Instructions such as the PLS, PLF, and □□□P instructions, which are executed only once when input conditions have been established, cannot be index qualified by using the IX to IXEND instruction loop.

(4) In cases where adding the qualification value causes the device number to exceed the device range, accurate processing will not be conducted.

(5) Do not execute the IX or IXEND instructions during online program changes of sequence programs (write during RUN).
Accurate processing will not be conducted if this happens.

(6) Qualification values are preset for random word devices as BIN values, and the initial device number for which qualification values have been set is designated by (S).

(7) Index qualification cannot be conducted in a program between the IX and IXEND instructions.

(8) An index qualification ladder established with the IX and IXEND instructions will be transformed into a ladder using the index register (Zn) during program expansion.
The configuration of the index qualification table and the corresponding index register numbers are as shown below:

	Device name	Index register number
(S)	Qualification value of timer (T)	Z0
(S)+1	Qualification value of counter (C)	Z1
(S)+2	Qualification value of input (X)	Z2
(S)+3	Qualification value of output (Y)	Z3
(S)+4	Qualification value of internal relay (M)	Z4
(S)+5	Qualification value of latch relay (L)	Z5
(S)+6	Qualification value of link relay (B)	Z6
(S)+7	Qualification value of edge relay (V)	Z7

	Device name	Index register number
(S)+8	Qualification value of data register (D)	Z8
(S)+9	Qualification value of link register (W)	Z9
(S)+10	Qualification value of file register (R)	Z10
(S)+11	Qualification value of buffer register I/O No. (U)	Z11
(S)+12	Qualification value of buffer register (G)	Z12
(S)+13	Qualification value of link direct device network No. (J)	Z13
(S)+14	Qualification value of file register (ZR)	Z14
(S)+15	Qualification value of pointer (P)	Z15

POINT

For index qualification ladders created with the IX and IXEND instructions, it is necessary to start peripheral devices up in the "general purpose" mode, and to conduct "program expansion".
When peripheral devices are started up by CPU type name (Q2A, Q2A-S1, Q3A, Q4A), and index qualification ladders are created by the IX and IXEND instruction, accurate processing will not be possible.

Operation Errors

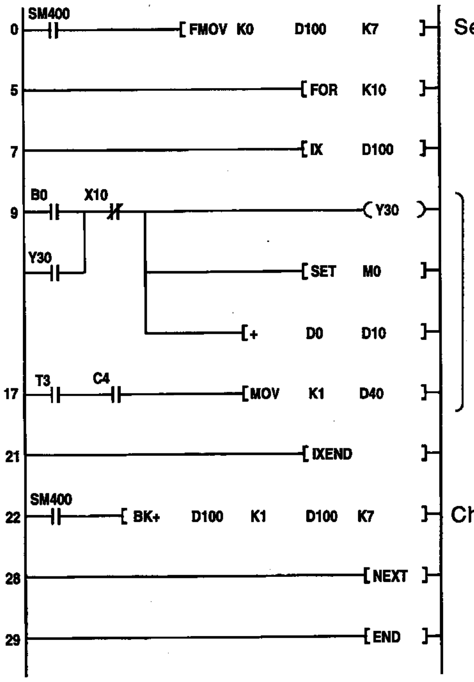
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The IX and IXEND instructions are not used together. (Error code: 4231)
- An END, FEND, GOEND, or STOP instruction is executed prior to the execution of the IXEND instruction, but after the execution of the IX instruction. (Error code: 4231)

Program Example

(1) The following program executes the same ladder 10 times, while changing device numbers.

[Ladder Mode]



Sets qualification values

Ladder which conducts device qualification

Changes qualification values

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	FMOV	K0
		D100
		K7
5	FOR	K10
7	IX	D100
9	LD	B0
10	OR	Y30
11	ANI	X10
12	OUT	Y30
13	SET	M0
14	+	D0
		D10
17	LD	T3
18	AND	C4
19	MOV	K1
		D40
21	IXEND	
22	LD	SM400
23	BK+	D100
		K1
		D100
		K7
28	NEXT	
29	END	

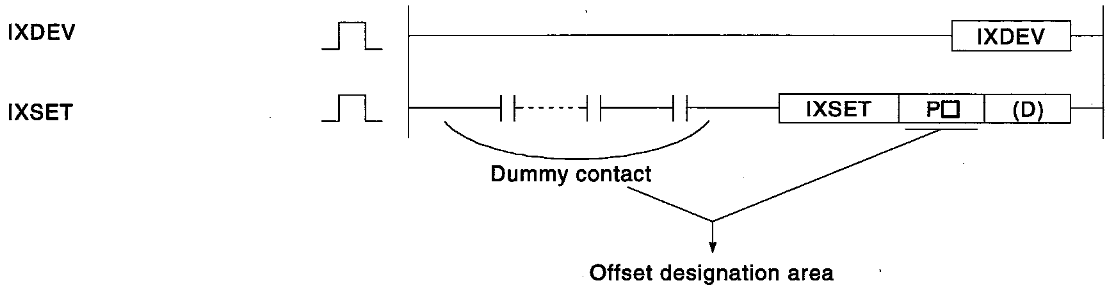
	Qualification value
D100	Qualification value of T
D101	Qualification value of C
D102	Qualification value of X
D103	Qualification value of Y
D104	Qualification value of M
D105	Qualification value of L
D106	Qualification value of B

1st time		2nd time		3rd time		10th time
B0	→	B1	→	B2	→	B9
X10	→	X11	→	X12	→	X19
Y30	→	Y31	→	Y32	→	X39
M0	→	M1	→	M2	→	M9
D0	→	D1	→	D2	→	D9
D10	→	D11	→	D12	→	D19
T3	→	T4	→	T5	→	T12
C4	→	C5	→	C6	→	T13
D40	→	D41	→	D42	→	D49

7.6.10 Designation of qualification values in index qualification of entire ladders

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				P
P□	—	—		—					o
(D)	—	o		—					—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
P□	First device number of devices (pointer only) where index qualification data is being stored	Device name
(D)	First device number of devices (something other than a pointer) which will store the index qualification data	BIN 16 bits

Functions

(1) The device offset value designated at the offset designation area is set at the index qualification table designated by (D).

(2) The value 0 will be entered if no designation value has been made.

(3) Word devices are also indicated by contact (word device bit designation).
Data register 10 (D0) is designated with D0, 0.
(Any value from 0 through A can be used for the bit number.)

(4) Designation is made according to the method described below.
(The symbol □ is where the offset value will be. The notation XX indicates random selection.)

Device	**T**	**C**	**X**	**Y**	**M**	**L**	**V**	**B**
Designation method	T□ ─┤ ├─	C□ ─┤ ├─	X□ ─┤ ├─	Y□ ─┤ ├─	M□ ─┤ ├─	L□ ─┤ ├─	V□ ─┤ ├─	B□ ─┤ ├─
Device	**D**	**W**	**R**	**U/G**		**G**	**J**	
Designation method	D□.XX ─┤ ├─	W□.XX ─┤ ├─	R□.XX ─┤ ├─	U□\G□.XX ─┤ ├─			J□\B□* ─┤ ├─	
Device	**ZR**	**P**						
Designation method	ZR□.XX ─┤ ├─	P□ set by device designation						

*: Devices following J□\ designate B, W, X, or Y, and the offset value is also set in correspondence with this.
If no offset value is desired, set this value as 0.

(5) If two offsets for two identical types of device have been set in the offset designation area, the last value set will be valid.

(6) The IXDEV and IXSET instructions should be treated as a pair.

(7) Any value from 0 to 32767 is valid for ZR.
(The offset value will be the remainder of the quotient of the designated device number divided by 32768.)

(8) Only LD or AND is valid for the dummy contact in the offset designation area. Any other instruction used here will be ignored.

Operation Errors

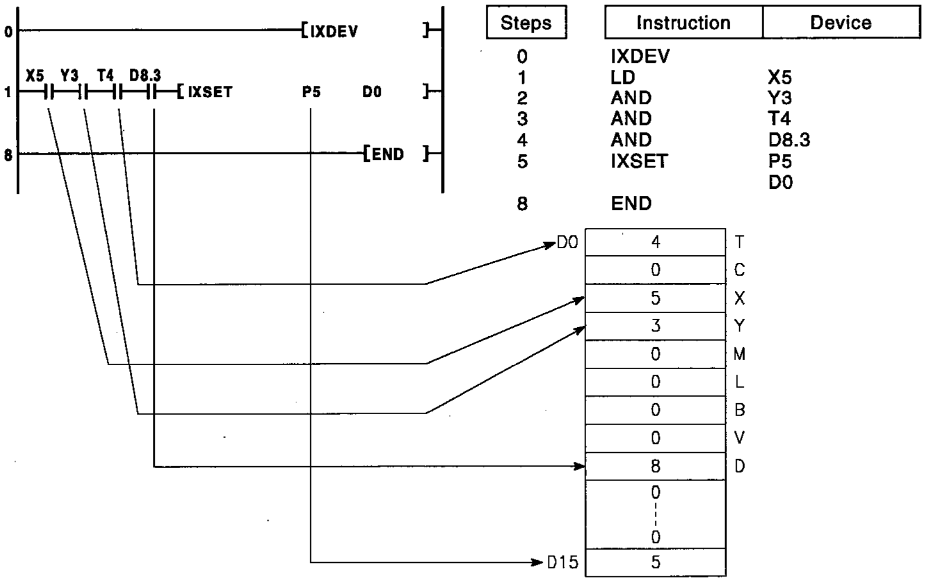
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The IXDEV and IXSET instructions have not been used as a pair.
(Error code: 4231)

Program Example

(1) The following program changes the index values for input (X), output (Y), data register (D) and pointer (P).

[Ladder Mode] [List Mode]



Steps	Instruction	Device
0	IXDEV	
1	LD	X5
2	AND	Y3
3	AND	T4
4	AND	D8.3
5	IXSET	P5 D0
8	END	

CONTENTS

7.7 Data Table Operation Instructions

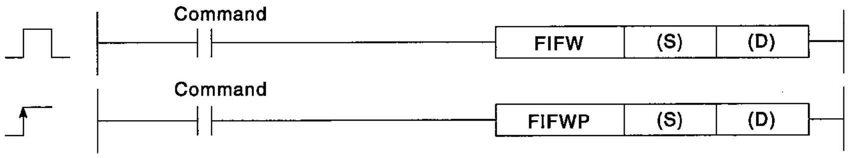
7.7.1 Writing data to the data table

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		o					—
(D)	—	o		—					—

[Instruction Symbol] [Execution Condition]

FIFW

FIFWP

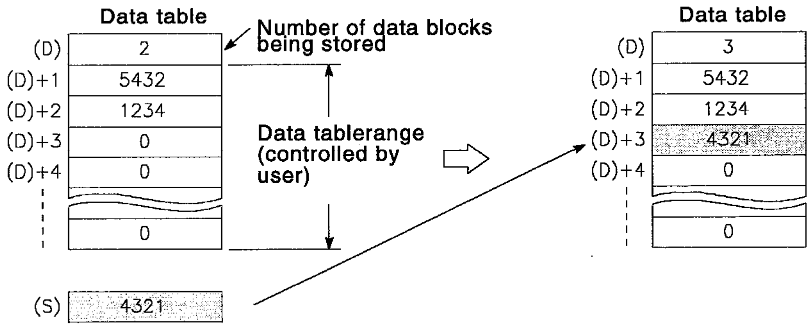


Set Data

Set Data	Meaning	Data Type
(S)	Data to write to table or number of device storing such data	BIN 16 bits
(D)	First number of table	

Functions

(1) Stores the 16-bit data designated by (S) in the data table designated by (D).
The number of data blocks being stored in the table is stored at (D), and the data designated by (S) is stored in sequence from (D)+1.



(2) The first time the FIFW instruction is executed, any values in the designated device should be cleared.

(3) The number of data blocks to be written in the data and the data table range should be controlled by the user.
(See Program Example (2))

Operation Errors

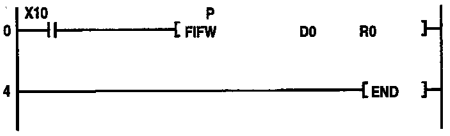
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data table range when the FIFW instruction was executed exceeds the relevant device range. (Error code: 4100)
- The value of (D) plus the number of registered data blocks exceeds the device range designated by (D). (Error code: 4101)

Program Example

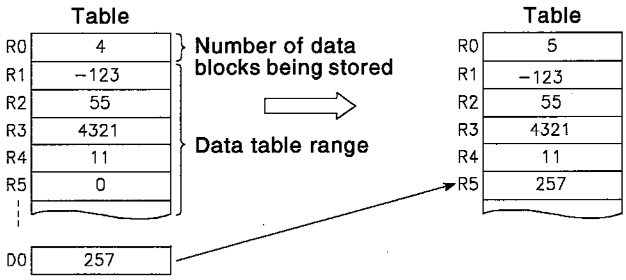
(1) The following program stores the data at D0 following R0 when X10 is ON.

[Ladder Mode]



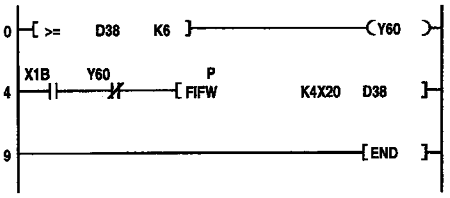
[List Mode]

Steps	Instruction	Device
0	LD	X10
1	FIFWP	D0 R0
4	END	



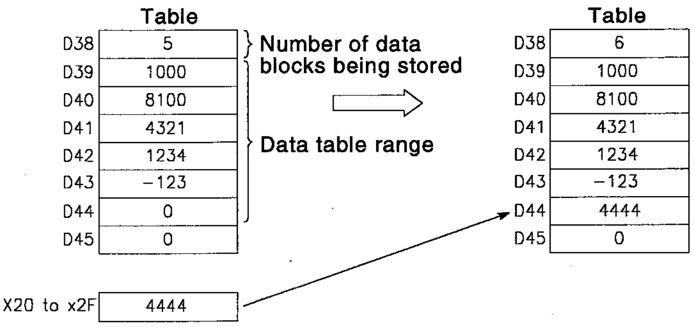
(2) The following program stores the data from X20 through X2F at the D38 through D44 table when X1B is ON, and, if there are more than 6 data blocks to be stored, turns Y60 ON and disables the FIFW instruction.

[Ladder Mode]



[List Mode]

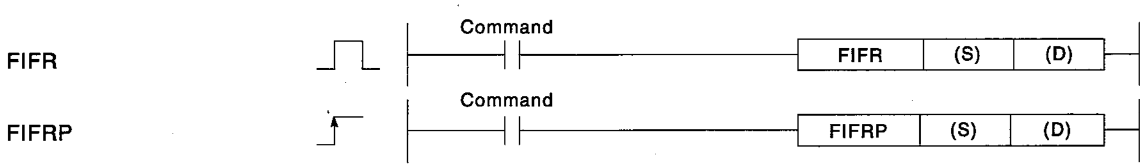
Steps	Instruction	Device
0	LD>=	D38 K6
3	OUT	Y60
4	LD	X1B
5	ANI	Y60
6	FIFWP	K4X20 D38
9	END	



7.7.2 Reading oldest data from tables

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	o	o		o				—	
(D)	—	o		—				—	

[Instruction Symbol] [Execution Condition]

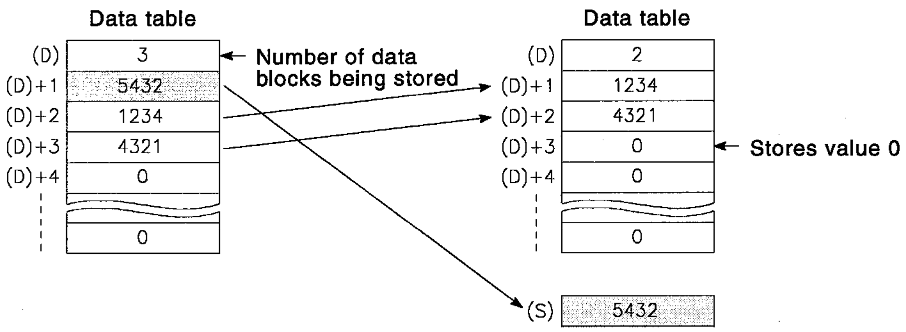


Set Data

Set Data	Meaning	Data Type
(S)	Initial number of storage device for data read from table	BIN 16 bits
(D)	Initial number of table	

Functions

(1) Stores the oldest data ((D)+1) input to the table designated by (D) at the device designated by (S).
After the execution of the FIFR instruction, the data in the table is all compressed up by one block.



(2) Users should attempt to avoid executing the FIFR instruction if the value stored at (D) is 0. (See Program Example (1))

Operation Errors

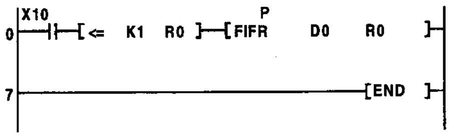
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The FIFR instruction was executed when the value at (D) was 0.
(Error code: 4100)

Program Example

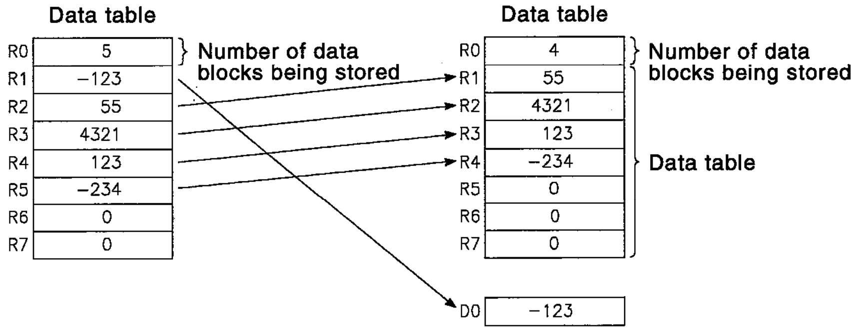
(1) The following program stores the data being held at R1 in the data table occupying from R0 to R7 at the destination D0 when X10 goes ON.

[Ladder Mode]



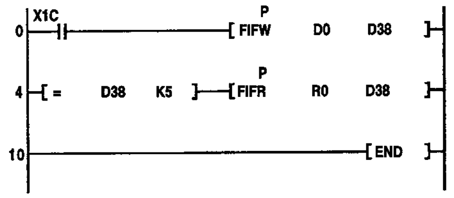
[List Mode]

Steps	Instruction	Device
0	LD	X10
1	AND<=	K1 R0
4	FIFRP	D0 R0
7	END	



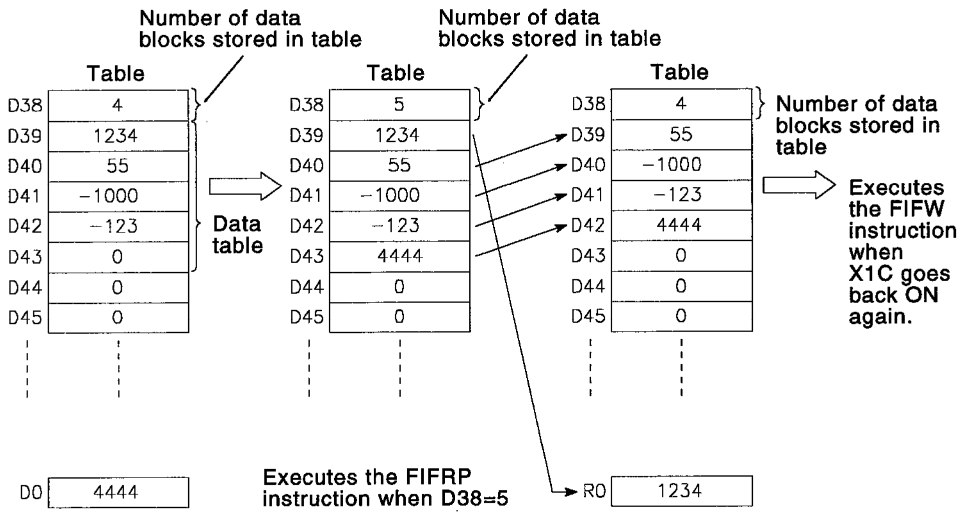
(2) The following program stores the data at D0 in the data table occupying from D38 to D43, and, when the table has 5 data entries, stores the data being held at D39 of the data table at R0, when X1C goes ON.

[Ladder Mode]



[List Mode]

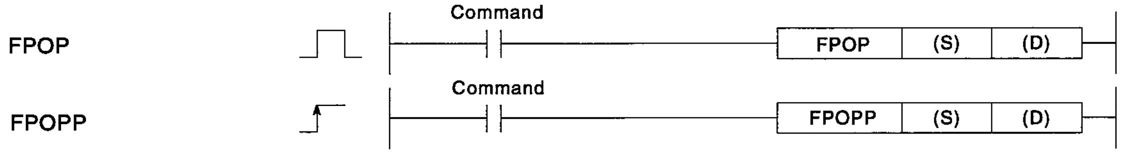
Steps	Instruction	Device
0	LD	X1C
1	FIFWP	D0 D38
4	LD=	D38 K5
7	FIFRP	R0 D38
10	END	



7.7.3 Reading newest data from data tables

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	o	o		o					—
(D)	—	o		—					—

[Instruction Symbol] [Execution Condition]

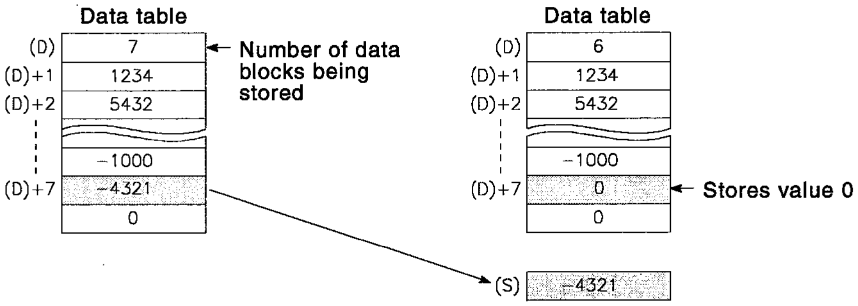


Set Data

Set Data	Meaning	Data Type
(S)	Initial number of storage device for data read from table	BIN 16 bits
(D)	Initial number of table	

Functions

(1) Stores the newest data input to the table designated by (D) at the device designated by (S).
After the execution of the FPOP instruction, the device holding the data read by the FPOP instruction is reset to 0.



(2) Users should attempt to avoid executing the FPOP instruction if the value stored at (D) is 0. (See Program Example (1))

Operation Errors

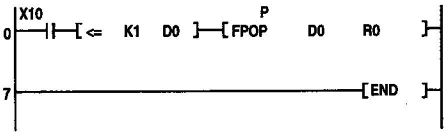
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The FPOP instruction was executed when the value of (D) was 0. (Error code: 4100)

Program Example

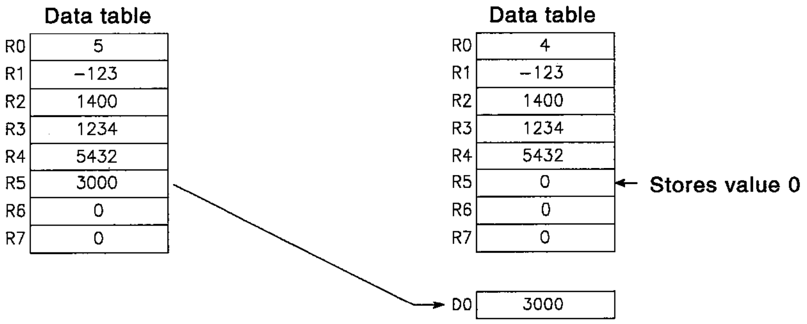
(1) The following program stores the last data that had been stored in the data table occupying from R0 to R7 at D0 when X10 goes ON.

[Ladder Mode]



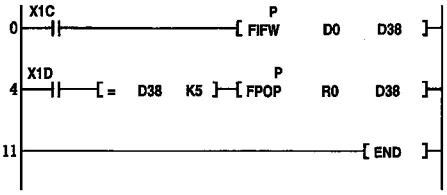
[List Mode]

Steps	Instruction	Device
0	LD	X10
1	AND<=	K1 D0
4	FPOPP	D0 R0
7	END	



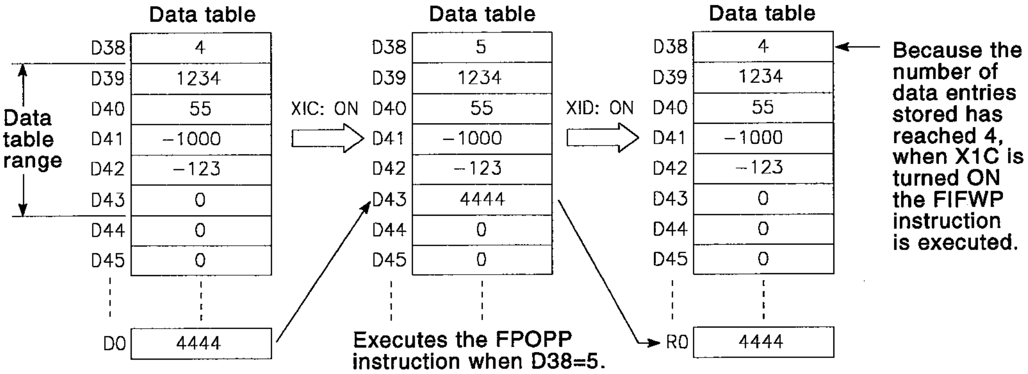
(2) The following program stores the data that had been at D0 in the data table occupying from D38 through D43 when X1C goes ON, and when the number of data entries in the table reaches 5, turns X1D ON, and stores the last data stored in the data table at R0.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	FIFWP	D0 D38
4	LD=	X1D
5	AND=	D38 K5
8	FPOPP	R0 D38
11	END	



7.7.4 Deleting and inserting data from and in data tables

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		o				—	—
(D)	—	o		—				—	—
n	o	o		o				o	—

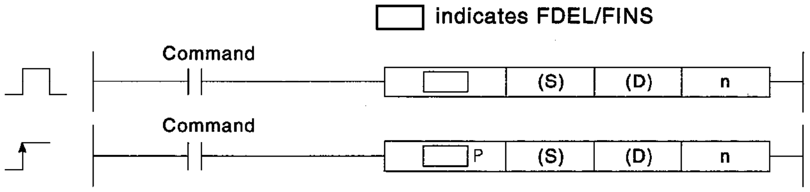
[Instruction Symbol] [Execution Condition]

FDEL

FINS

FDELP

FINSP



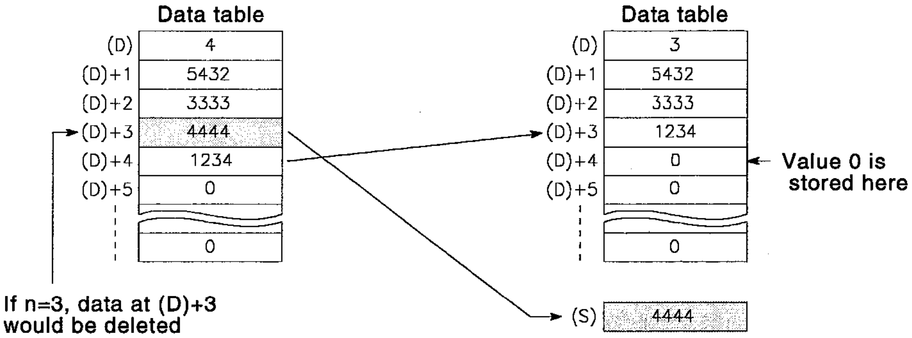
Set Data

Set Data	Meaning	Data Type
(S)	Inserted data or first device number of devices that will store inserted data First device number of devices that will store deleted data	BIN 16 bits
(D)	Initial number of table	
n	Table position for data for insertion or deletion	

Functions

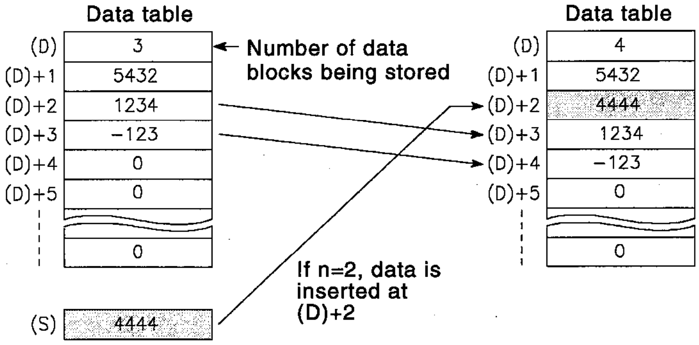
FDEL

(1) Deletes the nth block of data from the data table designated by (D), and stores it at the device designated by (S).
After the execution of the FDEL instruction, the data in the table following the deleted block is compressed forward by one block.



FINS

(1) Inserts the 16-bit data designated by (S) at the nth block of the data table designated by (D).
After the execution of the FINS instruction, the data in the table following the inserted block is all dropped one position.



Operation Errors

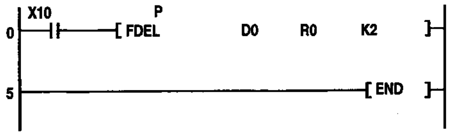
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The FDEL instruction has been executed even though there is no data at the nth position from (D). (Error code: 4100)
- The nth position from (D) in the case of the FINS instruction is larger than the actual number of data blocks in the table plus 1. (Error code: 4101)
- The value of n in the case of the FINS instruction exceeds the device range of the table (D). (Error code: 4101)

Program Example

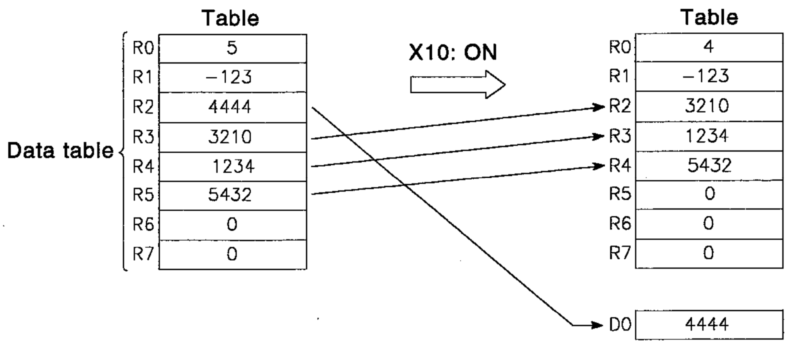
(1) The following program deletes the data from the 2nd position of the table occupying from R0 to R7, and stores this data at D0 when X10 goes ON.

[Ladder Mode]



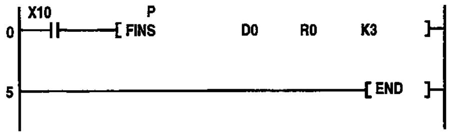
[List Mode]

Steps	Instruction	Device
0	LD	X10
1	FDELP	D0 R0 K2
5	END	



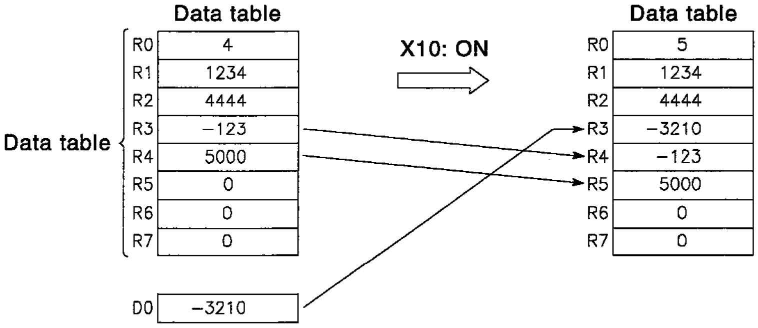
(2) The following program inserts the data at D0 at the 3rd block of the table occupying from R0 to R7 when X10 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X10
1	FINSP	D0 R0 K3
5	END	



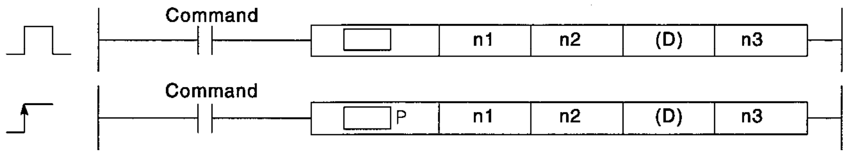
7.8 Buffer Memory Access Instruction

7.8.1 Special function modules 1- and 2-word data read operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
n1	o			o				o	o
n2	o			o				o	—
(D)	o			—				o	—
n3	o			o				o	—

[Instruction Symbol] [Execution Condition] □ indicates FROM/DFRO

FROM
DFRO
FROMP
DFROP



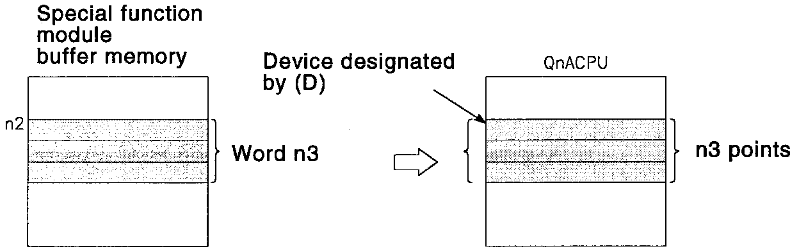
Set Data

Set Data	Meaning	Data Type
n1	Head I/O number of special function module	BIN 16 bits
n2	First address of data to read	BIN 16/32 bits
(D)	First device number of devices which will store read data	BIN 16 bits
n3	Number of read data • FROM(P) : 1 to 6144 • DFRO(P) : 1 to 3072	

Functions

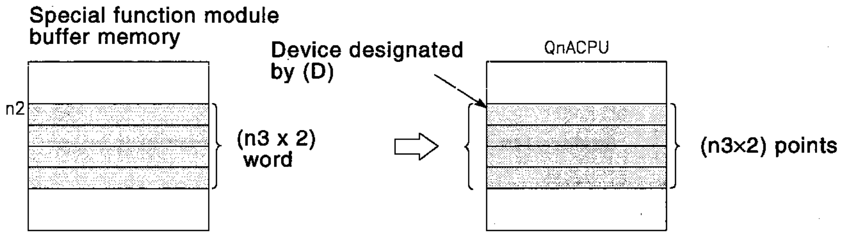
FROM

(1) This instruction is used to read word data as designated by n3 from the address specified by n2 in the buffer memory of the special function module specified by n1, then to store this data in devices starting from that designated by (D)



DFRO

(1) Reads (n3 x 2) words at the address designated by n2 from the buffer memory in the special function module designated by n1, and stores it in devices following that designated by (D).



POINT

Special function module devices can also be used to read data from special function module buffer memories.
See the QnACPU Programming Manual (Fundamentals) for further information concerning the special function modules.

Operation Errors

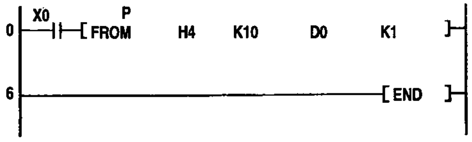
In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- There has been no exchange of signals with the special function module prior to the execution of the instruction. (Error code: 1412)
- An error has been detected in the special function module prior to the execution of the instruction. (Error code: 1402)
- The I/O number designated by n1 is not a special function module. (Error code: 2110)
- The n3 points of the device designated by (D) (2 x n3 points for DFRO) exceed the designated device range. (Error code: 4101)
- The address designated by n2 is outside the buffer memory range. (Error code: 4100)
- The address designated by n2 is an odd number address. (AJ71QC24N) (Error code: 4100)
- The number of read data exceeds 6144/3072. (Error code: 4101)

Program Example

(1) The following program example reads the digital value of CH1 of the A68AD mounted at I/O numbers 040 to 05F when X0 goes ON. (Reads 1 word of data from address 10 of the buffer memory)

[Ladder Mode]

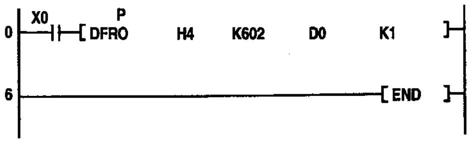


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	FROMP	H4
		K10
		D0
6	END	K1

(2) The following program reads the current value of the X-axis of the AD71 mounted at I/O numbers 040 to 05F to D0 and D1 when X0 is ON. (Reads 2 words of data from address 602, 603 of the buffer memory)

[Ladder Mode]

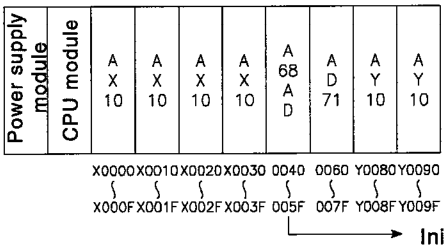


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	DFROP	H4
		K602
		D0
		K1
6	END	

REMARK

The value of n1 designates the upper 3 digits when the head I/O number of the slot in which the special function module is mounted is expressed in 4 digits in hexadecimal notation.



Initial input number K4 or H4 of the data read

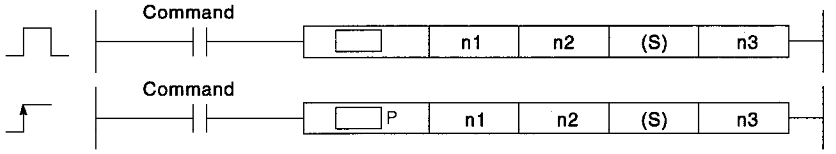
7.8.2 Special function modules 1- and 2-word data write operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
n1	o			o				o	o
n2	o			o				o	—
(D)	o			—				o	—
n3	o			o				o	—

[Instruction Symbol] [Execution Condition] □ indicates TO/DTO

TO
DTO

TOP
DTOP



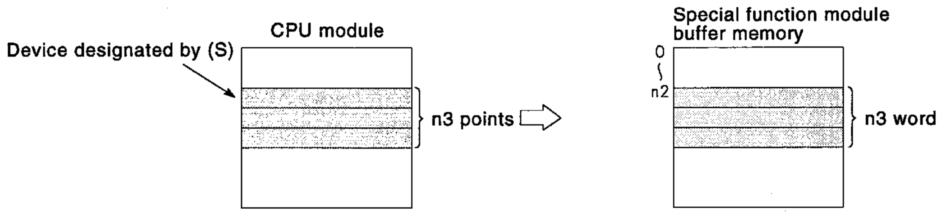
Set Data

Set Data	Meaning	Data Type
n1	Head I/O number of special function module	BIN 16 bits
n2	First address for the data write operation	
(S)	First device number of devices storing data to write	BIN 16/32 bits
n3	Write number of data • TO(P) : 1 to 6144 • DTO(P) : 1 to 3072	BIN 16 bits

Functions

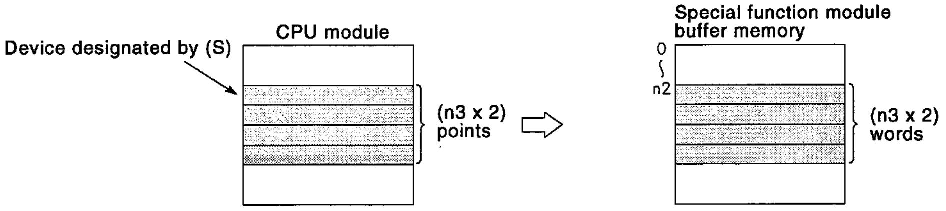
TO

(1) Writes data from the n3 point from the device designated by (S) to the area following the address designated by n2 of the buffer memory in the special function module designated by n1.



DTO

Writes the data of (n3 x 2) devices, starting from that designated by (S), to addresses starting from that designated by n2 in the buffer memory of the special function module designated by n1.



POINT

Data can be written from the buffer memory of a special function module by using special function module devices.
See the QnACPU Programming Manual (Fundamentals) for further information concerning the special function modules.

Operation Errors

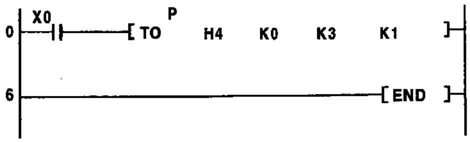
In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- There has been no exchange of signals with the special function module prior to the execution of the instruction. (Error code: 1402)
- An error has been detected in the special function module prior to the execution of the instruction. (Error code: 1402)
- The I/O number designated by n1 is not a special function module. (Error code: 2110)
- The n3 points of the device designated by (S) (2 x n3 points for DT0) exceed the designated device range. (Error code: 4101)
- The address designated by n2 is outside the buffer memory range. (Error code: 4100)
- The address designated by n2 is an odd number address. (AJ71QC24N) (Error code: 4100)
- The number of written data exceeds 6144/3072. (Error code: 4101)

Program Example

(1) The following program sets CH1 and CH2 of the A68AD mounted at I/O numbers 040 to 05F to "A/D conversion" when X0 is ON. (Writes "3" at buffer memory address.)

[Ladder Mode]

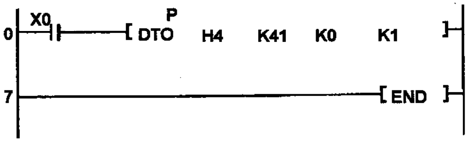


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	TOP	H4
		K0
		K3
6	END	K1

(2) The following program sets the current value of the X-axis of the AD71 mounted from 040 to 05F at 0 when X0 is ON. (Writes 0 at addresses 41 and 42 of the buffer memory.)

[Ladder Mode]

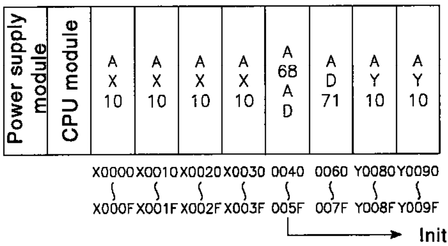


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	DTOP	H4
		K41
		K0
7	END	K1

REMARK

The argument n1 specifies the head I/O number of the slot in which the special function module is mounted, as the upper three digits when the number is expressed as a 4-digit hexadecimal value.



Initial input number K4 or H4 of the data read

7.9 Display Instructions

7.9.1 Print ASCII code instruction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—			o	o	—
(D)	o (Only Y)	—		—			o	—	—

[Instruction Symbol] [Execution Condition]

PR

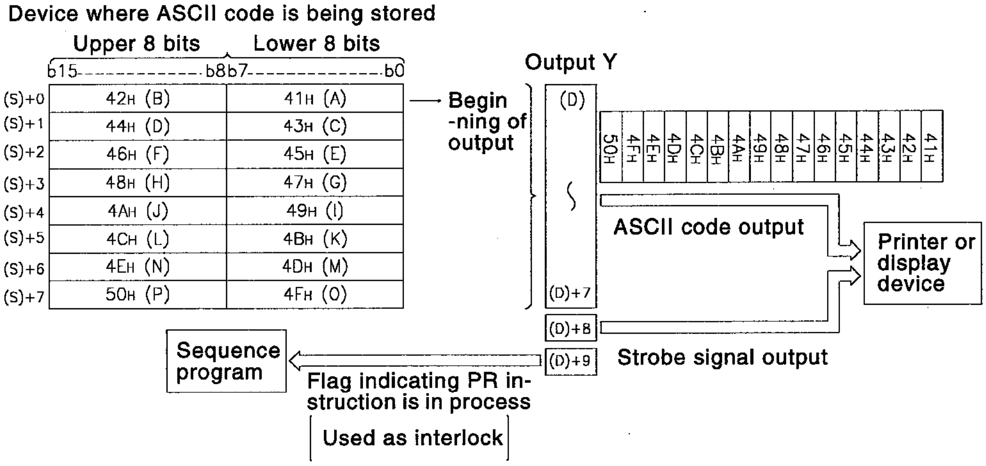
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where ASCII code is being stored	Character string
(D)	Initial number of output module to output ASCII code	Bit

Functions

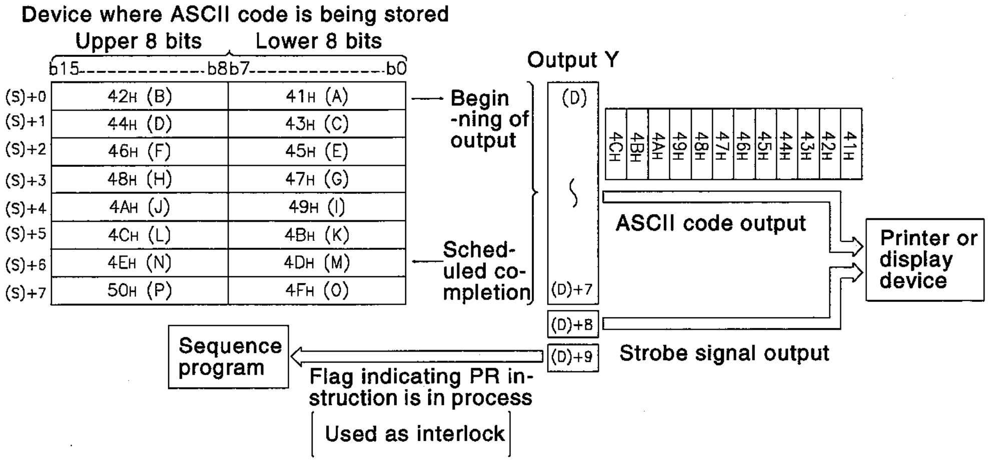
(1) Outputs ASCII code stored following the device designated by (S) to an output module designated by (D).
The number of characters output differs according to the ON/OFF status of SM701.

(a) If SM701 is ON, characters 8 points (16 characters) from the device designated by (S) will be operated on.



	b15 - b8	b7 - b0
(S)+0	42H (B)	41H (A)
(S)+1	44H (D)	43H (C)
(S)+2	46H (F)	45H (E)
(S)+3	48H (H)	47H (G)
(S)+4	4AH (J)	49H (I)
(S)+5	4CH (L)	4BH (K)
(S)+6	4EH (N)	4DH (M)
(S)+7	50H (P)	4FH (O)

(b) If SM701 is OFF, everything from the device designated by (S) to the 00H code will be the focus of the operation.



Device where ASCII code is being stored

	Upper 8 bits (b15-b8)	Lower 8 bits (b7-b0)
(S)+0	42H (B)	41H (A)
(S)+1	44H (D)	43H (C)
(S)+2	46H (F)	45H (E)
(S)+3	48H (H)	47H (G)
(S)+4	4AH (J)	49H (I)
(S)+5	4CH (L)	4BH (K)
(S)+6	4EH (N)	4DH (M)
(S)+7	50H (P)	4FH (O)

(2) The number of points used by the output module is 10 points from the Y address designated by (D).

(3) Output signals from the output module are transmitted at the rate of 30 ms per character.
For this reason, the time required to the completion of the transmission of the designated number of characters (n) will be 30 ms x n (ms).
The PR instruction is based on a 10 ms interrupt, and processing includes data output, strobe signal ON, strobe signal OFF, and processing and processing time is thus continuous with the execution of other instructions.

(4) In addition to the ASCII code, the output module also outputs a strobe signal (10 ms ON, 20 ms OFF) from the (D)+8 device.

(5) Following the execution of the PR instruction, the PR instruction execution flag ((D)+9 device) remains ON until the completion of the transmission of the designated number of characters.

(6) The PR and PRC instructions can be used multiple times, but it is preferable to establish an interlock with the PR instruction execution flag ((D)+9 device) so they will not be ON simultaneously.

(7) If the contents of the device where the ASCII code is being stored are changed during the output of the ASCII code, the updated data will be output.

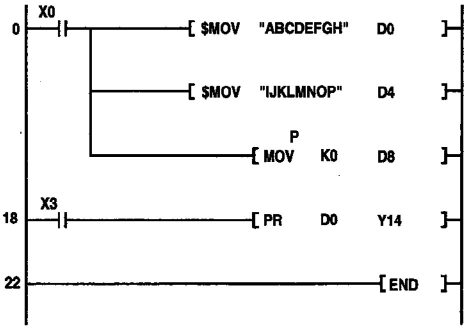
Operation Errors

(1) There are no operation errors associated with the PR instruction.

Program Example

(1) The following program converts the string "ABCDEFGHIJKLMNOP" to ASCII code when X0 goes ON and stores it from D0 through D7, and then outputs the ASCII code at D0 through D7 to Y14 through Y1D when X3 goes ON.

[Ladder Mode]

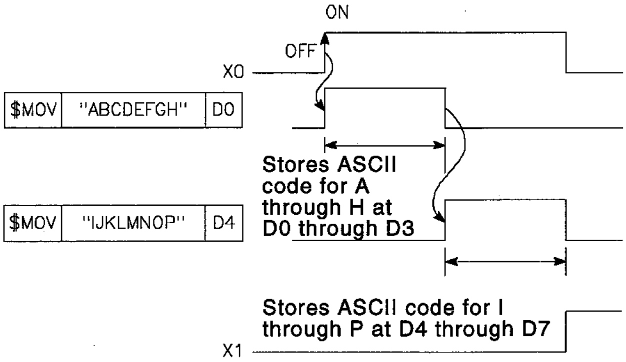


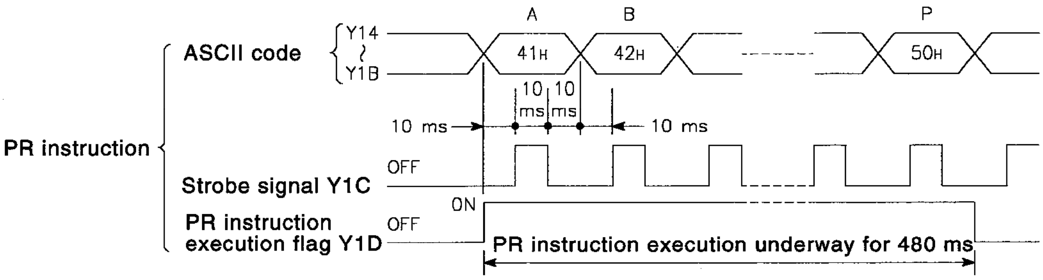
Converts "ABCDEFGHIJKLMNOP" to ASCII code when X0 goes ON, and stores it from D0 through D7.

Outputs the ASCII code at D0 through D7 to Y14 through Y1D when X1 goes ON.

[List Mode]

Steps	Instruction	Device
0	LD	X0
1	$MOV	"ABCDEFGH" D0
8	$MOV	"IJKLMNOP" D4
15	MOVP	K0 D8
18	LD	X0
19	PR	D0 Y14
22	END	



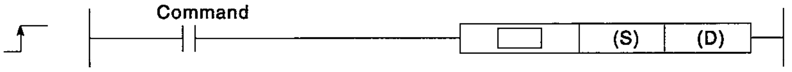


7.9.2 Print comment instruction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				P, I, J, U
(S)	o	o		o			—	—	o
(D)	o (Only Y)	—		—			—	—	—

[Instruction Symbol] [Execution Condition]

PRC



Set Data

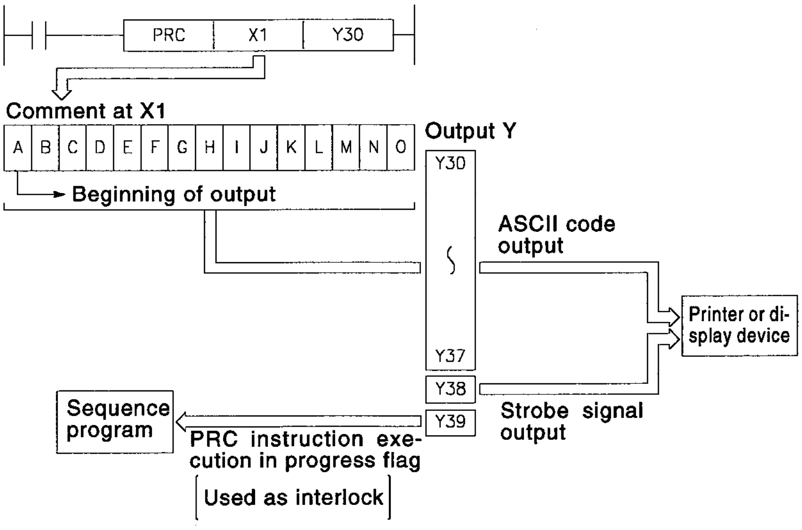
Set Data	Meaning	Data Type
(S)	First device number of devices that will print comment	
(D)	Head I/O number of the output module that will output comment	

Functions

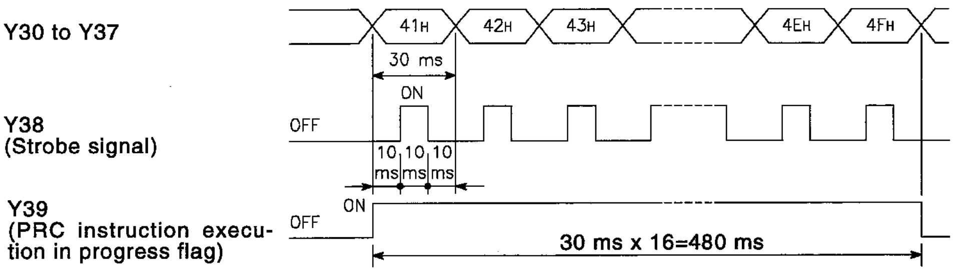
(1) Outputs comment (ASCII code) at device designated by (S) to output module designated by (D).
The number of characters output differs according to the ON/OFF status of SM701.

- When SM701 is OFF: Comment is 32 characters
- When SM701 is ON : Comment is the upper 16 characters

The number of points used by the output module is 10 points from the Y address designated by (D).



[Timing Chart]



(2) Output signals from the output module are transmitted at the rate of 30 ms per character.
For this reason, the time required to the completion of the transmission of the designated number of characters will be 30 ms x n (ms).
The PRC instruction is based on a 10 ms interrupt, and processing includes data output, strobe signal ON, strobe signal OFF, and processing and processing time is thus continuous with the execution of other instructions.

(3) In addition to the ASCII code, the output module also outputs a strobe signal (10 ms ON, 20 ms OFF) from the (D)+8 device.

(4) Following the execution of the PRC instruction, the PRC instruction execution flag ((D)+9 device) remains ON until the completion of the transmission of the designated number of characters.

(5) The PRC instruction can be used multiple times, but it is preferable to establish an interlock with the PRC instruction execution flag ((D)+9 device contact) so they will not be ON simultaneously.

(6) If no comments have been registered at the device designated by (S), processing will not be performed.

POINTS

(1) Device comments used by the PRC instruction use the comment file stored in the IC memory card.
They cannot use the comment file in internal memory.

(2) The comment file used by the PRC instruction is set at the "PC File Setting" option in the parameter mode.
If no comment file has been set for use by the PC file setting, it will not be possible to output device comments with the PRC instruction.

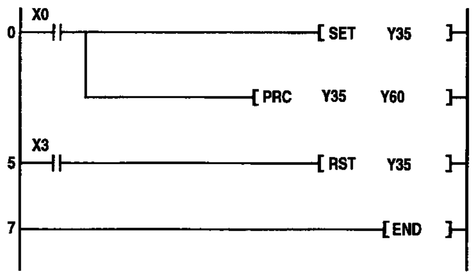
Operation Errors

(1) There are no operation errors associated with the PRC instruction.

Program Example

The following program turns Y35 ON and at the same time outputs the comment at Y35 to Y60 through Y69.

[Ladder Mode]



[List Mode]

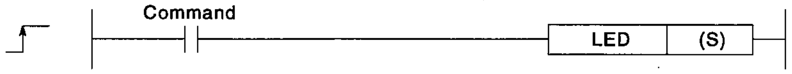
Steps	Instruction	Device
0	LD	X0
1	SET	Y35
2	PRC	Y35 Y60
5	LD	X3
6	RST	Y35
7	END	

7.9.3 ASCII code LED display instruction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—			—	o	—

[Instruction Symbol] [Execution Condition]

LED

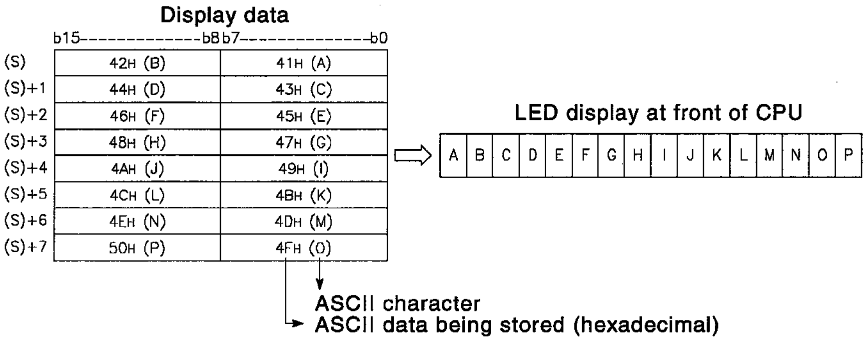


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where display data is being stored	Character string

Functions

(1) Displays the ASCII data (16 characters) being stored 8 points from the device designated by (S) at the LED display at the front of the CPU.



	b15 – b8	b7 – b0
(S)	42H (B)	41H (A)
(S)+1	44H (D)	43H (C)
(S)+2	46H (F)	45H (E)
(S)+3	48H (H)	47H (G)
(S)+4	4AH (J)	49H (I)
(S)+5	4CH (L)	4BH (K)
(S)+6	4EH (N)	4DH (M)
(S)+7	50H (P)	4FH (O)

(2) If no ASCII data is being stored at the 8 points from the device designated by (S), the following will take place:

(a) T, C, D, W.... Blank

(b) R............ Impossible to say what will be displayed here (Will be blank if the file register (R) has been cleared)

(3) The following characters can be displayed by the LED display:

- Numeral 0 to 9
- Letters A through Z (upper case letters)
- Special symbols <, >, =, *, /, ,,, +, -

(4) Use the $MOV instruction in sequence programs to convert alphanumeric ASCII data.

(5) If the LED instruction is executed by the Q2ACPU(S1), it will return no operation.

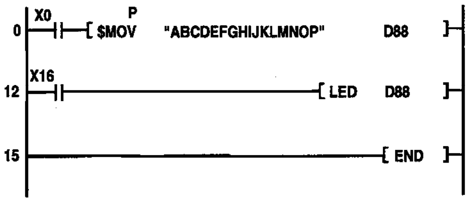
Operation Errors

(1) There are no operation errors associated with the LED instruction.

Program Example

(1) The following program converts the string "ABCDEFGHIJKLMNOP" to ASCII code when X8 goes ON and stores it from D88 through D95, and then outputs the ASCII code at D88 through D95 to the LED display at the front of the CPU when X16 goes ON.

[Ladder Mode]



Converts the 16 characters from A through P to ASCII code and stores from D88 through D95.

Displays the ASCII data from D88 through D95 at the LED display.

[List Mode]

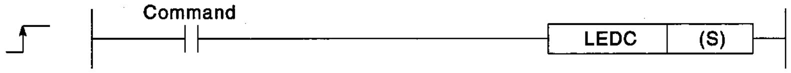
Steps	Instruction	Device
0	LD	X8
1	$MOVP	"ABCDEFGHIJKLMNOP" D88
12	LD	X16
13	LED	D88
15	END	

7.9.4 LED display instruction for comments

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				BL, BL\S, BL\TR
(S)	o	o		o			—	—	o

[Instruction Symbol] [Execution Condition]

LEDC



Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices to display comment	Device name

Functions

(1) Displays the comment (16 characters) being stored at device designated by (S) at the LED display at the front of the CPU.

(2) If there are no comments attached to the device designated by (S), or if the comments are outside the comment range designation, the following will take place:

Designated by (S)		LED Operation
Within the comment range	Comment attached	Displays comment at LED display
	Comment not attached	Clears the LED display
Outside the comment range		No operation (LED display does not change)

(3) An accurate display will not be made if the comment includes characters that cannot be displayed at the LED display.
The following characters can be displayed by the LED display:

- Numerals 0 to 9
- Letters A through Z (upper case letters)
- Special symbols <, >, =, *, /, ,, +, -

(4) The LED display can display up to 16 characters.
If there are more than 16 characters in the device comment, the first 16 characters in the comment will be displayed.

(5) If the operation is executed by the Q2A(-S1), it will return no operation.

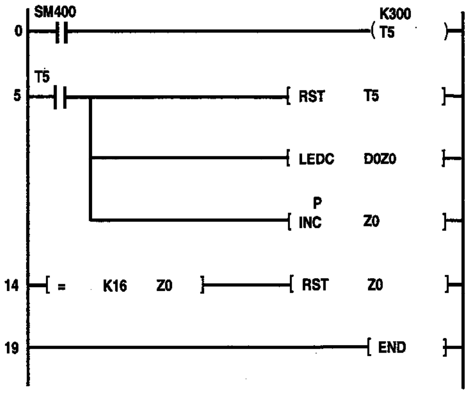
Operation Errors

(1) There are no operation errors associated with the LEDC instruction.

Program Example

(1) The following program displays comments from D0 through D15 in 30-second intervals.

[Ladder Mode]



Counts 30 seconds.

Displays the comments fromD(0+Z0) at the LED display in front of the CPU.

Z+1 is executed when T5 goes from OFF to ON

Z0 is set to 0 when it reaches 16.0 seconds.

[List Mode]

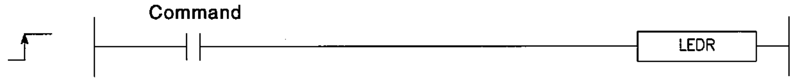
Steps	Instruction	Device
0	LD	SM400
1	OUT	T5 K300
5	LD	T5
6	RST	T5
10	LEDC	D0Z0
12	INCP	Z0
14	LD=	K16 Z0
17	RST	Z0
19	END	

7.9.5 Error display and annunciator reset instruction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]

LEDR



Functions

Resets the self-diagnosis error display so that annunciator display or operation can be continued.
With one execution of this instruction, either error display or annunciator is reset.

(1) Operation when self-diagnosis error is generated

(a) If the self-diagnosis errors is one which allows continued operation
If the self-diagnosis error being displayed is one that will allow continued operation of the CPU, the "ERROR" LED at the front of the CPU and the error display will be reset.
It will be necessary to reset SM1 and SD0 at the user program, because they do not reset automatically at these times.
Since the cause of the error displayed at this time has a higher priority over annunciator, no action for resetting the annunciator is taken.

(b) When a battery error has been generated
If the LEDR instruction is executed after the battery has been replaced, the "BAT. ARM" LED at the front of the CPU and the error display will be reset.
SM51 goes OFF at this time.

(2) Operations when an annunciator (F) is ON.

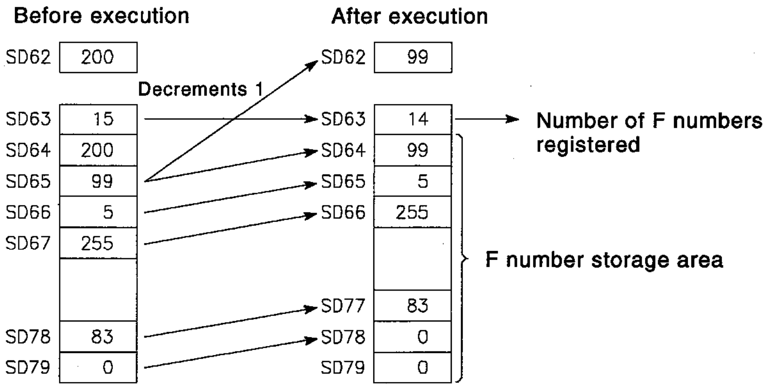
(a) When the CPU has no LED display
The following operations will be conducted when the LEDR instruction is executed:

1) "USER" LED flickers, and is turned off

2) The annunciators (F) stored in SD62 and SD64 are reset, and the F numbers for SD65 through SD79 are moved up.

3) The data newly stored at SD64 is transmitted to SD62.

4) The data at SD63 is decremented by 1.
   However, if SD63 is 0, it is left at 0.



(b) For CPUs with an LED display at the front
The following operations will be conducted when the LEDR instruction is executed:

1) The F number being displayed at the front of the CPU will be reset.

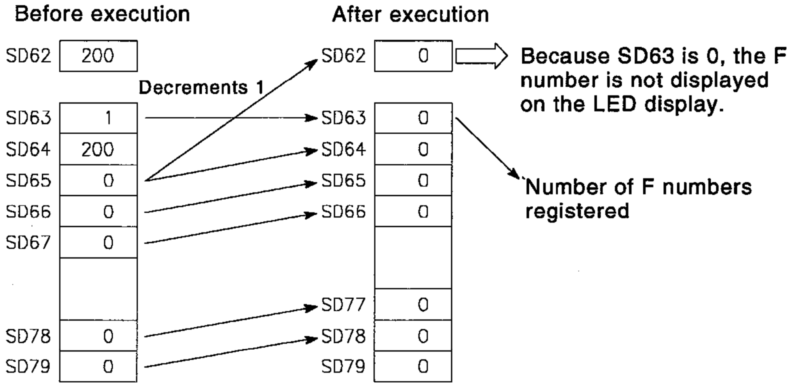
2) "USER" LED flickers, and is turned off

3) The annunciators (F) stored in SD62 and SD64 are reset, and the F numbers for SD65 through SD79 are compressed forwards.

4) The data newly stored at SD64 is transmitted to SD62.

5) The data at SD63 is decremented by 1.
   However, if SD63 is 0, it is left at 0.

6) The F number being stored at SD62 is displayed at the LED display.
   However, if the value of SD63 is 0, nothing will be displayed.



REMARKS

(1) The defaults for the error item numbers set in special registers SD207 to SD209 and order of priority are given in the table below.

Order of Priority	Error Item No. (Hex.)	Description	Remark
1	1	AC DOWN	Power supply cut
2	2	UNIT VERIFY ERR. FUSE BREAK OFF P. UNIT ERROR	I/O module verify error Blown fuse Special function module verify error
3	3	OPERATIN ERROR LINK PARA. ERROR SFCP OPE. ERROR SFCP EXE. ERROR	Operation error Link parameter error SFC instruction operation error SFC program execution error
4	4	ICM.OPE. ERROR FILE OPE. ERROR EXTEND INST. ERROR	Memory card operation error File access error Extend instruction error
5	5	PRG.TIME OVER	Constant scan setting time over error Low-speed execution monitoring tome over error
6	6	CHK instruction	
7	7	Annunciators	
8	8	LED instruction	
9	9	BATTERY ERR.	
10	A	Clock data	

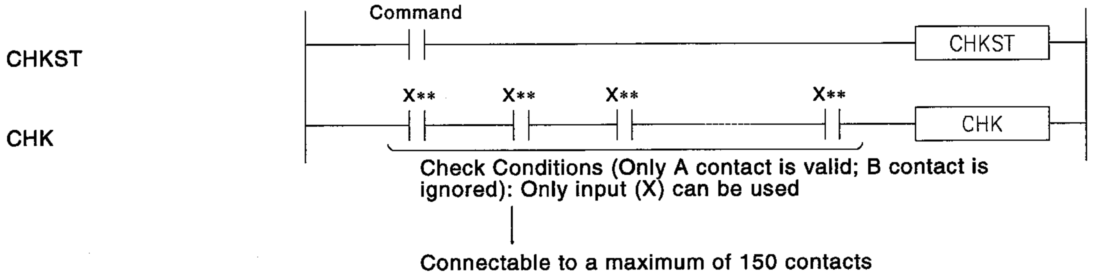
(2) If the highest priority is given to the annunciator, it can be reset with priority by the LEDR instruction.

7.10 Debugging and Failure Diagnosis Instructions

7.10.1 Special format failure checks

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]



Check Conditions (Only A contact is valid; B contact is ignored): Only input (X) can be used

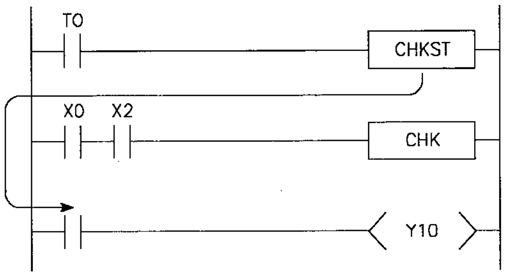
Connectable to a maximum of 150 contacts

Functions

CHKST

(1) The CHKST instruction is the instruction that starts the CHK instruction.
If the command for the CHKST instruction is OFF, execution jumps to the step following the CHK instruction.
If the command for the CHKST instruction is ON, the CHK instruction is executed.

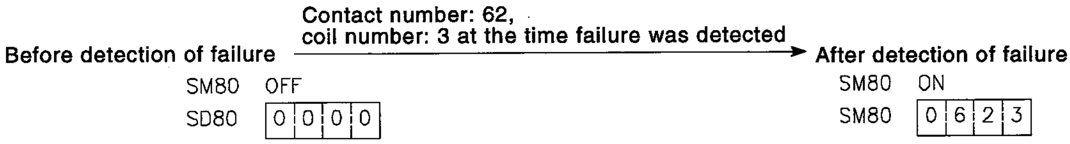
When M0 is OFF, program jumps to the instruction following the CHK instruction.



CHK

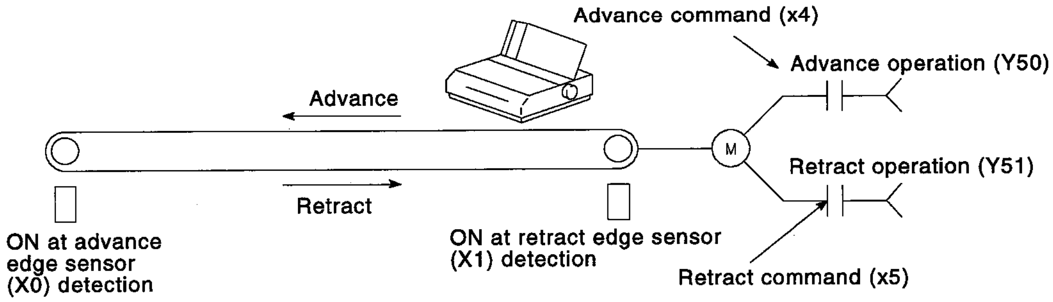
(1) The CHK instruction is the instruction used for the bidirectional operation as shown on the following page to confirm the nature of the system failure.

(a) When the CHK instruction is executed, a failure diagnosis check is conducted with the designated check conditions, and if a failure is detected, SM80 is turned ON, and the failure number is stored at SD80 as a BCD value.
The contact number where the failure was discovered is stored at the upper 3 digits of SD80 (see (3)), and the coil number where the failure was detected (see (2)) is stored at the lower digit of SD80.

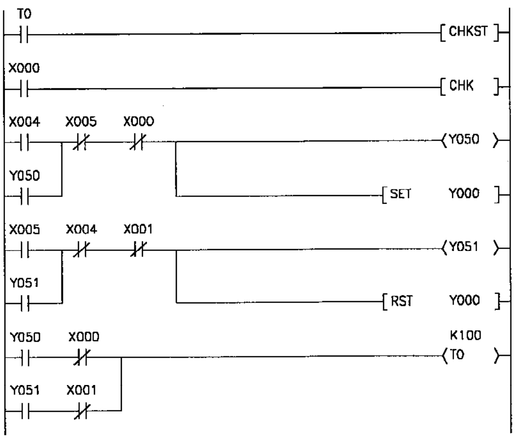


(b) The contact instruction prior to the CHK instruction does not control the execution of the CHK instruction, but rather sets the check conditions.

Example



(c) A ladder such as the one shown below can be created to perform a cycle time over check for the system shown above:



If T0 is OFF, program jumps to the instruction following the CHK instruction.
If T0 is ON, CHK instruction is executed.

Conveyor 1 advances

Internal relay used to detect failures is turned ON

Conveyor 1 retracts

Internal relay used to detect failures is turned OFF

Cycle time check timer

(d) The following points should be taken into consideration when creating a ladder for use with the CHK instruction:

1) The contact numbers for the advance edge detection sensor and the retract edge detection sensor (X□) must always be continuous. Further, the contact number (X□) for the advance edge detection sensor should be lower than that for the retract edge.

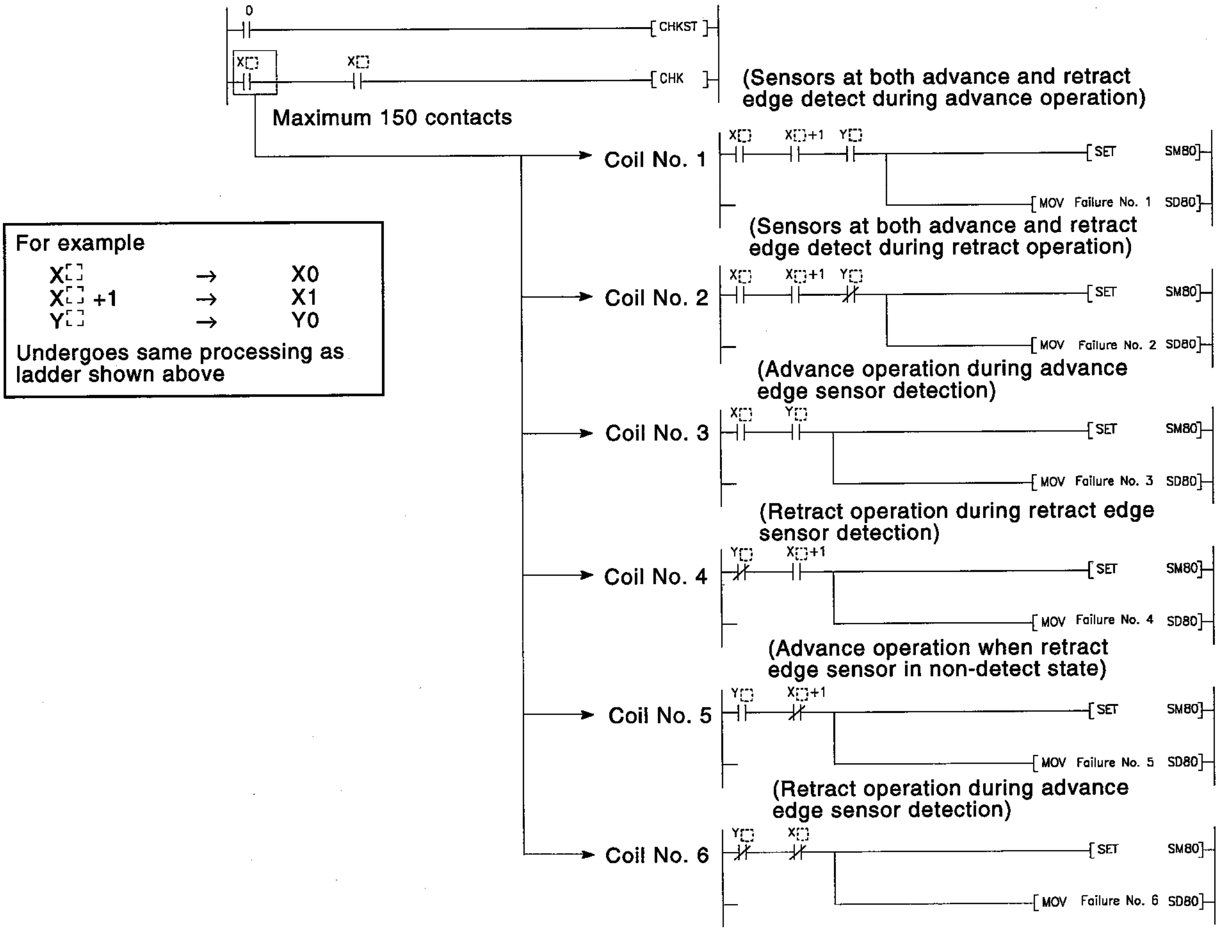
2) Controls for the advance edge detection sensor contact number (X□) and output with the identical number (Y□)* are as follows:

When advance operation is in progress . . . . . turn ON
When retract operation is in progress . . . . . . turn OFF

REMARK

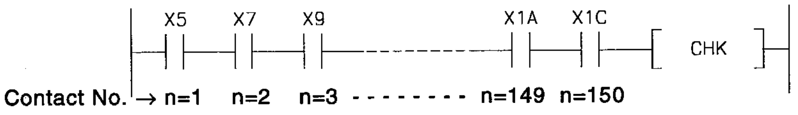
*: Output (Y□) is treated as an internal relay, and cannot be output to an external device.

(2) Depending on the designated contact, the CHK instruction undergoes processing identical to that shown for the ladder below:



For example
X□ → X0
X□ +1 → X1
Y□ → Y0
Undergoes same processing as ladder shown above

(3) Numbers 1 through 150 from the vertical bus on the left side have been allocated as contact numbers during failure detection.

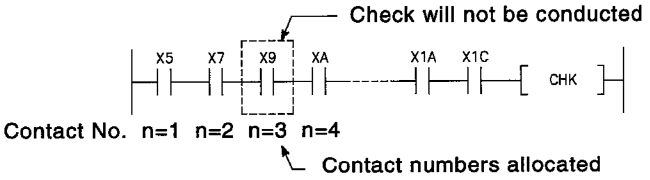


(4) Reset SM80 and SD80 prior to forcing the execution of the CHK instruction.
After the execution of the CHK instruction, it cannot be performed once again until SM80 and SD80 have been reset.
(The contents of SM80 and SD80 will be preserved until reset by user.)

(5) A CHKST instruction must be placed before the CHK instruction.

(6) The CHK instruction can be written at any step of the program.
It can be used in up to two locations in all program files being executed.
However, the CHK instruction cannot be used in more than 2 locations in a single program file.

(7) Place LD and AND instructions prior to the CHK instruction to establish a check condition.
Check conditions cannot be set using other contact instructions.
If a check condition has been set with LDI or ANI, the processing for the check condition they specify will not be conducted.
However, contact numbers during failure detection can also be allocated to the LDI and ANI instructions.



(8) The failure detection method will differ according to whether SM710 is ON or OFF.

(a) If SM710 is OFF, checks will be conducted of coil numbers 1 through 6 for each contact successively.
When the CHK instruction is executed, checks will be in order from coil No. 1 of contact No. 1, through coil No. 6, then move on to contact No. 2 and check the coils in order from No. 1.
The CHK instruction will be completed when coil No. 6 from contact No. n has been checked.

(b) If SM710 is ON, checks will be conducted of contact numbers 1 through n, in coil number order.
When the CHK instruction is executed, checks will begin with the ladder for coil No. 1, in order from contact No. 1 until contact No. n, then move on to the coil No. 2 ladder and begin from contact No. 1.
The CHK instruction will be completed when a check has been made through contact No. n of coil No. 6.

(9) If more than one failure is detected, the number of the first failure detected will be stored.
Failure numbers detected after this will be ignored.

(10) The CHK instruction cannot be used by a low-speed execution program.
If a low-speed execution program has been set in a program file containing the CHK instruction, an operation error will be returned, and the QnACPU operation will be suspended.

Operation Errors

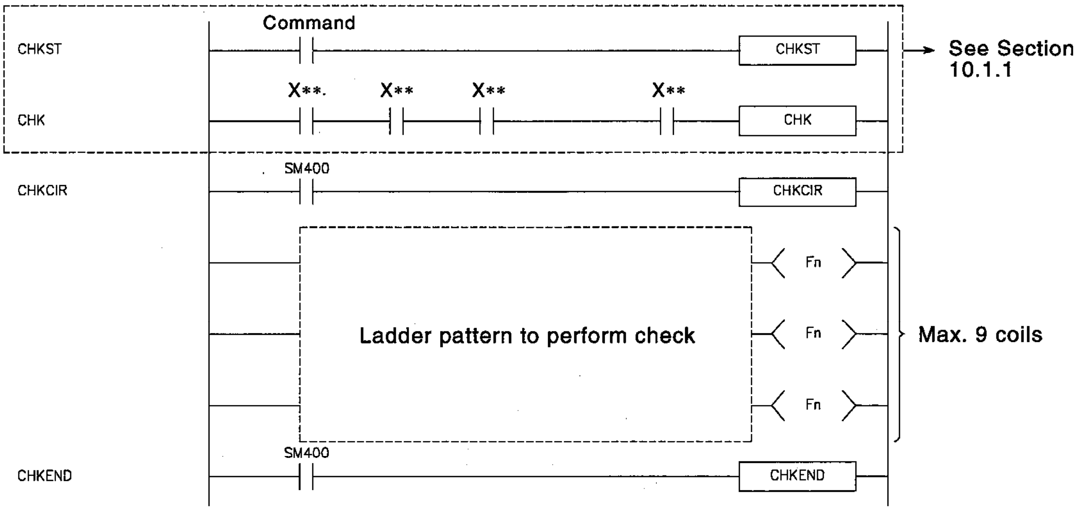
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- There is a parallel ladder (Error code: 4235)
- There is an NOP instruction (Error code: 4235)
- There are more than 150 contact instructions (Error code: 4235)
- A CHK instruction is not executed following the CHKST instruction (Error code: 4235)
- The CHK instruction is executed when no CHKST instruction has been executed (Error code: 4235)
- The CHKST and CHK instruction are used in a low-speed execution type program (Error code: 4235)

7.10.2 Changing check format of CHK instruction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]



Functions

CHKCIR, CHKEND

(1) The check ladder pattern that will be used in the CHK instruction can be updated to any format desired.
The actual failure checks are conducted with the CHKST and CHK instructions.

(2) Failure checks are conducted according to the check conditions designated by the CHK instruction and the ladder pattern described between the CHKCIR and CHKEND instructions.

REMARK

See Section 10.1.1 for more information on the CHKST and CHK instructions.

POINT

When using the CHKCIR to CHKEND instructions and changing the check format of the CHK instruction, it is necessary start the peripheral device in the "General Mode", to create a program, and to conduct program expansion.
If the peripheral device has been started up with a CPU model name (Q2A, Q2AS1, Q3A, or Q4A), and then an attempt is made to create a check format update ladder for the CHK instruction by using the CHKCIR to CHKEND instructions, accurate processing will not be possible.

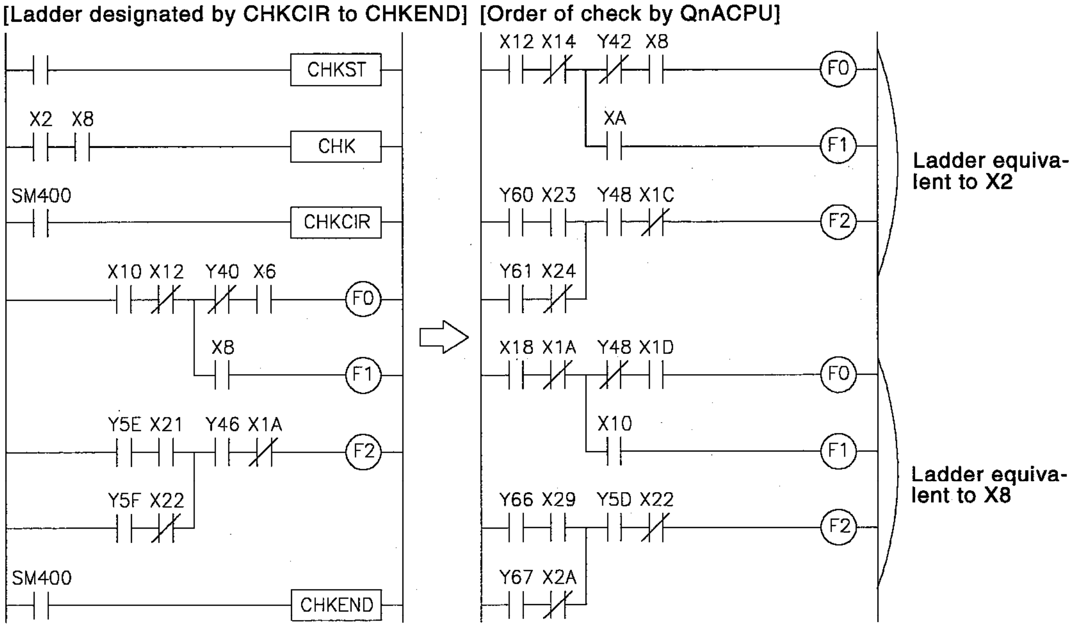
(a) The device numbers indicated at check conditions (X2 and X8 in the figure below) will assume index qualification values for the individual device numbers (with the exception of annunciators (F)) described in the ladder patterns.

Example: X10 in the [ ] in the figure below would be as follows:

When corresponding to check condition X2
.... Processing performed by X12
When corresponding to check condition X8
.... Processing performed by X18

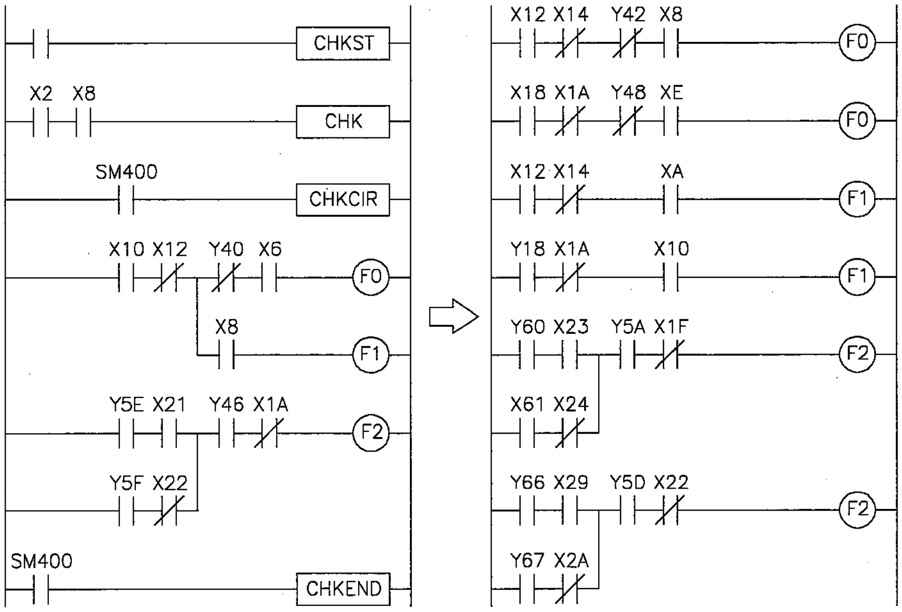
However, the order in which failure detection is carried out differs depending on whether SM710 is ON or OFF.

1) If SM710 is OFF, checks will be conducted of coil numbers 1 through the end for each contact successively.



2) If SM710 is ON, checks will be conducted of contact numbers 1 through the end, in coil number order.

[Ladder designated by CHKCIR to CHKEND] [Order of check by QnACPU]



(b) Failure checks check the ON/OFF status of OUT F□ by using the ladder pattern in the various check conditions.
In all check conditions, SM80 will be turned ON if even one of the OUT F□ is ON in a ladder pattern.
Further, the error numbers (contact numbers and coil numbers) corresponding to the OUT F□ which were found to be ON will be stored from SD80 in BCD order.

(c) The instructions that can be used in ladder patterns are as follows:

Contacts... LD, LDI, AND, ANI, OR, ORI, ANB, ORB, MPS, MPP,MRD, and comparative operation instructions

Coil........... OUT F□

(d) The following devices can be used for ladder pattern contacts:

Input (X), Output (Y)

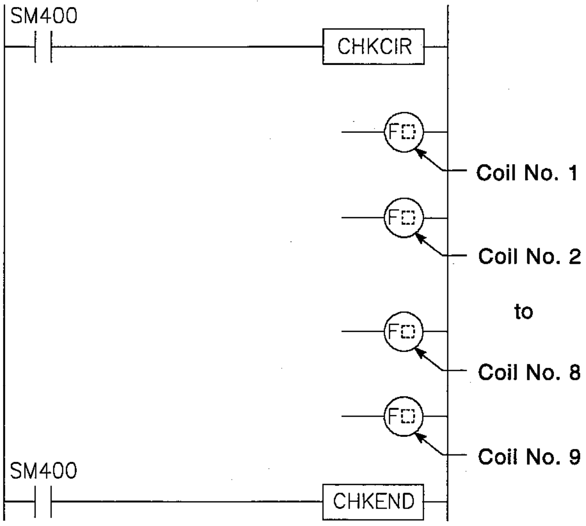
(e) Only annunciators (F) can be used in ladder pattern coils.
However, because they are used as dummies, any random setting from F0 can be used for them.
Further, they can overlap with no difficulties.

(f) ON/OFF controls can be performed without error if an annunciator (F) used during the execution of the CHK instruction has the same number as an annunciator (F) used in some other context than the CHK instruction.
They will be treated differently during the CHK instruction than they are in the different context.

(g) Because annunciators (F) used by the CHK instruction do not actually go ON or OFF, they will not go ON even when monitored by a peripheral device.

(h) A ladder pattern can be created with a maximum of 256 steps. Further, OUT F☐ can use up to a maximum of 9 coils.

(3) Coil numbers for ladders designated with the CHKCIR through CHKEND instructions are allocated coil numbers from 1 through 9, from top to bottom.



(4) The CHKCIR and CHKEND instructions can be written at any step in the program desired.
It can be used in up to two locations in all program files being executed. However, the CHKCIR and CHKEND instructions cannot be used in more than 1 location in a single program file.

(5) The CHKCIR and CHKEND instructions cannot be used in low-speed programs.
If a low-speed execution program has been set in a program file containing the CHKCIR or CHKEND instruction, an operation error will be returned, and the QnACPU operation will be suspended.

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The CHKCIR or CHKEND instruction appears three or more times in all program files. (Error code: 4235)
- The CHKCIR or CHKEND instruction appears two or more times in a single program file. (Error code: 4235)
- The CHKEND instruction is not executed following the execution of the CHKCIR instruction. (Error code: 4230)
- The CHKEND instruction is executed although no CHKCIR instruction has been executed. (Error code: 4230)
- The CHKST and CHK instruction are used in a low-speed type program. (Error code: 4235)
- There are 10 or more F☐ instances in a ladder pattern. (Error code: 4235)

- A ladder pattern has 257 or more steps. (Error code: 4235)
- A device has been encountered which cannot be used in a ladder pattern. (Error code: 4235)
- Index qualification has been conducted on a ladder pattern device. (Error code: 4235)

POINT

Generation of the following errors during program expansion at a peripheral device can make expansion impossible:

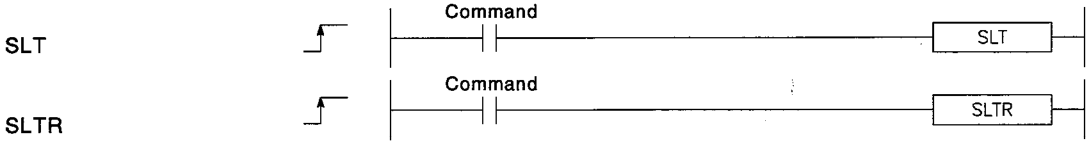
- A device has been encountered which cannot be used in a ladder pattern.
- Index qualification has been conducted on a ladder pattern device

Correct the ladder pattern if any of the above errors have occurred.

7.10.3 Setting and resetting status latch

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]



Functions

SLT

(1) When the SLT instruction is executed, the data set at set devices of the peripheral device at the time the SLT instruction is executed is stored in a file for status latch use in the IC memory card.

(2) The SLT instruction is valid only when SM806 is ON.
If SM806 has gone OFF, the SLT instruction can be executed, but will not be processed, and the status latch will not be executed.

(3) SM807 goes ON when the SLT instruction is executed, and when the status latch is completed, SM808 goes ON.
Additionally, the number of steps where the status latch was conducted is stored at SD807.

(4) The SLT instruction has been executed once, it will be ignored on the second and subsequent attempts at execution.
The SLT instruction can be re-enabled by the execution of the SLTR instruction.

(5) The results of the status latch operation can be monitored at the peripheral device.

SLTR

(1) The SLTR instruction is used to reset the SLT instruction.

(2) Execution of the SLTR instruction re-enables the SLT instruction.

REMARKS

(1) See the QnACPU User's Manual for more detailed information on the status latch.

(2) For information on status latch execution and monitoring results at peripheral devices, see the SW0IVD-GPPQ Operating Manual (Online).

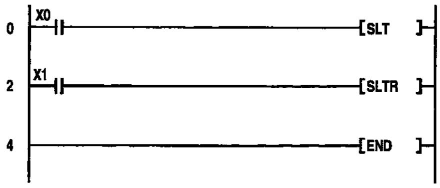
Operation Errors

(1) There are no operation errors associated with the SLT or SLTR instructions.

Program Example

(1) The following program executes the SLT instruction when X0 goes ON, and resets it with the SLTR instruction when X1 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	SLT	
2	LD	X1
3	SLTR	
4	END	

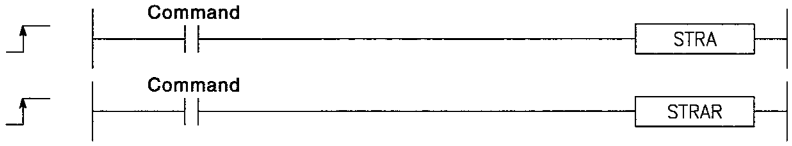
7.10.4 Setting and resetting sampling trace

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]

STRA

STRAR



Functions

STRA

(1) The sampling trace instruction is used to store sampling trace data set at a peripheral device for a designated number of cycles in a sampling trace file in an IC memory card when SM800, SM801, and SM802 are ON.
When the STRA instruction is executed, SM803 is turned ON, and after sampling has been conducted for the set number of sampling trace cycles after the execution of the STRA instruction, the data is checked, and the sampling trace operation is suspended.

(2) If SM801 is turned OFF during the execution of a sampling trace, the sampling is suspended.

(3) After the completion of the STRA instruction, sampling trace is finished and SM805 is turned ON.

(4) After STRA has been executed once, it is ignored the second and subsequent times it appears.
Run the STRAR instruction to make the STRA instruction valid once again.

STRAR

(1) The STRAR instruction is used to reset the STRA instruction.

(2) Running the STRAR instruction will turn SM801 through SM805 OFF.

(3) Run the STRAR instruction to make the STRA instruction valid once again.

REMARKS

(1) See the QnACPU User's Manual for more detailed information on the sampling trace operation.

(2) For information on the execution of sampling trace at peripheral devices, see the SW0IVD-GPPQ Operating Manual (Online).

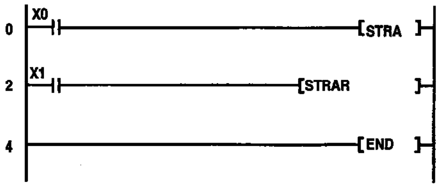
Operation Errors

(1) There are no operation errors associated with the STRA instruction.

Program Example

(1) The following program example runs the STRA instruction when X0 goes ON, and uses the SRTA instruction to reset the SRTA instruction when X1 goes ON.

[Ladder Mode]



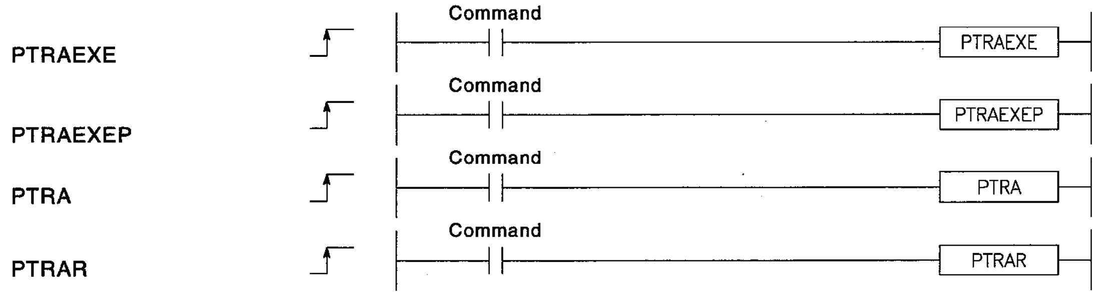
[List Mode]

Steps	Instruction	Device
0	LD	X0
1	STRA	
2	LD	X1
3	STRAR	
4	END	

7.10.5 Execution, setting, and resetting of program trace

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]



Functions

PTRA

(1) Program trace involves storing program trace data set at a peripheral device for the number of times set when SM810, SM811, and SM812 go ON in a program trace file at the IC memory card.

(2) When the PTRA instruction is executed, SM813 is turned ON, and after sampling is taken for the number of program traces set for the post-PTRA instruction execution operation, the data is latched and the program trace operation is suspended.

(3) If SM811 goes OFF during the execution of a program trace execution, the sampling is suspended.

(4) After the execution of the PTRA instruction, and at the completion of the program trace, SM815 will go ON.

(5) When the PTRA is executed once, it will be ignored from the second and subsequent times it is encountered.
The PTRA instruction can be re-enabled by the execution of the PTRAR instruction.

(6) The results of the program trace operation can be monitored at a peripheral device.

PTRAR

(1) The PTRAR instruction is used to reset the PTRA instruction.

(2) When PTRAR is executed, SM811 through SM815 go OFF.

(3) When the PTRAR instruction is executed, PTRA is enabled once again.

PTRAEXE

(1) When the program trace command goes ON, program trace is executed.

(2) If SM811 goes OFF during the execution of a program trace, sampling is suspended.

(3) If the program trace command is OFF, there will be no operation.

REMARKS

(1) See the QnACPU User's Manual for more detailed information on the program trace operation.

(2) For information on the execution of program trace at peripheral devices, see the SW0IVD-GPPQ Operating Manual (Online).

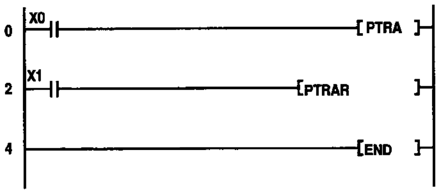
Operation Errors

(1) There are no operation errors associated with the PTRAEXE(P), PTRA, or PTRAR instructions.

Program Example

(1) The following program executes the PTRA instruction when X0 goes ON, and when X1 goes ON, resets the PTRA instruction by use of the PTRAR instruction.

[Ladder Mode]



[List Mode]

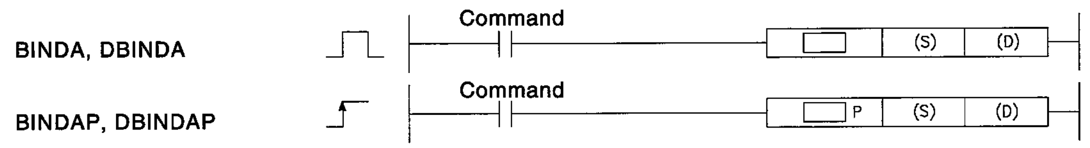
Steps	Instruction	Device
0	LD	X0
1	PTRA	
2	LD	X1
3	PTRAR	
4	END	

7.11 Character String Processing Instructions

7.11.1 Conversion from BIN 16-bit or 32-bit to decimal ASCII

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		o					—
(D)	—	o		—					—

[Instruction Symbol] [Execution Condition]



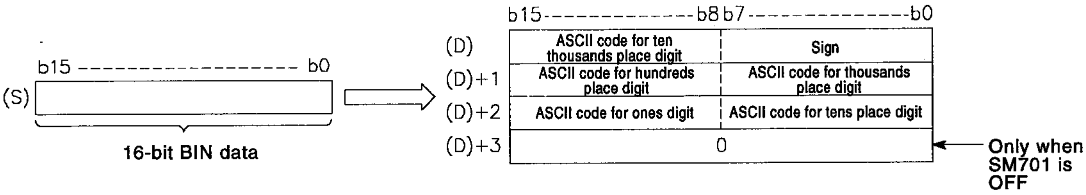
Set Data

Set Data	Meaning	Data Type
(S)	BIN data to be converted to ASCII	BIN 16- or 32-bit
(D)	First device number of devices where conversion results will be stored	Character string

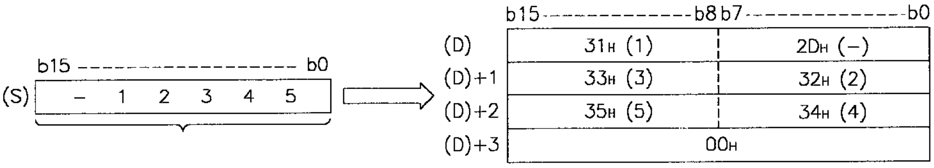
Functions

BINDA

(1) Converts each individual number of BIN 16-bit data designated by (S) that is being displayed in hexadecimal to its respective ASCII code, and stores starting from the device number designated by (D).



For example, if -12345 has been designated at (S), the following will be stored from (D) onward:



(2) The BIN data designated at (S) can be in the range from -32768 to 32767.

(3) The operation results stored at (D) are as follows:

(a) The sign "20$_H$" will be stored if the BIN data is positive, and the sign "2D$_H$" will be stored if it is negative.

(b) The sign "20$_H$" will be stored for the leading zeros of effective digits. (Zero suppression is conducted.)

$\underbrace{0\ 0}\underbrace{3\ 2\ 5}$

Number of significant digits

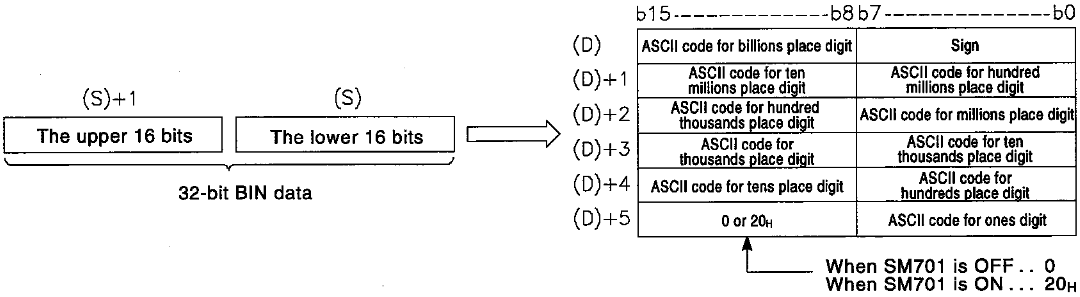
Becomes 20$_H$

(c) The storage of data at devices specified by (D)+3 differs depending on the ON/OFF status of SM701 (output number of characters conversion signal)

When SM701 is OFF ...... Stores "0"
When SM701 is ON .... Does not change

DBINDA

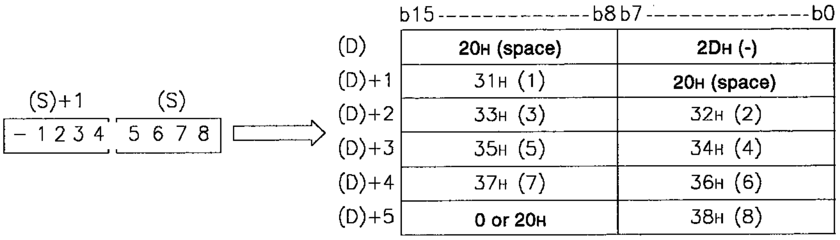
(1) Converts BIN 32-bit data designated by (S) to the ASCII code for each number at each position when displayed in decimal notation, and stores following the device designated by (D).



	b15 ------------- b8	b7 -------------- b0
(D)	ASCII code for billions place digit	Sign
(D)+1	ASCII code for ten millions place digit	ASCII code for hundred millions place digit
(D)+2	ASCII code for hundred thousands place digit	ASCII code for millions place digit
(D)+3	ASCII code for thousands place digit	ASCII code for ten thousands place digit
(D)+4	ASCII code for tens place digit	ASCII code for hundreds place digit
(D)+5	0 or 20$_H$	ASCII code for ones digit

When SM701 is OFF .. 0
When SM701 is ON ... 20$_H$

For example, if the value -12345678 has been designated by (S), the following would be stored following (D):



	b15 ------------- b8	b7 -------------- b0
(D)	20$_H$ (space)	2D$_H$ (-)
(D)+1	31$_H$ (1)	20$_H$ (space)
(D)+2	33$_H$ (3)	32$_H$ (2)
(D)+3	35$_H$ (5)	34$_H$ (4)
(D)+4	37$_H$ (7)	36$_H$ (6)
(D)+5	0 or 20$_H$	38$_H$ (8)

(2) BIN data designated by (S) can be between -2147483648 to 2147483647.

(3) The operations results stored at (D) will be stored in the following way:

(a) The sign "20$_H$" will be stored if the BIN data is positive, and the sign "2D$_H$" will be stored if it is negative.

(b) The sign "20$_H$" will be stored for the leading zeros of effective digits. (Zero suppression is conducted.)

$\underbrace{0\ 0}\underbrace{1\ 2\ 0\ 3\ 4\ 5\ 6\ 0}$

Becomes 20$_H$ Number of significant digits

(c) The data stored at the upper 8 bits of the device designated by (D)+5 will differ depending on the ON/OFF status of SM701 (number of characters to output select signal).

When SM701 is OFF ...... Stores "0"
When SM701 is ON .... Stores "20H"

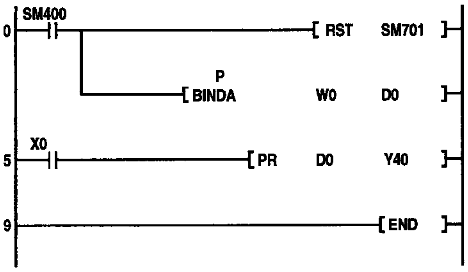
Operation Errors

There are no operation errors associated with the BINDA(P) or DBINDA(P) instructions.

Program Example

(1) The following example program uses the PR instruction to output the 16-bit BIN data W0 value by decimal to Y40 through Y48 as ASCII.

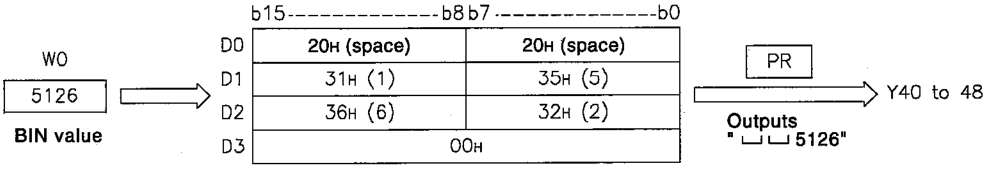
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	RST	SM701
2	BINDAP	W0 D0
5	LD	X0
6	PR	D0 Y40
9	END	

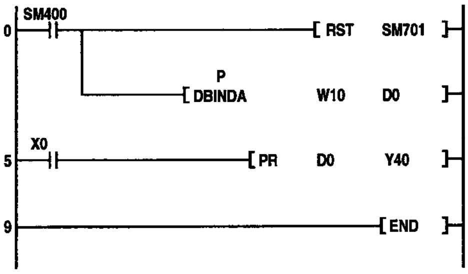
* Conducts ASCII output of Y40 through Y48 by using the PR instruction when X0 goes ON.
Because SM701 is OFF, The PR instruction will output ASCII code until 00H is encountered.



	b15-------------b8	b7--------------b0
D0	20H (space)	20H (space)
D1	31H (1)	35H (5)
D2	36H (6)	32H (2)
D3	00H	

(2) The following program uses the PR instruction to output the decimal value of the 32-bit BIN data at W10 through 11 in ASCII code to Y40 through 48.

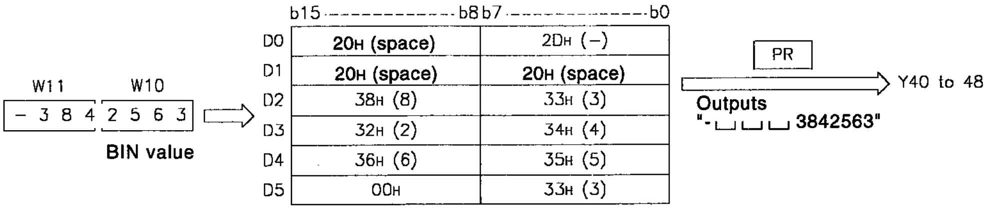
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	RST	SM701
2	DBINDAP	W10 D0
5	LD	X0
6	PR	D0 Y40
9	END	

* Conducts ASCII output of Y40 through Y48 by using the PR instruction when X0 goes ON.
Because SM701 is OFF, the PR instruction will output ASCII code until 00H is encountered.



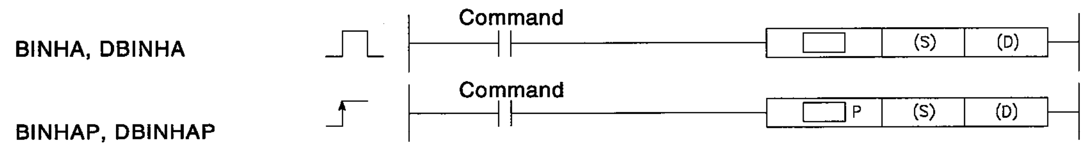
	b15------------b8	b7--------------b0
D0	20H (space)	2DH (–)
D1	20H (space)	20H (space)
D2	38H (8)	33H (3)
D3	32H (2)	34H (4)
D4	36H (6)	35H (5)
D5	00H	33H (3)

7.11.2 Conversion from BIN 16-bit or 32-bit data to hexadecimal ASCII

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		o					—
(D)	—	o		—					—

[Instruction Symbol] [Execution Condition]

□ indicates BINHA/DBINHA



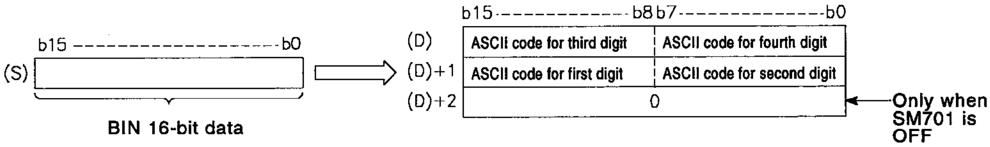
Set Data

Set Data	Meaning	Data Type
(S)	BIN data to be converted to ASCII	BIN 16- or 32-bit
(D)	First device number of devices where conversion results will be stored	Character string

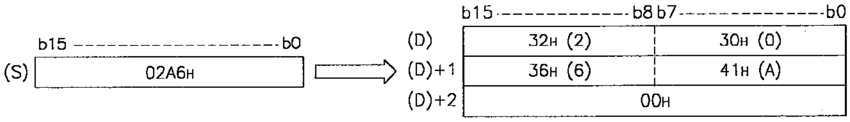
Functions

BINHA

(1) Converts each individual number of BIN 16-bit data designated by (S) that is being displayed in hexadecimal to its respective ASCII code, and stores starting from the device number designated by (D).



For example, if 02A6H has been designated by (S), it will be stored as follows:



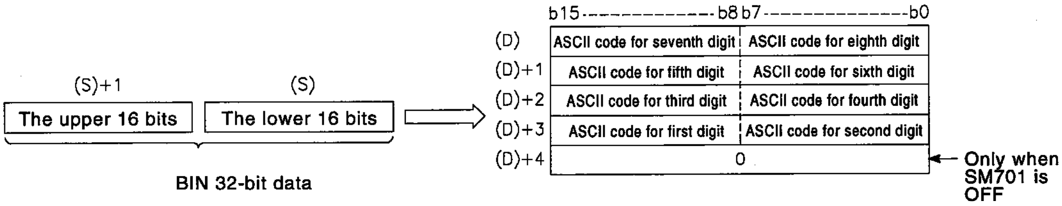
(2) The BIN data designated by (S) can be in the range from 0_H to $FFFF_H$.

(3) The operation results stored at (D) are processed as 4-digit hexadecimal values.
For this reason, zeros which are significant digits on the left side of the value are processed as "0". (No zero suppression is conducted.)

(4) The storage of the data in the device designated by (D)+2 will differ depending on the ON/OFF status of SM701 (number of characters to output select signal).
When SM701 is OFF.... Stores "0"
When SM701 is ON..... Does not change

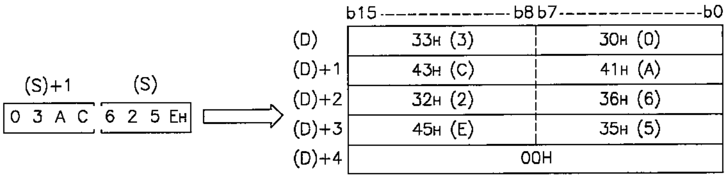
DBINHA

(1) Converts BIN 32-bit data designated by (S) to the ASCII code for each number at each position when displayed in hexadecimal notation, and stores following the device designated by (D).



	b15 ------ b8	b7 ------ b0
(D)	ASCII code for seventh digit	ASCII code for eighth digit
(D)+1	ASCII code for fifth digit	ASCII code for sixth digit
(D)+2	ASCII code for third digit	ASCII code for fourth digit
(D)+3	ASCII code for first digit	ASCII code for second digit
(D)+4	0	

For example, if the value 03AC625EH has been designated by (S), it would be stored following (D) in the following manner:



	b15 ------ b8	b7 ------ b0
(D)	33H (3)	30H (0)
(D)+1	43H (C)	41H (A)
(D)+2	32H (2)	36H (6)
(D)+3	45H (E)	35H (5)
(D)+4	00H	

(2) The BIN data designated by (S) can be in the range from 0H to FFFFFFFFH.

(3) The operation results stored at (D) are processed as 8-digit hexadecimal values.
For this reason, zeros which are significant digits on the left side of the value are processed as "0". (No zero suppression is conducted.)

(4) The storage of the data in the device designated by (D)+2 will differ depending on the ON/OFF status of SM701 (number of characters to output select signal).
When SM701 is OFF.... Stores "0"
When SM701 is ON..... Does not change

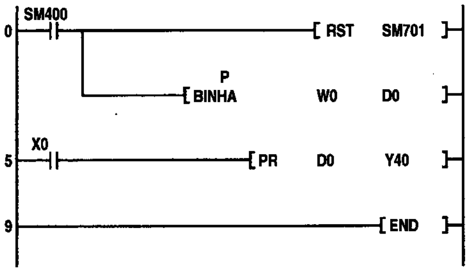
Operation Errors

There are no operation errors associated with the BINHA(P) or DBINHA(P) instructions.

Program Example

(1) The following program uses the PR instruction to output the hexadecimal value of the 16-bit BIN data at W0 in ASCII code to Y40 through 48.

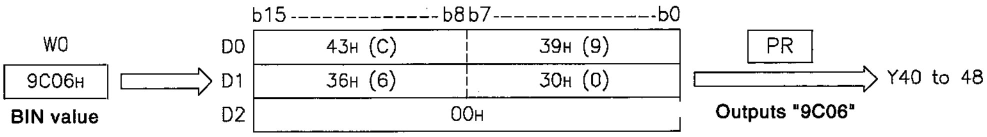
[Ladder Mode]



[List Mode]

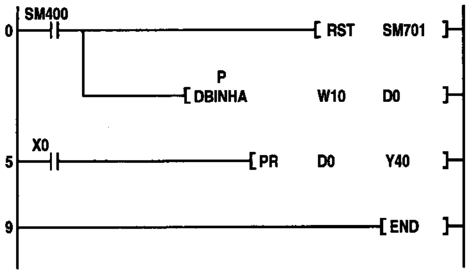
Steps	Instruction	Device
0	LD	SM400
1	RST	SM701
2	BINHAP	W0 D0
5	LD	X0
6	PR	D0 Y40
9	END	

* Outputs ASCII to Y40 through 48 by the PR instruction when X0 goes ON. Because SM701 is OFF, The PR instruction will output ASCII code until 00H is encountered.



(2) The following program uses the PR instruction to output the hexadecimal value of the 32-bit BIN data at W10 and 11 to Y40 through 48.

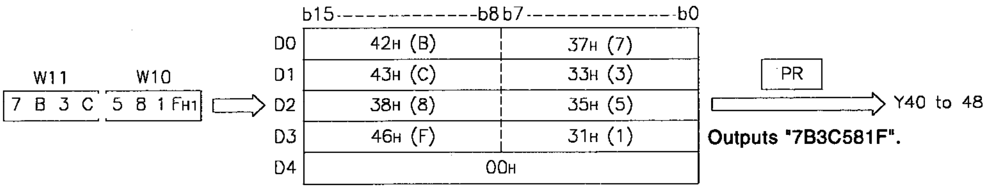
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	RST	SM701
2	DBINHAP	W10 D0
5	LD	X0
6	PR	D0 Y40
9	END	

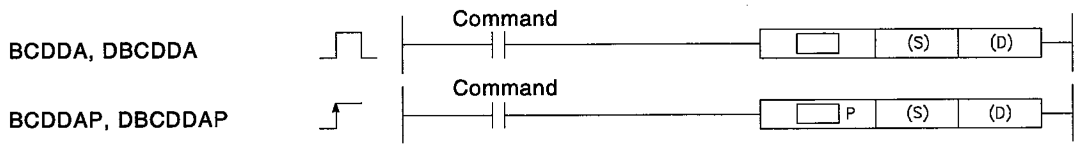
* Outputs ASCII to Y40 through 48 by the PR instruction when X0 goes ON. Because SM701 is OFF, The PR instruction will output ASCII code until 00H is encountered.



7.11.3 Conversion from BCD 4-digit and 8-digit to decimal ASCII data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		o					—
(D)	—	o		—					—

[Instruction Symbol] [Execution Condition] □ indicates BCDDA/DBCDDA



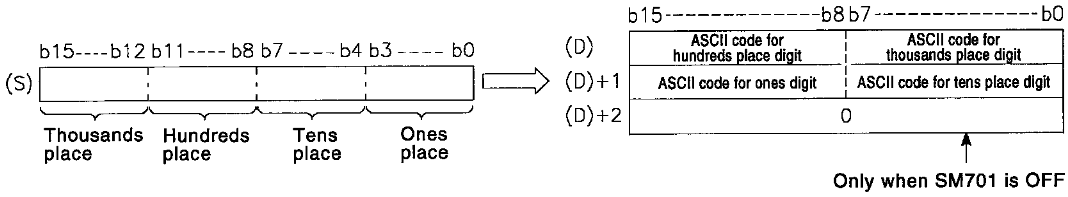
Set Data

Set Data	Meaning	Data Type
(S)	BCD data which will be converted to ASCII	Word
(D)	First device number of devices where conversion results will be stored	Character string

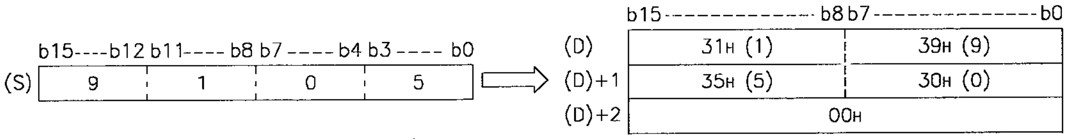
Functions

BCDDA

(1) Converts the individual numbers in the BCD 4-digit data designated by (S) to their respective ASCII codes, and stores starting from the device designated by (D).

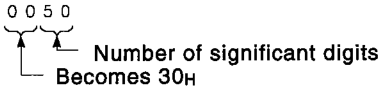


For example, if (S) were designating the value 9105, the results of the operation would be stored at (D) in the following manner:



(2) The BCD data designated by (S) can be in the range of from 0 to 9999.

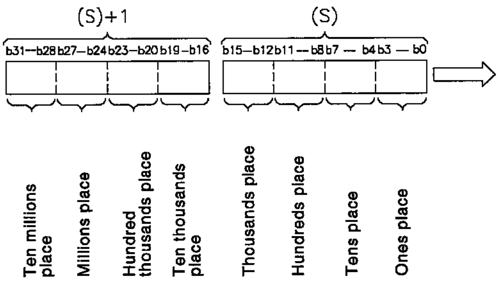
(3) Any significant digit zeros on the left side of the operation results as stored at (D) will be stored as "30H". (No zero suppression is conducted.)



(4) The data to be stored at the device designated by (D)+2 will differ depending on the ON/OFF status of SM701 (number of characters to output select signal).
When SM701 is OFF.... Stores "0"
When SM701 is ON..... Does not change

DBCDDA

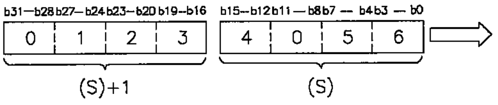
(1) Converts the individual numbers in the BCD 8-digit data designated by (S) to their respective ASCII codes, and stores following the device designated by (D).



	b15------------b8	b7------------b0
(D)	ASCII code for millions place digit	ASCII code for ten millions place digit
(D)+1	ASCII code for ten thousands place digit	ASCII code for hundred thousands place digit
(D)+2	ASCII code for hundreds place digit	ASCII code for thousands place digit
(D)+3	ASCII code for ones digit	ASCII code for tens place digit
	0	

←Only when SM701 is OFF

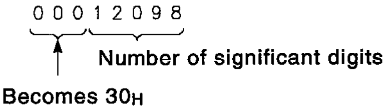
For example, if the value 01234056 were designated by (S), the operation result would be stored following (D) in the following manner:



	b15------------b8	b7------------b0
(D)	31H (1)	30H (0)
(D)+1	33H (3)	32H (2)
(D)+2	30H (0)	34H (4)
(D)+3	36H (6)	35H (5)
(D)+4	00H	

(2) The BCD data designated by (S) can be in the range of 0 to 99999999

(3) Any significant digit zeros on the left side of the operation results as stored at (D) will be stored as "30H". (No zero suppression is conducted.)



(4) The data to be stored at the device designated by (D)+4 will differ depending on the ON/OFF status of SM701 (number of characters to output select signal).
When SM701 is OFF.... Stores "0"
When SM701 is ON..... Does not change

Operation Errors

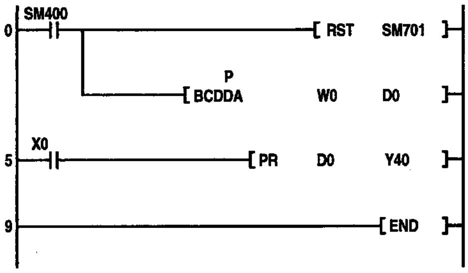
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data at (S) during the operation of the BCDDA instruction is outside the range of from 0 to 9999. (Error code: 4100)
- The data at (S) during the operation of the DBCDDA instruction is outside the range of from 0 to 99999999. (Error code: 4100)

Program Example

(1) The following program uses the PR instruction to convert BCD 4-digit data (the value at W0) to decimal, and outputs it in ASCII format to Y40 through 48.

[Ladder Mode]

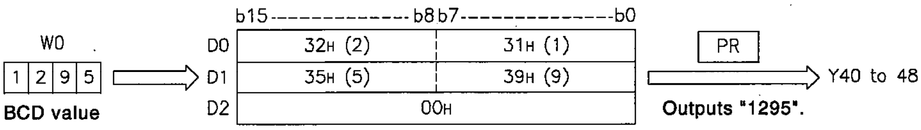


[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	RST	SM701
2	BCDDAP	W0 D0
5	LD	X0
6	PR	D0 Y40
9	END	

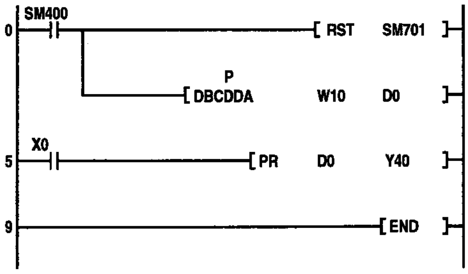
[Operation]

Conducts ASCII output of Y40 through Y48 by using the PR instruction when X0 goes ON.
Because SM701 is OFF, The PR instruction will output ASCII code until 00H is encountered.



(2) The following program uses the PR instruction to convert BCD 8-digit data (the values at W10 and 11) to decimal, and outputs it in ASCII format to Y40 through 48.

[Ladder Mode]

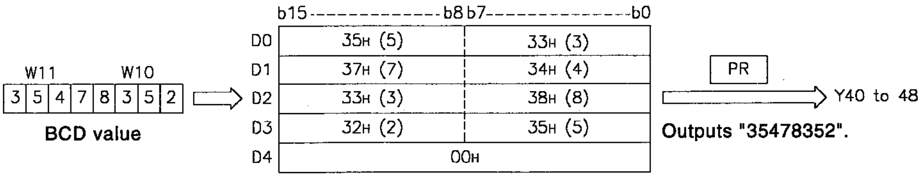


[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	RST	SM701
2	DBCDDAP	W10 D0
5	LD	X0
6	PR	D0 Y40
9	END	

[Operation]

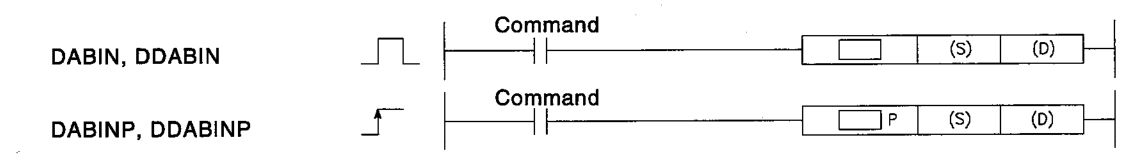
Conducts ASCII output of Y40 through Y48 by using the PR instruction when X0 goes ON.
Because SM701 is OFF, The PR instruction will output ASCII code until 00H is encountered.



7.11.4 Conversion from decimal ASCII to BIN 16-bit and 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—
(D)	o	o		o				—	—

[Instruction Symbol] [Execution Condition] □ indicates DABIN/DDABIN



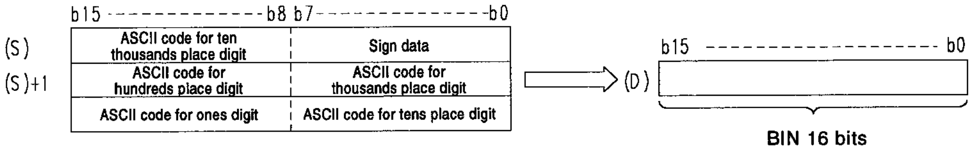
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where ASCII data that will be converted to BIN values is being stored	Character string
(D)	First device number of devices where conversion results will be stored	BIN 16- or 32-bit

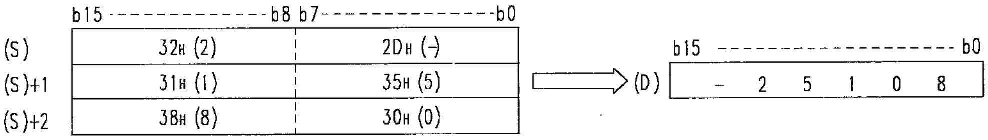
Functions

DABIN

(1) Converts decimal ASCII data stored following the device number designated by (S) into BIN 16-bit data, and stores it in the device number designated by (D).



For example, if the ASCII code following (S) were for the value -25018H, the operation result would be stored at (D) in the following manner:



(2) The ASCII data designated by from (S) to (S)+2 can be in the range of from -32768 to 32767

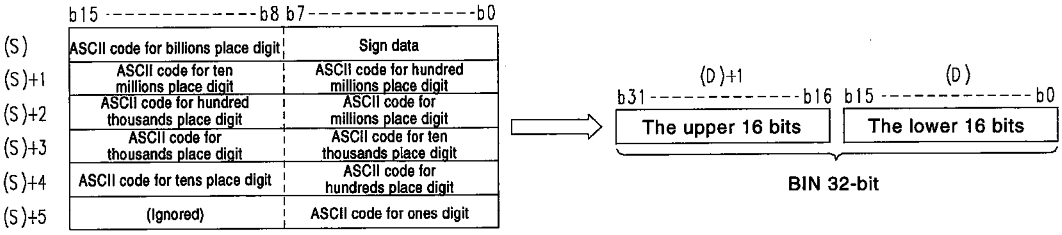
(3) The sign "20H" will be stored if the BIN data is positive, and the sign "2DH" will be stored if it is negative.

(4) ASCII code can be set for each position within the range from "30H" to "39H".

(5) If the ASCII code set for individual positions is "20H" or "00H," it will be processed as "30H".

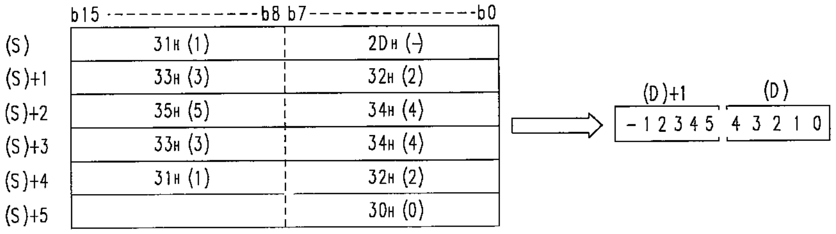
DDABIN

(1) Converts decimal ASCII data stored following the device number designated by (S) into BIN 32-bit data, and stores it in the device number designated by (D).



	b15 ------------ b8	b7 ------------ b0
(S)	ASCII code for billions place digit	Sign data
(S)+1	ASCII code for ten millions place digit	ASCII code for hundred millions place digit
(S)+2	ASCII code for hundred thousands place digit	ASCII code for millions place digit
(S)+3	ASCII code for thousands place digit	ASCII code for ten thousands place digit
(S)+4	ASCII code for tens place digit	ASCII code for hundreds place digit
(S)+5	(Ignored)	ASCII code for ones digit

For example, if the ASCII code following (S) were for the value -1234543210H, the operation result would be stored at (D)+1 and (D) in the following manner:



	b15 ------------ b8	b7 ------------ b0
(S)	31H (1)	2DH (-)
(S)+1	33H (3)	32H (2)
(S)+2	35H (5)	34H (4)
(S)+3	33H (3)	34H (4)
(S)+4	31H (1)	32H (2)
(S)+5		30H (0)

(2) The ASCII data designated by (S) to (S)+5 can be in the range of from -2147483648 to 2147483647.
Further, data stored at the upper bytes of (S)+5 will be ignored.

(3) If the value of the conversion data is positive, the "sign" data will be "20H"; if it is negative, this will be "2DH".

(4) ASCII code can be set for each position within the range from "30H" to "39H".

(5) If the ASCII code set for individual positions is "20H" or "00H", it will be processed as "30H".

Operation Errors

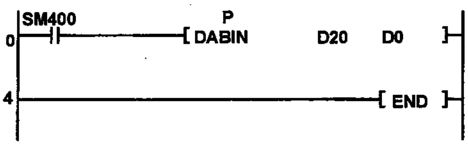
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The ASCII code designated by (S) through (S)+5 for the individual numbers is something other than "30_H" through "39_H", "20_H", or "00_H". (Error code: 4100)
- The ASCII data designated by (S) through (S)+5 is outside the ranges shown below: (Error code: 4100)
  When DABIN instruction is used..... -32768 to 32767
  When DDABIN instruction is used ... -2147483648 to 2147483647

Program Example

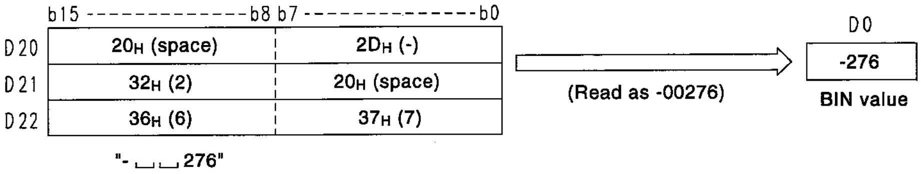
(1) The following program converts the decimal, 5-digit ASCII data and sign set at D20 through D22 to BIN values, and stores the result at D0.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DABINP	D20 D0
4	END	

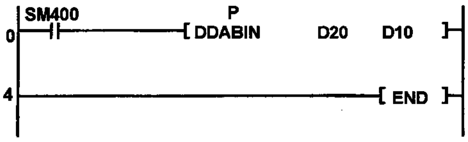


	b15 ------------- b8	b7 ------------- b0
D20	20_H (space)	$2D_H$ (-)
D21	32_H (2)	20_H (space)
D22	36_H (6)	37_H (7)

"- ␣␣276"

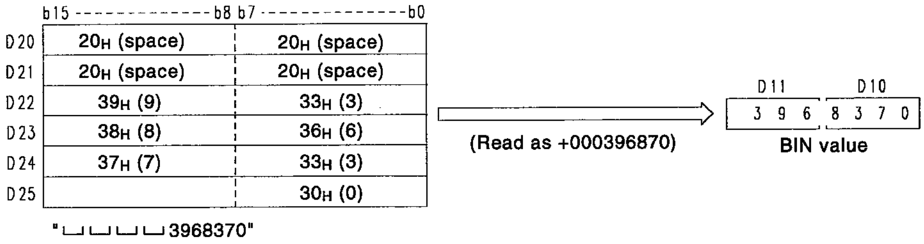
(2) The following program converts the decimal, 10-digit ASCII data and sign set at D20 through D25 to BIN values and stores the result at D10 and D11.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DDABINP	D20 D0
4	END	



	b15 ------------- b8	b7 ------------- b0
D20	20_H (space)	20_H (space)
D21	20_H (space)	20_H (space)
D22	39_H (9)	33_H (3)
D23	38_H (8)	36_H (6)
D24	37_H (7)	33_H (3)
D25		30_H (0)

"␣␣␣␣3968370"

7.11.5 Conversion from hexadecimal ASCII to BIN 16-bit and 32-bit data

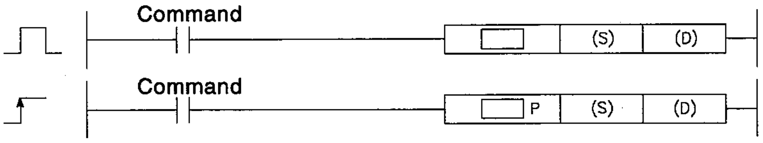
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—
(D)	o	o		o				—	—

[Instruction Symbol] [Execution Condition]

□ indicates HABIN/DHABIN

HABIN, DHABIN

HABINP, DHABINP



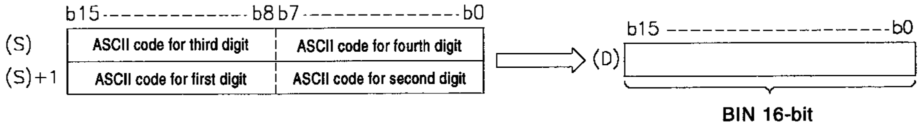
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where ASCII data that will be converted to BIN values is being stored	Character string
(D)	First device number of devices where conversion results will be stored	BIN 16- or 32-bit

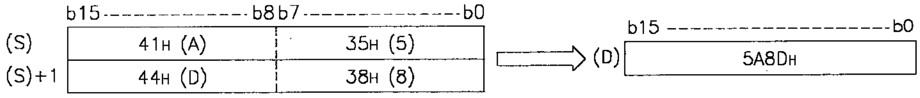
Functions

HABIN

(1) Converts hexadecimal ASCII data stored following the device number designated by (S) into BIN 16-bit data, and stores it in the device number designated by (D).



For example, if the ASCII code following (S) were for the value 5A8DH, the operation result would be stored at (D) in the following manner:

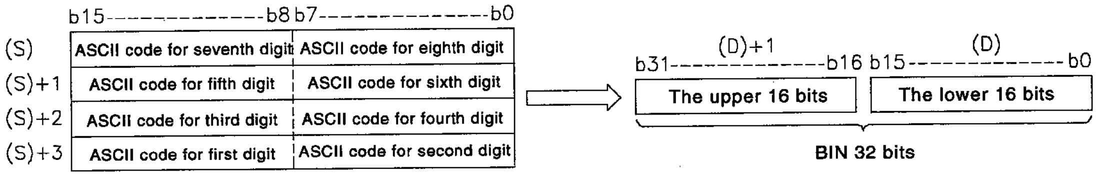


(2) The ASCII data designated by (S) to (S)+1 can be in the range of from 0000H to FFFFH.

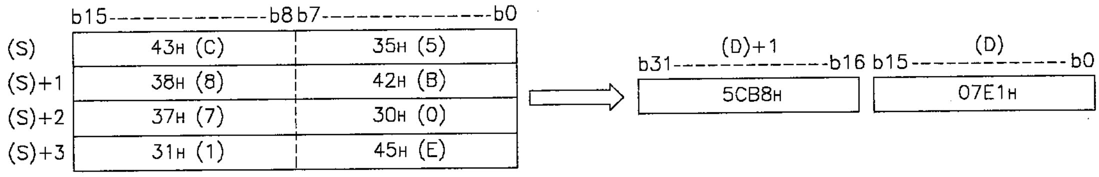
(3) The ASCII codes can be in the range of "30H" to "39H" and from "41H" to "46H".

DHABIN

(1) Converts hexadecimal ASCII data stored following the device number designated by (S) into BIN 32-bit data, and stores it in the device number designated by (D).



For example, if the ASCII code following (S) were for the value 5CB807E1H, the operation result would be stored at (D)+1 and (D) in the following manner:



(2) The ASCII data designated by (S) to (S)+3 can be in the range of from 00000000H to FFFFFFFFH.

(3) The ASCII codes can be in the range of from "30H" to "39H" and from "41H" to "46H".

Operation Errors

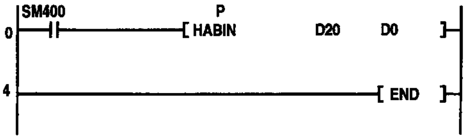
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The ASCII codes for the individual numbers designated by (S) through (S)+3 are outside the range of from "30H" to "39H" and from "41H" to "46H". (Error code: 4100)

Program Example

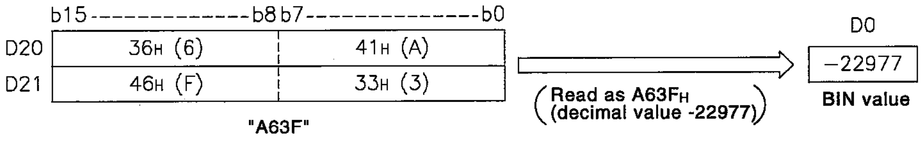
(1) The following program converts the hexadecimal, 4-digit ASCII data set at D20 and D21 to BIN data, and stores the result at D0.

[Ladder Mode]



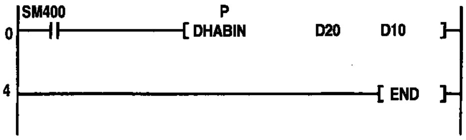
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	HABINP	D20
		D0
4	END	



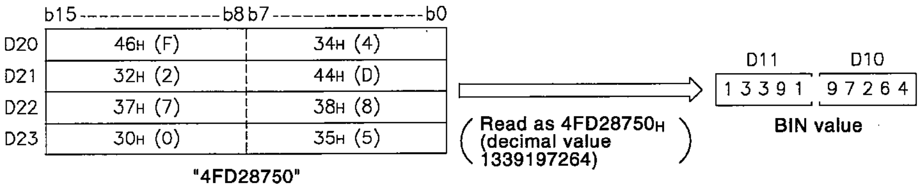
(2) The following program converts the hexadecimal, 8-digit ASCII data set at D20 through D23 to BIN values, and stores the result at D10 and D11.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DHABINP	D20
		D0
4	END	

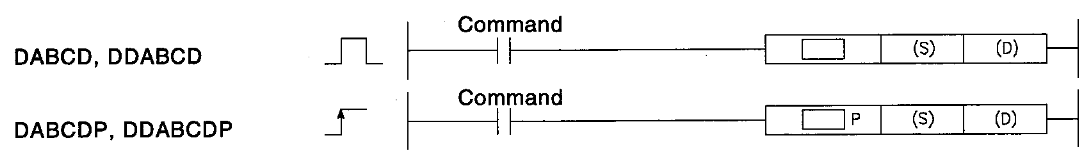


7.11.6 Conversion from decimal ASCII to BCD 4-digit or 8-digit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J◻\◻		Special Function Module U◻\G◻	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—
(D)	o	o		o				—	—

[Instruction Symbol] [Execution Condition]

◻ indicates DABCD/DDABCD



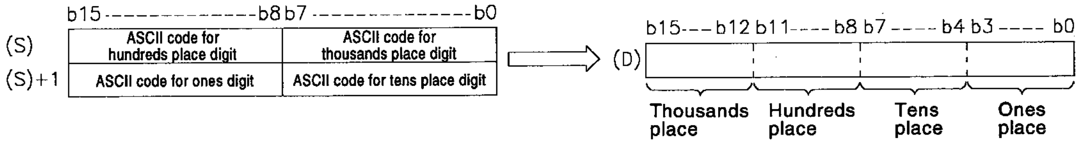
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where ASCII data to be converted to BCD values is being stored	Character string
(D)	First device number of devices where conversion results will be stored	BCD 4-digit/ 8-digit

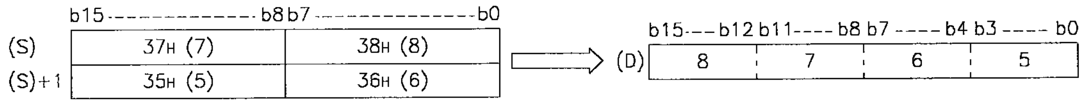
Functions

DABCD

(1) Converts hexadecimal ASCII data being stored following device number designated by (S) to 4-digit BCD data, and stores at device number designated by (D).



For example, if the ASCII code following (S) were designated as the value 8765H, the operation results would be stored at (D) in the following manner:



(2) The ASCII data designated by (S) through (S)+1 can be in the range of from 0 to 9999.

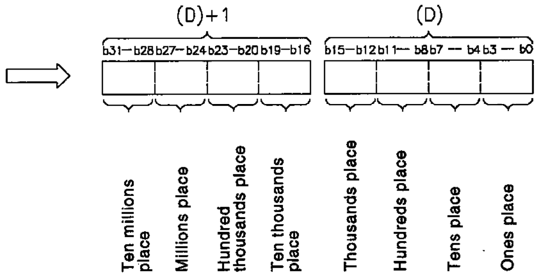
(3) The ASCII code set at each digit can be in the range of from "30H" to "39H".

(4) If ASCII code for individual digits is "20H" or "00H", it is processed as "30H".

DDABCD

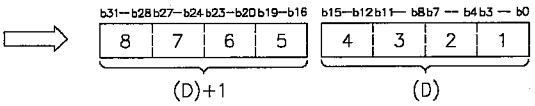
(1) Converts hexadecimal ASCII data being stored following device number designated by (S) to 8-digit BCD data, and stores at device number designated by (D).

	b15 ---------- b8	b7 ---------- b0
(S)	ASCII code for millions place digit	ASCII code for ten millions place digit
(S)+1	ASCII code for ten thousands place digit	ASCII code for hundred thousands place digit
(S)+2	ASCII code for hundreds place digit	ASCII code for thousands place digit
(S)+3	ASCII code for ones digit	ASCII code for tens place digit



For example, if the ASCII code following (S) were designated as the value 87654321H, the operation results would be stored at (D)+1 and (D) in the following manner:

	b15 ---------- b8	b7 ---------- b0
(S)	37H (7)	38H (8)
(S)+1	35H (5)	36H (6)
(S)+2	33H (3)	34H (4)
(S)+3	31H (1)	32H (2)



(2) The ASCII data designated at (S) through (S)+3 can be in the range of from 0 to 99999999.

(3) The ASCII code set at each digit can be in the range of from "30H" to "39H".

(4) If ASCII code for individual digits is from "20H" to "00H", it is processed as "30H".

Operation Errors

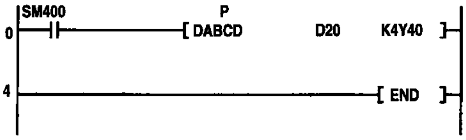
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- There are characters within the data at (S) that are outside the 0 to 9 range. (Error code: 4100)

Program Example

(1) The following program converts the hexadecimal ASCII data set from D20 to D22 to BCD 4-digit data, and outputs the results to Y40 through Y4F.

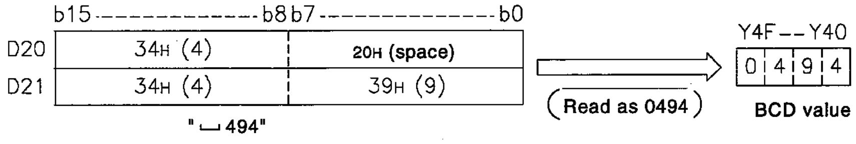
[Ladder Mode]



Outputs converted BCD value to display device

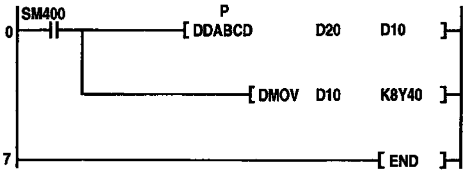
[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DABCDP	D20 K4Y40
4	END	



(2) The following program converts the hexadecimal ASCII data set at D20 through D23 into 8-digit BCD data, stores the result at D10 and D11, and also outputs it to from Y40 through Y5F.

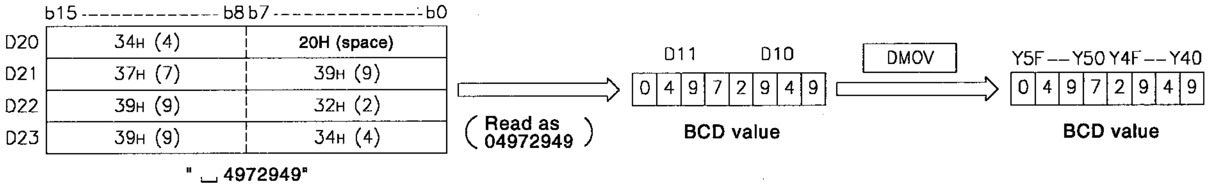
[Ladder Mode]



Outputs converted BCD value to display device

[List Mode]

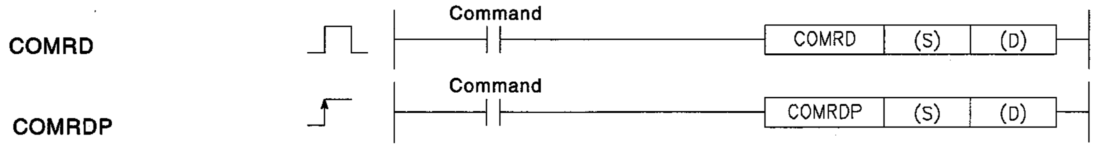
Steps	Instruction	Device
0	LD	SM400
1	DDABCDP	D20 D10
4	DMOV	D10 K8Y40
7	END	



7.11.7 Reading device comment data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				BL\S, BL\TR, BL, P, I, J, U
(S)	o	o		o			—		o
(D)	—	o		—			—		—

[Instruction Symbol] [Execution Condition]

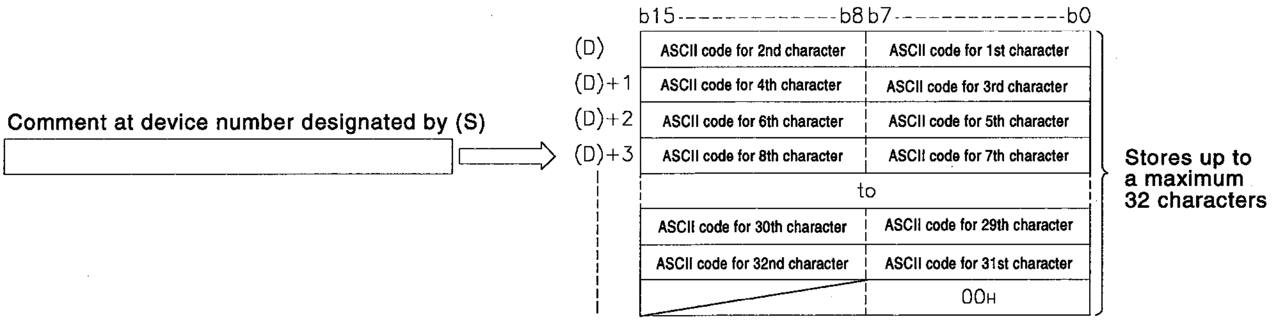


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where comment to be read is registered	Device name
(D)	First device number of devices to store read comment data	Character string

Functions

(1) Reads the comment at the device number designated by (S), and stores it as ASCII code following the device number designated by (D).



For example, if the comment for the device designated by (S) were "No. 1␣LINE␣START," the operation results would be stored following (D) as follows:

Comment at (S)

NO.1␣LINE␣START ⇨

	b15------b8	b7------b0
(D)	FH (O)	4EH (N)
(D)+1	31H (1)	2EH (.)
(D)+2	4CH (L)	**20H (space)**
(D)+3	4EH (N)	49H (I)
(D)+4	**20H (space)**	45H (E)
(D)+5	54H (T)	53H (S)
(D)+6	52H (R)	41H (A)
(D)+7	00H	54H (T)

(2) The only device numbers that can be designated by (S) are those that are set within the comment range.

(3) If no comment has been registered for the device designated by (S), all of the characters for the comment would be processed as "20H" (space).

(4) The device number plus 1 where the final (D) character is being stored will differ depending on the ON/OFF status of SM701 (number of characters to output select signal).

When SM701 is OFF : Stores "0"

When SM701 is ON : Does not change

Operation Errors

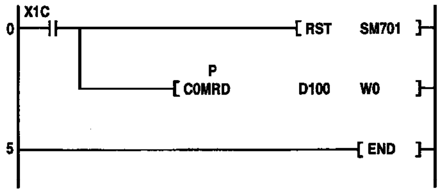
(1) In the following cases an error is returned, and the error flag goes ON.

- The device number designated by (S) is not within the comment range setting. (Error code: 4100)

Program Example

(1) The following program stores the comments set at D100 in W0 as ASCII when X1C goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	RST	SM701
2	COMRDP	D100 W0
5	END	

Comment at D100.

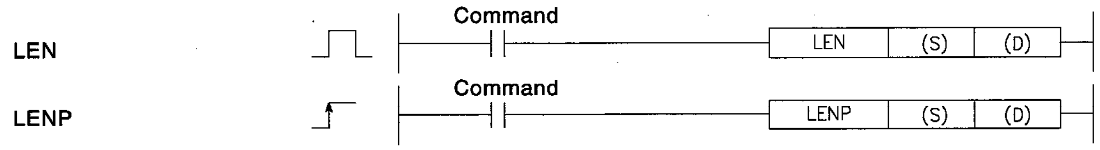
ALINE␣TARGET␣ ⇨

	b15------b8	b7------b0
W0	4CH (L)	41H (A)
W1	4EH (N)	49H (I)
W2	**20H (space)**	45H (E)
W3	41H (A)	54H (T)
W4	57H (G)	52H (R)
W5	54H (T)	55H (E)
W6	**20H (space)**	**20H (space)**
W7	00H	

7.11.8 Character string length detection

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				U
(S)	—	o		—				o	—
(D)	o	o		o				—	—

[Instruction Symbol] [Execution Condition]

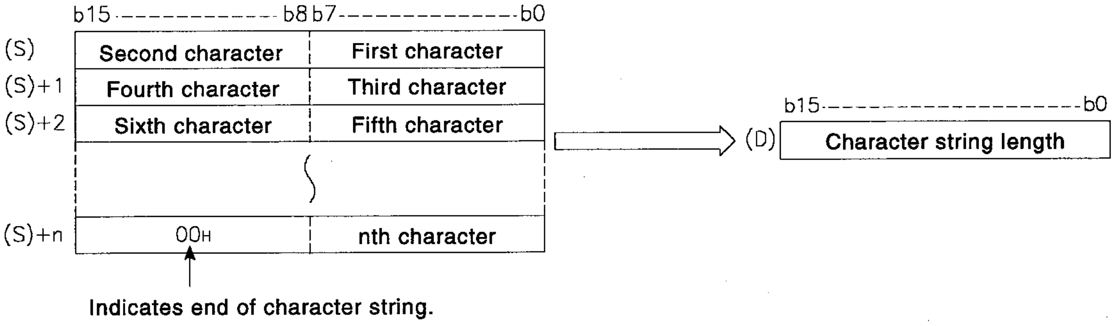


Set Data

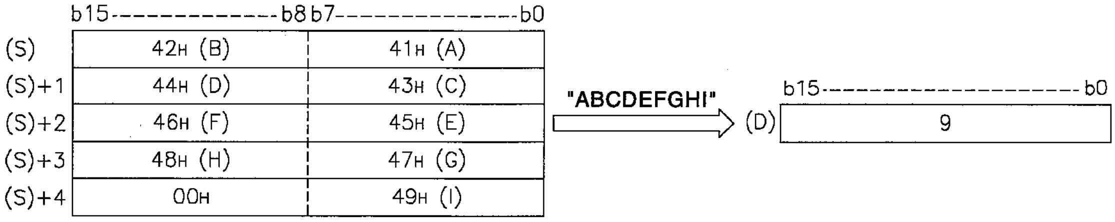
Set Data	Meaning	Data Type
(S)	First device number of devices storing character string	Character string
(D)	Number of device to store detected character string length	BIN 16 bits

Functions

(1) Detects length of character string designated by (S) and stores following device number designated by (D).
Processes the data from the device number designated by (S) to the device number storing "00H" as a character string.



For example, if the value "ABCDEFGHI" were being stored from (S) onwards, the value 9 would be stored at (D).



	b15 ------------ b8	b7 ------------ b0
(S)	42H (B)	41H (A)
(S)+1	44H (D)	43H (C)
(S)+2	46H (F)	45H (E)
(S)+3	48H (H)	47H (G)
(S)+4	00H	49H (I)

Operation Errors

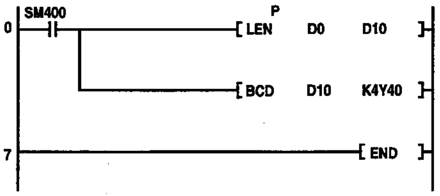
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- There is no "00H" set within the relevant device range following the device number designated by (S). (Error code: 4100)

Program Example

(1) The following program outputs the length of the character string from D0 to Y40 through Y4F as BCD 4-digit values.

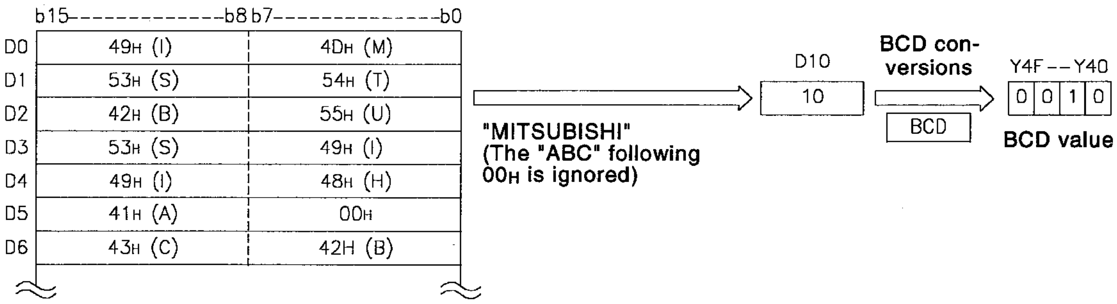
[Ladder Mode]



Outputs the length of the character string to the display device

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	LENP	D0 D10
4	BCD	D10 K4Y40
7	END	



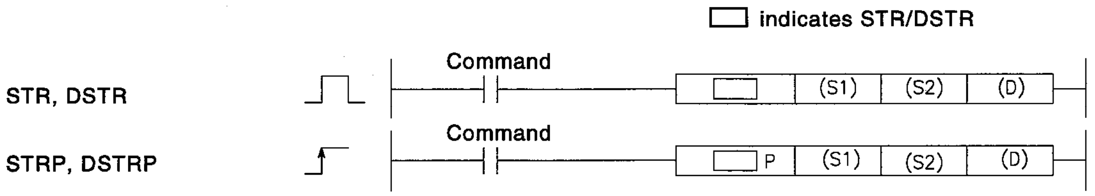
	b15 ------------ b8	b7 ------------ b0
D0	49H (I)	4DH (M)
D1	53H (S)	54H (T)
D2	42H (B)	55H (U)
D3	53H (S)	49H (I)
D4	49H (I)	48H (H)
D5	41H (A)	00H
D6	43H (C)	42H (B)

"MITSUBISHI"
(The "ABC" following 00H is ignored)

7.11.9 Conversion from BIN 16-bit or 32-bit to character string

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o	o		o				—	—
(S2)	o	o		o				o	—
(D)	—	o		—				—	—

[Instruction Symbol] [Execution Condition]



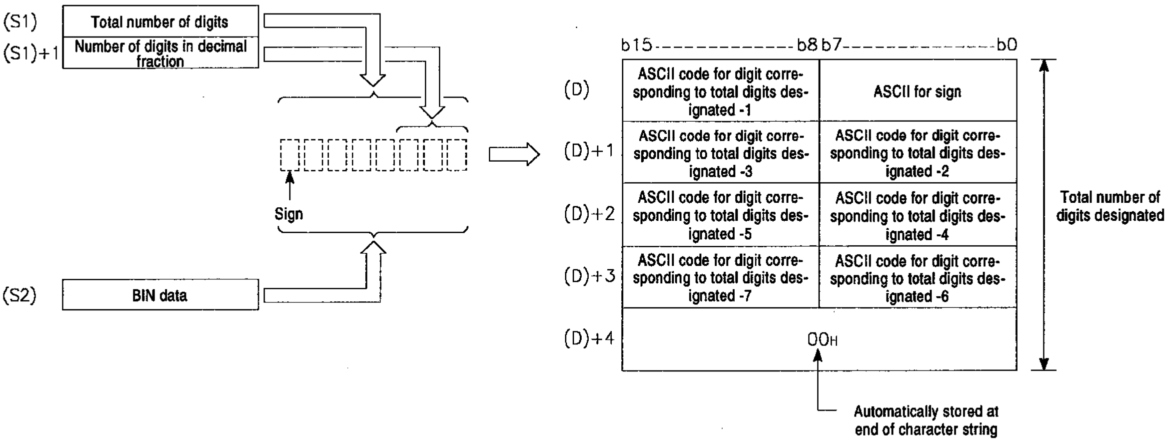
Set Data

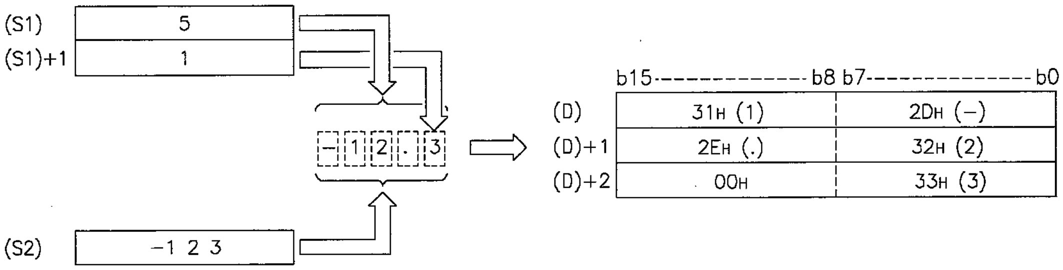
Set Data	Meaning	Data Type
(S1)	First device number of devices storing the number of digits of the numerical value to convert.	BIN 16 bits
(S2)	BIN data to be converted	BIN 16/32 bits
(D)	First number of devices storing converted character strings	Character string

Functions

STR

(1) Adds a decimal point to the BIN 16-bit data designated by (S2) at the location designated by (S1), converts the data to character string data, and stores it starting from the device number designated by (D).





(2) The total number of digits that can be designated by (S1) is from 2 to 8.

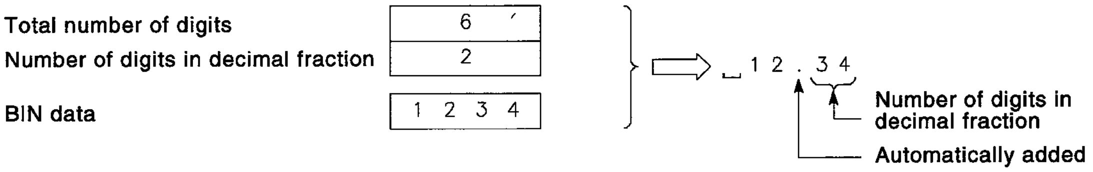
(3) The number of digits that can be designated by (S1)+1 as a part of the decimal fraction is from 0 to 5.
However, the number of digits following the decimal point must be smaller than or equal to the total number of digits minus 3.

(4) BIN data in the range between -32768 and 32767 can be designated at (S2).

(5) After conversion, character string data is stored at the device number following (D) as indicated below:

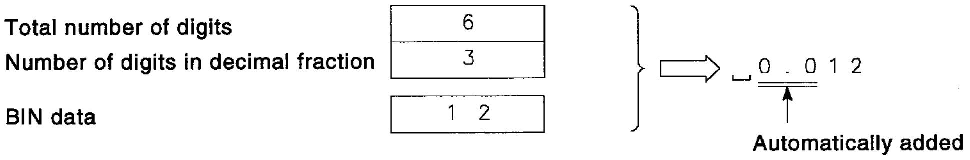
(a) The sign "20H" (space) will be stored if the BIN data is positive, and the sign "2DH" (minus sign -) will be stored if it is negative.

(b) If the setting for the number of digits after the decimal fraction is anything other than "0", 2EH (.) will automatically be stored at the position before the first of the specified number of digits.

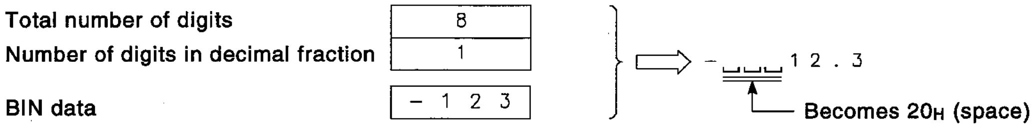


If the number of digits in the decimal fraction is 0, "2EH" (.) is not stored with the value.

(c) If the total number of digits following the decimal fraction is greater than the number of BIN data digits, a zero will be added automatically and the number converted by shifting to the right, so that it would become "0. □□□□□".



(d) If the total number of digits excluding the sign and the decimal point is greater than the number of BIN data digits, "20H" (space) will be stored between the sign and the numeric value.



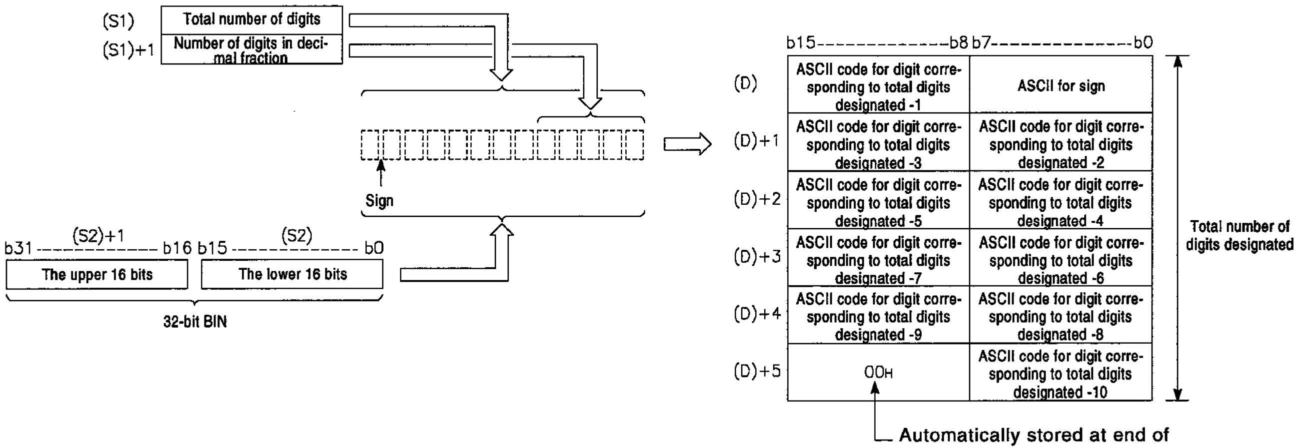
If the number of BIN digits is greater, an error will be returned.

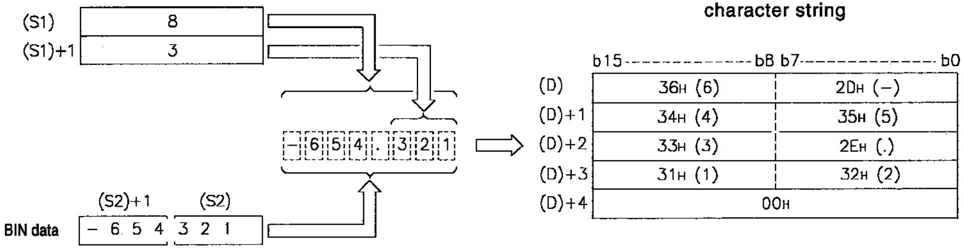
(e) The value "00H" is automatically stored at the end of the converted character string.

Functions

DSTR

(1) Adds a decimal point to the BIN 32-bit data designated by (S2) at the location designated by (S1), converts the data to character string data, and stores it following the device number designated by (D).





(2) The total number of digits that can be designated by (S1) is from 2 to 13.

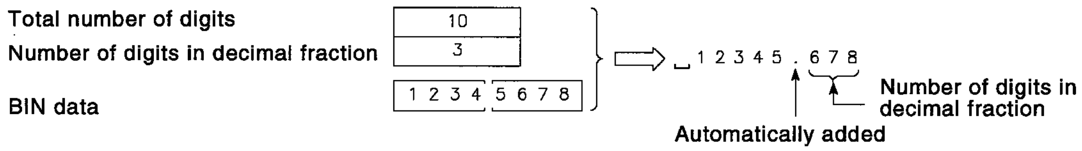
(3) The number of digits that can be designated by (S1)+1 as a part of the decimal fraction is from 0 to 10.
However, the number of digits following the decimal point must be smaller than or equal to the total number of digits minus 3.

(4) The BIN data that can be designated by (S1) and (S2)+1 is within the range of from -2147483648 to 2147483647.

(5) Character string data following conversion is stored in a device number as indicated below:

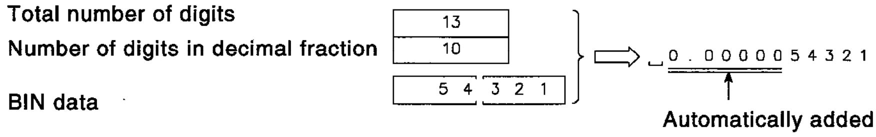
(a) The sign "20_H" (space) will be stored if the BIN data is positive, and the sign "$2D_H$" (minus sign -) will be stored if it is negative.

(b) If the setting for the number of digits after the decimal point is anything other than "0", $2E_H$ (.) will automatically be stored at the position before the first of the specified number of digits.

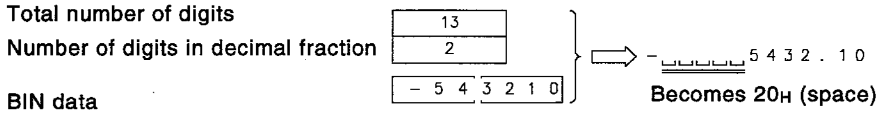


If the number of digits in the decimal fraction is 0, "$2E_H$" (.) is not stored with the value.

(c) If there are more digits in the decimal fraction than in the number of digits of BIN data, a 0 will be added automatically, the digits will be shifted to the right, and the conversion value will be "0. □□□□".



(d) If the total number of digits excluding the sign and the decimal point is greater than the number of BIN data digits, "20_H" (space) will be stored between the sign and the numeric value.



If the number of BIN digits is greater, an error will be returned.

(e) The value "00_H" is automatically stored at the end of the converted character string.

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The total number of digits designated by (S1) is outside the ranges shown below. (Error code: 4100)

  When the STR instruction is in use .... 2 to 8

  When the DSTR instruction is in use ...... 2 to 13

- The number of digits designated as a part of the decimal fraction by (S1)+1 is outside the range shown below. (Error code: 4100)

  When the STR instruction is in use .... 0 to 5

  When the DSTR instruction is in use ...... 0 to 10

- The relationship between the total number of digits designated by (S1) and the number of digits in the decimal fraction designated by (S1)+1 is not as shown below: (Error code: 4100)

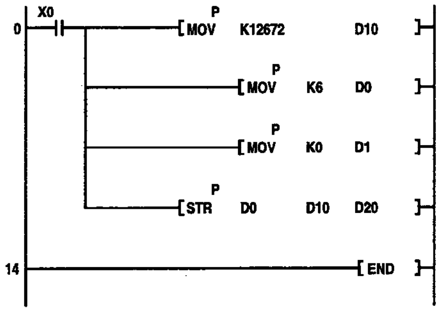
  Total number of digits minus 3 is equal to or larger than the number of digits in the decimal fraction.

- The number of digits designated by (S1) plus the sign and the number of digits in the decimal fraction is smaller than the number of digits in the BIN data designated by (S2). (Error code: 4100)
- The device range where the character string designated by (D) will be stored exceeds the relevant device range. (Error code: 4100)

Program Example

(1) The following program converts the BIN 16-bit data stored at D10 when X0 goes ON in accordance with the digit designation of D0 and D1, and stores the result from D20 to D23.

[Ladder Mode]



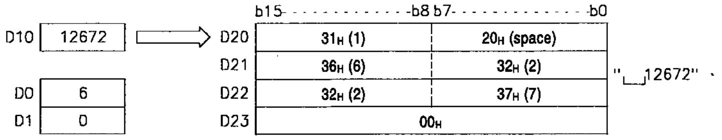
Data setting

Setting total number of digits

Setting number of digits following decimal point

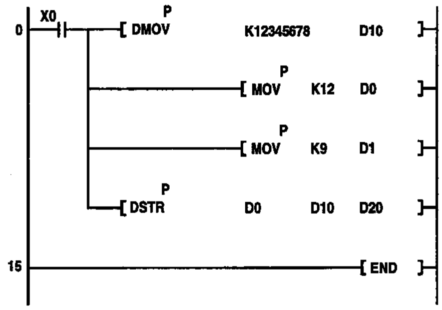
[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MOVP	K12672 D10
4	MOVP	K6 D0
7	MOVP	K0 D1
10	STEP	D0 D10
14	END	D20



(2) The following program converts the BIN 32-bit data stored at D10 and D11 when X0 goes ON in accordance with the digit designation of D0 and D1, and stores the result at from D20 to D26.

[Ladder Mode]



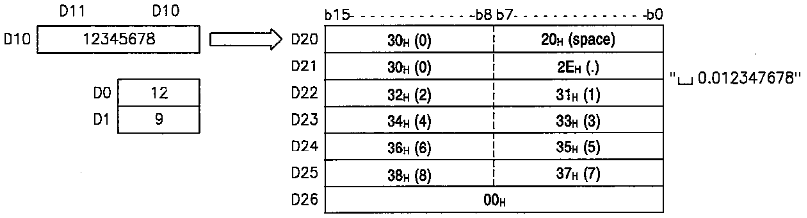
Data setting

Setting total number of digits

Setting number of digits following decimal point

[List Mode]

Steps	Instruction	Device
0	LD	X0
1	DMOVP	K12345678 D10
5	MOVP	K12 D0
8	MOVP	K9 D1
11	DSTRP	D0 D10
15	END	D20



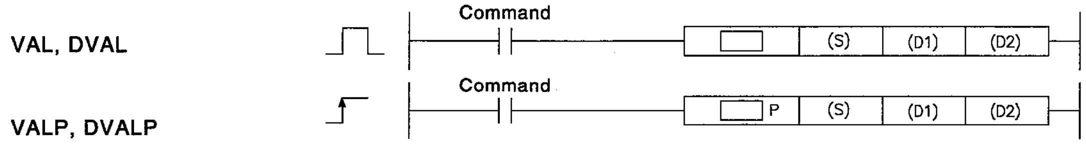
	b15 - b8	b7 - b0
D20	30H (0)	20H (space)
D21	30H (0)	2EH (.)
D22	32H (2)	31H (1)
D23	34H (4)	33H (3)
D24	36H (6)	35H (5)
D25	38H (8)	37H (7)
D26	00H	

7.11.10 Conversion from character string to BIN 16-bit or 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—
(D1)	o	o		—				—	—
(D2)	o	o		o				—	—

[Instruction Symbol] [Execution Condition]

□ indicates VAL/DVAL



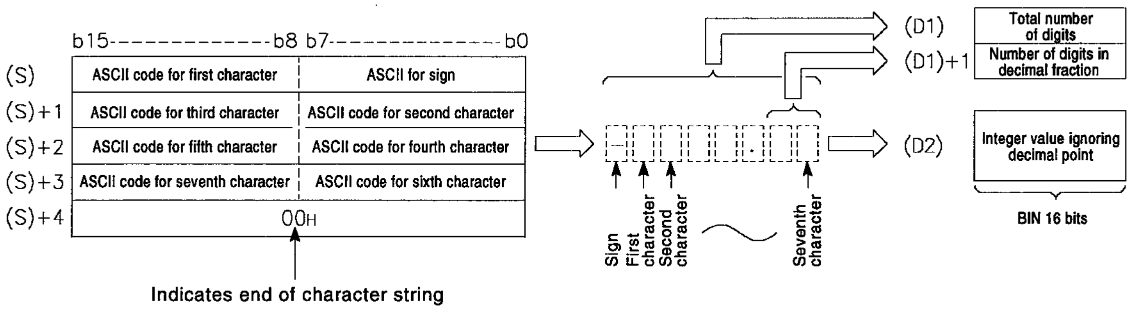
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where character string to convert to BIN data is being stored	Character string
(D1)	First device number of devices that will store the number of BIN data digits after conversion	BIN 16 bits
(D2)	Initial number of the device to store BIN data after conversion	BIN 16/32 bits

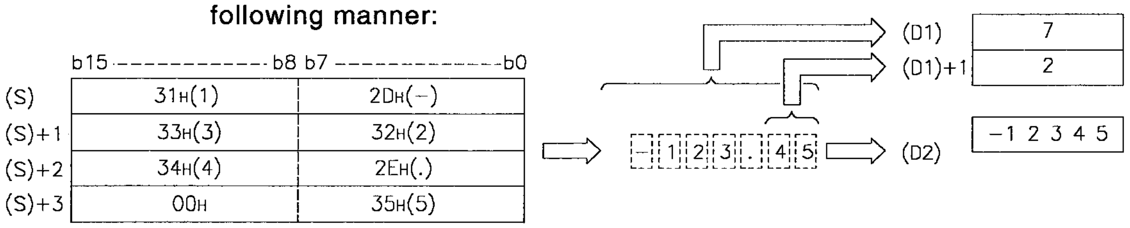
Functions

VAL

(1) Converts character strings stored in the device numbers starting from that designated at (S) to BIN 16-bit data, and stores the number of digits and BIN data in (D1) and (D2).
For conversions from character strings to BIN, all data from the device number designated by (S) to the point of the device number where "00H" is being stored will be treated as character strings.



For example, if the character string "-123.45" following (S) has been designated, the operation result would be stored at (D1) and (D2) in the following manner:



(2) The total number of characters that can be designated as a character string at (S) is from 2 to 8.

(3) From 0 to 5 characters from the character string designated at (S) can become the decimal fraction part.
However, this number must not exceed the total number of digits minus 3.

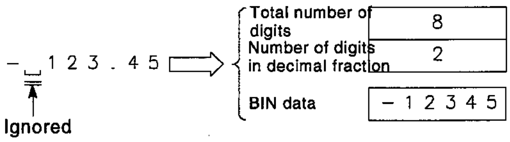
(4) The numerical value of character strings that can be converted to BIN values is from -32768 to 32767, when the decimal point is ignored.
Numerical value character strings, excluding the sign and the decimal point, can be designated only within the range from "30_H" to "39_H".

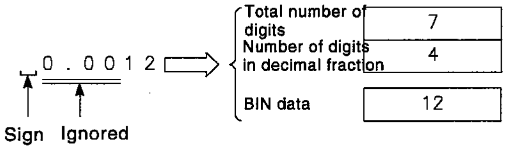
(5) The sign "20_H" will be stored if the numerical value is positive, and the sign "$2D_H$" will be stored if it is negative.

(6) "$2E_H$" is set for the decimal point.

(7) The total number of digits stored at (D1) amounts to all characters expressing numerical values (including signs and decimal points).
The characters following the decimal point stored at (D1)+1 include the number of characters from "$2E_H$ (.)" onward.
The BIN data stored at (D2) is the character string ignoring the decimal point that has been converted to BIN data.

(8) In cases where there is a "20_H (space)" or a "30_H (0)" between the sign and the first numerical value other than a zero in the character string designated at (S), the "20_H" or "30_H" will be ignored during conversion to BIN values.



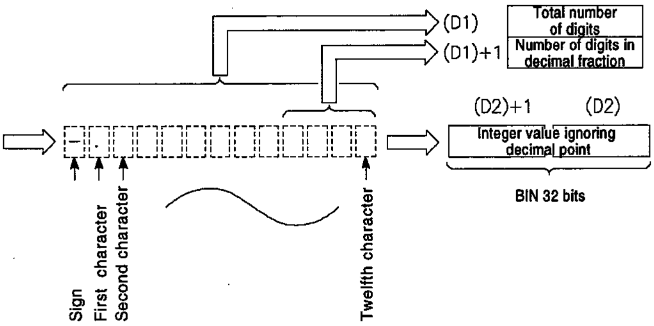


DVAL

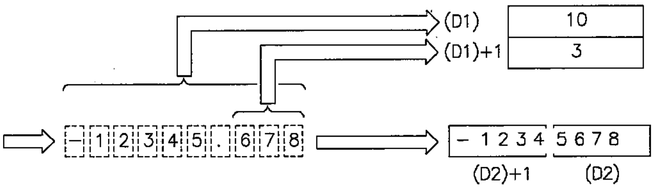
(1) Converts character string being stored following the device number designated by (S) to BIN 32-bit data, and stores at (D1) and (D2). For conversions from character strings to BIN, all data from the device number designated by (S) to the point of the device number where "00H" is being stored will be treated as character strings.

	b15------------b8	b7------------b0
(S)	ASCII code for first character	ASCII for sign
(S)+1	ASCII code for third character	ASCII code for second character
(S)+2	ASCII code for fifth character	ASCII code for fourth character
(S)+3	ASCII code for seventh character	ASCII code for sixth character
(S)+4	ASCII code for ninth character	ASCII code for eighth character
(S)+5	ASCII code for eleventh character	ASCII code for tenth character
(S)+6	00H	ASCII code for twelfth character

Indicates end of character string



	b15------------b8	b7------------b0
(S)	31H (1)	2DH (−)
(S)+1	33H (3)	32H (2)
(S)+2	35H (5)	34H (4)
(S)+3	36H (6)	2EH (.)
(S)+4	38H (8)	37H (7)
(S)+5	00H	



(2) The total number of characters in the character string indicated by (S) is from 2 to 13.

(3) From 0 to 10 characters in the character string indicated by (S) can be the decimal fraction part.
However, this number must not exceed the total number of digits minus 3.

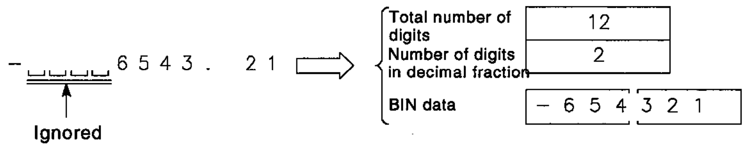
(4) Numerical character strings that can be converted to BIN values are those that, when the decimal point is ignored, fall within the range of from -2147483648 to 2147483647.
Numerical value character strings, excluding the sign and the decimal point, can be designated only within the range from "30H" to "39H".

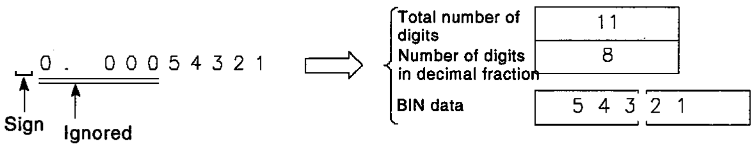
(5) The sign "20H" will be stored if the numerical value is positive, and the sign "2DH" will be stored if it is negative.

(6) "2EH" is set for the decimal point.

(7) The total number of digits stored at (D1) amounts to all characters expressing numerical values (including signs and decimal points).
The characters following the decimal point stored at (D1)+1 include the number of characters from "2EH (.)" onward.
The BIN data stored at (D2) is the character string ignoring the decimal point that has been converted to BIN values.

(8) In cases where there is a "20_H (space)" or a "30_H (0)" between the sign and the first numerical value other than a zero in the character string designated at (S), the "20_H" or "30_H" will be ignored during conversion to BIN values.





Operation Errors

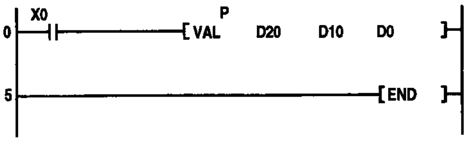
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The number of characters in the character string designated by (S1) falls outside the ranges shown below: (Error code: 4100)
  When VAL instruction is in use .... From 2 to 8 characters
  When DVAL instruction is in use... From 2 to 13 characters
- The number of characters in the decimal fraction portion of the character string designated by (S) falls outside the ranges shown below: (Error code: 4100)
  When VAL instruction is in use .... 0 to 5
  When DVAL instruction is in use... 0 to 10
- The total number of characters in the character string designated by (S) and the number of characters in the decimal fraction part stand in a relationship that is outside the range indicated below: (Error code: 4100)
  Total number of characters minus 3 is equal to or greater than the number of characters in the decimal fraction part
- An ASCII code other than "20_H" or "$2D_H$" has been set for the sign. (Error code: 4100)
- An ASCII code other than from "30_H" to "39_H" or "$2E_H$" (decimal point) has been set as a digit for one of the individual numbers. (Error code: 4100)
- There has been more than one decimal point set in the value. (Error code: 4100)
- The value of the BIN value when converted falls outside the following ranges: (Error code: 4100)
  When VAL instruction is in use .... -32768 to 32767
  When DVAL instruction is in use... -2147483648 to 2147483647
- The ASCII code "00_H" has been set somewhere between the device number designated by (S) and the final device number of the relevant device range. .... (Error code: 4101)

Program Example

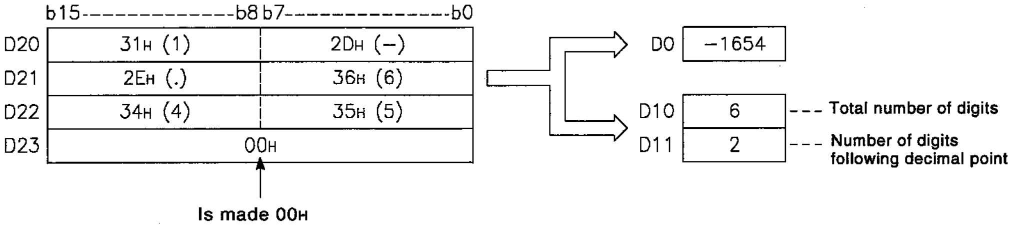
(1) The following program reads the character string data stored from D20 through D22 as an integer, converts it to a BIN value, and stores it at D0 when X0 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	VALP	D20 D10 D0
5	END	



	b15 - - - - - - - - - - - - b8	b7 - - - - - - - - - - - - - b0
D20	31H (1)	2DH (−)
D21	2EH (.)	36H (6)
D22	34H (4)	35H (5)
D23	00H	

Is made 00H

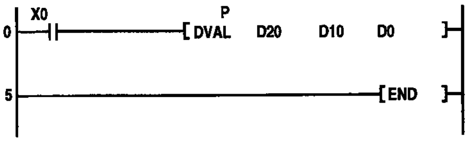
D0 −1654

D10 6 --- Total number of digits

D11 2 --- Number of digits following decimal point

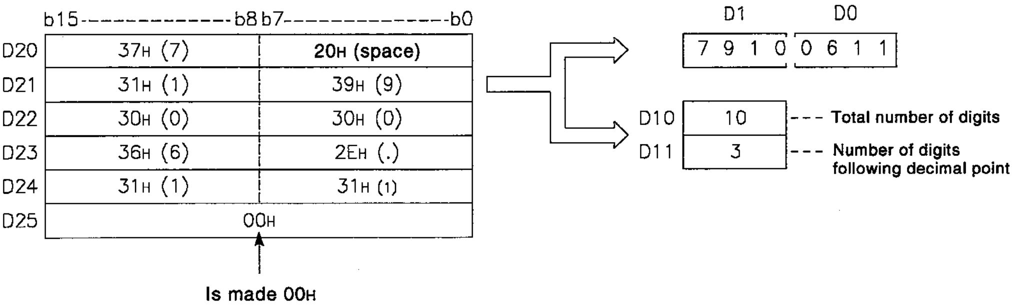
(2) The following program reads the character string data stored from D20 through D24 as an integer, converts it to a BIN value, and stores it at D0 when X0 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	ESTRP	D0 R10 D10
5	END	



	b15 - - - - - - - - - - - - b8	b7 - - - - - - - - - - - - - b0
D20	37H (7)	20H (space)
D21	31H (1)	39H (9)
D22	30H (0)	30H (0)
D23	36H (6)	2EH (.)
D24	31H (1)	31H (1)
D25	00H	

Is made 00H

D1	D0
7 9 1 0	0 6 1 1

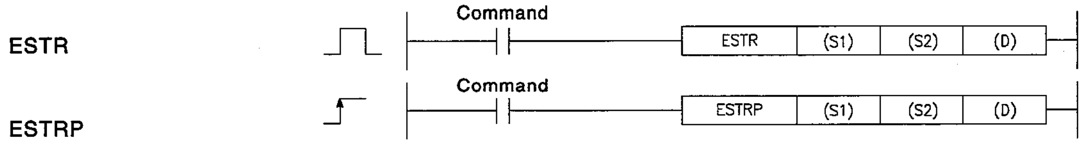
D10 10 --- Total number of digits

D11 3 --- Number of digits following decimal point

7.11.11 Conversion from floating decimal point to character string data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—	o		—	o	—
(S1)	—	o		—	—		—	—	—
(D)	—	o		—	—		—	—	—

[Instruction Symbol] [Execution Condition]



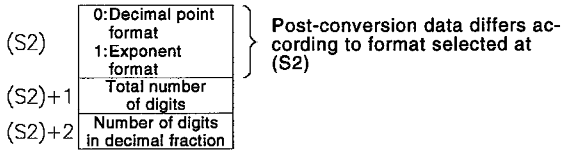
Set Data

Set Data	Meaning	Data Type
(S1)	Floating decimal point data to convert or the initial number of the device where such data is being stored	Real number
(S2)	First device number of devices where display designation for numerical value to convert is being stored	BIN 16 bits
(D)	First device number of devices storing converted character strings	Character string

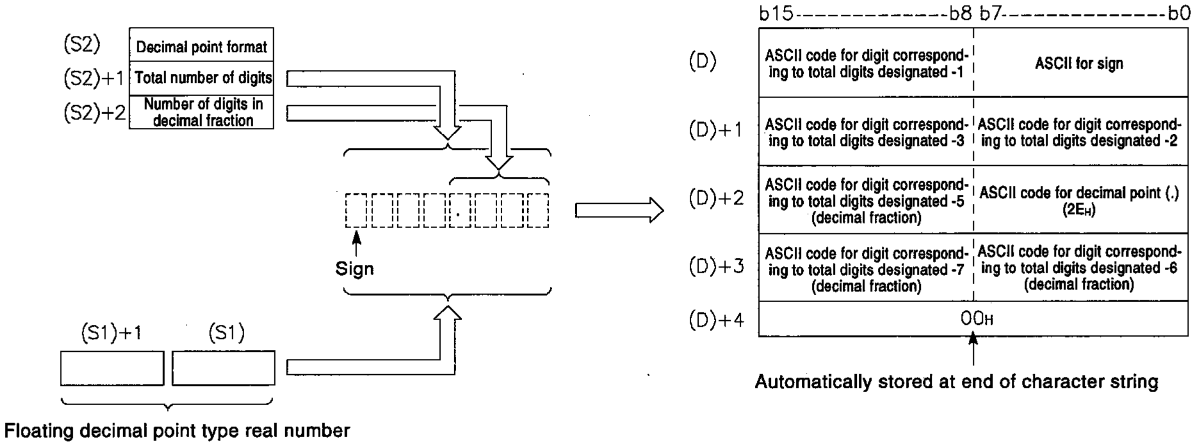
Functions

(1) Converts floating decimal point type real number data being stored at the device designated by (S1) to character string in accordance with the display designation stored following the device designated by (S2), and stores the result following the device number designated by (D).

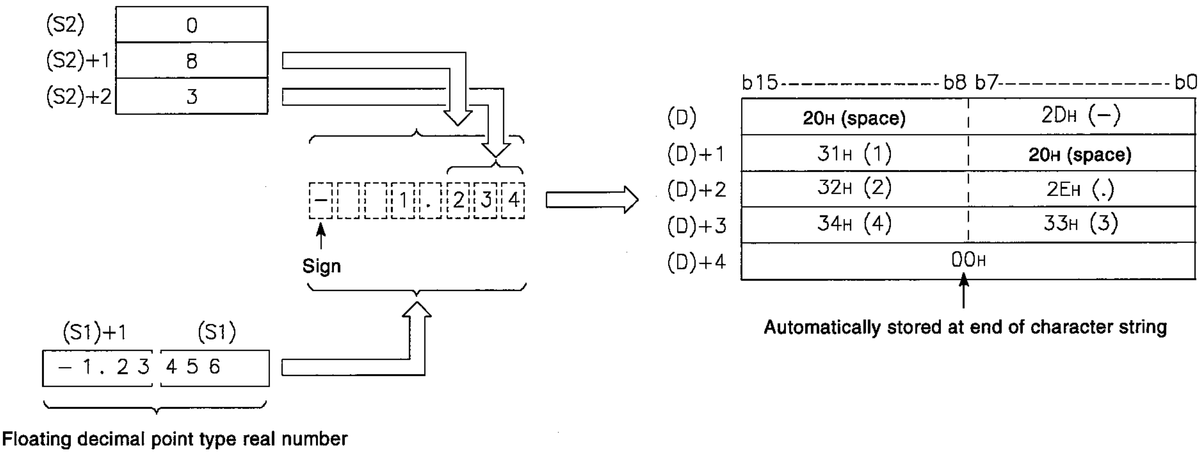
(2) The post-conversion data will differ depending on the display designation designated by (S2).



When using decimal point format



For example, in a case where there were 8 total digits, with 3 digits coming in the decimal fraction part, and the value designated was -1.23456, the operation result would be stored following (D) in the following manner:



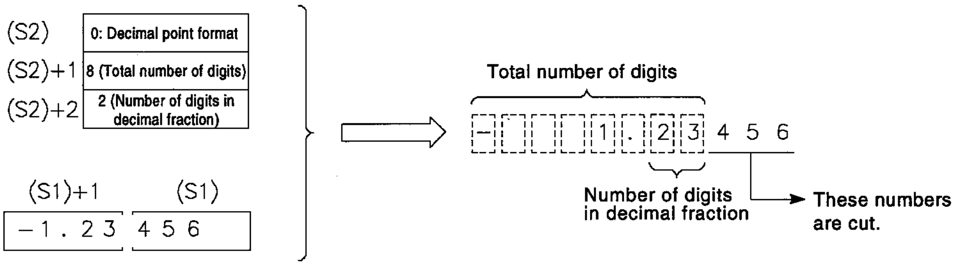
(a) The total number of digits that can be designated by (S2)+1 is as shown below:
When the number of digits following the decimal point is "0"
. . . Total digits ≥ 2
When the number of digits following the decimal point is some value other than "0"
. . . Total digits ≥ (number of digits following decimal point + 3)

(b) The number of digits that can be designated to follow the decimal point is from 0 to 7.
However, the number of digits after the decimal point should be equal to or smaller than the total number of digits minus three.

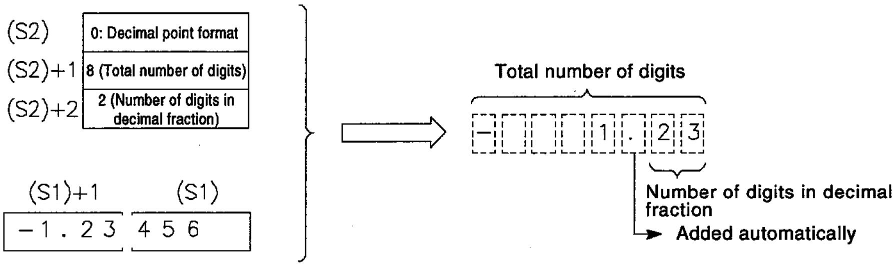
(c) After conversion, character string data is stored at the device number following (D) as indicated below:

1) The sign "20H (space)" will be stored if the floating decimal point type real number is positive, and the sign "2DH (minus sign -)" will be stored if it is negative.

2) In cases where there is floating decimal point real number data within the range of the number of digits to follow the decimal point, and the decimal values have not been stored, the lower level decimal values will be cut.

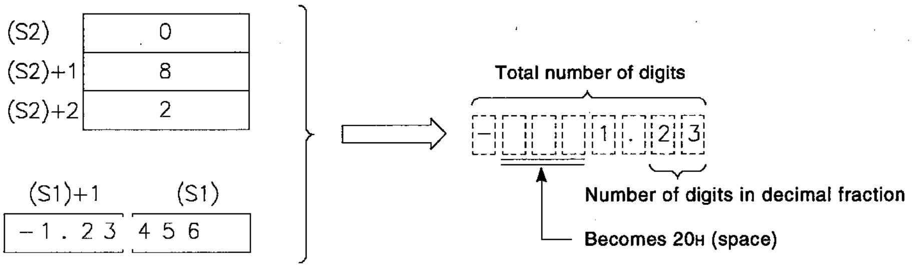


3) If the number of digits following the decimal point has been set at any value other than "0", "2EH (.)" will automatically be stored at the position before the first of the specified number of digits.



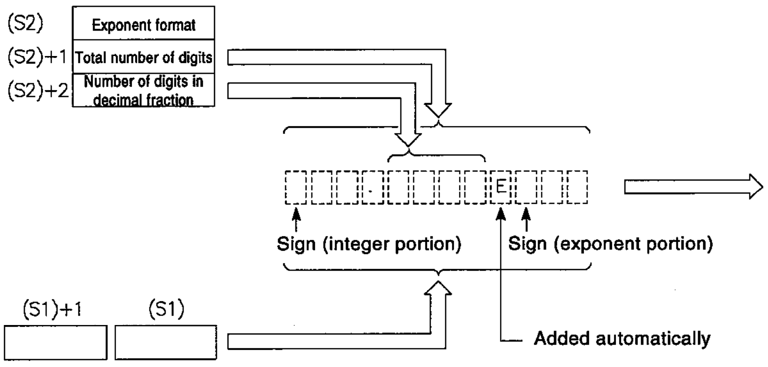
If the number of digits in the decimal fraction part of the number is "0", the ASCII code "2EH (.)" will not be stored.

4) If the total number of digits, excluding the sign, the decimal point and numbers following the decimal point, is greater than the part before the decimal point of the floating decimal point type real number data, "20H (space)" will be stored between the sign and the part before the decimal point of the number.



5) The value "00H" is automatically stored at the end of the converted character string.

When using exponent format

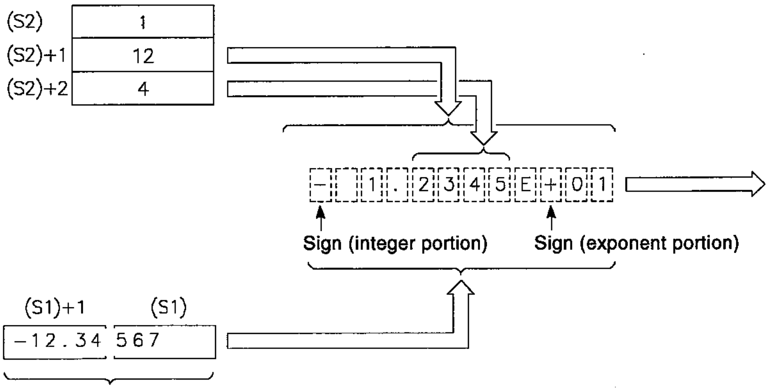


Floating decimal point type real number

	b15 ------------ b8	b7 ------------ b0
(D)	ASCII code for digit corresponding to total digits designated -1	ASCII for sign
(D)+1	ASCII code for decimal point (.) (2 EH)	ASCII code for digit corresponding to total digits designated -2
(D)+2	ASCII code for digit corresponding to total digits designated -5(decimal fraction)	ASCII code for digit corresponding to total digits designated -4(decimal fraction)
(D)+3	ASCII code for digit corresponding to total digits designated -7 (decimal fraction)	ASCII code for digit corresponding to total digits designated -6 (decimal fraction)
(D)+4	ASCII for sign (exponent portion)	45H (E)
(D)+5	ASCII code for digit corresponding to total digits designated -11 (exponent portion)	ASCII code for digit corresponding to total digits designated -10 (exponent portion)
(D)+6	00H	

Automatically stored at end of character string

For example, in a case where there were 12 total digits and 4 of these were in the decimal fraction portion, and the value designated was -12.34567, the operation results would be stored following (D) in the following manner:



Floating decimal point type real number

	b15 ------------ b8	b7 ------------ b0
(D)	20H (space)	2DH (-)
(D)+1	2EH (.)	31H (1)
(D)+2	33H (3)	32H (2)
(D)+3	35H (5)	34H (4)
(D)+4	2CH (+)	45H (E)
(D)+5	31H (1)	30H (0)
(D)+6	00H	

Automatically stored at end of character string

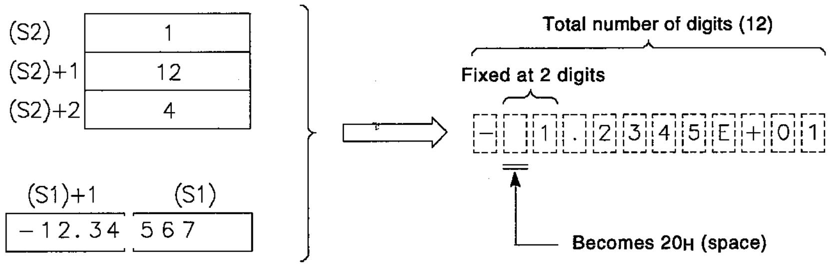
(a) The total number of digits that can be designated by (S2)+1 is as shown below:
When the number of digits following the decimal point is "0".
... Total digits ≥ 2
When the number of digits following the decimal point is some value other than "0"
... Total digits ≥ (Number of digits in decimal fraction + 7)

(b) The number of digits that can be designated to follow the decimal point is from 0 to 7.
However, the number of digits in the decimal fraction portion should be equal to or less than the total number of digits minus 7.

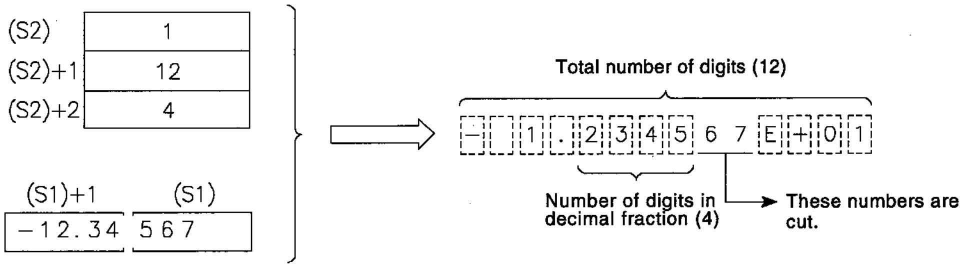
(c) After conversion, character string data is stored at the device number following (D) as indicated below:

1) If the floating decimal point type real number data is positive in value, the sign before the integer will be stored as ASCII code "20H (space)", and if it is a negative value, the sign will be stored as "2DH (-)".

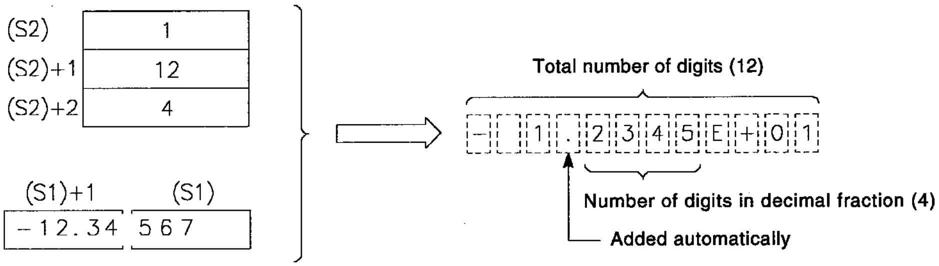
2) The integer portion is fixed at 2 digits.
If the integer portion is only 1 digit, the ASCII code "20H (space)" will be stored between the sign and the integer.



3) If the decimal fraction portion of the floating decimal point type real number is longer than can be contained within the range for the total number of digits, the trailing numbers of the decimal fraction will be cut from the result.



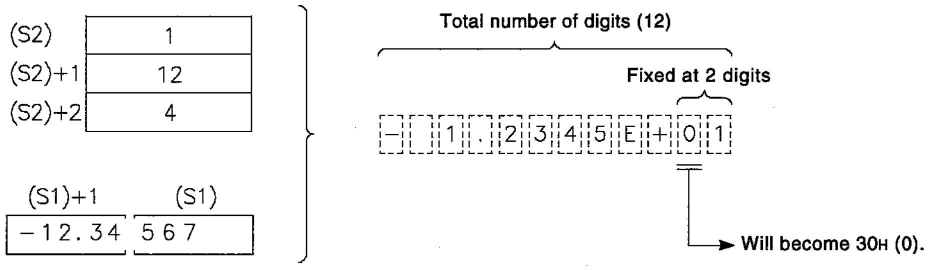
4) If the number of digits following the decimal point has been set at any value other than "0", "2EH (.)" will automatically be stored at the position before the first of the specified number of digits.



If the number of digits in the decimal fraction portion of the number is "0", the ASCII code "2EH (.)" will not be stored.

5) The ASCII code "2CH (+)" will be stored as the sign for the exponent portion of the value if the exponent is positive in value, and the code "2DH (-)" will be stored if the exponent is a negative value.

6) The exponent portion is fixed at 2 digits.
If the exponent portion is only 1 digit, the ASCII code "30H (0)" will be stored between the sign and the exponent portion of the number.



7) The value "00H" is automatically stored at the end of the converted character string.

Operation Errors

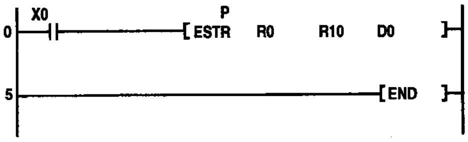
(1) In the following cases an operation error occurs and the error flag goes ON.

- The (S1) value is not "0", or it is not within the range indicated below: (Error code: 4100)
  $\pm 2^{-127} \leq S1 < \pm 2^{128}$
- The format designated by (S2) was neither 0 nor 1. (Error code: 4100)
- The total number of digits designated by (S2)+1 was outside the ranges shown below: (Error code: 4100)
  When using decimal point format
    When the number of digits following the decimal point is "0"
    .... Total digits ≥ 2
    When the number of digits following the decimal point is some value other than "0"
    .... Total digits ≥ (Number of digits in decimal fraction + 3)
  When using exponent format
    When the number of digits following the decimal point is "0"
    .... Total digits ≥ 2
    When the number of digits following the decimal point is some value other than "0"
    .... Total digits ≥ (Number of digits in decimal fraction + 7)
- The number of digits designated for the decimal fraction portion of the value by (S2)+2 was outside the ranges indicated below: (Error code: 4100)
  When using decimal point format
    Number of digits in decimal fraction portion is smaller than or equal to the total number of digits minus 3.
  When using exponent format
    Number of digits in decimal fraction portion is smaller than or equal to the total number of digits minus 7.
- The device range to store the character string designated by (D) exceeds the relevant device range. (Error code: 4101)

Program Example

(1) The following program converts the floating decimal point type real number data which had been stored at R0 and R1 in accordance with the conversion designation that is being stored at R10 through R12, and stores the result following D0 when X0 goes ON.

[Ladder Mode]

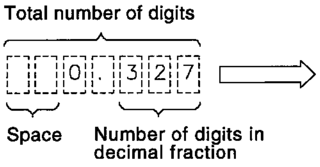


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ESTRP	R0 R10 D0
5	END	

R10	0	Conversion format
R11	7	Total number of digits
R12	3	Number of digits in decimal fraction

R1	R0
0 . 0 3 2 7	4 5 7

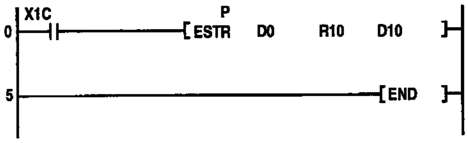


	b15 - b8	b7 - b0
D0	20H (space)	20H (space)
D1	2EH (.)	30H (0)
D2	32H (2)	33H (3)
	00H	37H (7)

Automatically stored

(2) The following program converts the floating decimal point type real number data which had been stored at D0 and D1 in accordance with the conversion designation that is being stored at R10 through R12, and stores the result following D0 when X0 goes ON.

[Ladder Mode]

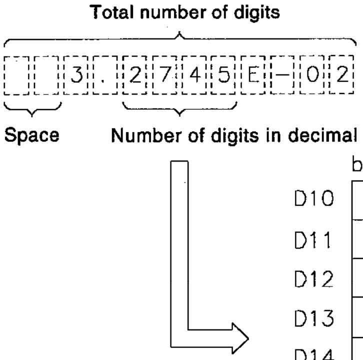


[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	ESTRP	D0 R10 D10
5	END	

R10	1 (exponent format)	Conversion format
R11	12	Total number of digits
R12	4	Number of digits in decimal fraction

R1	R0
0 . 0 3 2 7	4 5 7 8



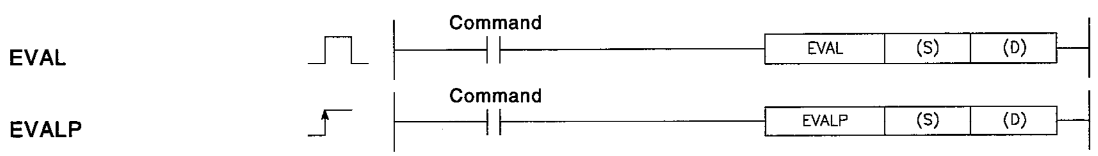
	b15 - b8	b7 - b0
D10	20H (space)	20H (space)
D11	2EH (.)	33H (3)
D12	37H (7)	32H (2)
D13	35H (5)	34H (4)
D14	2DH (–)	45H (E)
D15	32H (2)	30H (0)
D16	00H	

Automatically stored

7.11.12 Conversion from character string to floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	—	—	—	o	—
(D)	—	o		—	o	o	—	—	—

[Instruction Symbol] [Execution Condition]



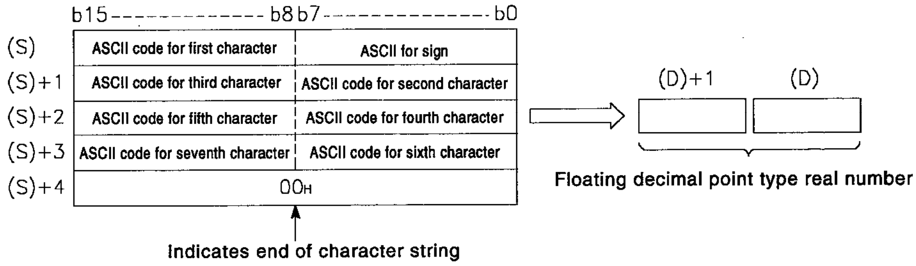
Set Data

Set Data	Meaning	Data Type
(S)	Character string data to be converted to floating decimal point type real number data, or first device number of devices where such data is being stored	Character string
(D)	First device number of devices that will store floating decimal point type real number data after conversion	Real number

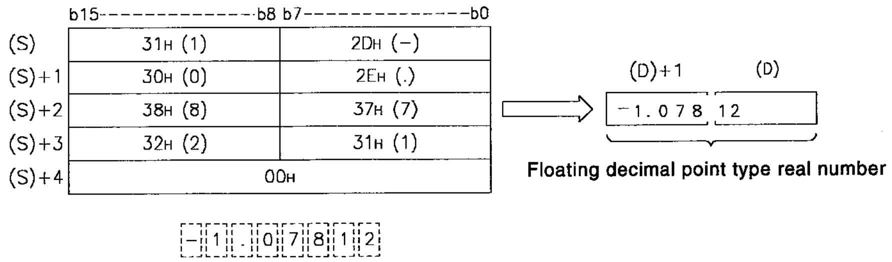
Functions

(1) Converts character string being stored following the device number designated by (S) to floating decimal point type real number, and stores result at device designated by (D).

(2) The designated character string can be converted to floating decimal point type real number data either in the decimal point format or the exponent format.



(a) When using decimal point format



(b) When using exponent format

	b15------------b8	b7------------b0
(S)	20_H (space)	$2D_H$ (–)
(S)+1	$2E_H$ (.)	31_H (1)
(S)+2	32_H (2)	33_H (3)
(S)+3	31_H (1)	30_H (0)
(S)+4	$2C_H$ (+)	45_H (E)
(S)+5	30_H (0)	31_H (1)
(S)+6	00_H	

(D)+1	(D)
–1.320	1E+10

Floating decimal point type real number

`–  1.3201E+10`

(3) Excluding the sign, decimal point, and exponent portion of the result, 6 digits of the character string designated by (S) to be converted to a floating decimal point type real number will be effective; everything from the 7th digit on will be cut from the result.

(a) When using decimal point format

	b15------------b8	b7------------b0
(S)	20_H (space)	$2D_H$ (–)
(S)+1	31_H (1)	20_H (space)
(S)+2	33_H (3)	$2E_H$ (.)
(S)+3	31_H (1)	30_H (0)
(S)+4	36_H (6)	35_H (5)
(S)+5	31_H (1)	38_H (8)
(S)+6	00_H	32_H (2)

(D)+1	(D)
–1. 30	1 5 6

Floating decimal point type real number

`–   1.90156812`

(b) When using exponent format

	b15------------b8	b7------------b0
(S)	20_H (space)	$2D_H$ (–)
(S)+1	$2E_H$ (.)	31_H (1)
(S)+2	35_H (5)	33_H (3)
(S)+3	33_H (3)	30_H (0)
(S)+4	31_H (1)	34_H (4)
(S)+5	45_H (E)	32_H (2)
(S)+6	30_H (0)	$2D_H$ (–)
(S)+7	00_H	32_H (2)

(D)+1	(D)
–1.350	34E –2

Floating decimal point type real number

`–  1.3503412E–02`

These numbers are cut.

(4) If, in the character string designated by (S), there is the ASCII code "20_H (space)" or "30_H (0)" within the string not including the initial zero, these will be ignored when the conversion is done.

	b15------------b8	b7------------b0
(S)	20_H (space)	$2D_H$ (–)
(S)+1	31_H (1)	30_H (0)
(S)+2	32_H (2)	$2E_H$ (.)
(S)+3	31_H (1)	33_H (3)
(S)+4	00_H	

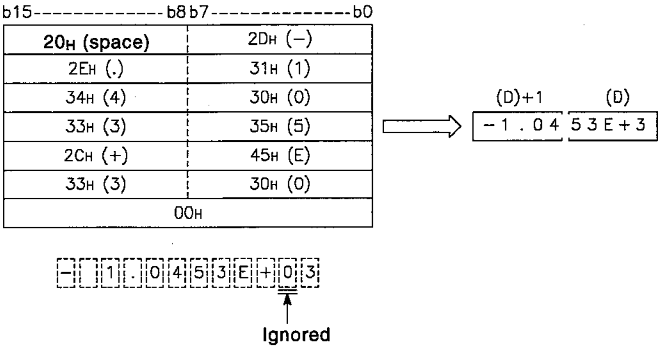
(D)+1	(D)
1 . 2	3 1

Floating decimal point type real number

`  01.231`

Ignored

(5) In a case where the ASCII code "30H (0)" exists between the character "E" and a number in an exponent format character string, the "30H" would be ignored when the conversion was performed.



(6) A maximum of up to 24 characters can be set for a character string.

Operation Errors

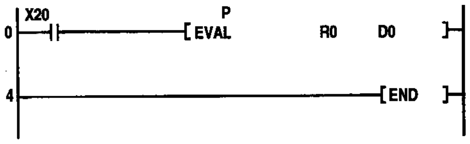
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The integer portion or the decimal fraction portion contains a character other than one in the range of from "30H (0)" to "39H (9)". (Error code: 4100)
- There are two or more occurrences of the character "2EH" (.) in the character string designated by (D). (Error code: 4100)
- The exponent portion of the value contains characters other than "45H (E), 2CH (+)" or "45H (E), 2DH (-)", or there is more than one exponent. (Error code: 4100)
- Data after conversion is not 0, or $1.0 \times 2^{-127} \leq$ (absolute value) 1.0×2^{128}. (Error code: 4100)
- The code "00H" does not appear in the range from (S) to the relevant device. (Error code: 4101)
- The number of characters in the character string is either 0 or more than 24. (Error code: 4100)

Program Example

(1) The following program converts the character string stored following R0 to a floating decimal point type real number, and stores the result at D0 and D1 when X20 goes ON.

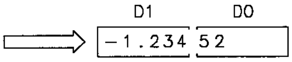
[Ladder Mode]

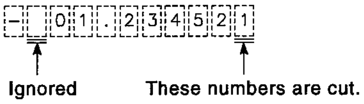


[List Mode]

Steps	Instruction	Device
0	LD	X20
1	EVALP	R0 D0
4	END	

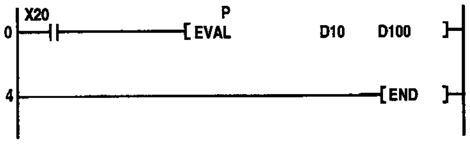
	b15------------b8	b7------------b0
R0	20H (space)	2DH (−)
R1	31H (1)	30H (0)
R2	32H (2)	2EH (.)
R3	34H (4)	33H (3)
R4	32H (2)	35H (5)
R5	00H	31H (1)





(2) The following program converts the character string stored following D10 to a floating decimal point type real number, and stores the result at D100 and D101 when X20 is ON.

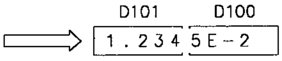
[Ladder Mode]

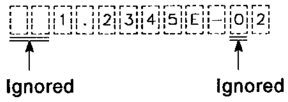


[List Mode]

Steps	Instruction	Device
0	LD	X20
1	EVALP	D0 D100
4	END	

	b15------------b8	b7------------b0
D10	20H (space)	20H (space)
D11	2EH (.)	31H (1)
D12	33H (3)	32H (2)
D13	35H (5)	34H (4)
D14	2DH (−)	45H (E)
D15	32H (2)	30H (0)
D16	00H	

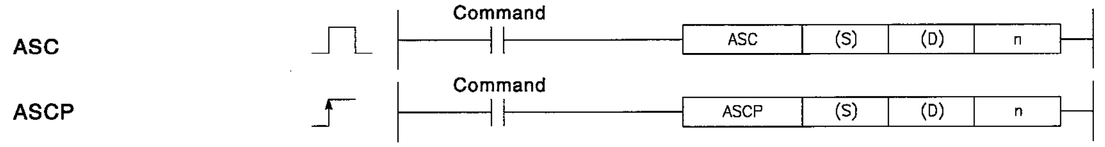




7.11.13 Conversion from hexadecimal BIN to ASCII

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—					—
(D)	—	o		—					—
n	o	o		o					—

[Instruction Symbol] [Execution Condition]

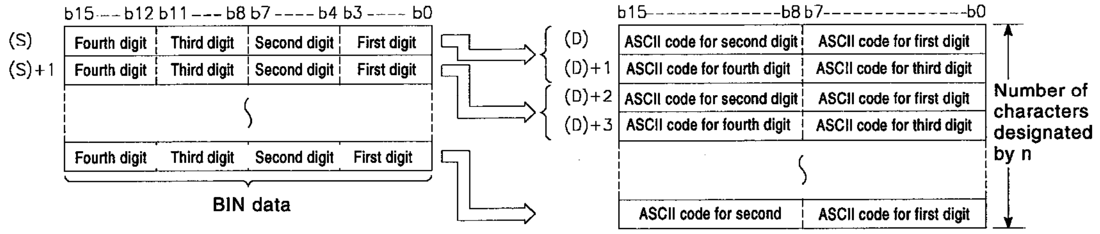


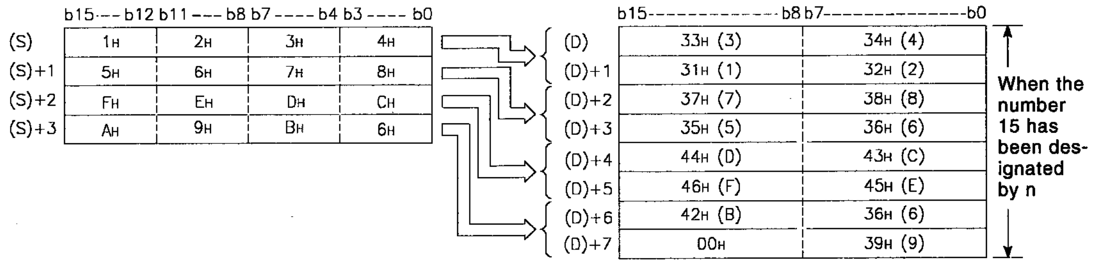
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices storing BIN data to be converted to character strings	BIN 16 bits
(D)	First device number of devices storing converted character strings	Character string
n	Number of characters stored	BIN 16 bits

Functions

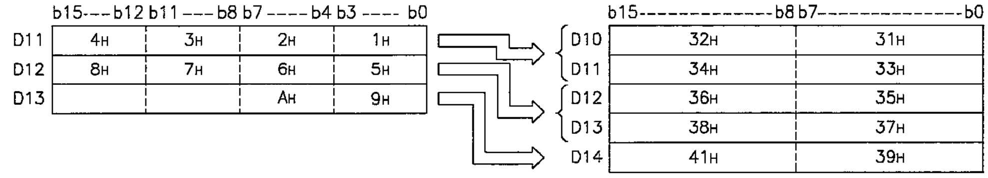
(1) Converts BIN 16-bit data stored following the device number designated by (S) to hexadecimal ASCII, and stores the result within the range for the number of characters designated by n following the device number designated by (D).





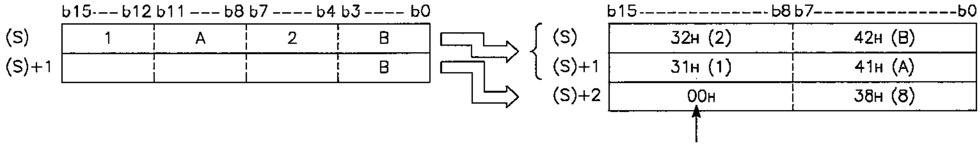
(2) The use of n to set the number of characters causes the BIN data range designated by (S) and the character string storage device range designated by (D) to be set automatically.

(3) Processing will be performed accurately even if the device range where BIN data to be converted is being stored overlaps with the device range where the ASCII data will be stored after conversion.



(4) If an odd number of characters has been designated by n, the ASCII code "00H" will be automatically stored in the upper 8 bits of the final device in the range where the character string is to be stored.

When 5 characters have been designated by n



Automatically stored

(5) If the number of characters designated by n is "0", conversion processing will not be conducted.

Operation Errors

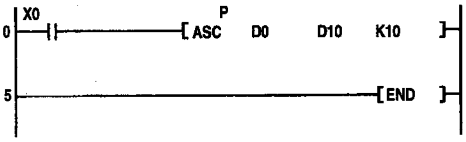
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range for the number of characters designated by n following the device number designated by (S) exceeds the relevant device range. (Error code: 4101)
- The range for the number of characters designated by n following the device number designated by (D) exceeds the relevant device range. (Error code: 4101)

Program Example

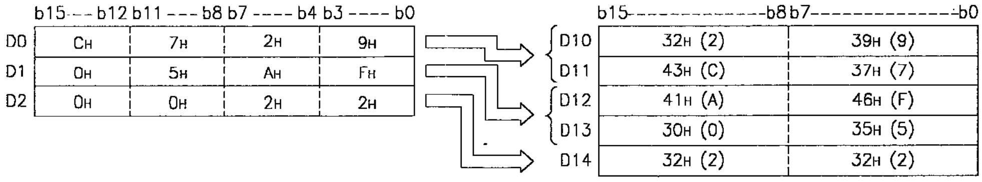
(1) The following program reads the BIN data being stored at D0 as hexadecimal values, converts them to a character string, and stores the result from D10 through D14 when X0 is turned ON.

[Ladder Mode]



[List Mode]

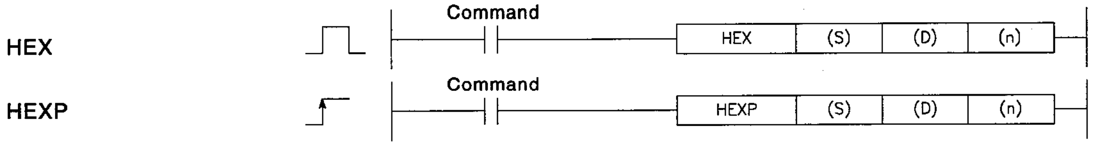
Steps	Instruction	Device
0	LD	X0
1	ASCP	D0 D10 K10
5	END	



7.11.14 Conversion from ASCII to hexadecimal BIN

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—					—
(D)	—	o		—					—
n	o	o		o					—

[Instruction Symbol] [Execution Condition]

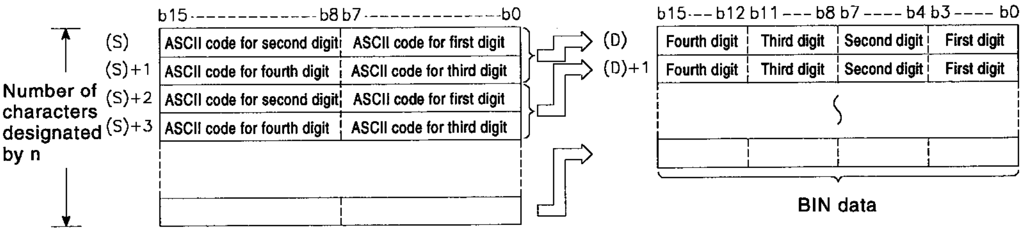


Set Data

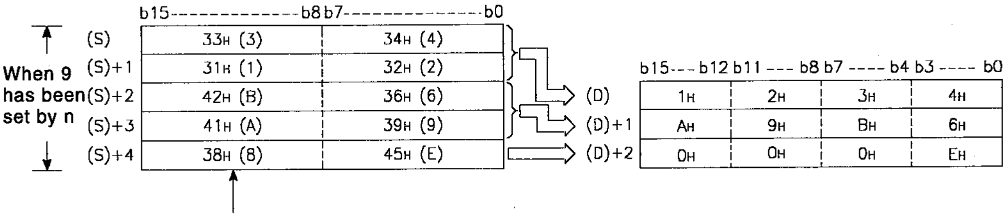
Set Data	Meaning	Data Type
(S)	First device number of devices where character string to convert to BIN data is being stored	Character string
(D)	First device number of devices that will store converted BIN data	BIN 16 bits
n	Number of characters to convert	BIN 16 bits

Functions

(1) Converts the number of characters of hexadecimal ASCII data designated by n following the device number designated by (S) into BIN values and stores following the device number designated by (D).



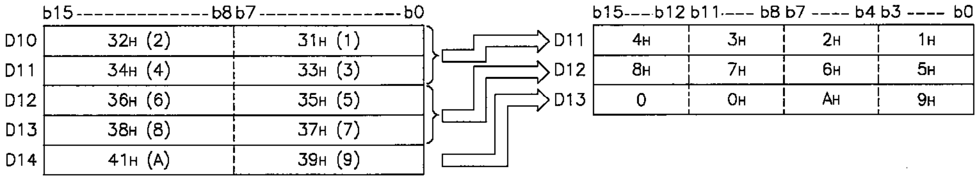
For example, if the number 9 has been designated by n, the operation would be as follows:



Because a 9-character unit has been designated, the "38H" does not change.

(2) The use of n to set the number of characters causes the character string range designated by (S) and the BIN data storage device range designated by (D) to be set automatically.

(3) Accurate processing will be conducted even in cases where the range of devices where the ASCII code to be converted is being stored overlaps with the range of devices that will store the converted BIN data.

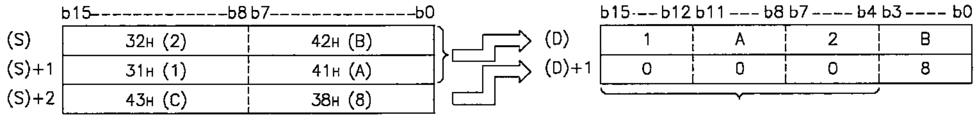


	b15 ---- b8	b7 ---- b0
D10	32H (2)	31H (1)
D11	34H (4)	33H (3)
D12	36H (6)	35H (5)
D13	38H (8)	37H (7)
D14	41H (A)	39H (9)

	b15 --- b12	b11 --- b8	b7 ---- b4	b3 ---- b0
D11	4H	3H	2H	1H
D12	8H	7H	6H	5H
D13	0	0H	AH	9H

(4) If the number of characters designated by n is not divisible by 4, "0" will be automatically stored after the designated number of characters in the final device number of the devices which are storing the converted BIN values.

When 5 characters have been designated by n



	b15 ---- b8	b7 ---- b0
(S)	32H (2)	42H (B)
(S)+1	31H (1)	41H (A)
(S)+2	43H (C)	38H (8)

	b15 --- b12	b11 --- b8	b7 ---- b4	b3 ---- b0
(D)	1	A	2	B
(D)+1	0	0	0	8

The value "0" is automatically stored outside the range of the number of characters.

(5) If the number of characters designated by n is "0", conversion processing will not be conducted.

(6) ASCII code that can be designated by (S) includes from "30H" to "39H" and from "41H" to "46H".

Operation Errors

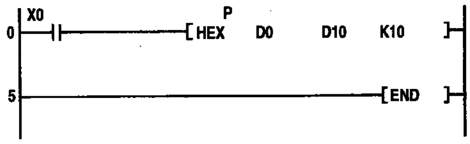
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Characters outside the hexadecimal character string (that is, characters that are not in the range of from "30H" to "39H", or from "41H" to "46H") have been set by (S). (Error code: 4100)
- The range for the number of characters designated by n following the device number designated by (S) exceeds the relevant device range. (Error code: 4101)
- The range for the number of characters designated by n following the device number designated by (D) exceeds the relevant device range. (Error code: 4101)
- The number of characters designated by n is a negative value.

Program Example

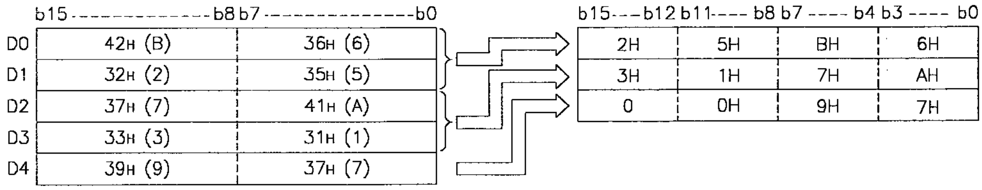
(1) The following program converts the character string being stored from D0 through D4 to BIN data and stores the result from D10 through D14 when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	HEXP	D0 D10 K10
5	END	



	b15 ---- b8	b7 ---- b0
D0	42H (B)	36H (6)
D1	32H (2)	35H (5)
D2	37H (7)	41H (A)
D3	33H (3)	31H (1)
D4	39H (9)	37H (7)

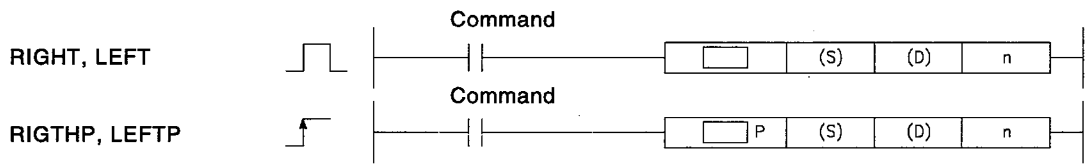
b15 --- b12	b11 --- b8	b7 ---- b4	b3 ---- b0
2H	5H	BH	6H
3H	1H	7H	AH
0	0H	9H	7H

7.11.15 Extracting character string data from the right or left

<table>
<tr><th rowspan="3">Set Data</th><th colspan="10">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System, User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th colspan="2">Constant</th><th>Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th><th>K, H</th><th>$</th><th>U</th></tr>
<tr><td>(S)</td><td>—</td><td colspan="2">o</td><td colspan="4">—</td><td>—</td><td>o</td><td>—</td></tr>
<tr><td>(D)</td><td>—</td><td colspan="2">o</td><td colspan="4">—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>n</td><td>o</td><td colspan="2">o</td><td colspan="4">o</td><td>o</td><td>—</td><td>—</td></tr>
</table>

[Instruction Symbol] [Execution Condition]

□ indicates RIGHT/LEFT



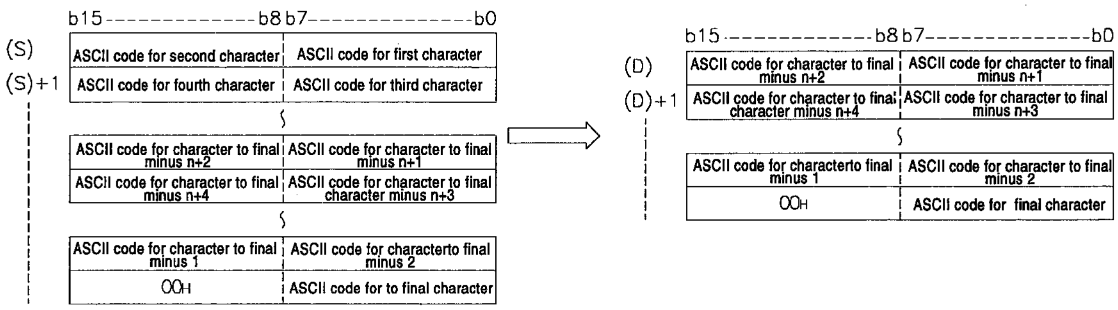
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where character string is being stored	Character string
(D)	First device number of devices where a character string of n number of characters from the right or left of (S) will be stored	
n	Number of characters to extract	BIN 16 bits

Functions

RIGHT

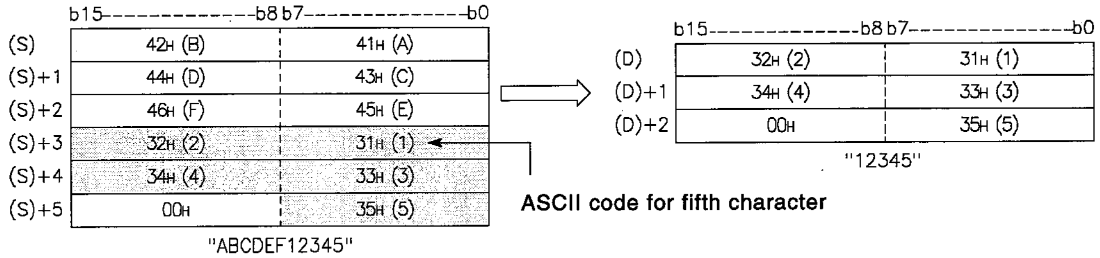
(1) Stores n number of characters from the right side of the character string (the end of the character string) being stored in devices starting from that whose number is designated by (S), in devices starting from that whose number is designated by (D).



(S)	ASCII code for second character	ASCII code for first character
(S)+1	ASCII code for fourth character	ASCII code for third character
	⁐	
	ASCII code for character to final minus n+2	ASCII code for character to final minus n+1
	ASCII code for character to final minus n+4	ASCII code for character to final character minus n+3
	⁐	
	ASCII code for character to final minus 1	ASCII code for characterto final minus 2
	00H	ASCII code for to final character

(D)	ASCII code for character to final minus n+2	ASCII code for character to final minus n+1
(D)+1	ASCII code for character to final character minus n+4	ASCII code for character to final minus n+3
	⁐	
	ASCII code for characterto final minus 1	ASCII code for character to final minus 2
	00H	ASCII code for final character

When n=5



	b15 – b8	b7 – b0
(S)	42H (B)	41H (A)
(S)+1	44H (D)	43H (C)
(S)+2	46H (F)	45H (E)
(S)+3	32H (2)	31H (1)
(S)+4	34H (4)	33H (3)
(S)+5	00H	35H (5)

"ABCDEF12345"

	b15 – b8	b7 – b0
(D)	32H (2)	31H (1)
(D)+1	34H (4)	33H (3)
(D)+2	00H	35H (5)

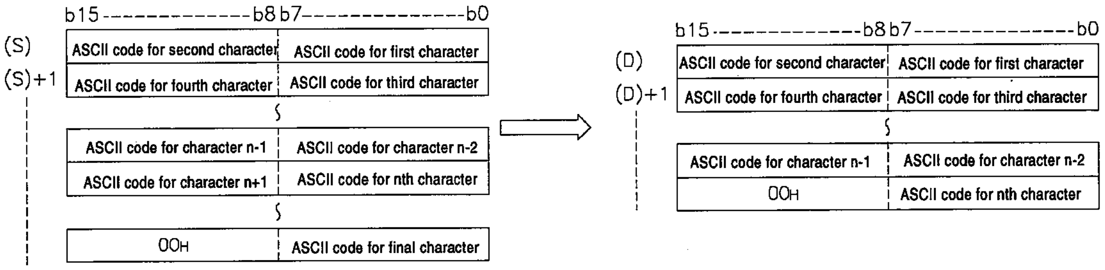
"12345"

ASCII code for fifth character

(2) If the number of characters designated by n is "0", the NULL code (00H) will be stored at (D).

LEFT

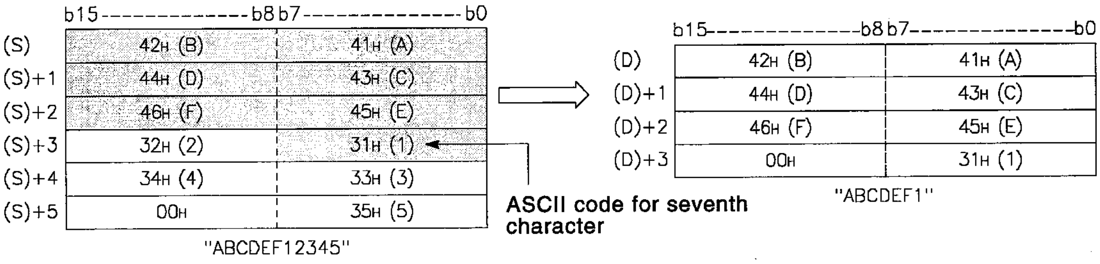
(1) Stores n number of characters from the left side of the character string (the beginning of the character string) being stored in devices starting from that whose number is designated by (S), in devices starting from that whose number designated by (D).



	b15 – b8	b7 – b0
(S)	ASCII code for second character	ASCII code for first character
(S)+1	ASCII code for fourth character	ASCII code for third character
	∫	∫
	ASCII code for character n-1	ASCII code for character n-2
	ASCII code for character n+1	ASCII code for nth character
	∫	∫
	00H	ASCII code for final character

	b15 – b8	b7 – b0
(D)	ASCII code for second character	ASCII code for first character
(D)+1	ASCII code for fourth character	ASCII code for third character
	∫	∫
	ASCII code for character n-1	ASCII code for character n-2
	00H	ASCII code for nth character

When n=7



	b15 – b8	b7 – b0
(S)	42H (B)	41H (A)
(S)+1	44H (D)	43H (C)
(S)+2	46H (F)	45H (E)
(S)+3	32H (2)	31H (1)
(S)+4	34H (4)	33H (3)
(S)+5	00H	35H (5)

"ABCDEF12345"

	b15 – b8	b7 – b0
(D)	42H (B)	41H (A)
(D)+1	44H (D)	43H (C)
(D)+2	46H (F)	45H (E)
(D)+3	00H	31H (1)

"ABCDEF1"

ASCII code for seventh character

(2) If the number of characters designated by n is "0", the NULL code (00H) will be stored.

Operation Errors

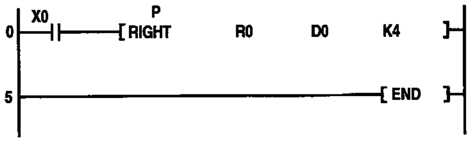
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The value of n exceeds the number of characters designated by (S). (Error code: 4101)
- The range of n characters from (D) exceeds the relevant device. (Error code: 4101)

Program Example

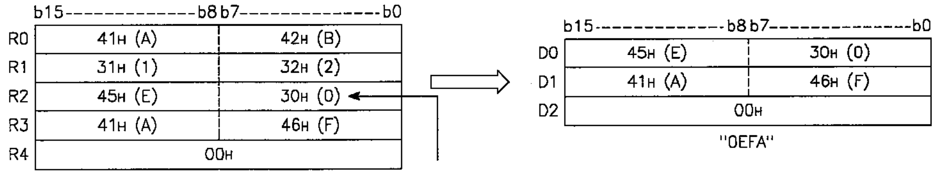
(1) The following program stores 4 characters of data from the right side of the character string being stored at R0 and stores it at D0 when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	RIGHTP	R0 D0 K4
5	END	



	b15–b8	b7–b0
R0	41H (A)	42H (B)
R1	31H (1)	32H (2)
R2	45H (E)	30H (0)
R3	41H (A)	46H (F)
R4	00H	

"BA210EFA"

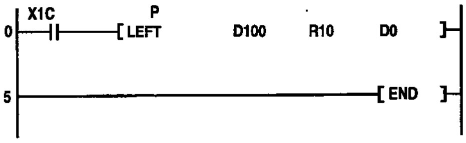
	b15–b8	b7–b0
D0	45H (E)	30H (0)
D1	41H (A)	46H (F)
D2	00H	

"0EFA"

ASCII code for fourth character

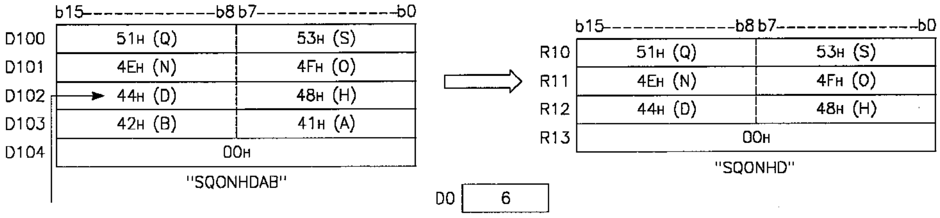
(2) The following program stores the number of characters corresponding to the value being stored in D0 from the left of the character string data being stored at D100 following R10 when X1C goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	LEFTP	D100 R10 D0
5	END	



	b15–b8	b7–b0
D100	51H (Q)	53H (S)
D101	4EH (N)	4FH (O)
D102	44H (D)	48H (H)
D103	42H (B)	41H (A)
D104	00H	

"SQONHDAB"

	b15–b8	b7–b0
R10	51H (Q)	53H (S)
R11	4EH (N)	4FH (O)
R12	44H (D)	48H (H)
R13	00H	

"SQONHD"

D0 6

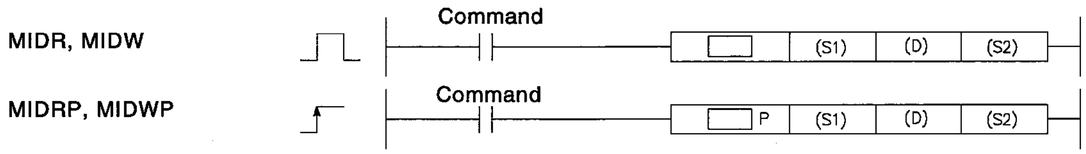
ASCII code for 6th character

7.11.16 Random selection from and replacement in character strings

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—				o	—
(D)	—	o		—				—	—
(S2)	o	o		o				—	—

[Instruction Symbol] [Execution Condition]

□ indicates MIDR/MIDW



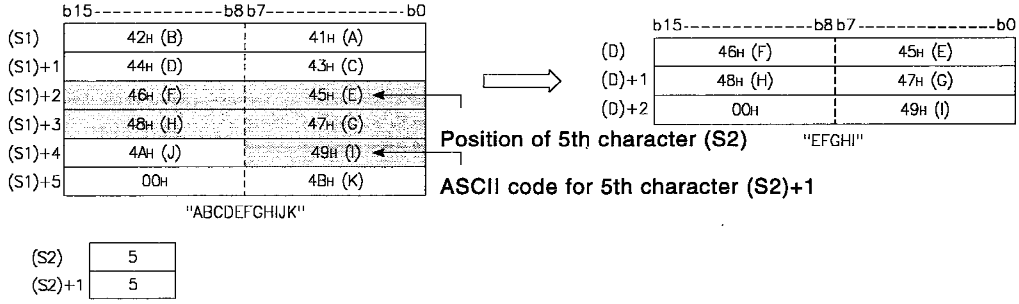
Set Data

Set Data	Meaning	Data Type
(S1)	First device number of devices where character string is being stored	Character string
(D)	First device number of devices that will store the character string data of the operation result	
(S2)	First device number of devices that will store the location of the first character and the number of characters • (S2)+1: Location of first character • (S2)+1: Number of characters	BIN 16 bits

Functions

MIDR

(1) Stores, starting from the device number designated by (D), the number of characters designated by (S2)+1 from the position designated by (S2) from the left of the character string being stored starting from the device number designated by (S1).



	b15-----b8	b7-----b0
(S1)	42H (B)	41H (A)
(S1)+1	44H (D)	43H (C)
(S1)+2	46H (F)	45H (E)
(S1)+3	48H (H)	47H (G)
(S1)+4	4AH (J)	49H (I)
(S1)+5	00H	4BH (K)

"ABCDEFGHIJK"

	b15-----b8	b7-----b0
(D)	46H (F)	45H (E)
(D)+1	48H (H)	47H (G)
(D)+2	00H	49H (I)

"EFGHI"

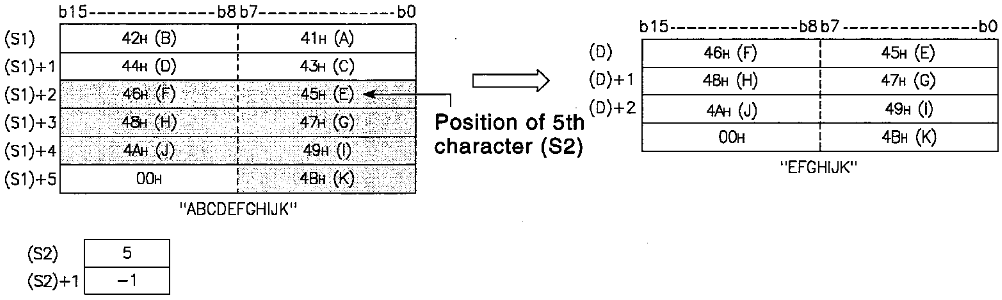
Position of 5th character (S2)

ASCII code for 5th character (S2)+1

(S2)	5
(S2)+1	5

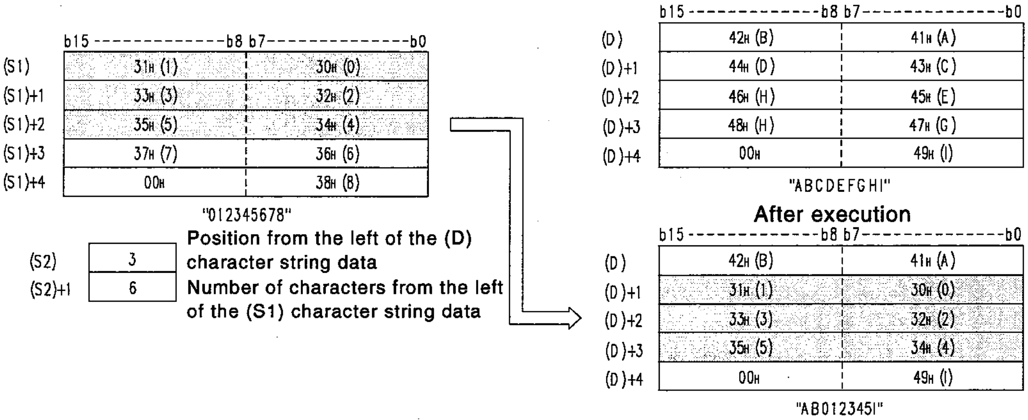
(2) No processing will be conducted if the number of characters designated by (S2)+1 is "0".

(3) If the number of characters designated by (S2)+1 is "-1", stores the data up to the final character designated by (S) starting from the device designated by (D).



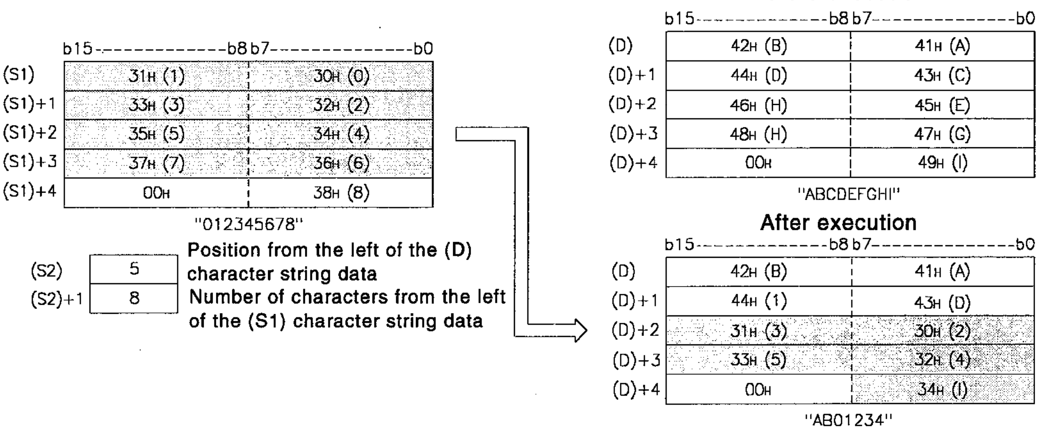
MIDW

(1) Stores the number of characters of data designated by (S2)+1 from the left of the character string data being stored starting from the device number designated by (S1), at a location starting from the position designated by (S2) from the left of the character string data being stored starting from the device number designated by (D).



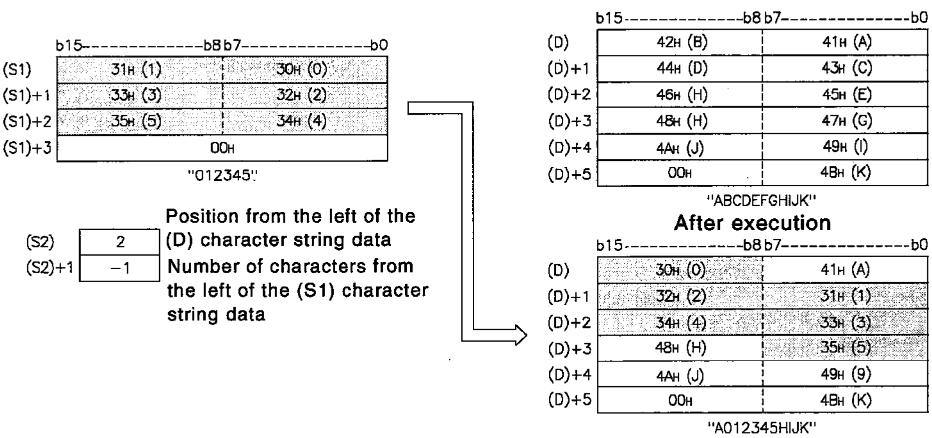
(2) No processing will be conducted if the number of characters designated by (S2)+1 is 0.

(3) If the number of characters designated by (S2)+1 exceeds the final character from the character string data designated by (D), data will be stored up to the final character.



Characters from "35_H" (5) through "37_H" (7) will not be stored.

(4) If the number of characters designated by (S2)+1 is "-1," stores up to the final character data designated by (S1) starting from the device designated by (D).



	b15 - - - b8	b7 - - - b0
(S1)	31H (1)	30H (0)
(S1)+1	33H (3)	32H (2)
(S1)+2	35H (5)	34H (4)
(S1)+3	00H	

"012345"

(S2)	2	Position from the left of the (D) character string data
(S2)+1	−1	Number of characters from the left of the (S1) character string data

Before execution

	b15 - - - b8	b7 - - - b0
(D)	42H (B)	41H (A)
(D)+1	44H (D)	43H (C)
(D)+2	46H (H)	45H (E)
(D)+3	48H (H)	47H (G)
(D)+4	4AH (J)	49H (I)
(D)+5	00H	4BH (K)

"ABCDEFGHIJK"

After execution

	b15 - - - b8	b7 - - - b0
(D)	30H (0)	41H (A)
(D)+1	32H (2)	31H (1)
(D)+2	34H (4)	33H (3)
(D)+3	48H (H)	35H (5)
(D)+4	4AH (J)	49H (9)
(D)+5	00H	4BH (K)

"A012345HIJK"

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

For MIDR instruction

- The value of (S2) exceeds the number of characters designated by (S1). (Error code: 4101)
- The (S2)+1 number of characters from position (D) exceeds the (D) device range. (Error code: 4101)

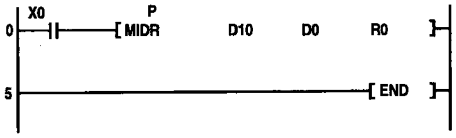
For MIDW instruction

- The (S2) value exceeds the number of characters for (D). (Error code: 4101)
- The (S2)+1 value exceeds the number of characters for (S1). (Error code: 4101)

Program Example

(1) The following program stores the 3rd character through the 6th character from the left of the character string stored starting from D10 at devices starting from D0 when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MIDRP	D10 D0 R0
5	END	

	b15------------b8	b7-------------b0
D10	41H (A)	42H (B)
D11	31H (1)	32H (2)
D12	46H (E)	33H (3)
D13	00H	45H (D)

"BA213ED"

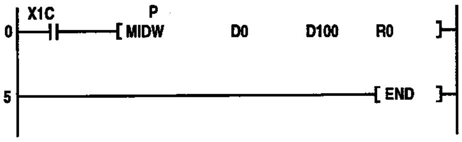
R0	3
R1	4

	b15------------b8	b7-------------b0
D0	31H (1)	32H (2)
D1	46H (E)	33H (3)
D2	00H	

"213E"

(2) The following program stores 4 characters of the character string data being stored in devices starting from D0 at the position starting from the 3rd character from the left of the character string data in devices starting from D100 when X1C goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	MIDWP	D0 D100 R0
5	END	

	b15------------b8	b7-------------b0
D0	31H (1)	32H (2)
D1	45H (E)	46H (F)
D2	33H (3)	30H (0)
D3	00H	

"21FE03"

R0	3
R1	4

Before execution

	b15------------b8	b7-------------b0
D100	53H (S)	55H (U)
D101	59H (Y)	43H (C)
D102	31H (1)	5AH (Z)
D103	42H (B)	30H (0)
D104	00H	

"USCYZ10B"

After execution

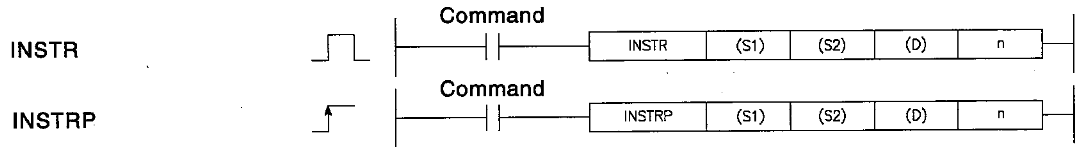
	b15------------b8	b7-------------b0
D100	53H (S)	55H (U)
D101	31H (1)	32H (2)
D102	45H (E)	46H (F)
D103	42H (B)	30H (0)
D104	00H	

"US21FE0B"

7.11.17 Character string search

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant		Other
	Bit	Word		Bit	Word			K, H	$	U
(S1)	—	o		—				—	o	—
(S2)	—	o		—				—	o	—
(D)	o	o		o				—	—	—
n	o	o		o				o	—	—

[Instruction Symbol] [Execution Condition]



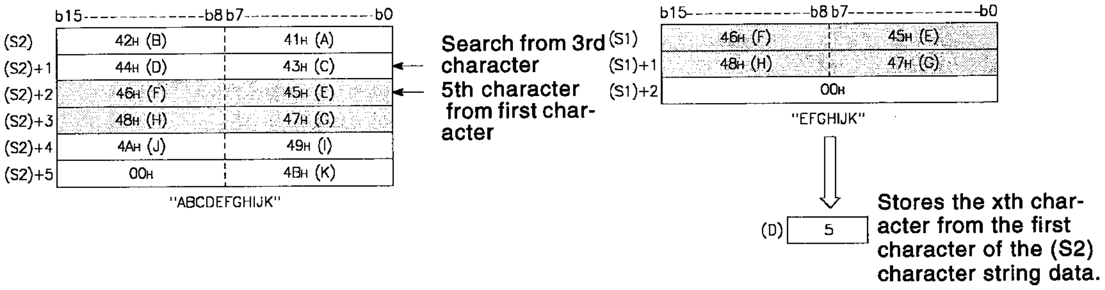
Set Data

Set Data	Meaning	Data Type
(S1)	First device number of devices where search character string is being stored	Character string
(S2)	First device number of devices where character string being searched is being stored	
(D)	First device number of devices that will store search results	BIN 16 bits
n	Position at which search is initiated	

Functions

(1) Searches character string data from the nth character from the left of the character string data being stored in devices starting from that whose device number designated by (S2) for the character string data being stored in devices starting from that whose device number is designated by (S1), and stores the search results at the device designated by (D). The search results store the xth character from the first character of the character string data designated by (S2).

When n=3



Search from 3rd character

5th character from first character

Stores the xth character from the first character of the (S2) character string data.

(2) If there is no matching character string data, stores "0" at (D).

(3) No processing will be conducted if n is a negative value or "0".

Operation Errors

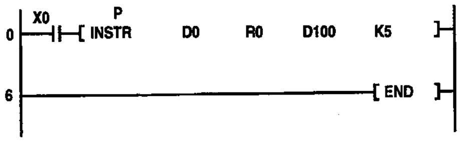
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The value of n exceeds the number of characters for (S2). (Error code: 4100)

Program Example

(1) The following program searches from the 5th character from the left of the character string data stored in devices starting from R0 for the character string data in devices starting from D0, and stores the results at D100 when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	INSTRP	D0 R0 D100 K5
6	END	

	b15 ... b8	b7 ... b0
R0	49_H (I)	43_H (C)
R1	33_H (3)	32_H (2)
R2	32_H (2)	31_H (1)
R3	49_H (I)	43_H (C)
R4	00_H	$4D_H$ (M)

"CI2312CIM"

Not searched because the position from which the search begins is 5

Searches from the 5th character

	b15 ... b8	b7 ... b0
D0	49_H (I)	43_H (C)
D1	33_H (3)	32_H (2)
D2	00_H	

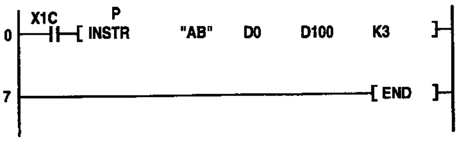
"CI23"

D100 0

Stores "0" because there is no matching character string.

(2) The following program searches from the 3rd character from the left of the character string data being stored in devices starting from D0 for the character string data "AB", and stores the results of the search at D100 when X1C goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	INSTRP	"AB" D0 D100 K3
7	END	

	b15 ... b8	b7 ... b0
D0	32_H (2)	31_H (1)
D1	34_H (4)	33_H (3)
D2	42_H (B)	41_H (A)
D3	36_H (6)	35_H (5)
D4	42_H (B)	41_H (A)
D5	00_H	

"1234AB56AB"

Searches for "AB" from 3rd character

5th character from first characterllowing types of basic instructions are availables.

D100 5

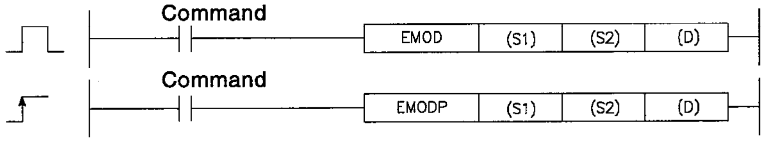
7.11.18 Floating decimal point to BCD

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant		Other
	Bit	Word		Bit	Word			K, H	E	
(S1)	—	o		—	o		—	—	o	—
(S2)	o	o		o	o		o	o	—	—
(D)	—	o		—	—		—	—	—	—

[Instruction Symbol] [Execution Condition]

EMOD

EMODP

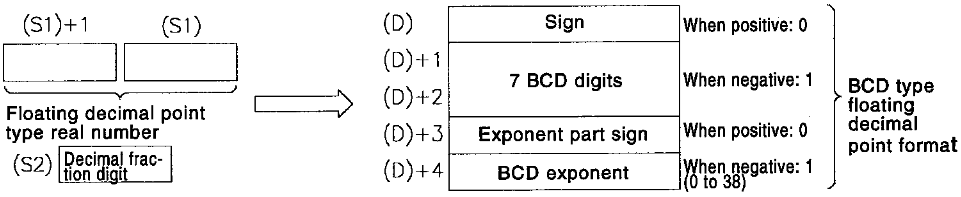


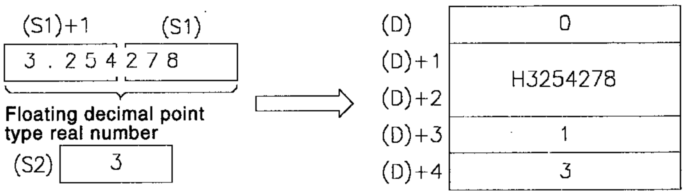
Set Data

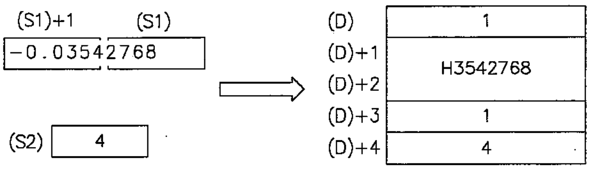
Set Data	Meaning	Data Type
(S1)	Floating decimal point type real number data or first device number of devices where floating decimal point type real number data is being stored	Real number
(S2)	Decimal fraction part digit data or first device number of devices where such data is being stored	BIN 16 bits
(D)	First device number of devices that will store the BCD analyzed data	

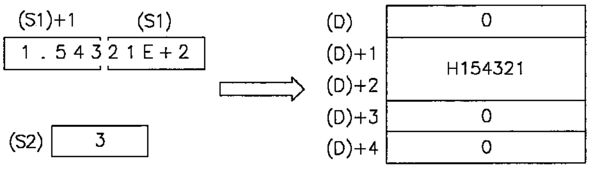
Functions

(1) Breaks down the floating decimal point type real number data being stored at the device designated by (S1) into the BCD type floating decimal point format on the basis of the decimal fraction digits being stored at the device designated by (S2), and stores the result starting from the device designated by (D).

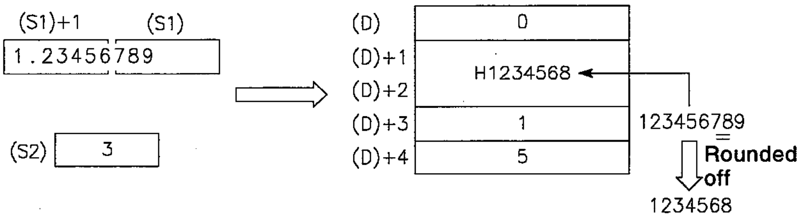








(2) The 8th digit of the significant digits being stored at (D)+1 and (D)+2 is rounded off to make a 7-digit number.



Operation Errors

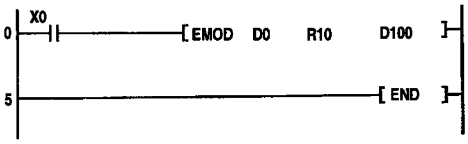
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The decimal fraction digit designated by (S2) is outside the range of from 0 to 7. (Error code: 4100)
- The device range designated by (D) exceeds the range of the relevant device. (Error code: 4101)

Program Example

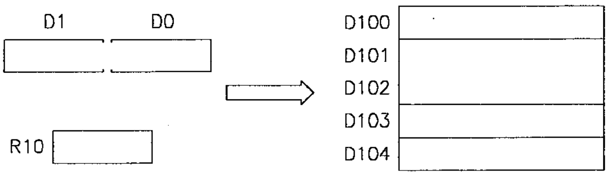
(1) The following program breaks down the floating decimal point type real number data being stored at D0 and D1 on the basis of the decimal fraction corresponding to the value being stored at R10, and stores the results at devices starting from D100 when X0 goes ON.

[Ladder Mode]



[List Mode]

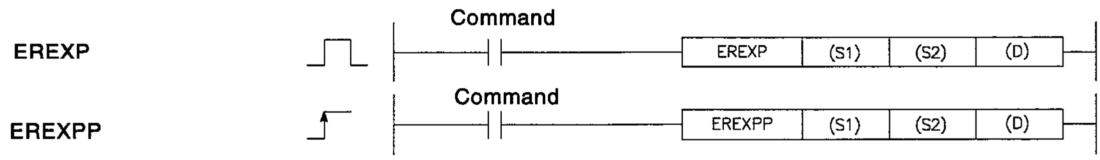
Steps	Instruction	Device
0	LD	X0
1	EMOD	D0 R10 D100
5	END	



7.11.19 From BCD format data to floating decimal point

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
(S1)	—	o		—	—		—	—	—
(S2)	o	o		o	o		o	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

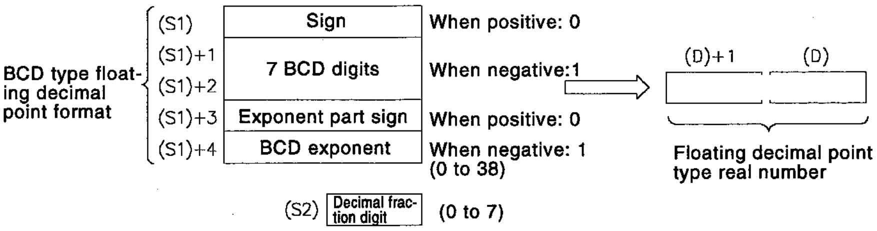


Set Data

Set Data	Meaning	Data Type
(S1)	First device number of devices where BCD type floating decimal point format data is being stored	BIN 16 bits
(S2)	Decimal fraction digit data or the device where decimal fraction digit data is being stored	BIN 16 bits
(D)	Device that will store floating decimal point type real number data	Real number

Functions

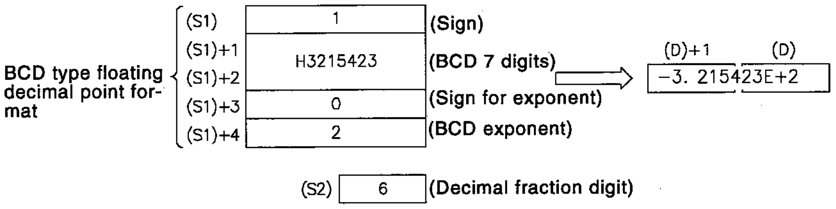
(1) Converts the BCD type floating decimal point format data being stored in devices starting from that whose number is designated by (S1) to floating decimal point type real number data based on the decimal fraction digit being stored at the device designated by (S2), and stores the result at devices starting from that whose device number is designated by (D).



(2) The sign at (S1) and the sign for the exponent part at (S1)+3 is set at 0 for a positive value and at 1 for a negative value.

(3) A value of from 0 to 38 can be set for (S1)+4 BCD exponent.

(4) From 0 to 7 decimal fraction digits can be set.



Operation Errors

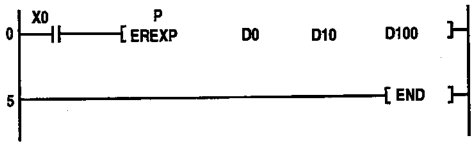
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The format designation made by (S1) is something other than 0 or 1. (Error code: 4100)
- The BCD data designated by (S1)+1 and (S1)+2 exceeds 8 digits. (Error code: 4100)
- The format designation made by (S1)+3 is something other than 0 or 1. (Error code: 4100)
- The exponent data designated by (S1)+4 is outside the range of from 0 to 38. (Error code: 4100)
- The decimal fraction digit designated by (S2) is outside the range of from 0 to 7. (Error code: 4100)

Program Example

(1) The following program converts the BCD type floating decimal point format data being stored in devices starting from D0 to floating decimal point type real number data based on the decimal fraction digit being stored at D10, and stores the result at D100 and D101 when X0 goes ON.

[Ladder Mode]



[List Mode]

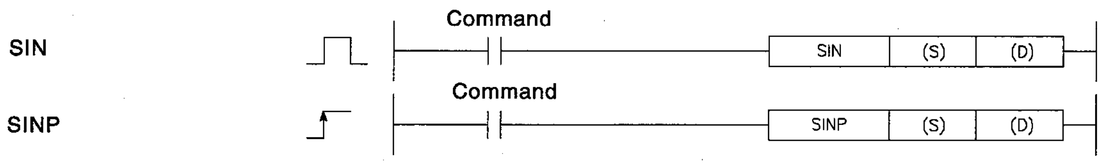
Steps	Instruction	Device
0	LD	X0
1	EREXPP	D0 D10 D100
5	END	

7.12 Special Function Instructions

7.12.1 SIN operation on floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

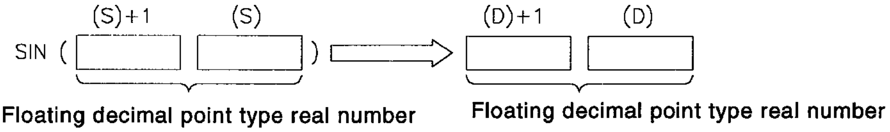


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where angle data on which SIN(sine) operation will be performed is being stored.	Real number
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Calculates SIN (sine) value of angle designated at (S) and stores the operation result in the device number designated at (D).



(2) Angles designated at (S) are set in radian units (degrees x π/180).
For conversion between degrees and radian values, see the RAD and DEG instructions.

Operation Errors

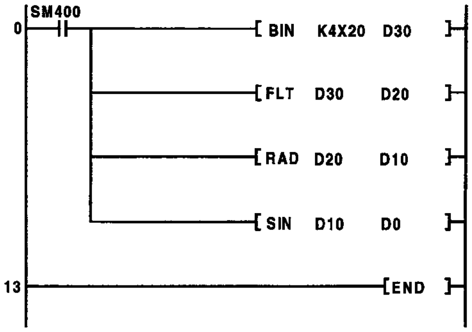
- When the specified device contains -0 (Q4ARCPU)
(Operation error does not occur even if -0 is stored if SM707 is turned on.)

(Error code: 4100)

Program Example

(1) The following program coducts a SIN operation on the angles stored in the four BCD digits from X20 through X2F and stores the results at D0 and D1 as floating decimal point type real numbers.

[Ladder Mode]



Input angles for SIN operation (1)

Convert input angles to floating decimal point type real numbers (2)

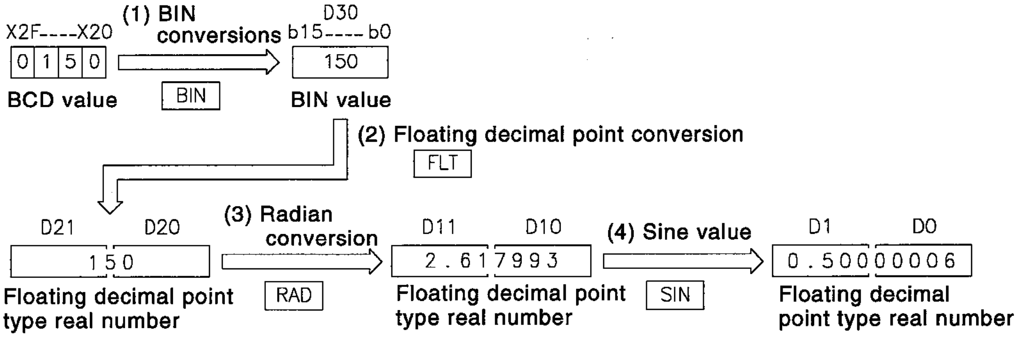
Convert angles to radian values (3)

Perform sine operation on converted radian values (4)

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	BIN	K4X20 D30
4	FLT	D30 D20
7	RAD	D20 D10
10	SIN	D10 D0
13	END	

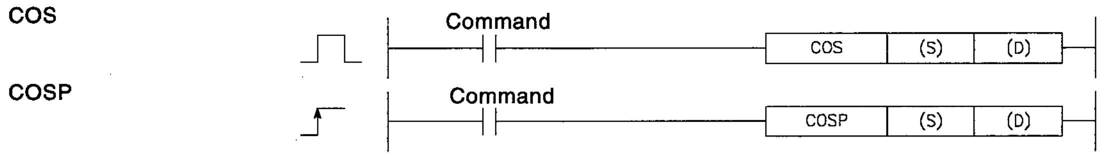
[Operations involved when X20 through X2F designate a value of 150]



7.12.2 COS operation on floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

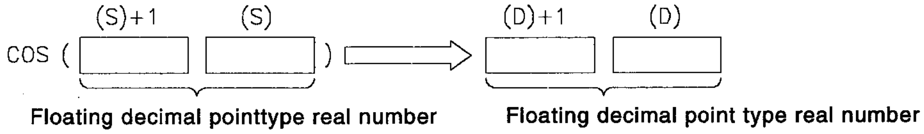


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where angle data on which cosine (COS) operation will be performed is being stored	Real number
(D)	First device number of the devices where the operation result is stored	

Function

(1) Performs cosine (COS) on angle designated by (S) and stores operation result at device number designated by (D).



(2) Angles designated at (S) are set in radian units (degrees x π/180). For conversion between degrees and radian values, see the RAD and DEG instructions.

Operation Errors

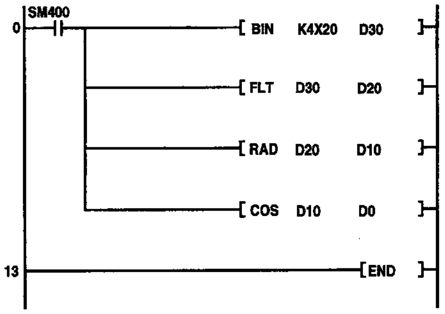
- When the specified device contains -0 (Q4ARCPU)
  (Operation error does not occur even if -0 is stored if SM707 is turned on.)

(Error code: 4100)

Program Example

(1) The following program performs a COS operation on the angle data designated by the four BCD digits from X20 through X2F, and stores results as floating decimal point type real numbers at D0 and D1.

[Ladder Mode]



Input angle data for COS operation (1)

Convert input angles to floating decimal point type real numbers (2)

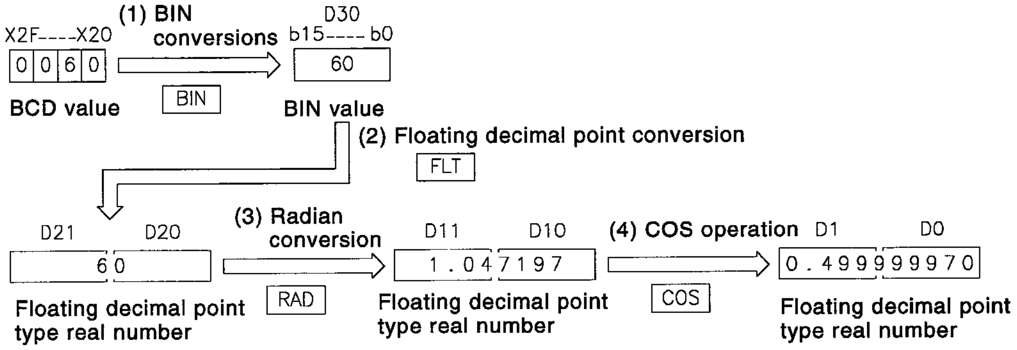
Convert angles to radian values (3)

COS operation based on converted radian value (4)

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	BIN	K4X20 D30
4	FLT	D30 D20
7	RAD	D20 D10
10	COS	D10 D0
13	END	

[Operations involved when X20 through X2F designate a value of 60]



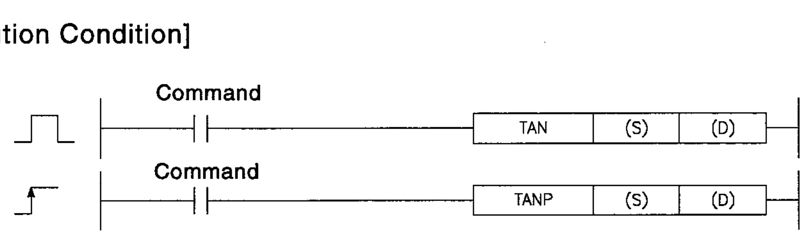
7.12.3 TAN operation on floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

TAN

TANP

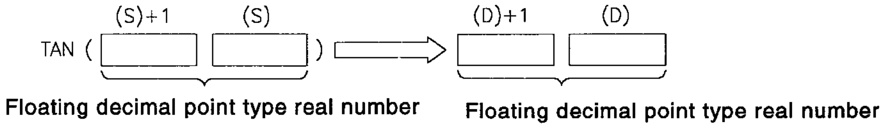


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where angle data on which TAN (tangent) operation is to be performed is being stored	Real number
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Performs tangent (TAN) operation on angle data designated by (S), and stores operation result in device designated by (D).



(2) Angles designated at (S) are set in radian units (degrees x π/180). For conversion between degrees and radian values, see the RAD and DEG instructions.

(3) When angles designated by (S) are π/2 radians, or (3/2) π radians, an operation error will be generated in the calculation of the radian value, so care must be taken to avoid such errors.

Operation Errors

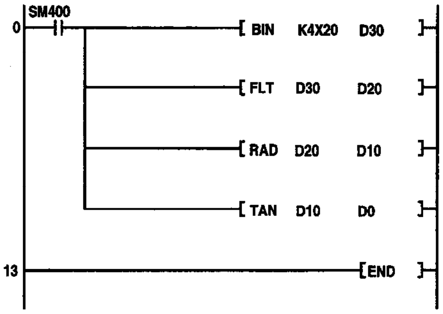
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Operation results are "0", or are outside the range shown below: (Error code: 4100)
  $\pm 2^{-127} \leq$ operation result 1 $< \pm 2^{128}$
- When the specified device contains -0 (Q4ARCPU)
  (Operation error does not occur even if -0 is stored if SM707 is turned on.)
  (Error code: 4100)

Program Example

(1) The following program performs a TAN operation on the angle data set by the 4 BIN digits from X20 through X2F, and stores the results as floating decimal point type real numbers at D0 and D1.

[Ladder Mode]



Input angle data for TAN operation (1)

Convert input angles to floating decimal point type real numbers (2)

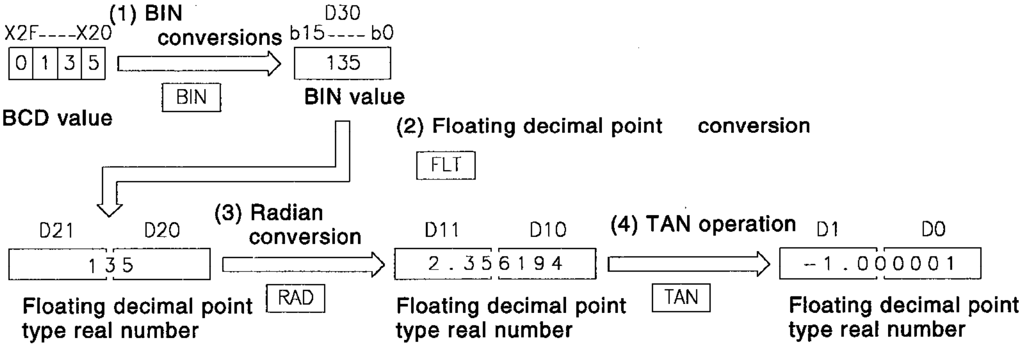
Convert angles to radian values (3)

TAN operation with converted radian value (4)

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	BIN	K4X20 D30
4	FLT	D30 D20
7	RAD	D20 D10
10	TAN	D10 D0
13	END	

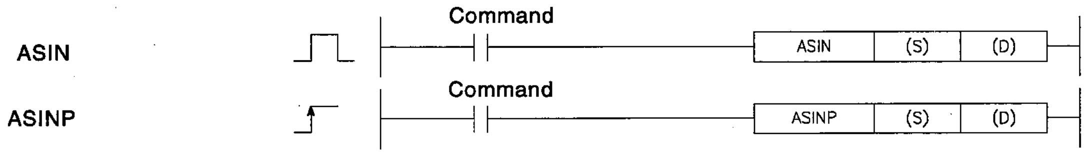
[Operations involved when X20 through X2F designate a value of 135]



7.12.4 SIN^{-1} operation on floating decimal point data

<table>
<tr><th rowspan="3">Set Data</th><th colspan="9">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System, User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th rowspan="2">Constant E</th><th rowspan="2">Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th></tr>
<tr><td>(S)</td><td>—</td><td colspan="2">o</td><td>—</td><td colspan="2">o</td><td>—</td><td>o</td><td>—</td></tr>
<tr><td>(D)</td><td>—</td><td colspan="2">o</td><td>—</td><td colspan="2">o</td><td>—</td><td>—</td><td>—</td></tr>
</table>

[Instruction Symbol] [Execution Condition]

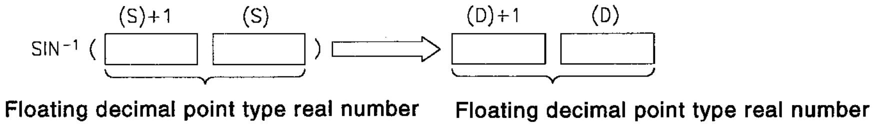


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where SIN value on which SIN^{-1} (inverse sine) operation is to be performed is being stored.	Real number
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Calculates angle from SIN value designated by (S), and stores operation results at word device designated by (D).



(2) The SIN value designated by (S) can be in the range from -1.0 to 1.0.

(3) The angle (operation result) stored at (D) is stored in radian units. For more information on the conversion between radian and angle data, see description of DEG and RAD instructions.

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

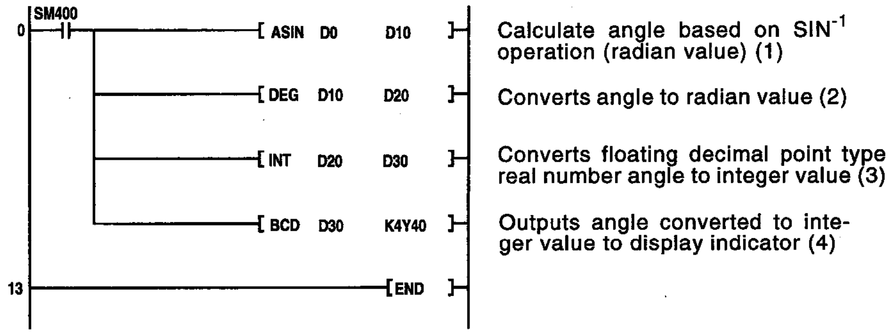
- The value designated by (S) is outside the range of from -1.0 to 1.0. (Error code: 4100)

- When the specified device contains -0 (Q4ARCPU)
  (Operation error does not occur even if -0 is stored if SM707 is turned on.)
  (Error code: 4100)

Program Example

(1) The following program seeks the inverse sine of the floating decimal point real number at D0 and D1, and outputs the angle to the 4 BCD digits at Y40 through Y4F.

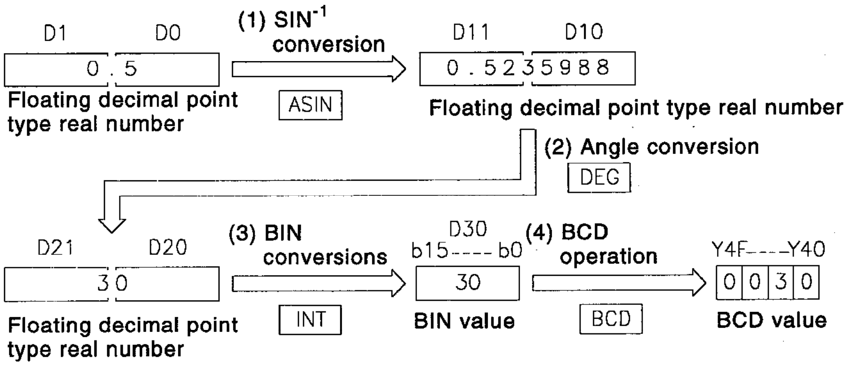
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	ASIN	D0 D10
4	DEG	D10 D20
7	INT	D20 D30
10	BCD	D30 K4Y40
13	END	

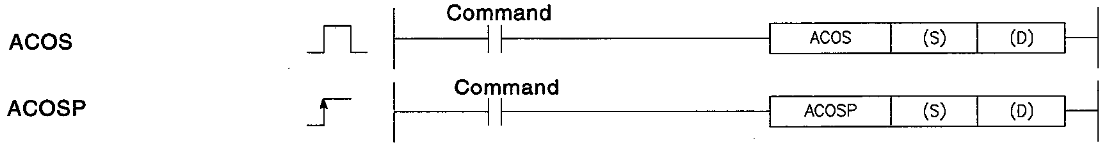
[Operations involved when the D0 and D1 value is 0.5]



7.12.5 COS^{-1} operation on floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

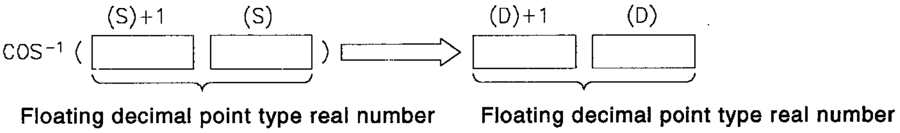


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where COS value on which COS^{-1} (inverse cosine) operation is to be performed is stored	Real number
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Calculates angle from COS value designated by (S), and stores operation result at word device designated by (D).



(2) The COS value designated by (S) can be in the range of from -1.0 to 1.0.

(3) The angle (operation result) stored at (D) is stored in radian units.
For more information on the conversion between radian and angle data, see description of DEG and RAD instructions.

Operation Errors

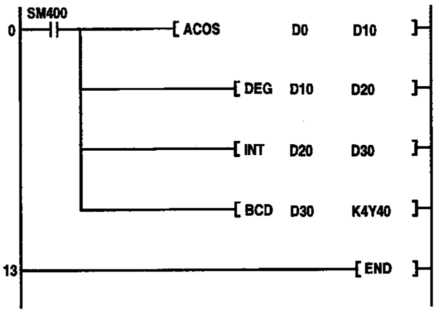
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The SIN value designated by (S) is outside the range of from -1.0 to 1.0. (Error code: 4100)
- When the specified device contains -0 (Q4ARCPU)
  (Operation error does not occur even if -0 is stored if SM707 is turned on.)
  (Error code: 4100)

Program Example

(1) The following program seeks the inverse cosine of the floating decimal point real number at D0 and D1, and outputs the angle to the 4 BCD digits at Y40 through Y4F.

[Ladder Mode]



Calculate angle based on COS^{-1} operation (radian value) (1)

Converts angle to radian value (2)

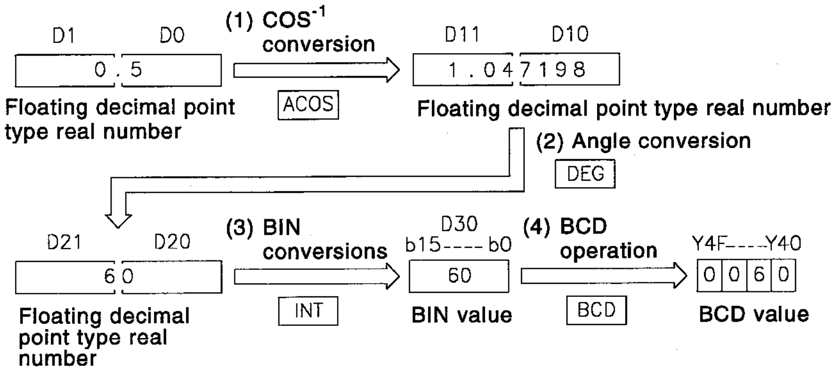
Converts floating decimal point type real number angle to integer value (3)

Outputs angle converted to integer value to display indicator (4)

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	ACOS	DO D10
4	DEG	D10 D20
7	INT	D20 D30
10	BCD	D30 K4Y40
13	END	

[Operations involved when the D0 and D1 value is 0.5]



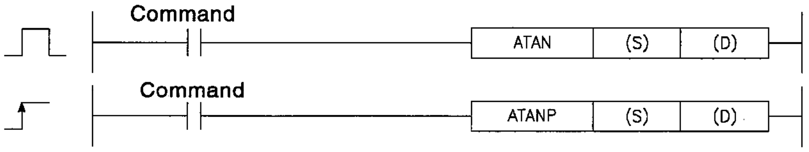
7.12.6 TAN^{-1} operation on floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

ATAN

ATANP

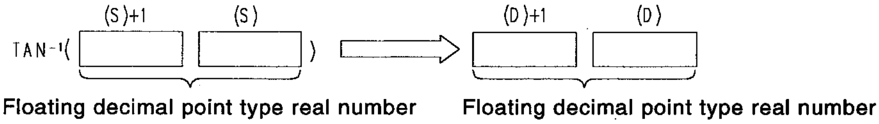


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices storing TAN value on which TAN^{-1} (inverse tangent) value will be performed	Real number
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Calculates angle from TAN value designated by (S), and stores operation results at word device designated by (D).



(2) The angle (operation result) stored at (D) is stored in radian units.
For more information on the conversion between radian and angle data, see description of DEG and RAD instructions.

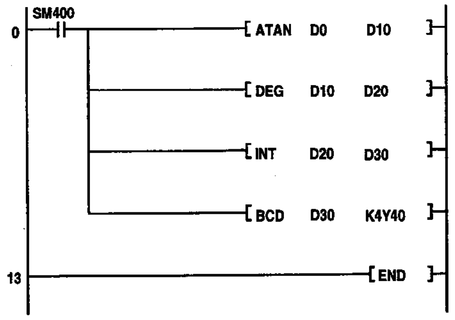
Operation Errors

- When the specified device contains -0 (Q4ARCPU)
(Operation error does not occur even if -0 is stored if SM707 is turned on.)
(Error code: 4100)

Program Example

(1) The following program seeks the inverse tangent of the floating decimal point real number at D0 and D1, and outputs the angle to the 4 BCD digits at Y40 through Y4F.

[Ladder Mode]



Calculation of angle by TAN^{-1} operation (radian value) (1)

Converts angle to radian value (2)

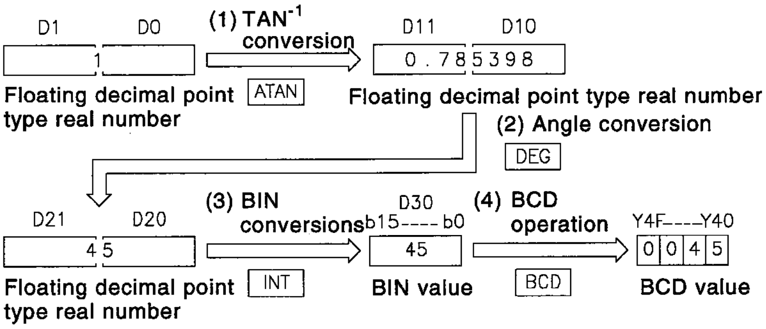
Converts floating decimal point type real number angle to integer value (3)

Outputs angle converted to integer value to display indicator (4)

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	ATAN	D0 D10
4	DEG	D10 D20
7	INT	D20 D30
10	BCD	D30 K4Y40
13	END	

[Operations involved when D0 and D1 value is 1]



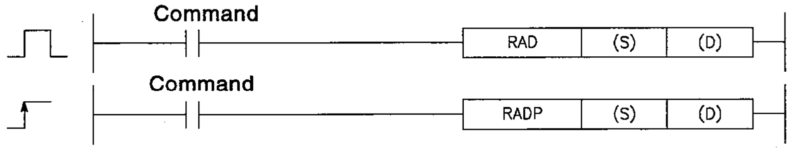
7.12.7 Conversion from floating decimal point angle to radian

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

RAD

RADP

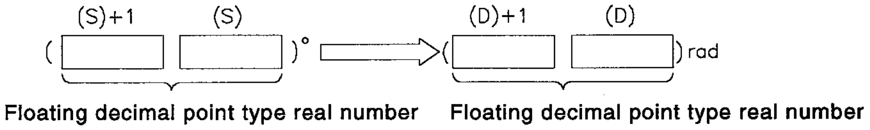


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices storing angle to be converted to radian	Real number
(D)	First device number of devices where radian values will be stored after conversion	

Functions

(1) Converts units of angle size from angle units designated by (S) to radian units, and stores result at device number designated by (D).



(2) Conversion from degree to radian units is performed according to the following equation:

$$\text{Radian unit} = \text{Degree unit} \times \frac{\pi}{180}$$

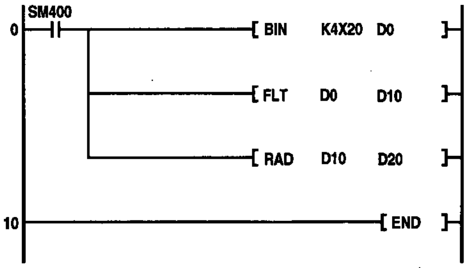
Operation Errors

- When the specified device contains -0 (Q4ARCPU)
  (Operation error does not occur even if -0 is stored if SM707 is turned on.)
  (Error code: 4100)

Program Example

(1) The following program converts the angle set by the 4 BCD digits at X20 through X2F to radians, and stores results as floating decimal point type real number at D20 and D21.

[Ladder Mode]



Input angle to be converted to radian (1)

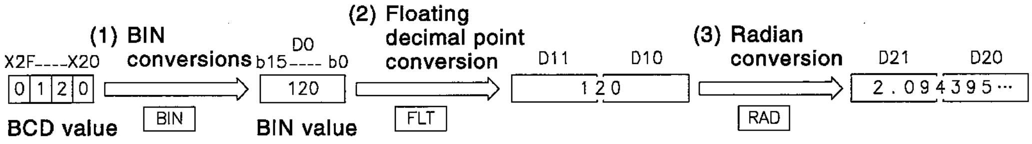
Convert input angles to floating decimal point type real numbers (2)

Convert angle to radian (3)

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	BIN	K4X20
		D0
4	FLT	D0
		D10
7	RAD	D10
		D20
10	END	

[Operations involved when X20 through X2F designate a value of 120]



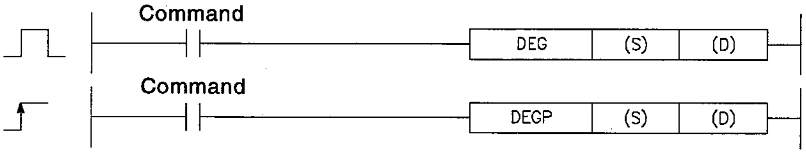
7.12.8 Conversion from floating decimal point radian to angle

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

DEG

DEGP

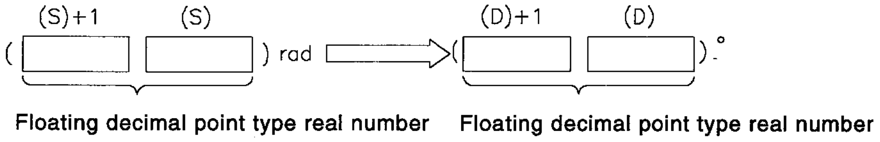


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices storing radian angle to be converted to degrees	Real number
(D)	First device number of devices where degree values will be stored after conversion	

Functions

(1) Converts units of angle size from radian units designated by (S) to angles, and stores result at device number designated by (D).



(2) The conversion from radians to angles is performed according to the following equation:

$$\text{Angle unit} = \text{Radian unit} \times \frac{180}{\pi}$$

Operation Errors

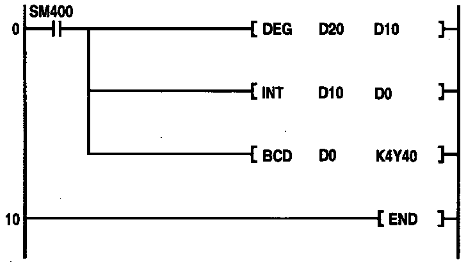
- When the specified device contains -0 (Q4ARCPU)
  (Operation error does not occur even if -0 is stored if SM707 is turned on.)

  (Error code: 4100)

Program Example

(1) The following program converts the floating decimal point type real number radian set by the 4 BCD digits at D20 and D21 to angles, and stores results as BCD value at Y40 through Y4F.

[Ladder Mode]



Converts radian value to angle (1)

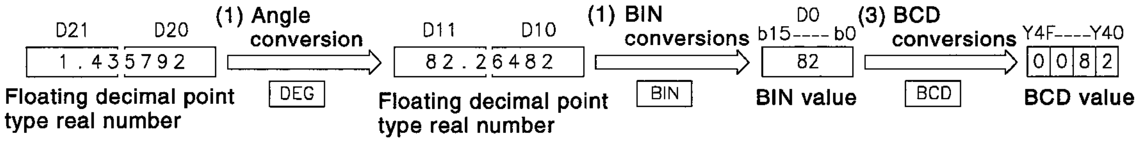
Converts floating decimal point type real number angle to integer value (2)

Outputs converted integer value to display indicator (3)

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DEG	D20 D10
4	INT	D10 D0
7	BCD	D0 K4Y40
10	END	

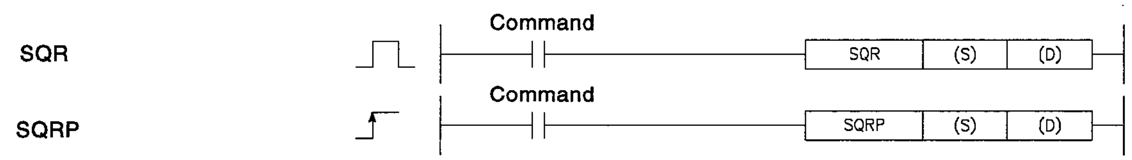
[Operations involved when the values at D20 and D21 are 1.435792]



7.12.9 Square root operations for floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

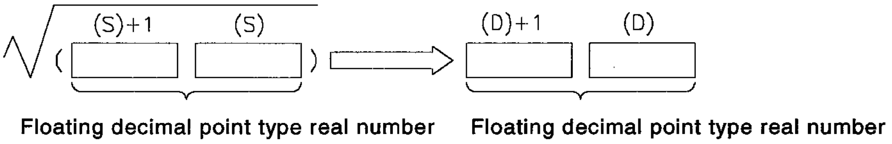


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices storing data on which square root operation will be conducted	Real number
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Calculates square root of value designated at (S), and stores the operation result in the device number designated at (D).



(2) Only positive values can be designated by (S).
(Operation cannot be performed on negative numbers.)

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

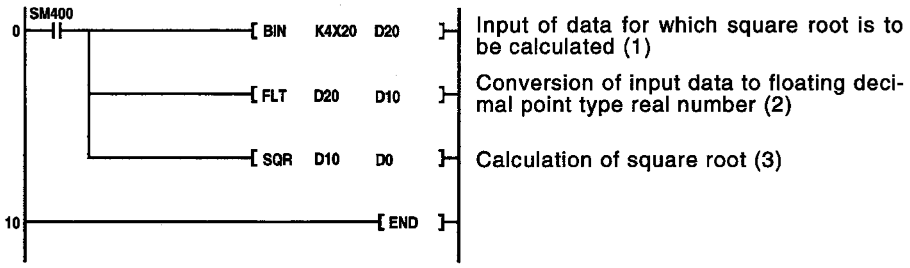
- The value designated by (S) is a negative number.
- When the specified device contains -0 (Q4ARCPU)
  (Operation error does not occur even if -0 is stored if SM707 is turned on.)

(Error code: 4100)

Program Example

(1) The following program seeks the square root of the value set by the 4 BCD digits from X20 through X2F, and stores the result as a floating decimal point type real number at D0 and D1.

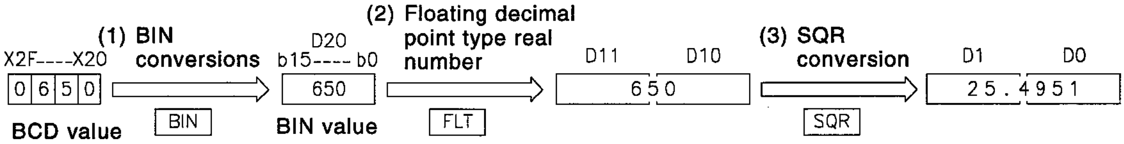
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	BIN	K4X20
		D20
4	FLT	D20
		D10
7	SQR	D10
		D0
10	END	

[Operations involved when value designated by X20 through X2F is 650]



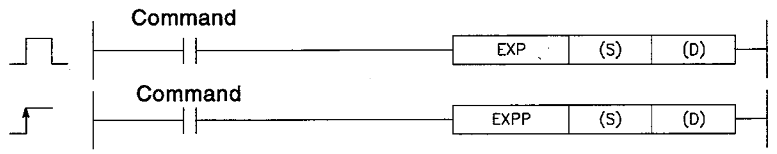
7.12.10 Exponent operations on floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

EXP

EXPP

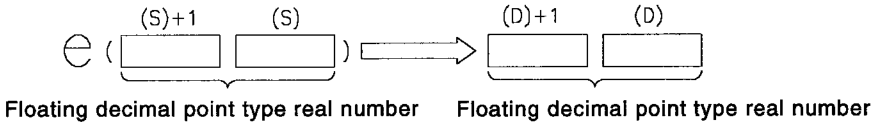


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices storing data on which exponent operation will be performed	Real number
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Calculates exponent for value designated by (S), and stores the results of the operation at the device designated by (D).



(2) Exponent operations are calculated taking the base (e) to be "2.71828".

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

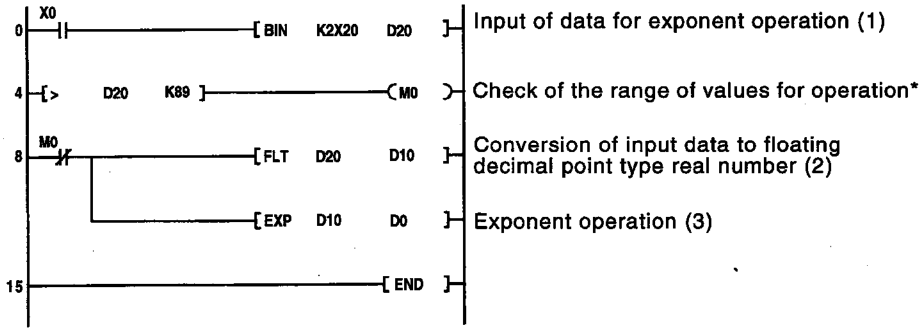
- Operation results are not within the range indicated below:
  $2^{-127} \leq 1$ operation results $1 < 2^{128}$
- When the specified device contains -0 (Q4ARCPU)
  (Operation error does not occur even if -0 is stored if SM707 is turned on.)

(Error code: 4100)

Program Example

(1) The following program performs an exponent operation on the value set by the 2 BCD digits at X20 through X27, and stores the results as a floating decimal point real number at D0 and D1.

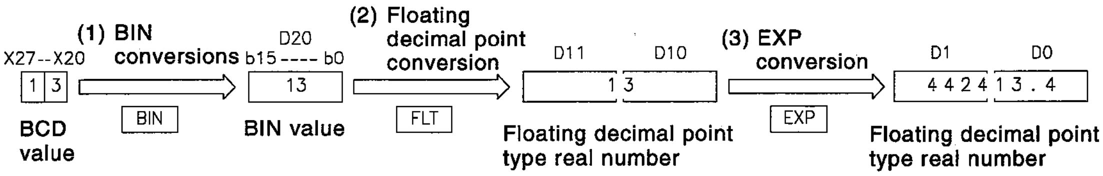
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	BIN	K2X20 D20
4	LD>	D20 K89
7	OUT	M0
8	LDI	M0
9	FLT	D20 D10
12	EXP	D10 D0
15	END	

[Operations involved when the value designated by X20 through X27 is 13]



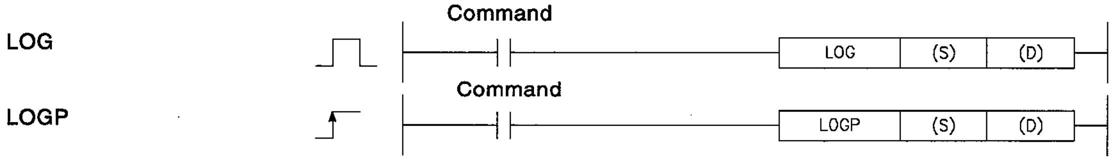
POINT

*: The operation result will be under 2^{129} if the BCD value of X20 through X27 is less than 89, from the calculation $\log_e 2^{129}$ = 89.4. Because setting a value of over 90 will return an operation error, turn M0 ON if a value of over 90 has been set to avoid the error.

7.12.11 Natural logarithm operations on floating decimal point data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant E	Other
	Bit	Word		Bit	Word				
(S)	—	o		—	o		—	o	—
(D)	—	o		—	o		—	—	—

[Instruction Symbol] [Execution Condition]

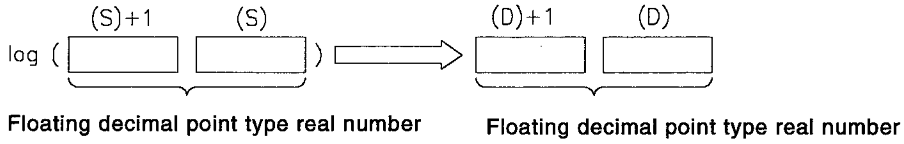


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices storing data for natural logarithm operation	Real number
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Calculates natural logarithm of value designated by (S) taking (e) as base, and stores operation results at device designated by (D).



(2) Only positive values can be designated by (S). (Operation cannot be performed on negative numbers.)

Operation Errors

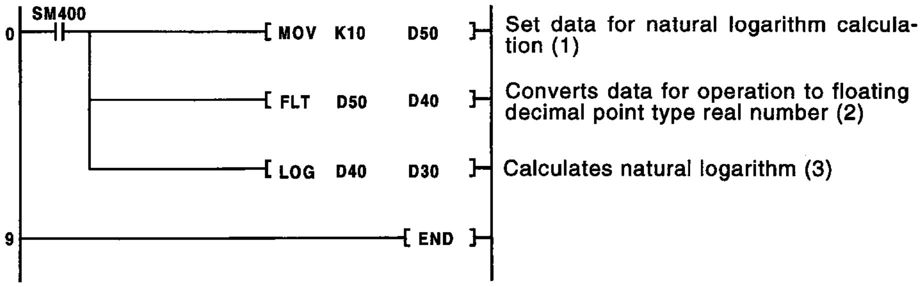
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The value designated by (S) is negative. (Error code: 4100)
- The value designated by (S) is 0 or does not conform to the following: $2^{-127} \leq$ (operation result) $< 2^{128}$ (Error code: 4100)
- When the specified device contains -0 (Q4ARCPU)
(Operation error does not occur even if -0 is stored if SM707 is turned on.)
(Error code: 4100)

Program Example

(1) The following program seeks the natural logarithm of the value "10" set by D50, and stores the result at D30 and D31.

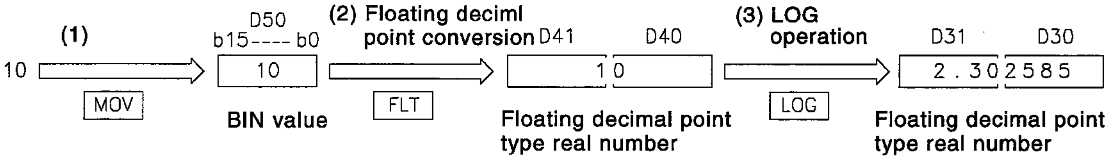
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	MOV	K10
		D50
3	FLT	D50
		D40
6	LOG	D40
		D30
9	END	

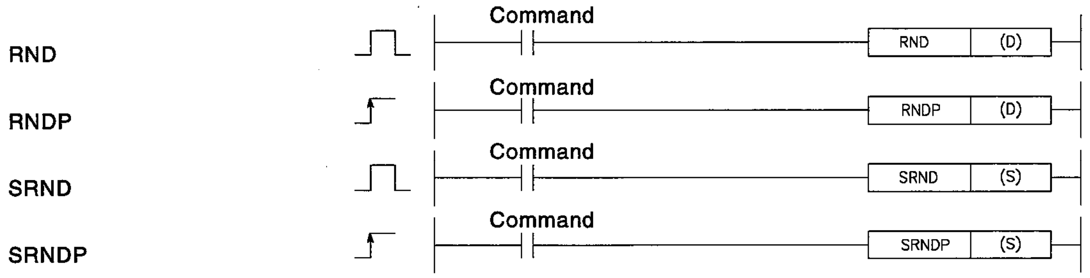
[Operation]



7.12.12 Random number generation and series updates

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)/(D)	o							—	—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(D)	First device number of devices to store random number	BIN 16 bits
(S)	Random number serial data or first device number of devices where such data is being stored	

Functions

RND

Generates random number of from 0 to 32767, and stores at device designated by (D).

SRND

Updates random number series according to the 16-bit BIN data being stored in device designated by (S).

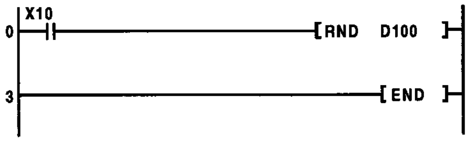
Operation Errors

There are no operation errors associated with the RND(P) or SRND(P) instructions.

Program Example

(1) The following program stores random number at D100 when X10 is ON.

[Ladder Mode]

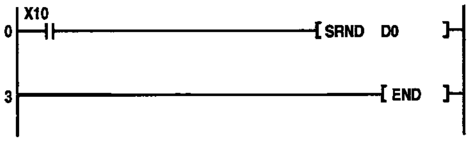


[List Mode]

Steps	Instruction	Device
0	LD	X10
1	RND	D100
3	END	

(2) The following program updates a random number series according to the contents of D0 when X10 goes ON.

[Ladder Mode]

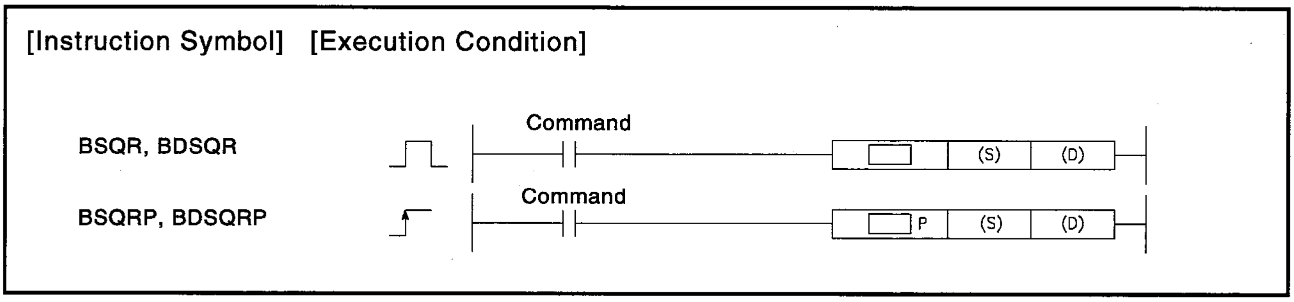


[List Mode]

Steps	Instruction	Device
0	LD	X10
1	SRND	D0
3	END	

7.12.13 BCD 4-digit and 8-digit square roots

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o							o	—
(D)	o							—	—



Set Data

Set Data	Meaning	Data Type
(S)	Data on which square root calculation will be performed, or number of device where such data is being stored	BCD 4-digit/8-digit
(D)	First device number of devices that will store square root calculation results	BCD 4 digits

Functions

BSQR

(1) Calculates square root of value designated at (S), and stores the operation result in the device number designated at (D).

$$\sqrt{(S)} = \overset{(D)}{\boxed{\text{Integer part}}} \,.\, \overset{(D)+1}{\boxed{\text{Decimal fraction part}}}$$

(2) Values that can be designated at (S) are BCD values with a maximum of four digits (from 0 to 9999).

(3) The operation results of (D) and (D)+1 are stored as their respective BCD values of between 0 and 9999.

(4) Operation results are consolidated as a value to the fifth decimal point. For this reason, the fourth decimal place has an error of +/-1.

BDSQR

(1) Performs square root operation on value designated by (S) and (S)+1, and the stores operation result at the device designated by (D).

$$\sqrt{(\underbrace{\boxed{(S)+1}\ \boxed{(S)}}_{\text{2-word data}})} = \overset{(D)}{\boxed{\text{Integer part}}} \,.\, \overset{(D)+1}{\boxed{\text{Decimal fraction part}}}$$

(2) A BCD value of a maximum of 8 digits (0 to 99999999) can be designated by (S) and (S)+1.

(3) The operation results of (D) and (D)+1 are stored as their respective BCD values of between 0 and 9999.

(4) Operation results are consolidated as a value to the fifth decimal point.
For this reason, the fourth decimal place has an error of ±1.

Operation Errors

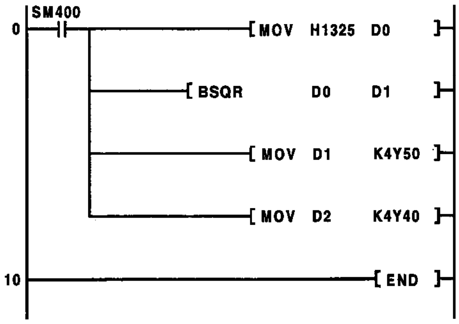
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data designated by (S) is not a BCD value. (Error code: 4100)

Program Example

(1) The following program calculates the square root of BCD value 1325 and outputs the integer part to the 4 BCD digits from Y50 through Y5F, and the decimal fraction part to the 4 BCD digits from Y40 through Y4F.

[Ladder Mode]



Setting of data for which square root is to be calculated (1)

Square root calculation (2)

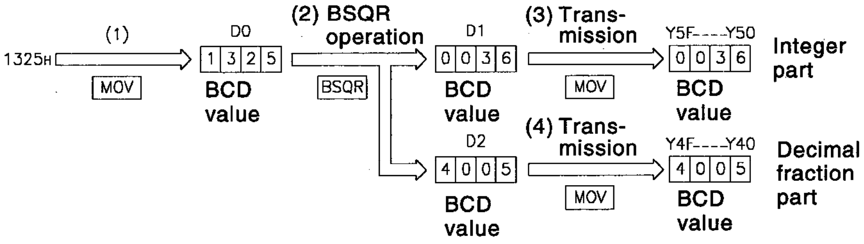
The integer part of the operation result is output to display device (3)

The decimal fraction part of the operation result is output to display device (4)

[List Mode]

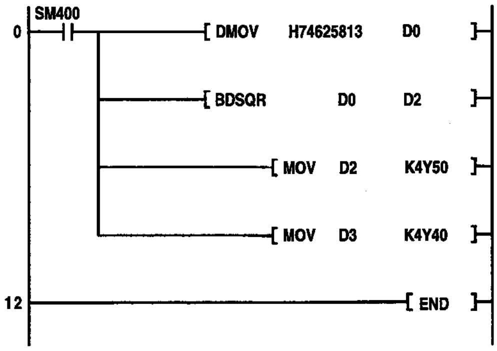
Steps	Instruction	Device
0	LD	SM400
1	MOV	H1325 D0
3	BSQR	D0 D1
6	MOV	D1 K4Y50
8	MOV	D2 K4Y40
10	END	

[Operation]



(2) The following program calculates the square root of BCD value 74625813 and outputs the integer part of the result to the 4 BCD digits at Y50 through Y5F, and the decimal fraction part to the 4 BCD digits from Y40 through Y4F.

[Ladder Mode]



Setting data for which square root will be calculated (BCD value) (1)

Square root calculation (2)

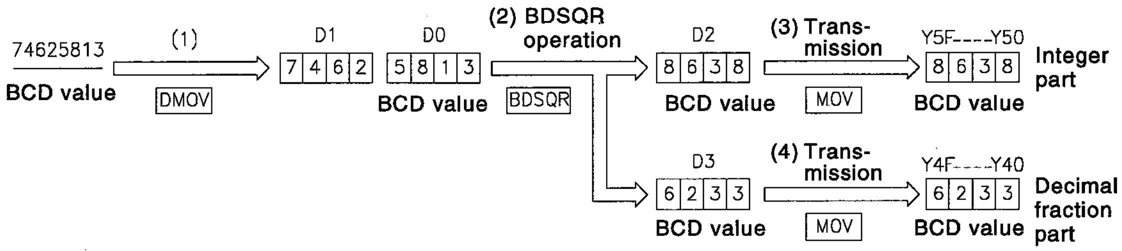
The integer of the operation result is output to display device (3)

The decimal fraction part of the operation result is output to display device (4)

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	DMOV	H74625813 D0
5	BDSQR	D0 D2
8	MOV	D2 K4Y50
10	MOV	D3 K4Y40
12	END	

[Operation]



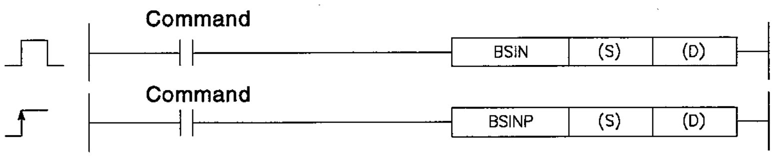
7.12.14 BCD type SIN operation

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		o					—
(D)	—	o		—					—

[Instruction Symbol] [Execution Condition]

BSIN

BSINP



Set Data

Set Data	Meaning	Data Type
(S)	Data on which SIN (sine) operation will be performed, or number of device where such data is being stored	BCD 4 digits
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Calculates the SIN (sine) value of the designated value (angle), and stores the sign of the operation result in the device designated at (D), and the operation result in the devices designated at (D)+1 and (D)+2.

$$\mathrm{SIN(S)} = \overset{(D)}{\boxed{\text{Sign}}}\ \overset{(D)+1}{\boxed{\text{Integer part}}}\ .\ \overset{(D)+2}{\boxed{\text{Decimal fraction part}}}$$

(2) The value designated at (S) is a BCD value which can be between 0 and 360 degrees (in units of degrees).

(3) The sign for the operation result stored in (D) will be "0" if the result is a positive value, and "1" if the result is a negative value.

(4) The operation results stored in (D)+1 and (D)+2 are BCD values between -1.000 and 1.000.

(5) Operation results are rounded off from the fifth decimal place.

Operation Errors

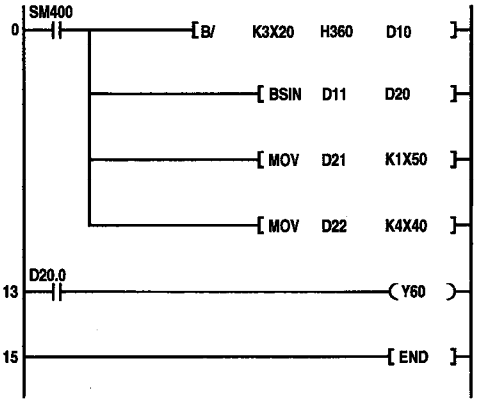
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data designated by (S) is not a BCD value. (Error code: 4100)
- The data designated by (S) is not in the range of from 0 to 360. (Error code: 4100)

Program Example

(1) The program example below calculates the SIN of 3-digit BCD data designated by X20 through X2B, and outputs a 1-digit BCD part before the decimal point to Y50 through Y53, and a 4-digit BCD fraction part to Y40 through Y4F.
Y60 is turned ON if the results of the operation are negative.
(If a value has been set at X20 through X2F that is greater than 360, it will be adjusted to be in the range from 0 to 360.)

[Ladder Mode]



The angle input will be processed so as to be within the 360 degree range.(1)

SIN value operation (2)

The integer of the operation result is output to display device (3)

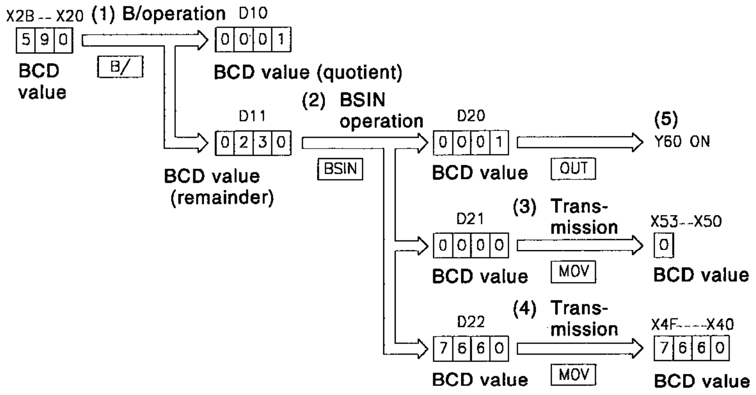
The decimal fraction part of the operation result is output to display device (4)

Sign of operation result is output at ON or OFF (5)

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	B/	K3X20 H360 D10
5	BSIN	D11 D20
8	MOV	D21 K1X50
11	MOV	D22 K4X40
13	LD	D20.0
14	OUT	Y60
15	END	

[Operation]



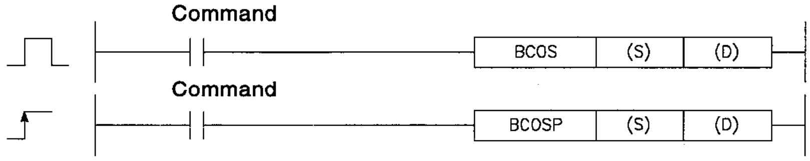
7.12.15 BCD type COS operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constan K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		o					—
(D)	—	o		—					—

[Instruction Symbol] [Execution Condition]

BCOS

BCOSP



Set Data

Set Data	Meaning	Data Type
(S)	Data on which COS (cosine) operation will be performed, or first device number of devices storing such data	BCD 4 digits
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Calculates COS (cosine) value of value (angle) designated by (S), then stores the sign for the operation result in the word device designated by (D), and the operation result in the word device designated by (D)+1 and (D)+2.

$$\mathrm{COS(S)} = \overset{\text{(D)}}{\boxed{\text{Sign}}}\ \overset{\text{(D)+1}}{\boxed{\text{Integer part}}}\ .\ \overset{\text{(D)+2}}{\boxed{\text{Decimal fraction part}}}$$

(2) The value designated at (S) is a BCD value which can be between 0 and 360 degrees (in units of degrees).

(3) The sign for the operation result stored in (D) will be "0" if the result is a positive value, and "1" if the result is a negative value.

(4) The operation results stored in (D)+1 and (D)+2 are BCD values between -1.000 and 1.000.

(5) Operation results are rounded off from the fifth decimal place.

Operation Errors

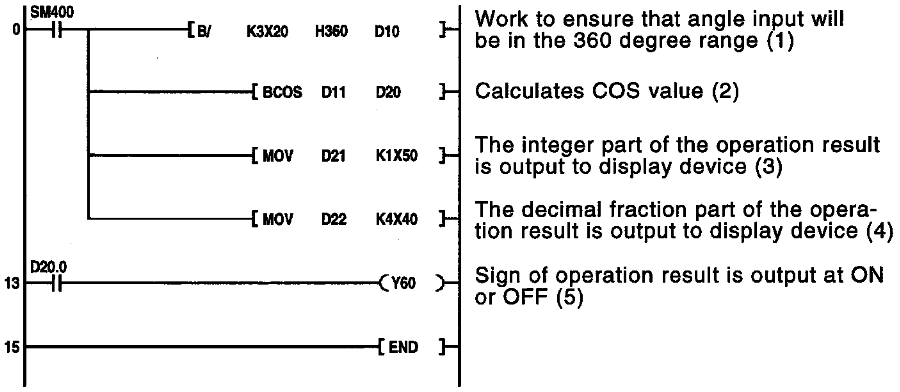
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data designated by (S) is not a BCD value. (Error code: 4100)
- The data designated by (S) is not in the range of from 0 to 360. (Error code: 4100)

Program Example

(1) The following program calculates the cosine of the data designated by the 3 BCD digits from X20 through X2B and outputs the integer part of the result to 1 BCD digit from Y50 through Y53, and the decimal fraction part of the result to the 4 BCD digits from Y40 through Y4F.
Y60 is turned ON if the results of the operation are negative.

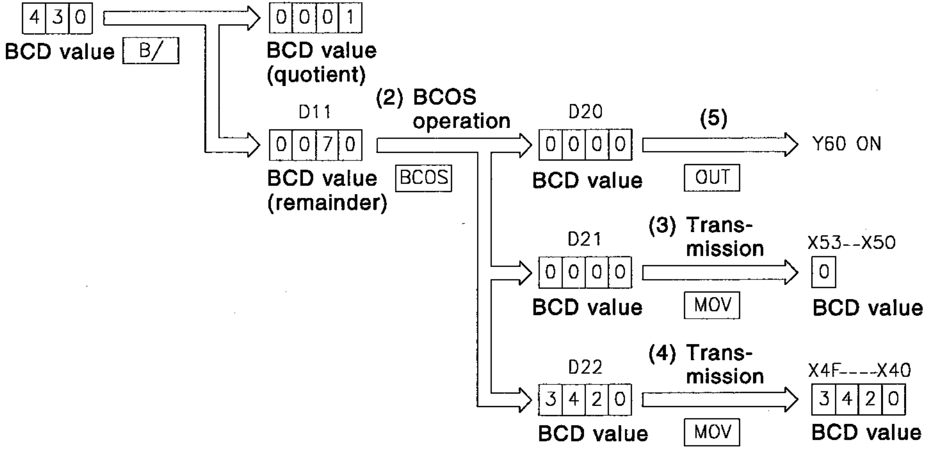
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	B/	K3X20 H360 D10
5	BCOS	D11 D20
8	MOV	D21 K1X50
11	MOV	D22 K4X40
13	LD	D20.0
14	OUT	Y60
15	END	

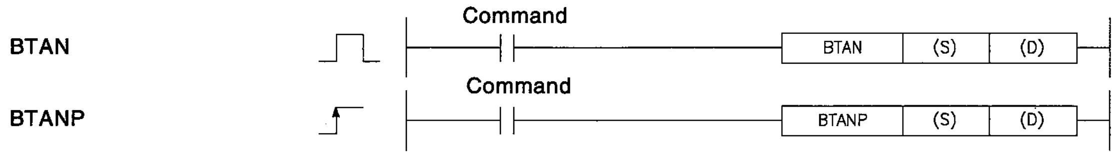
[Operation]



7.12.16 BCD type TAN operation

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o	o		o					—
(D)	—	o		—					—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	Data on which TAN (tangent) operation is to be performed, or first device number of devices where such data is being stored	BCD 4 digits
(D)	First device number of the devices where the operation result is stored.	

Functions

(1) Calculates TAN (tangent) value for value (angle) designated by (S), and stores the sign for the operation result in the word device designated by (D), and the operation result in the word device designated by (D)+1 and (D)+2.

$$\text{TAN(S)} = \overset{\text{(D)}}{\boxed{\text{Sign}}}\ \overset{\text{(D)+1}}{\boxed{\text{Integer part}}}\ .\ \overset{\text{(D)+2}}{\boxed{\text{Decimal fraction part}}}$$

(2) The value designated at (S) is a BCD value which can be between 0 and 360 degrees (in units of degrees).

(3) The sign for the operation result stored in (D) will be "0" if the result is a positive value, and "1" if the result is a negative value.

(4) The operation results stored at (D)+1 and (D)+2 are BCD values within the range of from -57.2900 and 57.2900.

(5) Operation results are rounded off from the fifth decimal place.

Operation Errors

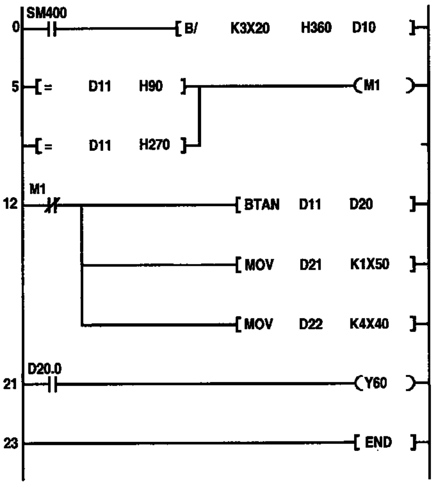
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data designated by (S) is not a BCD value. (Error code: 4100)
- The data designated by (S) is not in the range of from 0 to 360. (Error code: 4100)
- The data designated by (S) is 90° or 270°. (Error code: 4100)

Program Example

(1) The following program calculates the tangent of the data stored at the 3 BCD digits from X20 through X2B, and stores the integer part of the results at the 4 BCD digits from Y50 through Y53, and the decimal fraction part at the 4 BCD digits from Y40 through Y4F.
Y60 is turned ON if the results of the operation are negative.

[Ladder Mode]



Work to ensure that angle input will be in the 360 degree range (1)

Use M1 as interlock to ensure that operation will not be executed if angle input is 90° or 270°

TAN value operation (2)

The integer part of the operation result is output to the display device (3)

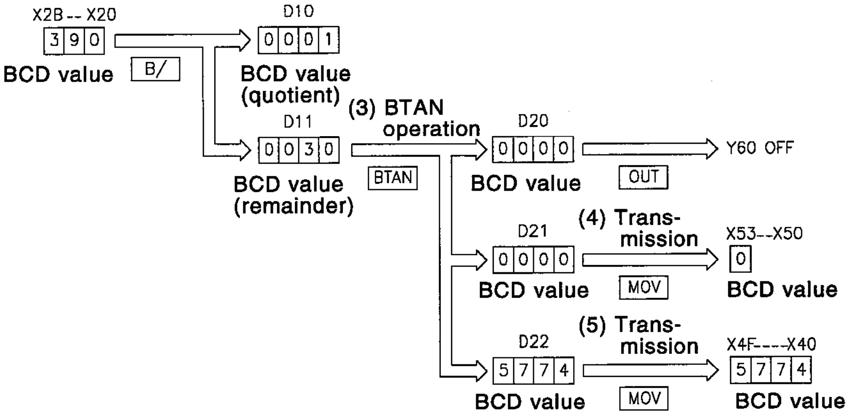
The decimal fraction part of the operation result is output to the display device (4)

Sign of operation result is output at ON or OFF (5)

[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	B/	K3X20 H360 D10
5	LD=	D11 H90
8	OR=	D11 H270
11	OUT	M1
12	LDI	M1
13	BTAN	D11 D20
16	MOV	D21 K1X50
19	MOV	D22 K4Y40
21	LD	D20.0
22	OUT	Y60
23	END	

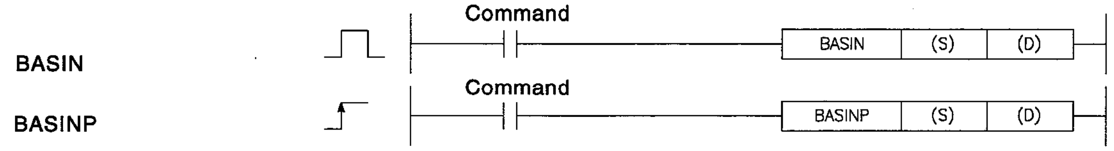
[Operations involved when X20 through X2B designate a value of 390]



7.12.17 BCD type SIN^{-1} operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				—	—
(D)	o	o		o				—	—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	Data on which inverse sine (SIN^{-1}) operation will be performed, or device number where such data is being stored	BCD 4 digits
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Performs inverse sine (SIN^{-1}) operation on value designated by (S) and stores operation results (angles) at device designated by (D).

$$\mathrm{SIN}^{-1} = (\overset{(S)}{\boxed{\text{Sign}}}\ \overset{(S)+1}{\boxed{\text{Integer part}}}\ .\ \overset{(S)+2}{\boxed{\text{Decimal fraction part}}}\ ) = (D)$$

(2) A sign for the operation data is set at (S).
If the operation data is a positive value, this is set at "0", and if it is a negative value, it is set at "1".

(3) The part before the decimal point and fraction part are stored at (S)+1 and (S)+2 respectively, as BCD values.
(Settings can be between 0 and 1.0000.)

(4) Operation results stored at (D) are BCD values between 0 and 90 degrees, and 270 and 360 degrees (degree units).

(5) Calculation results are a value from which the decimal fraction part has been rounded.

Operation Errors

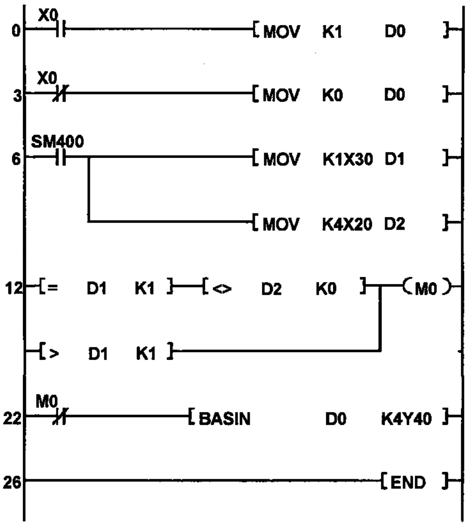
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data designated by (S) is not a BCD value. (Error code: 4100)
- Data designated by (S) is not in the range of from -1.000 to 1.000 (Error code: 4100)

Program Example

(1) The following program performs a SIN^{-1} operation on the sign (positive when X0 is OFF, and negative when X0 is ON), the BCD 1-digit integer part from X30 through X33 and the BCD 4-digit decimal fraction part from X20 through X2F, and outputs the calculated angle in 4 BCD digits from Y40 through Y4F.

[Ladder Mode]



Sets sign for SIN value (1)

Sets integer part of SIN value (2)

Sets decimal fraction part of SIN value (3)

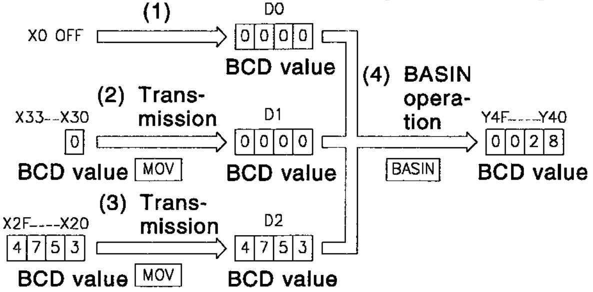
Performs range check of set SIN value

Performs SIN^{-1} operation, and outputs operation result to Y40 through Y4F (4)

[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MOV	K1
		D0
3	LDI	X0
4	MOV	K0
		D0
6	LD	SM400
7	MOV	K1X30
		D1
10	MOV	K4X20
		D2
12	LD=	D1
		K1
15	AND<>	D2
		K0
18	OR>	D1
		K1
21	OUT	M0
22	LDI	M0
23	BASIN	D0
		K4Y40
26	END	

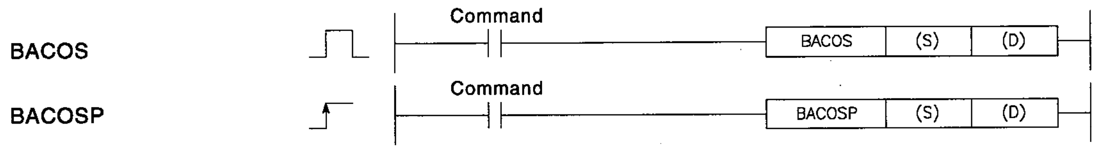
[Operations involved when X20 through X33 designates value of 0.4753]



7.12.18 BCD type COS^{-1} operation

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				—	—
(D)	o	o		o				—	—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	Data on which inverse cosine (COS^{-1}) operation will be performed, or first device number of devices where such data is being stored	BCD 4 digits
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Performs COS^{-1} (inverse cosine) operation on value designated by (S), and stores operation results at device designated by (D).

$$\mathrm{COS}^{-1} = (\overset{(S)}{\boxed{\text{Sign}}}\ \overset{(S)+1}{\boxed{\text{Integer part}}}\ .\ \overset{(S)+2}{\boxed{\text{Decimal fraction part}}}) = (D)$$

(2) A sign for the operation data is set at (S)
If the operation data is a positive value, this is set at "0", and if it is a negative value, it is set at "1".

(3) The part before the decimal point and fraction part are stored at (S)+1 and (S)+2 respectively, as BCD values.
(Settings can be between 0 and 1.0000.)

(4) The operation results stored at (D) will be a BCD value in the range of between 0 and 180° (degrees).

(5) Calculation results are a value from which the decimal fraction part has been rounded.

Operation Errors

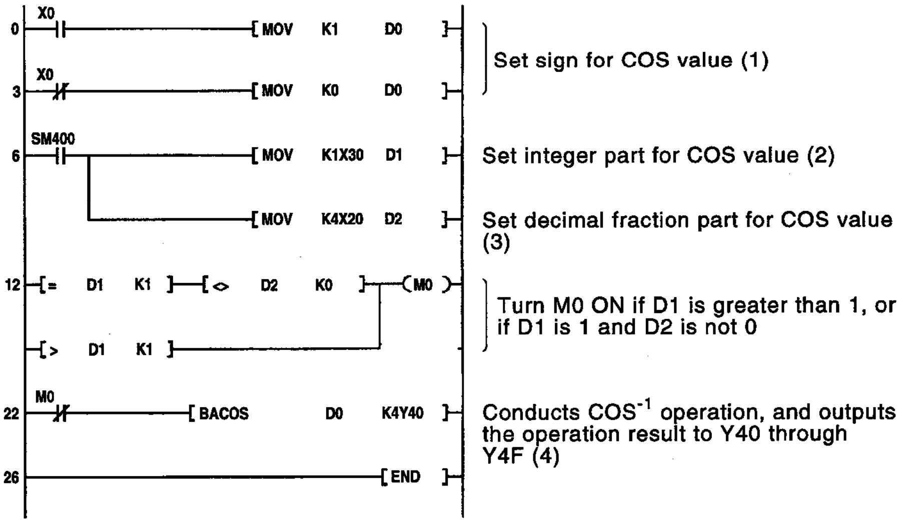
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The operation data designated by (S) is not a BCD value. (Error code: 4100)
- The operation data designated by (S) is not in the range of from -1.000 to 1.000. (Error code: 4100)

Program Example

(1) The following program performs a COS^{-1} operation on the sign (positive when X0 is OFF, and negative when X0 is ON), the BCD 1-digit integer part from X30 through X33 and the BCD 4-digit decimal fraction part from X20 through X2F, and outputs the calculated angle in 4 BCD digits from Y40 through Y4F).

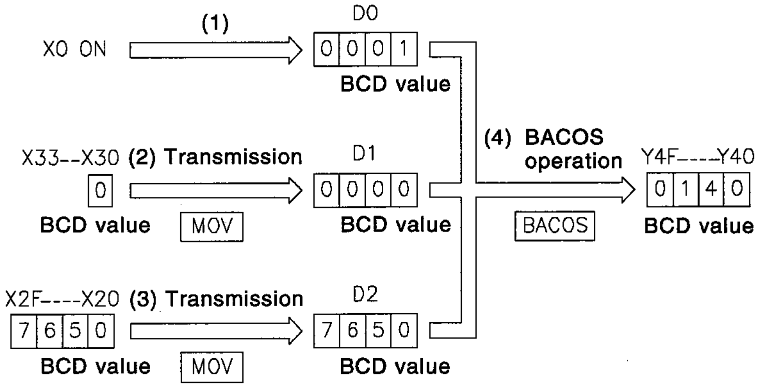
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MOV	K1 D0
3	LDI	X0
4	MOV	K0 D0
6	LD	SM400
7	MOV	K1X30 D1
10	MOV	K4X20 D2
12	LD=	D1 K1
15	AND<>	D2 K0
18	OR>	D1 K1
21	OUT	M0
22	LDI	M0
23	BACOS	D0 K4Y40
26	END	

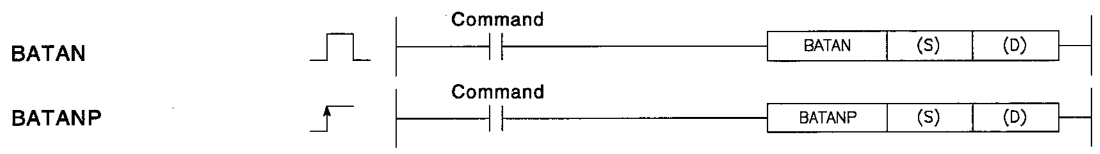
[Operations involved if X0 and X20 through X33 designate a value of -0.7650]



7.12.19 BCD type TAN^{-1} operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				—	—
(D)	o	o		o				—	—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(S)	Data on which TAN^{-1} (inverse tangent) operation is to be performed, or first device number of devices storing such data	BCD 4 digits
(D)	First device number of the devices where the operation result is stored	

Functions

(1) Performs TAN^{-1} (inverse tangent) on value designated by (S) and stores operation results (angles) at device designated by (D).

$$\mathrm{TAN}^{-1} = (\overset{(S)}{\boxed{\text{Sign}}} \overset{(S)+1}{\boxed{\text{Integer part}}} . \overset{(S)+2}{\boxed{\text{Decimal fraction part}}}) = (D)$$

(2) A sign for the operation data is set at (S)
If the operation data is a positive value, this is set at "0", and if it is a negative value, it is set at "1".

(3) The part before the decimal point and fraction part are stored at (S)+1 and (S)+2 respectively, as BCD values.
(Values from 0 to 9999.9999 can be set.)

(4) Operation results stored at (D) are BCD values between 0 and 90 degrees, and 270 and 360 degrees (degree units).

(5) Calçulation results are a value from which the decimal fraction part has been rounded.

Operation Errors

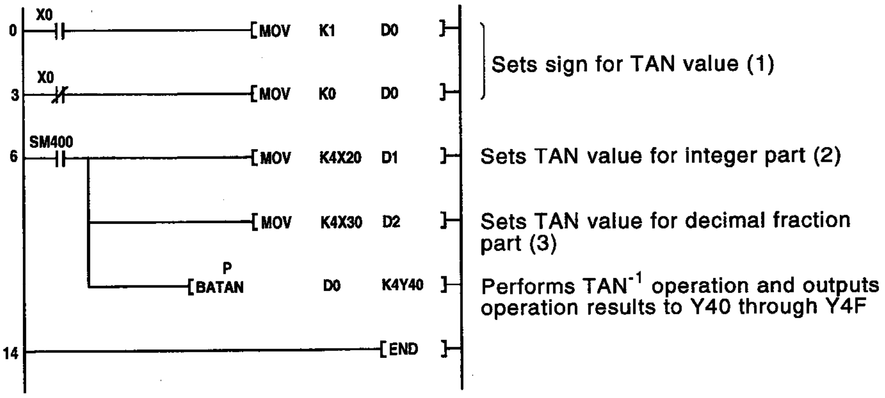
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The operation data designated by (S) is not a BCD value.
(Error code: 4100)

Program Example

(1) The following program performs a TAN^{-1} operation on the sign (positive when X0 is OFF, and negative when X0 is ON), the BCD 4-digit integer part from X20 through X2F and the BCD 4-digit decimal fraction part from X30 through X3F, and outputs the calculated angle in 4 BCD digits from Y40 through Y4F).

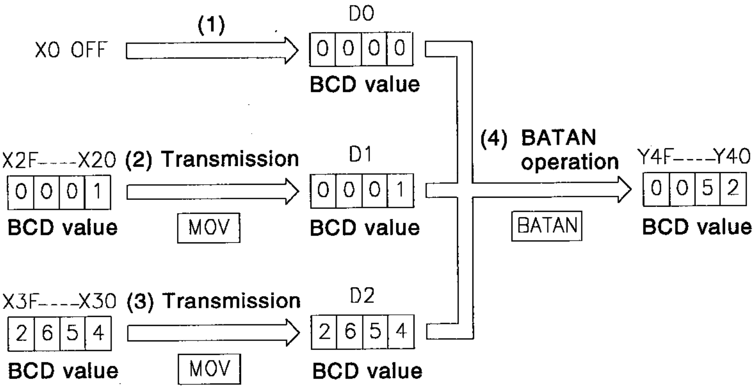
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MOV	K1
		D0
3	LDI	X0
4	MOV	K0
		D0
6	LD	SM400
7	MOV	K4X20
		D1
9	MOV	K4X30
		D2
11	BATANP	D0
		K4Y40
14	END	

[Operations involved when X0 and X20 through XF designate a value of 1.2654]



7.13 Data Control Instructions

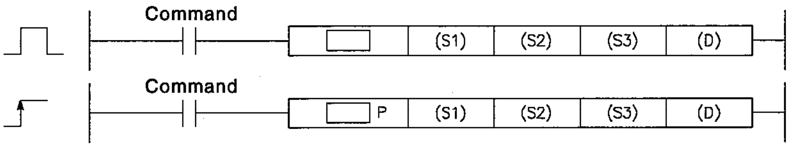
7.13.1 Upper and lower limit controls for BIN 16-bit and BIN 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o							o	—
(S2)	o							o	—
(S3)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition] □ indicates LIMIT/DLIMIT

LIMIT, DLIMIT

LIMITP, DLIMITP



Set Data

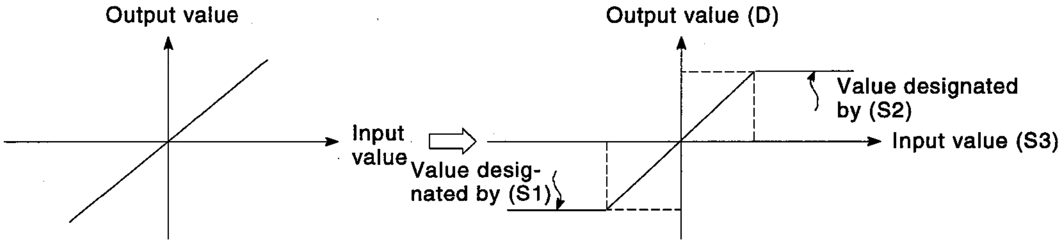
Set Data	Meaning	Data Type
(S1)	Lower limit value (minimum output threshold value)	BIN 16/32 bits
(S2)	Upper limit value (maximum output threshold value)	
(S3)	Control input value made by upper and lower limit controls	
(D)	First device number of devices where output value controlled by upper and lower limit controls will be stored	

Functions

LIMIT

(1) Input values designated by (S3) (BIN 16-bit values) are checked for confirmation that they are within the upper and lower limit values designated by (S1) and (S2), and depending on the results of this check, output values stored in the device designated at (D) are controlled. Output values are controlled in the following manner:

- (S1) $\boxed{\text{Lower limit value}}$ > (S3) $\boxed{\text{Input value}}$
  .... (S1) $\boxed{\text{Lower limit value}}$ → (D) $\boxed{\text{Output value}}$
- (S2) $\boxed{\text{Upper limit value}}$ < (S3) $\boxed{\text{Input value}}$
  .... (S2) $\boxed{\text{Upper limit value}}$ → (D) $\boxed{\text{Output value}}$
- (S1) $\boxed{\text{Lower limit value}}$ ≤ (S3) $\boxed{\text{Input value}}$ ≤ (S2) $\boxed{\text{Upper limit value}}$
  .... (S3) $\boxed{\text{Input value}}$ → (D) $\boxed{\text{Output value}}$



(2) Values in the range from -32768 and 32767 can be designated at (S1), (S2), and (S3).

(3) When control based only on upper limit values is performed, the lower limit value designated at (S1) is set at "-32678".

(4) When control based only on lower limit values is performed, the upper limit value designated at (S2) is set at "32767".

DLIMIT

(1) Controls output value stored at the device designated by ((D)+1, (D)) according to whether or not the input value (BIN 32-bit value) designated by ((S3), (S3)+1) is within the range of the upper and lower limit values designated by ((S2), (S2)+1).

(S1)+1 (S1) (S3)+1 (S3)
- Lower limit value > Input value

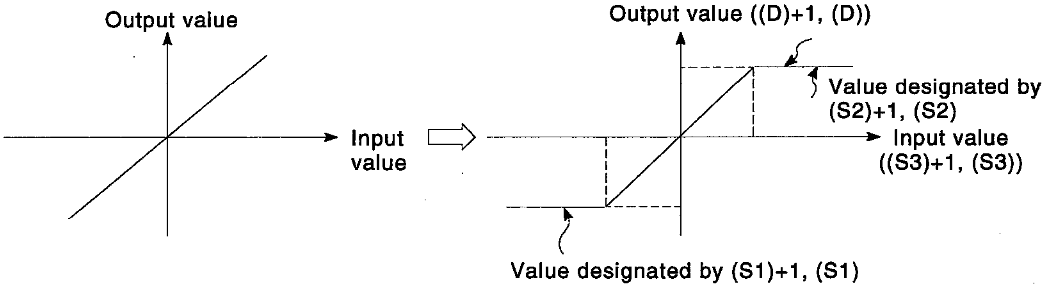
(S1)+1 (S1) (D)+1 (D)
.... Lower limit value → Output value

(S2+1) (S2) (S3+1) (S3)
- Upper limit value < Input value

(S2+1) (S2) (D+1) (D)
.... Upper limit value → Output value

(S1+1) (S1) (S3+1) (S3) (S2+1) (S2)
- Lower limit value ≤ Input value ≤ Upper limit value

(S3+1) (S3) (D+1) (D)
.... Input value → Output value



(2) The values designated by ((S1), (S1)+1), ((S2), (S2)+1), or ((S3), (S3)+1) are within the range of -2147483648 to 2147483647.

(3) To perform controls based only on the upper limit value, set the lower limit value designated by ((S1), (S1)+1) to "-2147483648".

(4) To perform controls based only on the lower limit value, set the upper limit value designated by ((S2), (S2)+1) to "2147483647".

Operation Errors

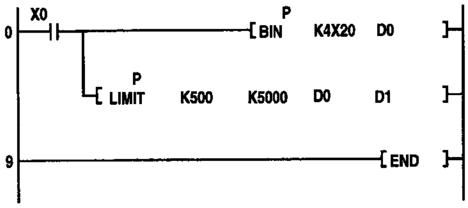
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The lower limit value designated by (S1) is larger than the upper limit value designated by (S2). (Error code: 4100)

Program Example

(1) The following program conducts limit controls from 500 to 5000 on the data set as BCD values from X20 through X2F, and stores the result at D1 when X0 goes ON.

[Ladder Mode] [List Mode]



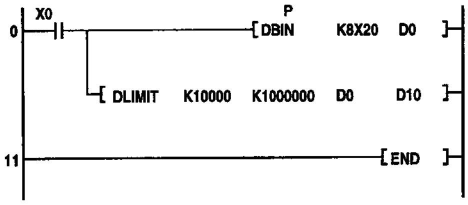
Steps	Instruction	Device
0	LD	X0
1	BINP	K4X20 D0
4	LIMITP	K500 K5000 D0 D1
9	END	

[Operation]

- D1 becomes 500 if D0 < 500.
  Example D0 = 400 → D1 = 500
- D1 becomes the value of D0 when 500 ≤ D0 ≤ 5000.
  Example D0 = 1300 → D1 = 1300
- D1 becomes 5000 when 5000 < D0.
  Example D0 = 9600 → D1 = 5000

(2) The following program conducts limit value controls from 10000 to 1000000 on the data set as BCD values from X20 through X3F when X0 goes ON.

[Ladder Mode] [List Mode]



Steps	Instruction	Device
0	LD	X0
1	DBINP	K8X20 D0
4	DLIMIT	K10000 K1000000 D0 D10
11	END	

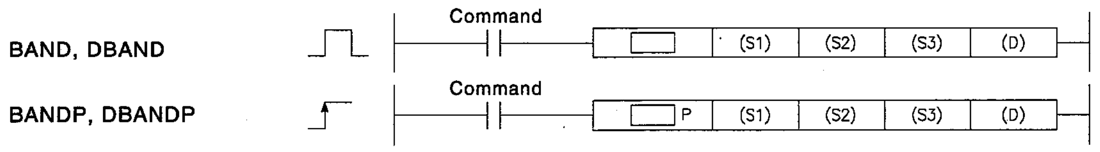
[Operation]

- (D11, D10) become 10000 if (D1, D0) are less than 10000.
  Example (D1, D0) = 400 → (D11, D10) = 10000
- (D11, D10) become the value of (D1, D0) if 10000 ≤ (D1, D0) ≤ 1000000.
  Example (D1, D0) = 345678 → (D11, D10) = 345678
- (D11, D10) become 1000000 if 1000000 < (D1, D0).
  Example (D1, D0) = 9876543 → (D11, D10) = 1000000

7.13.2 BIN 16-bit and 32-bit dead band controls

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o							o	—
(S2)	o							o	—
(S3)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition] ▭ indicates BAND/DBAND



Set Data

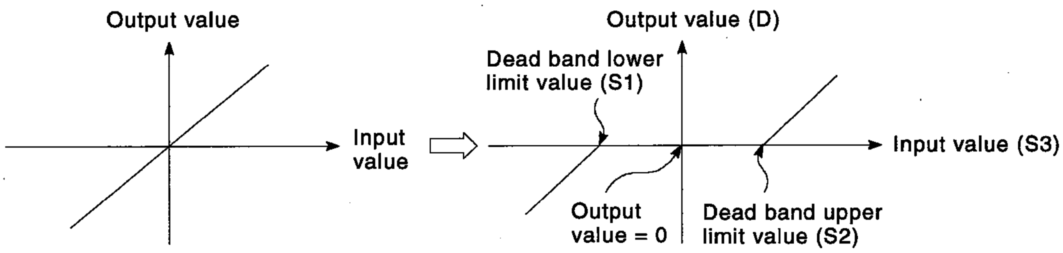
Set Data	Meaning	Data Type
(S1)	Lower limit value of dead band (no output band)	BIN 16/32 bits
(S2)	Upper limit value of dead band (no output band)	
(S3)	Input value controlled based on dead band control	
(D)	First device number of devices that store output value controlled by dead band controls	

Functions

BAND

(1) Controls output value stored at the device designated by (D) based on whether or not the input value (BIN 16-bit value) designated by (S3) is within the range of upper and lower limit dead band controls designated by (S1) and (S2).
The output value is controlled as follows:

- (S1) Lower limit value > (S3) Input value
  .... (S3) Input value - (S1) Lower limit value → (D) Output value
- (S2) Upper limit value < (S3) Input value
  .... (S3) Input value - (S2) Upper limit value → (D) Output value
- (S1) Lower limit value ≤ (S3) Input value ≤ (S2) Upper limit value
  .... 0 → (D) Output value



(2) The values that can be designated by (S1), (S2), and (S3) are in the range of from -32768 to 32767.

(3) The output value stored at (D) is a signed 16-bit BIN value. Therefore, if the operation results exceed the range of from -32768 to 32767, the following will take place:

Dead band lower limit value (S1) 10
Input value (S3) .... -32768
In these cases

Output value = -32768-10 = 8000_H-A_H = $7FF6_H$ = 32758

DBAND

(1) Controls output value stored at the device designated by (D) based on whether or not the input value (BIN 32-bit value) designated by ((S3), (S3)+1) is within the range of upper and lower limit dead band controls designated by ((S1), (S1)+1) and ((S2), (S2)+1). The output value is controlled as follows:

- $\overset{(S1)+1\ \ (S1)}{\boxed{\text{Lower limit value}}} > \overset{(S3)+1\ \ (S3)}{\boxed{\text{Input value}}}$

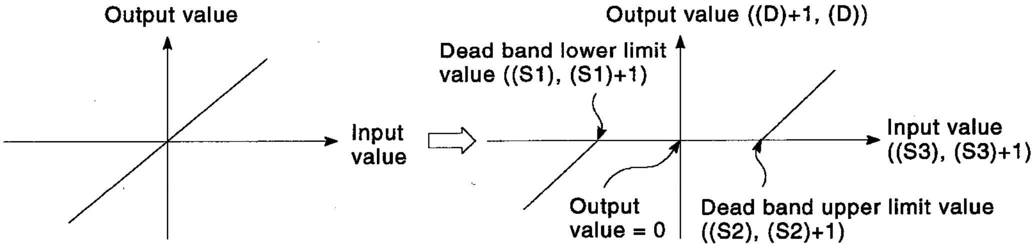
  .... $\overset{(S3)+1\ \ (S3)}{\boxed{\text{Input value}}} - \overset{(S1)+1\ \ (S1)}{\boxed{\text{Lower limit value}}} \rightarrow \overset{(D)+1\ \ (D)}{\boxed{\text{Output value}}}$

- $\overset{(S2)+1\ \ (S2)}{\boxed{\text{Upper limit value}}} < \overset{(S3)+1\ \ (S3)}{\boxed{\text{Input value}}}$

  .... $\overset{(S3)+1\ \ (S3)}{\boxed{\text{Input value}}} - \overset{(S2)+1\ \ (S2)}{\boxed{\text{Upper limit value}}} \rightarrow \overset{(D)+1\ \ (D)}{\boxed{\text{Output value}}}$

- $\overset{(S1)+1\ \ (S1)}{\boxed{\text{Lower limit value}}} \leq \overset{(S3)+1\ (S3)}{\boxed{\text{Input value}}} \leq \overset{(S2)+1\ \ (S2)}{\boxed{\text{Upper limit value}}}$

  .... $0 \rightarrow \overset{(D)+1\ \ (D)}{\boxed{\text{Output value}}}$



(2) The values designated by ((S1), (S1)+1), ((S2), (S2)+1), or ((S3), (S3)+1) are within the range of from -2147483648 to 2147483647.

(3) The output value stored at (D), (D)+1 is a signed 32-bit BIN value. Therefore, if the operation results exceed the range of from -2147483648 to 2147483647, the following takes place:

Dead band lower limit value ((S1), (S1)+1). . 1000 } In these
Input value ((S3), (S3)+1) . . . . -2147483648 } cases
Output value = -2147483648 - 1000 = 80000000H - 000003E8H
= 7FFFFC18H = 2147482648.

Operation Errors

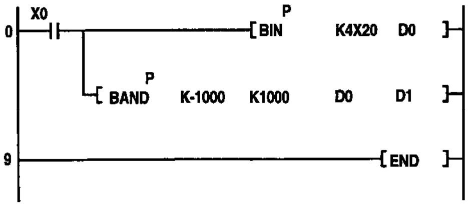
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The lower limit value designated by (S1) is greater than the upper limit value designated by (S2). (Error code: 4100)

Program Example

(1) The following program performs band control from -1000 to 10000 on the data set by the BCD value from X20 through X2F when X0 goes ON, and stores the result at D1.

[Ladder Mode]



[List Mode]

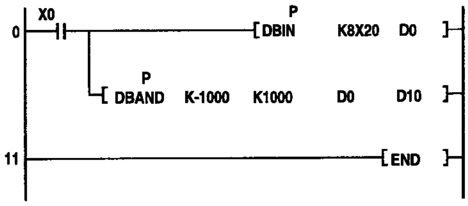
Steps	Instruction	Device
0	LD	X0
1	BINP	K4X20 D0
4	BANDP	K-1000 K1000 D0 D1
9	END	

[Operation]

- The value of (D0)-(-1000) is stored at D1 if D0 is smaller than (-1000).
  Example D0 = -2000 → D1 = -1000
- The value 0 is stored at D1 when -1000 ≤ D0 ≤ 1000.
  Example D0 = 500 → D1 = 0
- The value of (D0)-1000 is stored at D1 if 1000 is less than D0.
  Example D0 = 7000 → D1 = 6000

(2) The following program performs band control from -10000 to 10000 on the data set as BCD values from X20 through X3F, and stores the result at D10 and D11 when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	DBINP	K8X20 D0
4	DBANDP	K-10000 K10000 D0 D10
11	END	

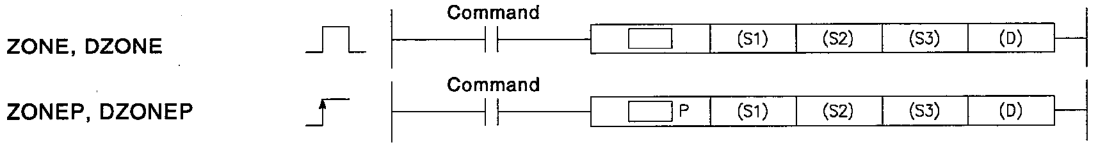
[Operation]

- The value (D1, D0)-(-10000) is stored at (D11, D10) if (D1, D0) is less than -10000.
  Example (D1, D0) = -12345 → (D11, D10) = -2345
- The value 0 is stored at (D11, D10) if -10000 ≤ (D1, D0) ≤ 10000.
  Example (D1, D0) = 6789 → (D11, D10) = 0
- The value (D1, D0)-10000 is stored at (D11, D10) if 10000 < (D1, D0).
  Example (D1, D0) = 50000 → (D11, D10) = 40000

7.13.3 Zone control for BIN 16-bit and BIN 32-bit data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S1)	o							o	—
(S2)	o							o	—
(S3)	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition] □ indicates ZONE/DZONE



Set Data

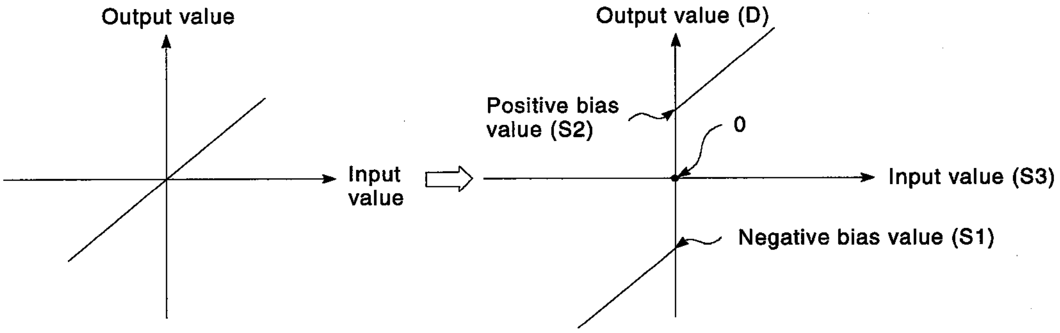
Set Data	Meaning	Data Type
(S1)	Negative bias value added to input value	BIN 16/32 bits
(S2)	Positive bias value added to input value	
(S3)	Input value for the purpose of conducting zone control	
(D)	First device number of devices that will store output value based on zone control	

Functions

ZONE

(1) Adds bias value designated by (S1) or (S2) to input value designated by (S3), and stores at device number designated by (D).
Bias values are calculated in the following manner:

- (S3) Input value < 0
  .... (S3) Input value - (S1) Negative bias value → (D) Output value
- (S3) Input value = 0
  .... 0 → (D) Output value
- (S3) Input value > 0
  .... (S3) Input value - (S2) Positive bias value → (D) Output value



(2) The values that can be designated by (S1), (S2), and (S3) are in the range of from -32768 to 32767.

(3) The output value stored at (D) is a signed 16-bit BIN value. Therefore, if the operation results exceed the range of -32768 to 32767, the following will take place:

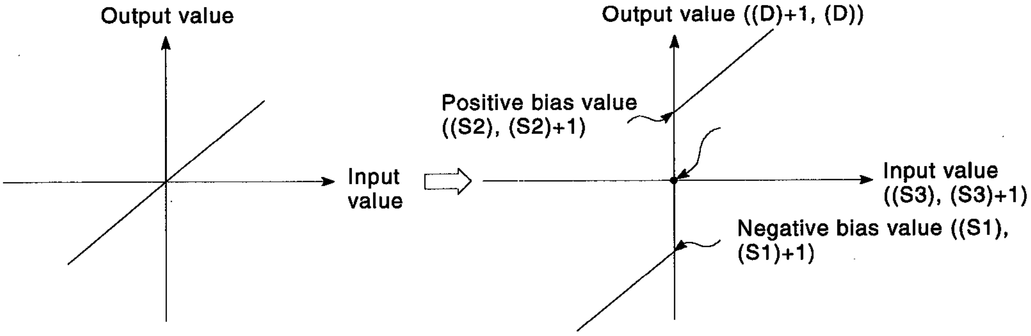
Negative bias value (S1) 100
Input value (S3) .... -32768
} In these cases

Output value $= -32768 + (-100) = 8000_H + FF9C_H = 7F9C_H$
$= 32668$

DZONE

(1) Adds bias value designated by ((S1), (S1)+1) or ((S2), (S2)+1) to input value designated by ((S3), (S3)+1), and stores the result at device number designated by ((D), (D)+1).
Addition of the bias value is performed as follows:

- (S3)+1 (S3) [Lower limit value] < 0
  .... (S3)+1 (S3) [Input value] + (S1)+1 (S1) [Negative bias value] → (D)+1 (D) [Output value]
- (S3)+1 (S3) [Upper limit value] = 0
  .... 0 → (D)+1 (D) [Output value]
- (S3)+1 (S3) [Lower limit value] > 0
  ..... (S3)+1 (S3) [Input value] + (S2)+1 (S2) [Positive bias value] → (D)+1 (D) [Output value]



(2) The values designated by ((S1), (S1)+1), ((S2), (S2)+1), or ((S3), (S3)+1) are within the range of from -2147483648 to 2147483647.

(3) The value stored at ((D), (D)+1) is a signed 32-bit BIN value. Therefore, if the operation results exceed the range of from -2147483648 to 2147483647, the following takes place:

Negative bias value ((S1), (S1)+1)......... 1000 } In these cases
Input value ((S3), (S3)+1) .... -2147483648 }
Output value = -2147483648 + (-1000)
= 80000000_H + $FFFFFC18_H$ = 7FFFFC18
= 2147482648

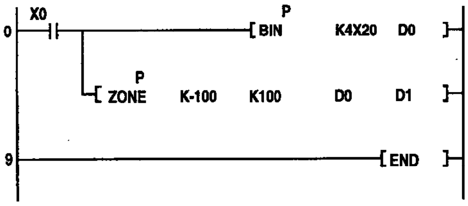
Operation Errors

(1) There are no operation errors associated with the ZONE(P) or DZONE(P) instructions.

Program Example

(1) The following program conducts zone controls of -100 to 100 on the data set as BCD values from X20 through X2F and stores the result at D1 when X0 goes ON.

[Ladder Mode]



[List Mode]

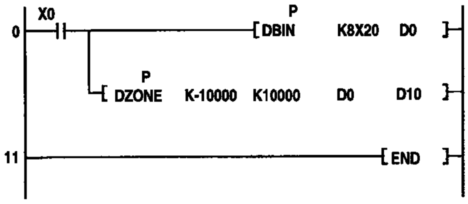
Steps	Instruction	Device
0	LD	X0
1	BINP	K4X20 D0
4	ZONEP	K-100 K100 D0 D1
9	END	

[Operation]

- The value (D0)+(-100) is stored at D1 if D0 is less than 0.
  Example D0 = -200 → D1 = -300
- The value 0 is stored at D1 if D0=0.
- The value of (D0)+100 is stored at D1 if 0 < D0.
  Example D0 = 700 → D1 = 800

(2) The following program conducts zone control from -10000 to 10000 on the data set as BCD values from X20 through X3F, and stores the result at D10 and D11 when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	DBINP	K8X20 D0
4	DZONEP	K-10000 K10000 D0 D10
11	END	

[Operation]

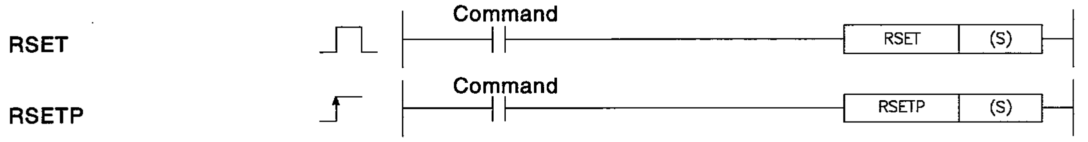
- The value (D1, D0)+(-10000) is stored at (D11, D10) if (D1, D0) is less than 0.
  Example (D1, D0) = -12345 → (D11, D10) = -22345
- The value 0 is stored at (D11, D10) if (D1, D0)=0.
- The value (D1, D0)+10000 is stored at (D11, D10) if 0 < (D1, D0).
  Example (D1, D0) = 50000 → (D11, D10) = 60000

7.14 File Register Switching Instructions

7.14.1 Switching file register numbers

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	○								—

[Instruction Symbol] [Execution Condition]

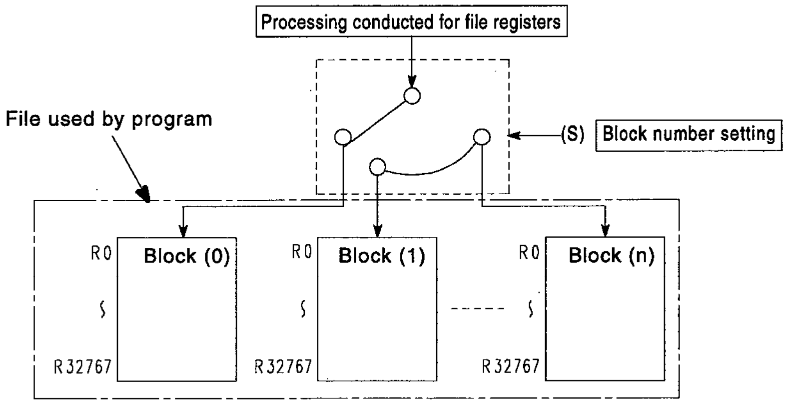


Set Data

Set Data	Meaning	Data Type
(S)	Block number data that will be switched or device number where block number data is being stored	BIN 16 bits

Functions

(1) Changes the file register block number used in the program to the block number stored in the device designated at (S).
Following the block number change, all file registers used in the sequence program are processed in relation to the file register of the block number after the change.



Operation Errors

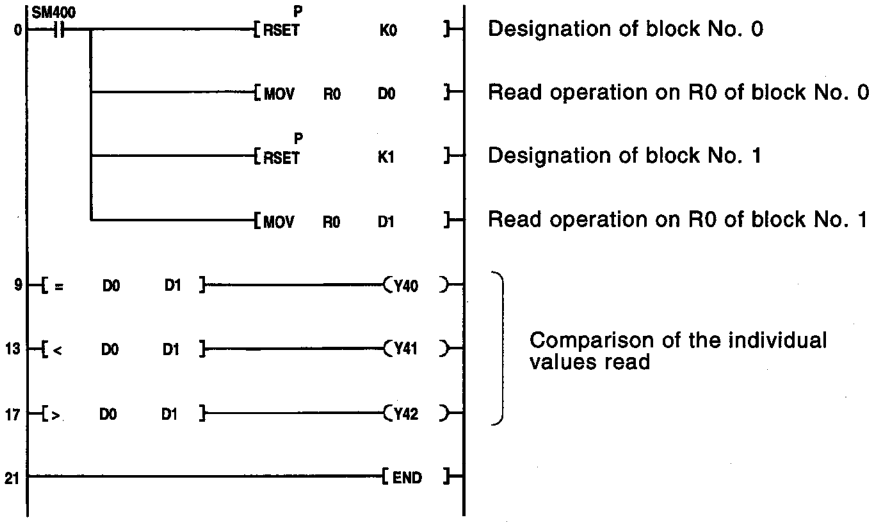
(1) In the following cases an operation error occurs and the error flag (SM0) goes ON.

- The block number designated by (S) does not exist. (Error code: 4100)
- The block number designated by (S) has no file register. (Error code: 2410)

Program Example

(1) The following program compares R0 of block No. 0 and block No. 1.

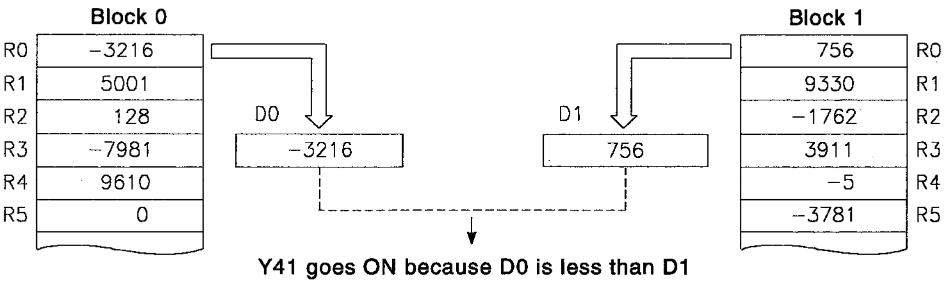
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	SM400
1	RSETP	K0
3	MOV	R0
		D0
5	RSETP	K1
7	MOV	R0
		D1
9	LD=	D0
		D1
12	OUT	Y40
13	LD<	D0
		D1
16	OUT	Y41
17	LD>	D0
		D1
20	OUT	Y42
21	END	

[Operation]



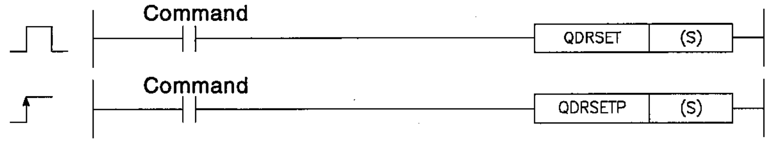
7.14.2 Setting files for file register use

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—

[Instruction Symbol] [Execution Condition]

QDRSET

QDRSETP

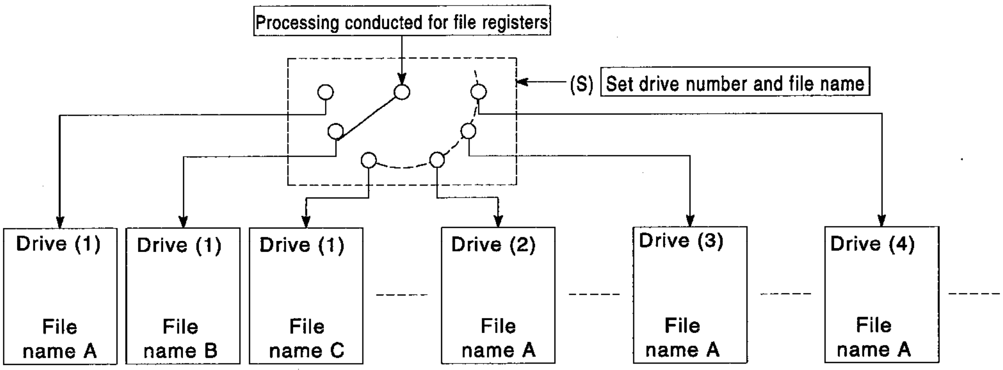


Set Data

Set Data	Meaning	Data Type
(S)	Drive number and character string data for file name where the file register file is to be set, or first device number of devices where the character string is being stored.	Character string

Functions

(1) Changes the file register file name used in the program to the file name being stored at the device designated by (S).
After the file name change, all file registers being used by the sequence program perform processing in relation to the file register of the block No. 0 of the file name after the change.
Block number switches are performed by the RSET instruction.



(2) Drive number can be designated from 1 to 4.
The drive number cannot be designated as drive 0 (internal memory).

(3) It is not necessary to designate the extension (.QDR) with the file name.

(4) A file name setting can be deleted by designating the NULL character (00_H) for the file name.

(5) File names designated with this instruction will be given priority even if a drive number and file name have been designated in the parameters.

Operation Errors

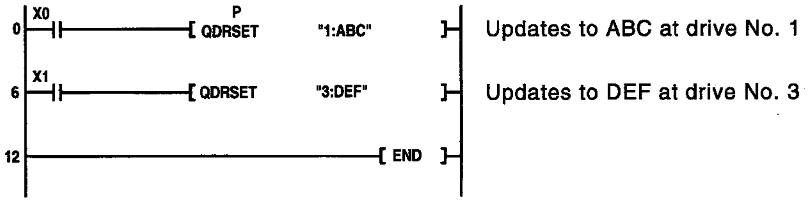
(1) In the following cases an operation error occurs and the error flag turns ON.

- File name does not exist at the drive number designated by (S).
(Error code: 2410)

Program Example

(1) The following program switches object file to file name ABC.QDR at drive No. 1 when X0 is ON, and to DEF.QDR at drive No. 3 when X1 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	QDRSETP	"1:ABC"
6	LD	X1
7	QDRSET	"3:DEF"
12	END	

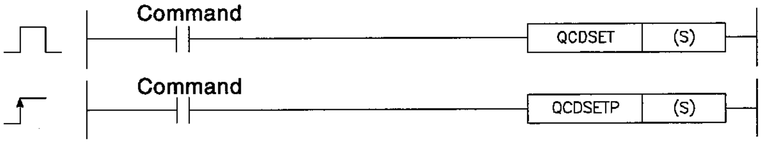
7.14.3 File setting for comments

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—

[Instruction Symbol] [Execution Condition]

QCDSET

QCDSETP

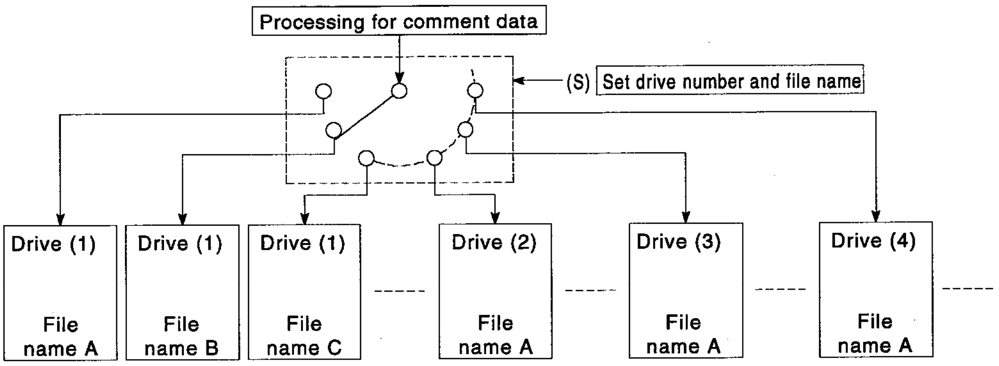


Set Data

Set Data	Meaning	Data Type
(S)	Drive number and character string data for file name where the comment file is to be set, or first device number of devices where the character string is being stored.	Character string

Functions

(1) Changes the file register file name used in the program to the file name being stored at the device designated by (S).
After the file name change, comment data being used by the sequence program perform processing in relation to the comment data of the file name after the change.



(2) Drive number can be designated from 1 to 4.
The drive number cannot be designated as drive 0 (internal memory).

(3) It is not necessary to designate the extension (.QCD) with the file name.

(4) A file name setting can be deleted by designating the NULL character (00_H) for the file name.

(5) File names designated with this instruction will be given priority even if a drive number and file name have been designated in the parameters.

Operation Errors

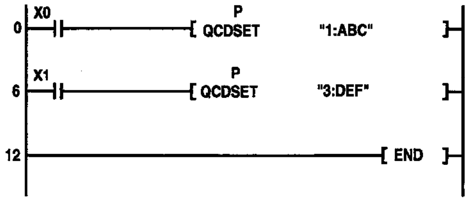
(1) In the following cases an operation error occurs and the error flag turns ON.

- File name does not exist at the drive number designated by (S). (Error code: 2410)

Program Example

(1) The following program switches object file to file name ABC.QDR at drive No. 0 when X0 is ON, and to DEF.QDR at drive No. 1 when X1 is ON.

[Ladder Mode]



Updates to ABC at drive No. 1

Updates to DEF at drive No. 3

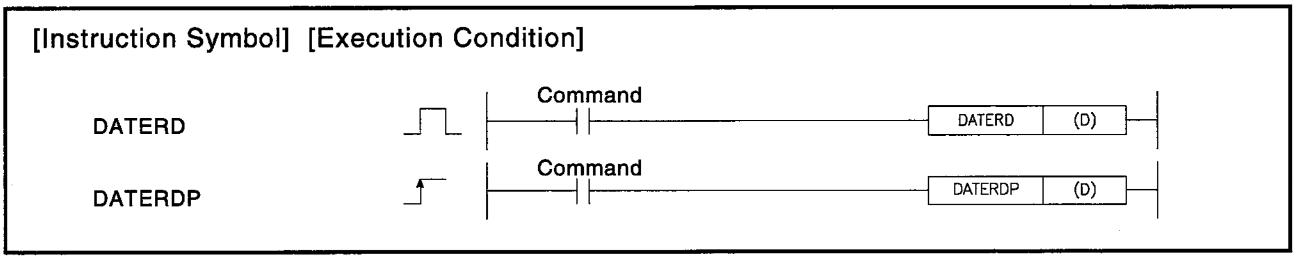
[List Mode]

Steps	Instruction	Device
0	LD	X0
1	QCDSETP	"1:ABC"
6	LD	X1
7	QCDSETP	"3:DEF"
12	END	

7.15 Clock Instructions

7.15.1 Reading clock data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	—	o		—					

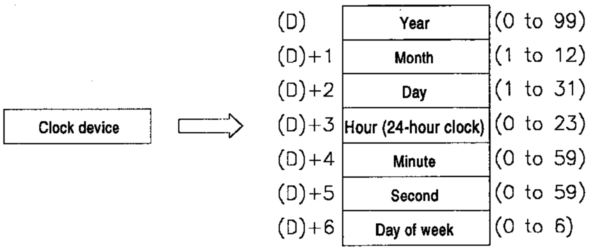


Set Data

Set Data	Meaning	Data Type
(D)	First device number of devices that will store clock data when read	BIN 16 bits

Functions

(1) Reads "year, month, day, minute, second, and day of week" from the QnACPU clock device, and stores as BIN value following device designated by (D).



(2) The year is stored only as the final two digits of the year.
The year 1995, for example, would be stored as "95".

(3) The "day of week" at (D)+6 is stored as 0 through 6 to represent the days Sunday through Saturday.

Day of week	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Stored data	0	1	2	3	4	5	6

(4) Compensation is made automatically for leap years.

Operation Errors

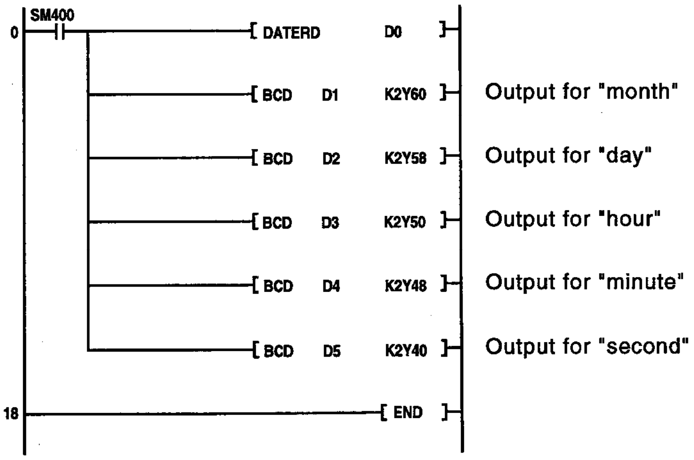
(1) There are no operation errors associated with the DATERD(P) instruction.

Program Example

(1) The following program outputs the following clock data as BCD values:

Month. . . . . . . Y60 to Y67
Day. . . . . Y58 to Y5F
Hour. . . . . Y50 to Y57
Minute . . . . . . Y48 to Y4F
Second . . . . . Y40 to Y47

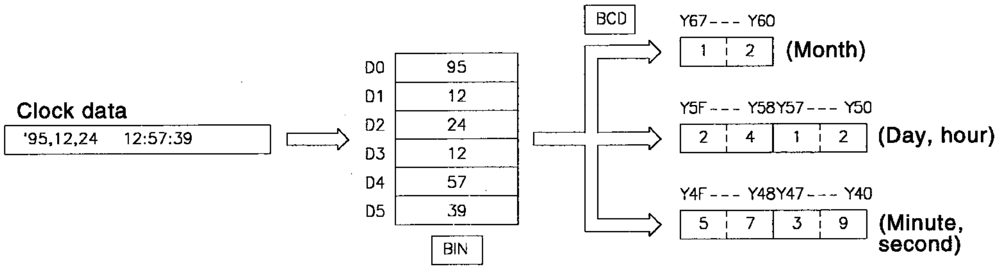
[Ladder Mode]



[List Mode]

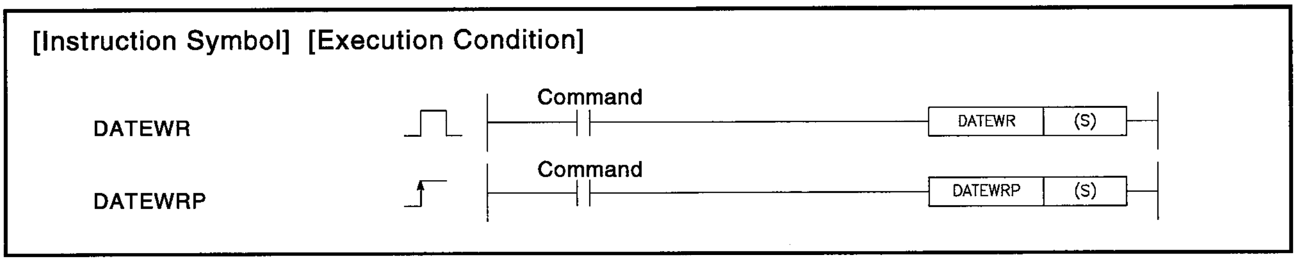
Steps	Instruction	Device
0	LD	SM400
1	DATERD	D0
3	BCD	D1 K2Y60
6	BCD	D2 K2Y58
9	BCD	D3 K2Y50
12	BCD	D4 K2Y48
15	BCD	D5 K2Y40
18	END	

[Operation]



7.15.2 Writing clock data

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	—	o		—					

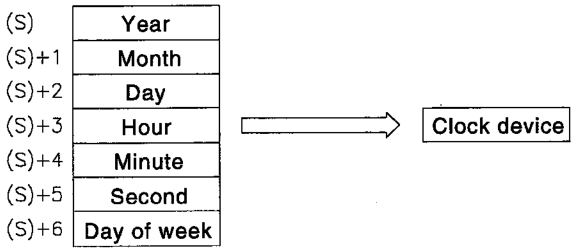


Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices that is storing clock data to be written to the clock device	BIN 16 bits

Functions

(1) Writes clock data being stored following the device number designated by (S) to the QnACPU clock device.



(2) Each item is set as a BIN value.

(3) The "year" designated by (S) is set as the final two digits of the four digit number.
The year 1995 would be set as "95".

(4) (S)+1 designates the "month" in values of from 1 through 12 (January through December).

(5) (S)+2 designates the "day" in values of from 1 through 31.

(6) (S)+3 designates the "hour" in values of from 0 to 23 (using 24-hour clock, from 0 hours to 23 hundred hours). (Uses the 24-hour clock.)

(7) (S)+4 designates the "minute" in values of from 0 to 59.

(8) (S)+5 designates the "second" in values of from 0 to 59.

(9) (S)+6 designates the "day of week" in values of from 0 to 6 (Sunday through Saturday).

Day of week	Sunday	Monday	Tuesday	Wedne-sday	Thurs-day	Friday	Satur-day
Stored data	0	1	2	3	4	5	6

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

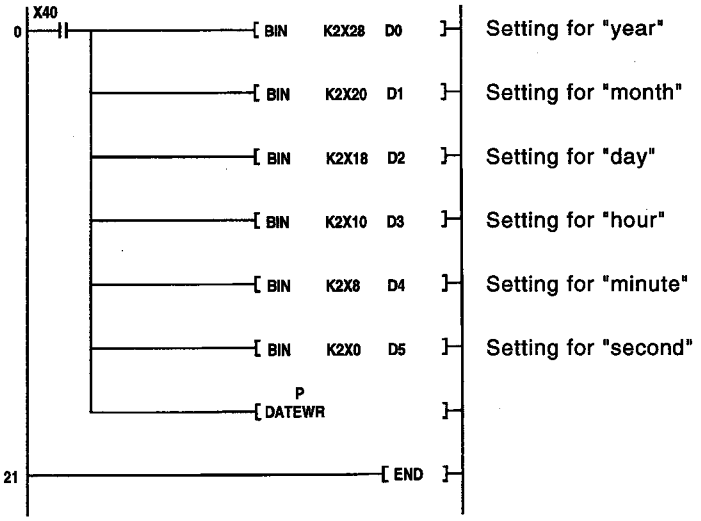
- Individual items of data have been set outside the setting range. (Error code: 4100)

Program Example

(1) The following program writes the following clock data to the clock device as BCD values when X40 goes ON.

Year...........	X28 to 2F	Hour	X10 to 17
Month	X20 to27	Minute...........	X8 to F
Day	X18 to 1F	Second	X0 to 7

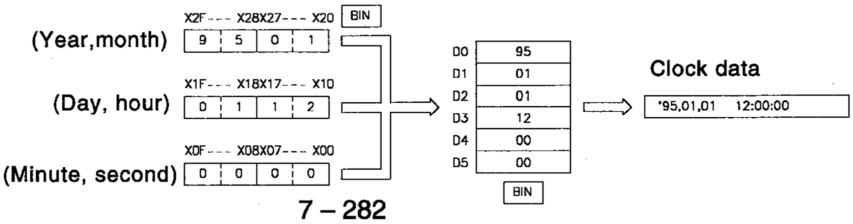
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X40
1	BIN	K2X28 D0
4	BIN	K2X20 D1
7	BIN	K2X18 D2
10	BIN	K2X10 D3
13	BIN	K2X8 D4
16	BIN	K2X0 D5
19	DATEWRP	D0
21	END	

[Operating]



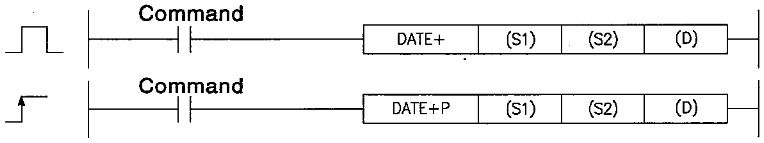
7.15.3 Clock data addition operation

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J◻\◻		Special Function Module U◻\G◻	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—					
(S2)	—	o		—					
(S)	—	o		—					

[Instruction Symbol] [Execution Condition]

DATE+

DATE+P



Set Data

Set Data	Meaning	Data Type
(S1)	Time data to be added to	BIN 16 bits
(S2)	Added time (clock) data	
(D)	First device number of devices where addition results of clock (time) data	

Functions

DATE+

(1) Adds time data being stored following the device number designated by (S2) to time data being stored in devices starting from that whose device number is designated by (S1), and stores the result of the addition in devices starting from that whose device number is designated by (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)

For example, adding the time 7:48:10 to 6:32:40 would result in the following operation:

(S1)	Hour: 6		(S2)	Hour: 7		(D)	Hour: 14
(S1)+1	Minute: 32	+	(S2)+1	Minute: 48	⇨	(D)+1	Minute: 20
(S1)+2	Second: 40		(S2)+2	Second: 10		(D)+2	Second: 50

(2) If the results of the addition of time exceed 24 hours, 24 hours will be subtracted from the sum to make the final operation result.
For example, if the time 20:20:20 were added to 14:20:30, the result would not be 34:40:50, but would instead be 10:40:50.

(S1)	Hour: 14		(S2)	Hour: 20		(D)	Hour: 10
(S1)+1	Minute: 20	+	(S2)+1	Minute: 20	⇨	(D)+1	Minute: 40
(S1)+2	Second: 30		(S2)+2	Second: 20		(D)+2	Second: 50

REMARK

See Section 7.15.2 for further information regarding the data that can be set for hours, minutes, and seconds.

Operation Errors

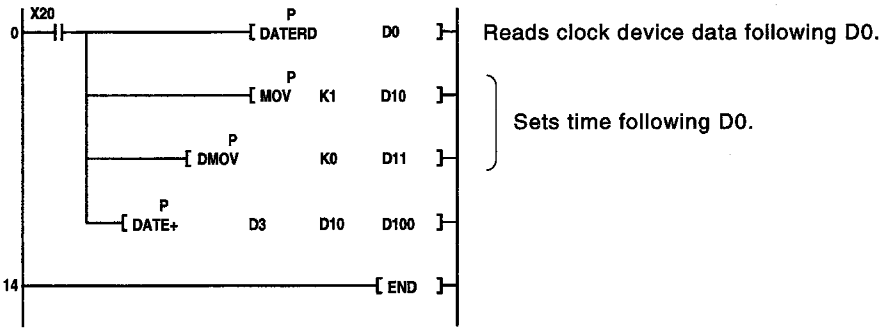
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data set by (S1) and (S2) is outside the setting range.

Program Example

(1) The following program adds 1 hour to the clock data read from the clock device, and stores the results following D100 when X20 is ON.

[Ladder Mode]

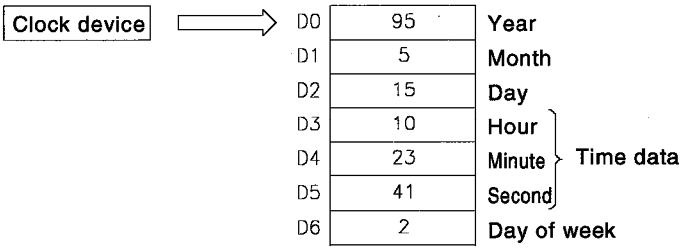


[List Mode]

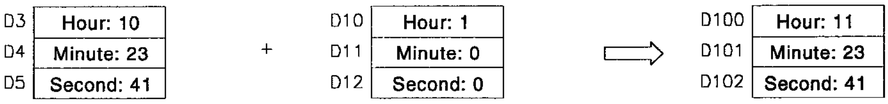
Steps	Instruction	Device
0	LD	X20
1	DATERDP	D0
3	MOVP	K1
		D10
6	DMOVP	K0
		D11
10	DATE+P	D3
		D10
		D100
14	END	

[Operation]

- Time data read operation triggered by DATERDP instruction.



- Addition triggered by DATE+P instruction.



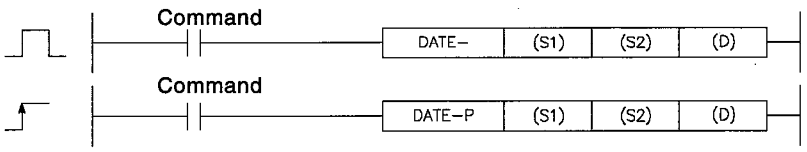
7.15.4 Clock data subtraction operation

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—					
(S2)	—	o		—					
(D)	—	o		—					

[Instruction Symbol] [Execution Condition]

DATE-

DATE-P



Set Data

Set Data	Meaning	Data Type
(S1)	First device number of devices where time data to be subtracted from is being stored	BIN 16 bits
(S2)	First device number of devices where subtraction time (clock time) data is being stored	
(D)	First device number of devices where clock time (time) data of results of subtraction operation is being stored	

Functions

(1) Subtracts time data being stored in devices starting from that whose number is designated by (S2) from the time data being stored in devices starting from that whose number is designated by (S1), and stores the subtraction result in devices starting from that whose number is designated by (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)

For example, if the clock time 3:50:10 were subtracted from the clock time 10:40:20, the operation would be performed as follows:

(S1)	Hour: 10	(0 to 23)		(S2)	Hour: 3	(0 to 23)		(D)	Hour: 6	(0 to 23)
(S1)+1	Minute: 40	(0 to 59)	−	(S2)+1	Minute: 50	(0 to 59)	⇨	(D)+1	Minute:50	(0 to 59)
(S1)+2	Second:	(0 to 59)		(S2)+2	Second: 10	(0 to 59)		(D)+2	Second:10	(0 to 59)

(2) If the subtraction results in a negative number, 24 will be added to the result to make a final operation result.
For example, if the clock time 10:42:12 were subtracted from 4:50:32, the result would not be -6:8:20, but rather would be 18:8:20.

(S1)	Hour: 4		(S2)	Hour: 10		(D)	Hour: 18
(S1)+1	Minute: 50	-	(S2)+1	Minute: 42	⇒	(D)+1	Minute: 8
(S1)+2	Second: 32		(S2)+2	Second: 12		(D)+2	Second: 20

REMARK

See Section 7.15.2 for further information regarding the data that can be set for hours, minutes, and seconds.

Operation Errors

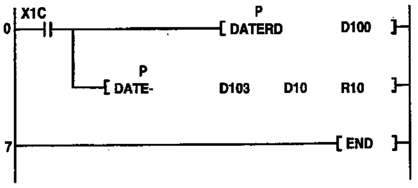
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data set by (S1) and (S2) is outside the setting range.

Program Example

(1) The following program subtracts the time data being stored in devices starting from D10 from the clock data read from the clock device when X1C goes ON, and stores the result at devices starting from R10.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X1C
1	DATERDP	D100
3	DATE-P	D103 D10 R10
7	END	

[Operation]

- Time data read operation triggered by DATERDP.

Clock device ⇒

D100	95	Year
D101	4	Month
D102	20	Day
D103	3	Hour } Time data
D104	21	Minute } Time data
D105	20	Second } Time data
D106	1	Day of week

- Subtraction as triggered by DATE-P instruction (when 10 hours, 40 minutes, and 10 seconds have been designated by D10 through D12).

D103	Hour: 3		D10	Hour: 10		R10	Hour: 16
D104	Minute: 21	-	D11	Minute: 40	⇒	R11	Minute: 41
D105	Second: 20		D12	Second: 10		R12	Second: 10

3:21:20 - 10:40:10 ⇒ -8:41:10 ⇒ 16:41:10

24 added to this value

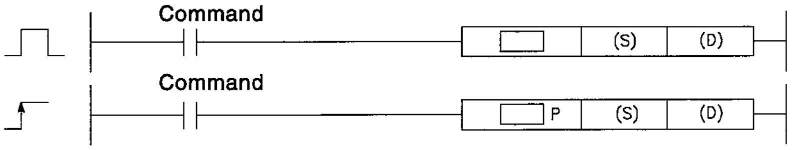
7.15.5 Changing time data formats

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				—	—
(D)	o	o		o				—	—
(S)	o	o		o				o	—
(D)	—	o		—				—	—

[Instruction Symbol] [Execution Condition]

SECOND, HOUR

SECONDP, HOURP



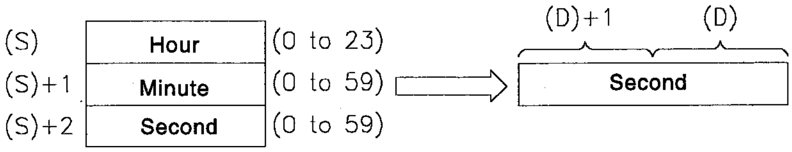
Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices storing clock data prior to change	BIN 16/32 bits
(D)	First device number of devices that will store clock data after change	BIN 16/32 bits

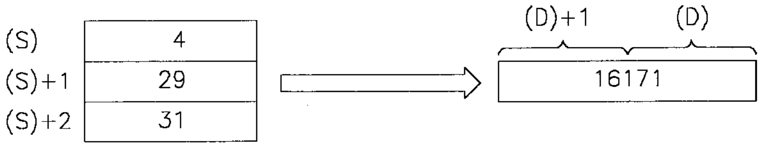
Functions

SECOND

(1) Converts time data being stored following the device designated by (S) to seconds, and stores the conversion results at the device designated by (D).

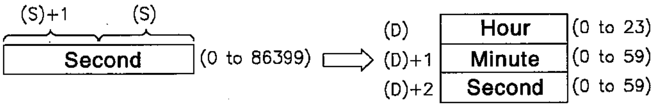


For example, if the value were 4 hours, 29 minutes, and 31 seconds, the conversion would be made as follows:

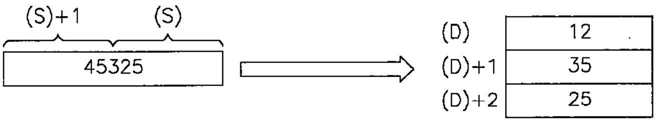


HOUR

(1) Converts the seconds data being stored at the device number designated by (S) to hours, minutes, and seconds, and stores the operation results following the device designated by (D).



For example, if 45325 seconds were the value designated, the conversion operation would be conducted as follows:



Operation Errors

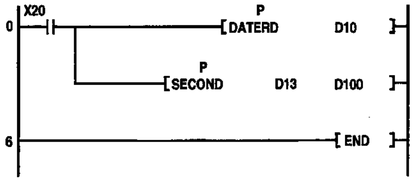
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The data designated by (S) is outside the acceptable range.
  (Error code: 4100)

Program Example

(1) The following program converts the clock time data read from the clock device when X20 goes ON into seconds, and stores the result at D100 and D101.

[Ladder Mode]

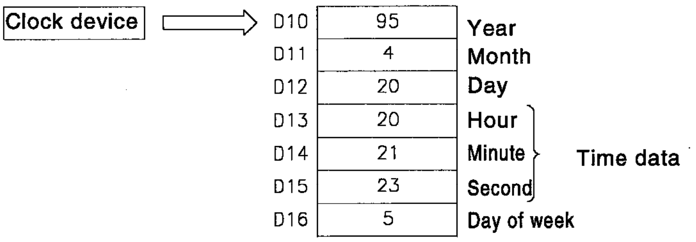


[List Mode]

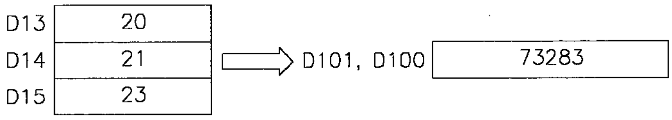
Steps	Instruction	Device
0	LD	X20
1	DATERDP	D10
3	SECONDP	D13 D100
6	END	

[Operation]

- Time data read operation triggered by DATERDP.

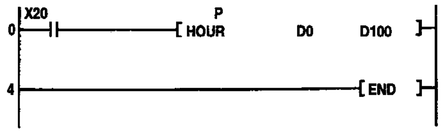


- Conversion to seconds as triggered by the SECONDP instruction.



(2) The following program converts the seconds being stored at D0 and D1 when X20 goes ON to an hour, minute, second format, and stores the result at devices starting from D100.

[Ladder Mode]

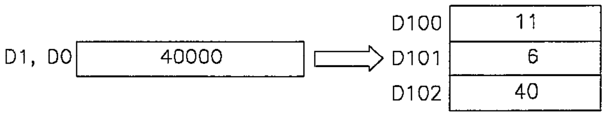


[List Mode]

Steps	Instruction	Device
0	LD	X20
1	HOURP	D0
		D100
4	END	

[Operation]

- Conversion to hour minute, and second format as triggered by the Hour-pinstruction (when the value 40000 seconds has been designated by D1 andD0).



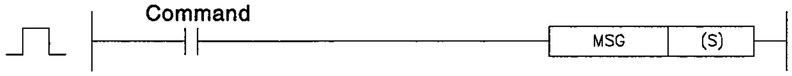
7.16 Peripheral Device Instructions

7.16.1 Message displays to peripheral devices

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—

[Instruction Symbol] [Execution Condition]

MSG

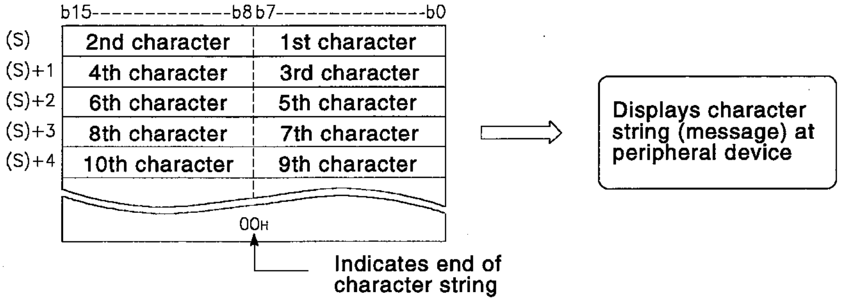


Set Data

Set Data	Meaning	Data Type
(S)	Character string data from which display will be made, or first device number of devices where character string data is being stored	Character string

Functions

(1) Displays the character string data being stored in devices starting from that whose device number is designated by (S) as a message at the peripheral device set in the terminal mode.
The end of the character string data is marked by 00H.



(2) The character string data displayed at the peripheral device can include a maximum of up to 64 characters.

(3) The character string data designated by (S) is stored from SD738 through SD773 (message storage area).

(4) When the MSG instruction is executed, SM738 (MSG instruction execution signal) is turned ON.
When SM738 goes ON, any other MSG instruction that is executed will not be processed.

(5) When a character string (message) is displayed at a peripheral device, SM738 goes OFF, and all character string data that had been stored from SD738 through SD773 is cleared (00H).

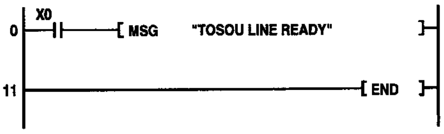
Operation Errors

(1) There are no operation errors associated with the MSG instruction.

Program Example

(1) The following program sends the message "TOSOU LINE READY" for display at a peripheral device when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MSG	"TOSOU LINE READY"
11	END	

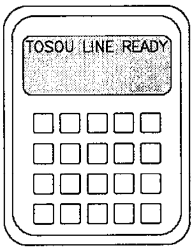
[Operation]

"TOSOU LINE READY"

⇩ When MSG instruction is executed

	b15-------------b8	b7--------------b0
SD738	4FH (O)	54H (T)
SD739	4FH (O)	53H (S)
SD740	20H (space)	55H (U)
SD741	49H (I)	4CH (L)
SD742	45H (E)	4EH (N)
SD743	52H (R)	20H (space)
SD744	41H (A)	45H (E)
SD745	59H (Y)	44H (D)
SD746	00H	

Display ⇨



REMARK

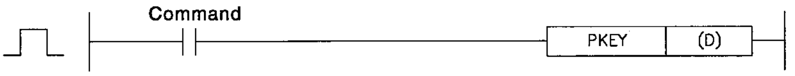
See peripheral device operating manual for information about how to operate in the terminal mode on the peripheral device in question.

7.16.2 Keyboard input from peripheral devices

Set Data	Usable Devices									
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
(D)	—	o		—					

[Instruction Symbol] [Execution Condition]

PKEY



Set Data

Set Data	Meaning	Data Type
(D)	First device number of devices that will store keyboard input	BIN 16 bits

Functions

(1) When the PKEY instruction is executed, 17 words are cleared from the device designated by (D), and SM736 (PKEY instruction execution in progress flag) goes ON.
Additionally, SM737 (keyboard input reception flag) goes OFF.
When the execution of the PKEY instruction is finished, keyboard input data from the peripheral device designated in the terminal mode is stored as ASCII code following the device designated by (D).

(2) When the PKEY instruction command goes OFF, SM736 and SM737 go OFF.

(3) When one character of keyboard input is received from the peripheral device, SM737 goes ON, and when the QnACPU stores the data from the keyboard input, SM737 goes OFF.
While SM737 is ON, keyboard input from peripheral devices will not be accepted.

(4) When "CR" is keyed in from the peripheral device, the ASCII code (0DH) of "CR" is appended at the end of the keyed in data to end the keying operation from the peripheral device.

(5) The data volume that can be input from peripheral devices is 32 characters.
If 32 characters have been input from the peripheral device, keyboard input will be stopped from the peripheral device even though "CR" may not have been input.

(6) The storage of data following the device designated by (D) is performed as shown below:

(D)	Counter	
(D)+1	2nd character	1st character
(D)+2	4th character	3rd character
(D)+3	6th character	5th character
(D)+15	30th character	29th character
(D)+16	32nd character	31st character

(D): Number of characters input through keyboard stored as BIN

(D)+1 to (D)+16: Maximum 16 words

(7) The PKEY instruction cannot be executed from two locations or more simultaneously.
When it is necessary to use the PKEY instruction from two or more locations, establish an interlock with SM736 (PKEY instruction execution in progress flag) so they will not be executed simultaneously.

Operation Errors

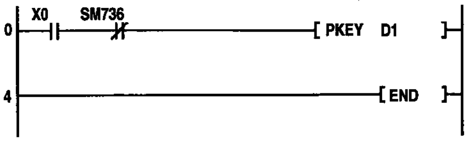
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- An attempt was made to store keyboard input data that exceeded the device range designated by (D). (Error code: 4101)

Program Example

(1) The following program inputs the characters "TOSOU LINE READY" following D1 from the peripheral device after X0 goes ON.

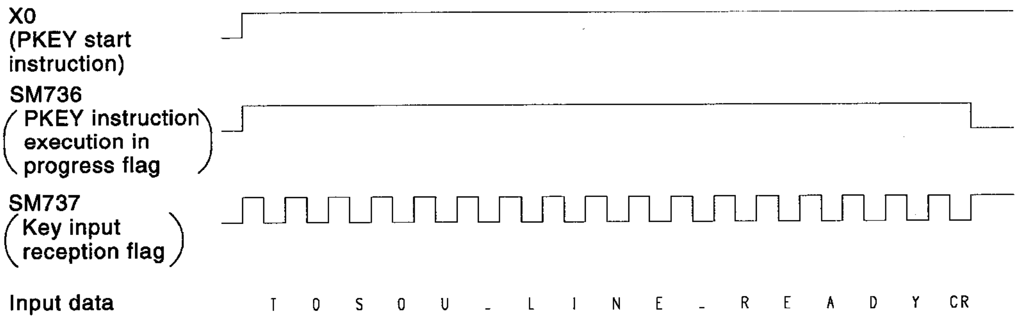
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ANI	SM736
2	PKEY	D1
4	END	

[Operation]



[Storage of keyboard input data]

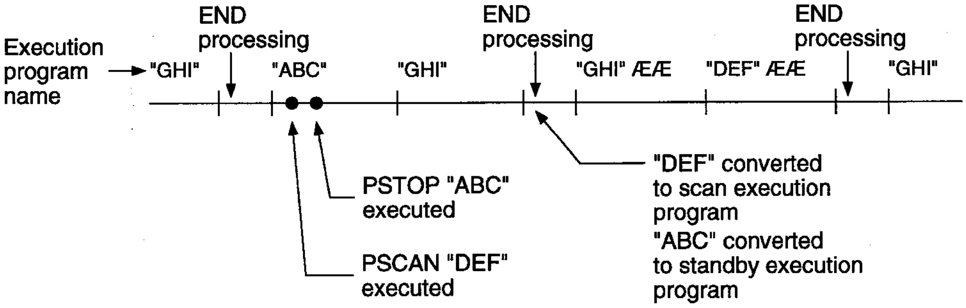
D1	16	
D2	$4F_H$	54_H
D3	$4F_H$	53_H
D4	20_H	55_H
D5	49_H	$4C_H$
D6	45_H	$4E_H$
D7	52_H	20_H
D8	41_H	45_H
D9	59_H	44_H
D10		00_H

7.17 Instructions for Program Controls

(1) Processing when the execution type is converted with the program control instruction is as follows.

Executed Instruction Execution type before change	PSCAN	PSTOP	POFF	PLOW
Scan execution type	No change-remains scan type execution.	Becomes standby type.	Output turned OFF in next scan. Becomes standby type from the next scan after that.	Becomes low-speed type.
Initial execution type	Becomes scan execution type.			
Standby type		No change-remains standby type.	No processing.	
Low-speed execution type	Low-speed execution type execution is stopped: becomes scan execution type from the next scan.(Execution from step 0)	Low-speed execution type execution is stopped: becomes standby type from next scan.	Low-speed type execution is stopped, and output is turned OFF in the next scan. Becomes standby type from the next scan after that.	No change - remains low-speed type.

(a) As program execution type conversions by PSCAN and PSTOP instructions occur at the END processing, such conversions are impossible during program execution. When different execution types have been set for the same program in the same scan, the execution type will be that specified by the execution switching command that was executed last.



REMARK

1) *: The order of GHI and DEF program execution is determined by the program settings parameters.

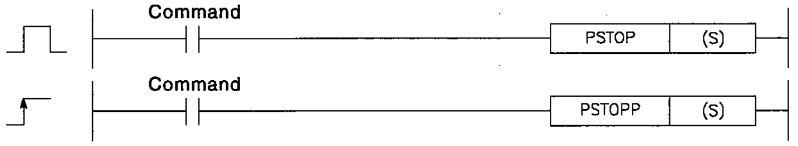
7.17.1 Program standby instruction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—

[Instruction Symbol] [Execution Condition]

PSTOP

PSTOPP



Set Data

Set Data	Meaning	Data Type
(S)	Character string data with program file name to be placed in the standby state, or first device number of devices where character string data is being stored.	Character string

Functions

(1) Places the file name program stored in the device designated by (S) in the standby state.

(2) All that can be done in the standby state is to run the program stored in internal memory (drive 0).

(3) The specified program is placed in the standby status when END processing is performed.

(4) This instruction will be given priority even in cases when a program execution type has been designated in the parameters.

(5) It is not necessary to designate the extension (.QPG) with the file name. (Only .QPG files will be acted on.)

Operation Errors

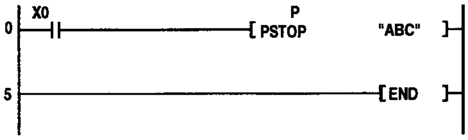
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The program with the designated file name does not exist. (Error code: 2410)

Program Example

(1) The following program places the program with the file name ABC in the standby state when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	PSTOPP	"ABC"
5	END	

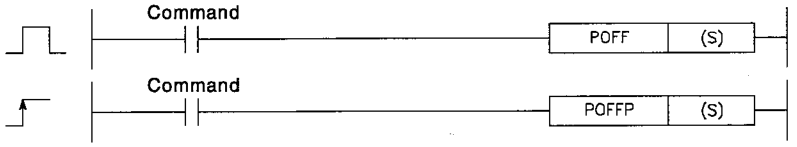
7.17.2 Program output OFF standby instruction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	○		—				○	—

[Instruction Symbol] [Execution Condition]

POFF

POFFP



Set Data

Set Data	Meaning	Data Type
(S)	First device number of the devices where the program file name designating standby status or the file name is stored	Character string

Functions

(1) Places the program with the file name stored in the device designated by (S) in the non-execution standby state.

(2) Only programs stored in internal memory (drive No. 0) can be made non-executionable and placed in the standby state.

(3) The designated program will undergo END processing, and be placed in the non-execution standby state.

(4) This instruction will be given priority even in cases when a program execution type has been designated in the parameters.

(5) It is not necessary to designate the extension (.QPG) with the file name.
(Only .QPG files will be acted on.)

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The program with the designated file name does not exist.
(Error code: 2410)

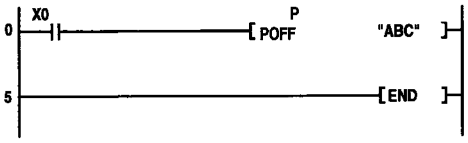
REMARK

The OUT instruction coil is turned OFF as a part of non-execution processing.

Program Example

(1) The following program makes the program with the file name ABC non-executionable and places it in the standby state when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	POFFP	"ABC"
5	END	

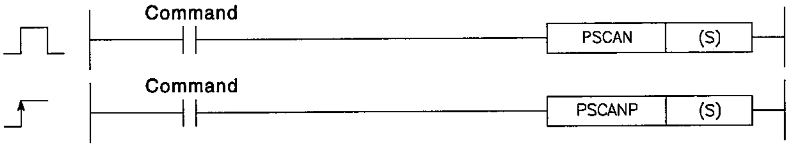
7.17.3 Program scan execution registration instruction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—

[Instruction Symbol] [Execution Condition]

PSCAN

PSCANP



Set Data

Set Data	Meaning	Data Type
(S)	Program file name to be placed in scan execution state, or first device number of devices where the file name is being stored	Character string

Functions

(1) Places the program with the file name being stored at the device designated by (S) in the scan execution state.

(2) Only programs stored in internal memory (drive No. 0) can be placed in the scan execution state.

(3) Designated programs assume the scan execution state with END processing.

(4) This instruction will be given priority even in cases when a program execution type has been designated in the parameters.

(5) It is not necessary to designate the extension (.QPG) with the file name.
(Only .QPG files will be acted on.)

Operation Errors

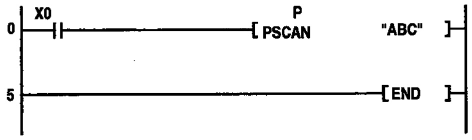
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The program with the designated file name does not exist.
(Error code: 2410)

Program Example

(1) The following program places the program with file name ABC in the scan execution state when X0 goes ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	PSCANP	"ABC"
5	END	

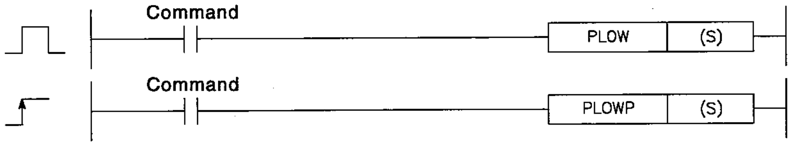
7.17.4 Program low speed execution registration instruction

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J::\::		Special Function Module U::\G::	Index Register Zn	Constant $	Other
	Bit	Word		Bit	Word				
(S)	—	o		—				o	—

[Instruction Symbol] [Execution Condition]

PLOW

PLOWP



Set Data

Set Data	Meaning	Data Type
(S)	File name of program to be designated for low-speed execution, or the first number of the device where the file name is being stored	Character string

Functions

(1) Places the program whose file name is being stored at the device designated by (S) in low-speed execution state.

(2) Only programs stored in internal memory (drive No. 0) can be placed in the low-speed execution state.

(3) Designated programs assume the low-speed execution state with END processing.

(4) This instruction will be given priority even in cases when a program execution type has been designated in the parameters.

(5) It is not necessary to designate the extension (.QPG) with the file name.
(Only .QPG files will be acted on.)

Operation Errors

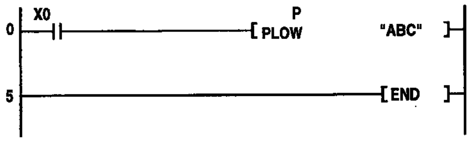
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The program with the designated file name does not exist. (Error code: 2410)
- There is a CHK instruction contained within the program whose file name has been designated. (Error code: 4235)

Program Example

(1) The following program places the program with the file name ABC in the low-speed execution state when X0 goes ON.

[Ladder Mode]



[List Mode]

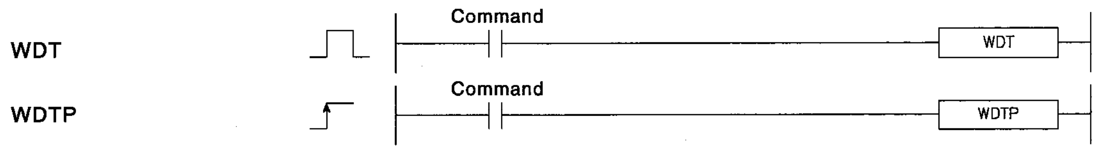
Steps	Instruction	Device
0	LD	X0
1	PLOWP	"ABC"
5	END	

7.18 Other Instructions

7.18.1 Resetting watchdog timer (WDT)

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
—	—								

[Instruction Symbol] [Execution Condition]

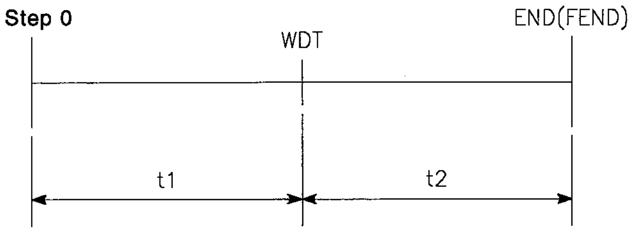


Functions

(1) Resets watchdog timer during the execution of a sequence program.

(2) Used in cases where the scan time exceeds the value set for the watchdog timer due to prevailing conditions.
If the scan time exceeds the watchdog timer setting value on every scan, change the watchdog timer settings at the peripheral device parameter settings.

(3) Take care that neither t1 from step 0 through the WDT instruction, nor t2 from WDT to the END (FEND) instruction exceeds the watchdog timer's setting value.



(4) The WDT instruction can be used two or more times during a single scan, but care should be taken in such cases, because of the time required until the output goes OFF during the generation of an error.

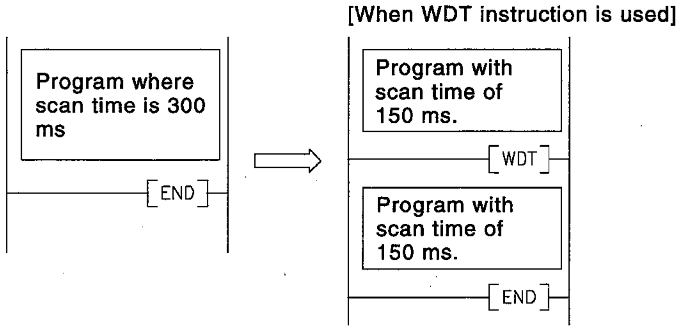
(5) Scan time values stored at the special register will not be cleared even if the WDT or WDTP instruction is executed.
Accordingly, there are times when the value for the scan time for the special register is greater than the value of the watchdog timer set at the parameters.

Operation Errors

(1) There are no operation errors associated with the WDT(P) instruction.

Program Example

(1) The following program has a watchdog timer setting of 200 ms, when due to the execution conditions program execution requires 300 ms from step 0 to the END (FEND) instruction.



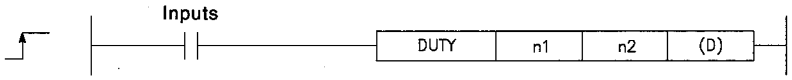
7.18.2 Timing pulse generation

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n1	o	o							—
n2	o	o							—
(D)	o*	—							—

*: Can only be used from SM420 through SM424 and SM430 through SM434

[Instruction Symbol] [Execution Condition]

DUTY

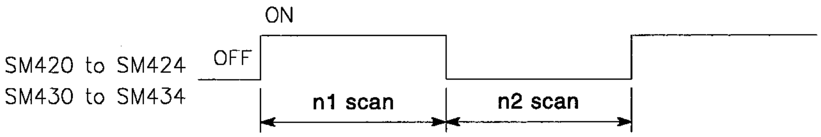


Set Data

Set Data	Meaning	Data Type
n1	Number of scans forcing ON	BIN 16 bits
n2	Number of scans forcing OFF	
(D)	Timing clock for use by user (SM420 through SM424, SM430 through SM434)	Bit

Functions

(1) Turns scan designated by n1 ON for user timing block designated by (D) (SM420 through SM424, SM430 through SM434), and scan designated by n2 OFF.



(2) Scan execution type programs use SM420 through SM424, and low-speed execution type programs use SM430 through SM434.

(3) The following will take place if both n1 and n2 have been set for 0:

(a) n=0......... SM420 through SM424 and SM430 through SM434 will stay OFF.

(b) n>0, n2=0 ... SM420 through SM424 and SM430 through SM434 will stay ON.

(4) The data designated by n1, n2, and (D) is registered with the system when the DUTY instruction is executed, and the timing pulse is turned ON and OFF by END processing.

Operation Errors

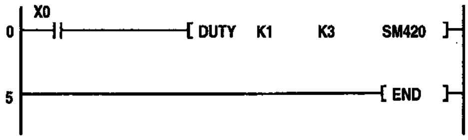
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The device designated by (D) is not from SM420 through SM424 or SM430 through SM434. (Error code: 4101)
- The values of n1 and n2 are less than 0. (Error code: 4100)

Program Example

(1) The following program turns SM420 ON for 1 scan, and OFF for 3 scans if X0 is ON.

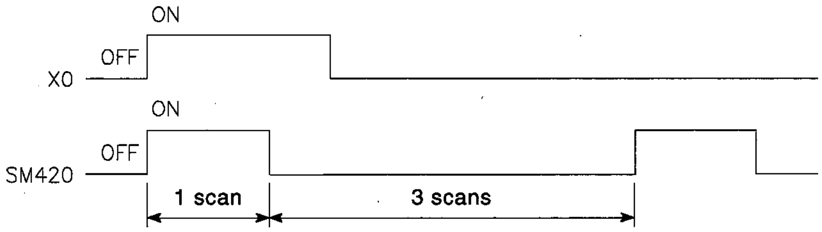
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	DUTY	K1 K3 SM420
5	END	

[Operation]



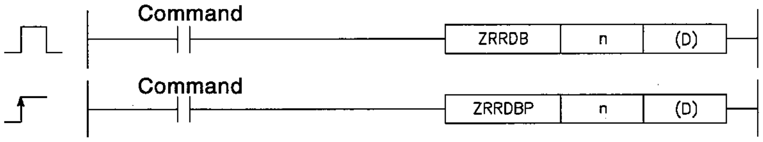
7.18.3 Direct 1-byte read from file register

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n	o							o	—
(D)	o							—	—

[Instruction Symbol] [Execution Condition]

ZRRDB

ZRRDBP

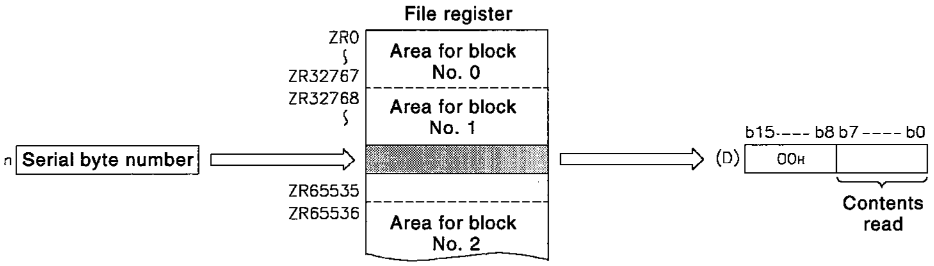


Set Data

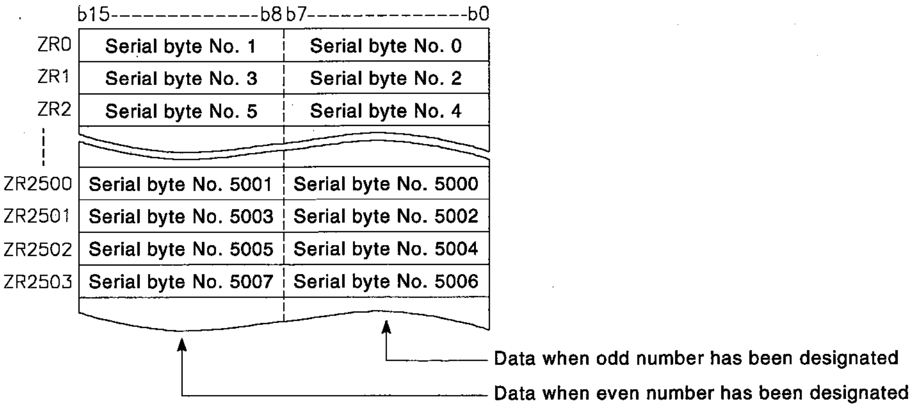
Set Data	Meaning	Data Type
n	Serial byte number for file register to be read	BIN 32 bits
(D)	Number of device that will store the read data	BIN 16 bits

Functions

(1) Reads the serial byte number designated by n that does not signify a block number, and stores at the lower 8 bits of the device designated by (D).
The upper 8 bits designated by (D) will become 00_H.

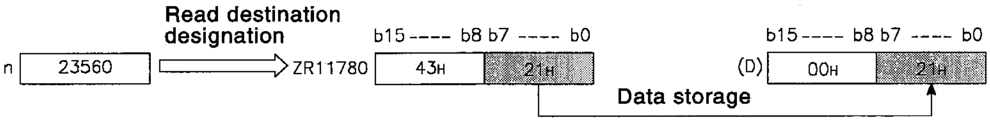


(2) The correspondence between file register numbers and serial byte numbers is as indicated below:

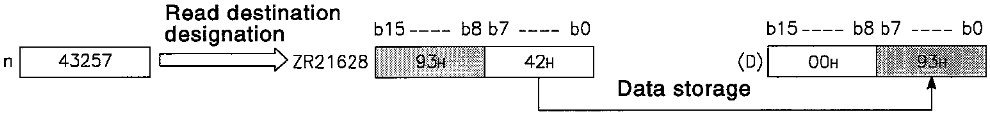


	b15 ---- b8	b7 ---- b0
ZR0	Serial byte No. 1	Serial byte No. 0
ZR1	Serial byte No. 3	Serial byte No. 2
ZR2	Serial byte No. 5	Serial byte No. 4
ZR2500	Serial byte No. 5001	Serial byte No. 5000
ZR2501	Serial byte No. 5003	Serial byte No. 5002
ZR2502	Serial byte No. 5005	Serial byte No. 5004
ZR2503	Serial byte No. 5007	Serial byte No. 5006

(a) If the value of n has been designated as 23560, the data at the lower 8 bits of ZR11780 will be read.



(b) If the value of n has been designated as 43257, the data at the upper 8 bits of ZR21628 will be read.



Operation Errors

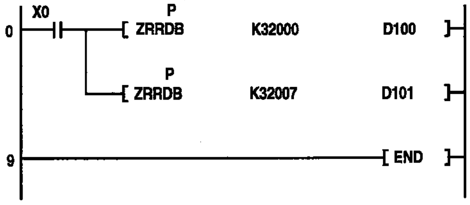
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- A device number (serial byte number) that exceeds the range of allowable designations has been designated. (Error code: 4101)

Program Example

(1) The following program reads the lower bits of ZR16000 and the upper bits of R16003, and stores results at D100 and D101 when X0 is ON.

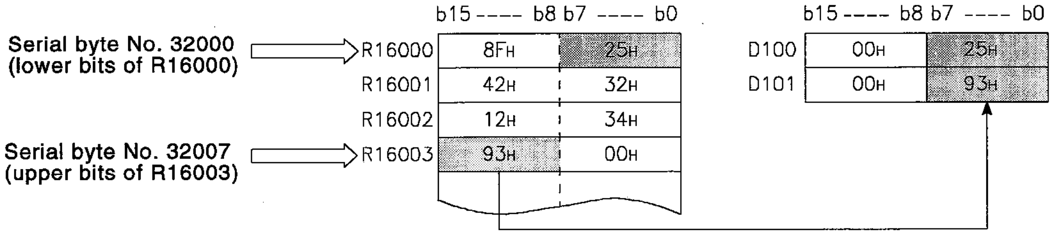
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ZRRDBP	K32000 D100
5	ZRRDBP	K32007 D101
9	END	

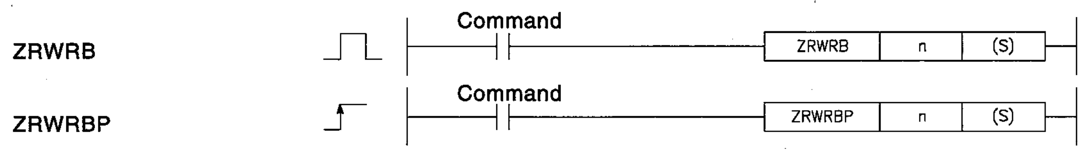
[Operation]



7.18.4 File register direct 1-byte write

<table>
<tr><th rowspan="3">Set Data</th><th colspan="9">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System, User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th rowspan="2">Constant K, H</th><th rowspan="2">Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th></tr>
<tr><td>n</td><td colspan="8">o</td><td>—</td></tr>
<tr><td>(S)</td><td colspan="8">o</td><td>—</td></tr>
</table>

[Instruction Symbol] [Execution Condition]

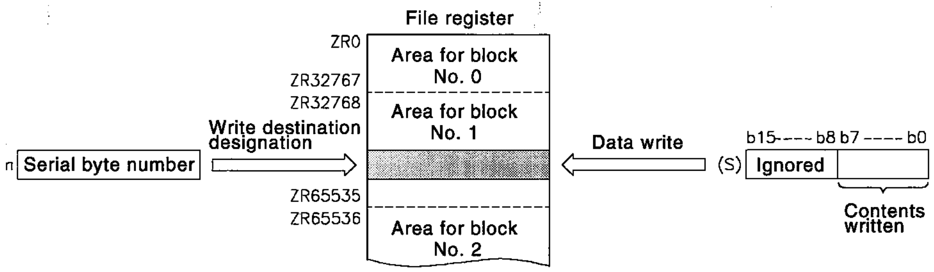


Set Data

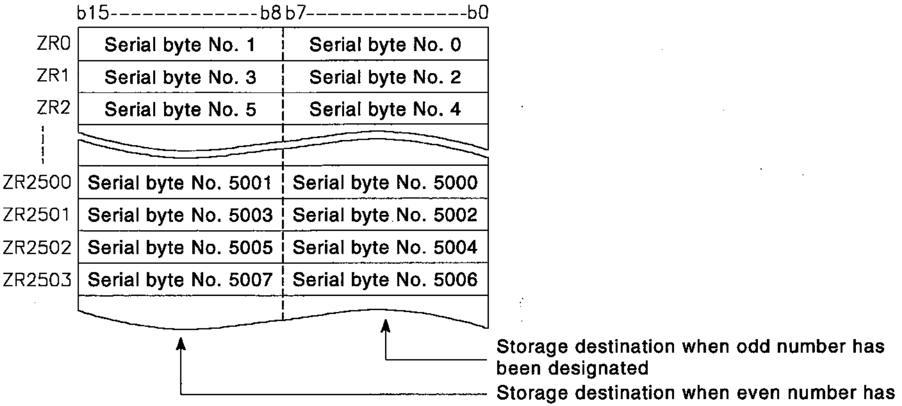
Set Data	Meaning	Data Type
n	Serial byte number of file register to be written	BIN 32 bits
(S)	Device number where data to be written is being stored	BIN 16 bits

Functions

(1) Writes the lower bits of data stored in the device designated by (S) that does not signify a block number to the file register of the serial byte number designated by n.
The upper 8 bits of data in the device designated by (S) are ignored.



(2) The correspondence between file register numbers and serial byte numbers is as indicated below:

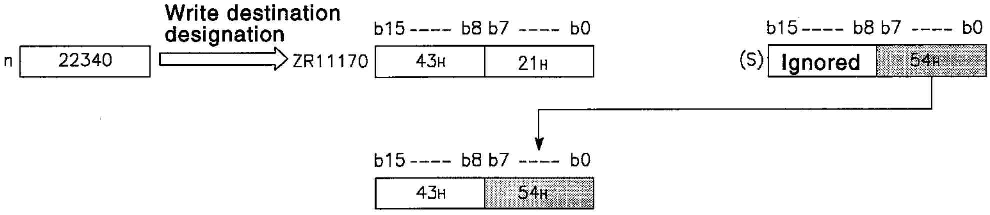


	b15------b8	b7------b0
ZR0	Serial byte No. 1	Serial byte No. 0
ZR1	Serial byte No. 3	Serial byte No. 2
ZR2	Serial byte No. 5	Serial byte No. 4
⋮		
ZR2500	Serial byte No. 5001	Serial byte No. 5000
ZR2501	Serial byte No. 5003	Serial byte No. 5002
ZR2502	Serial byte No. 5005	Serial byte No. 5004
ZR2503	Serial byte No. 5007	Serial byte No. 5006

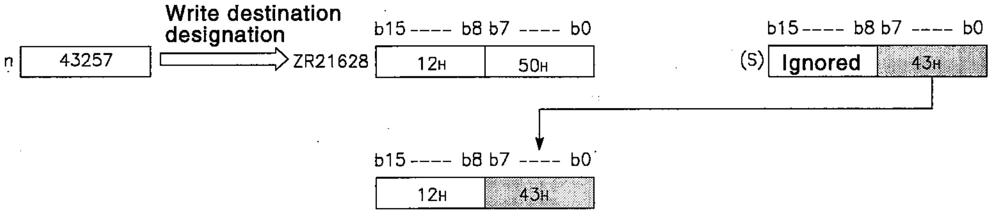
Storage destination when odd number has been designated

Storage destination when even number has been designated

If n has designated a value of 22340, the lower 8 bits of ZR11170 will be written to.



When n has designated a value of 43257



Operation Errors

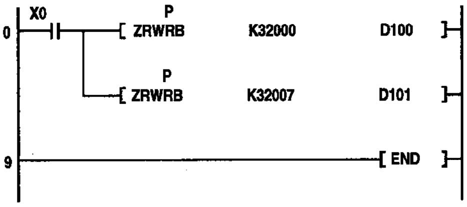
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- A device number (serial byte number) that exceeds the range of allowable designations has been designated. (Error code: 4101)

Program Example

(1) The following program writes the data at the lower bits of D100 and D107 to the lower bits of R16000 and the upper bits of R16003 when X0 goes ON.

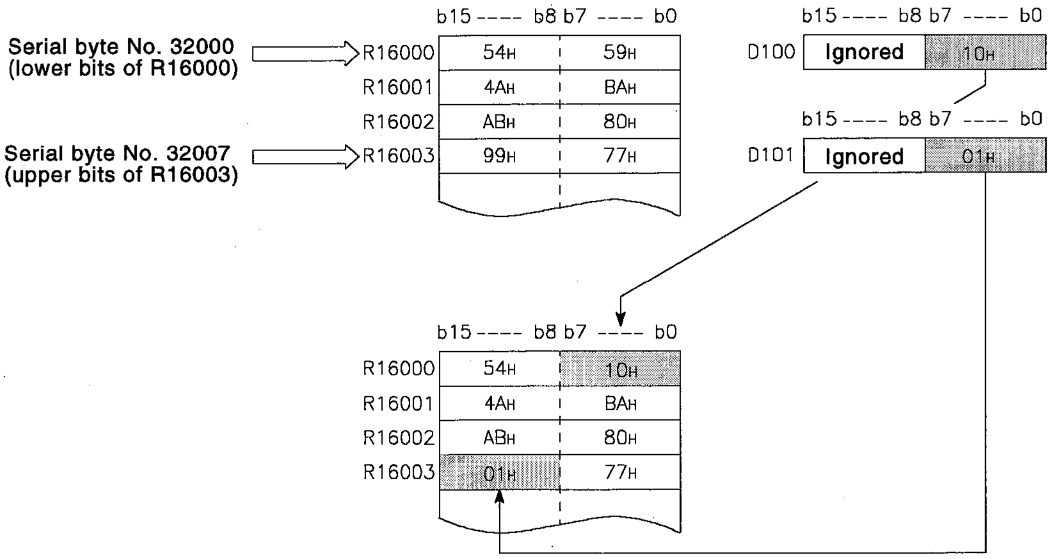
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ZRWRBP	K32000 D100
5	ZRWRBP	K32007 D101
9	END	

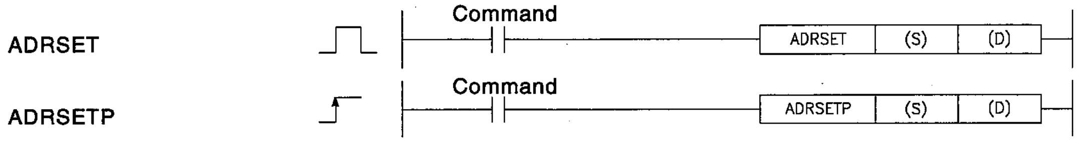
[Operation]



7.18.5 Indirect address read operations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	o			—					
(D)	o			—					

[Instruction Symbol] [Execution Condition]

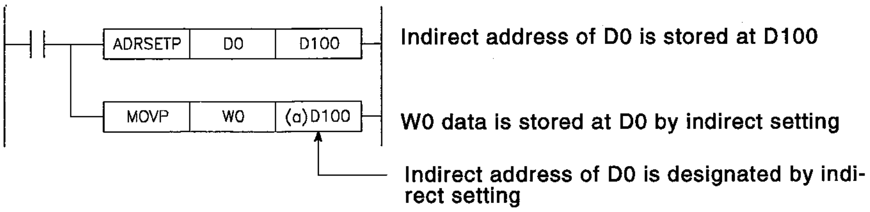


Set Data

Set Data	Meaning	Data Type
(S)	Number of device for indirect address read	Device name
(D)	Number of device that will store the indirect address of the device designated by (S)	BIN 32 bits

Functions

(1) Stores the indirect address of the device designated by (S) at (D)+1 and (D).
The address stored at the device designated by (D) is used when an indirect device address is performed by the sequence program.



(2) A bit device designation cannot be made at (S).

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- A device for which designation is not allowable has been designated. (Error code: 4100)

REMARK

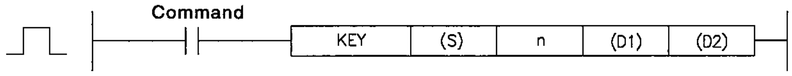
* : See Section 3.4 for more information on indirect settings.

7.18.6 Numerical key input from keyboard

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
(S)	o (Only x)	—		—			—		—
n	o	o		o			o		—
(D1)	—	o		—			—		—
(D2)	o	o		o			—		—

[Instruction Symbol] [Execution Condition]

KEY

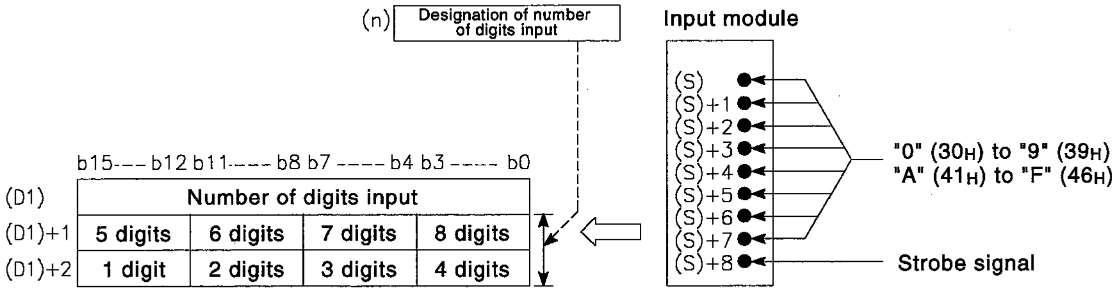


Set Data

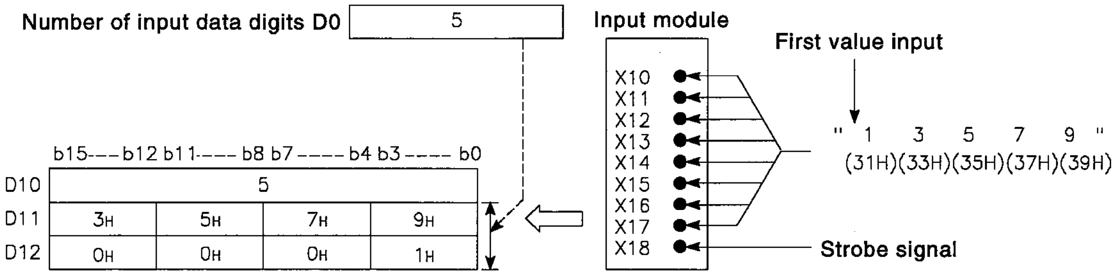
Set Data	Meaning	Data Type
(S)	First device number of devices (X) where numerical input will be made	Bit
n	Number of digits of numerical input to be made	BIN 16 bits
(D1)	First device number of devices that will store numbers input	BIN 16 bits
(D2)	Number of bit device to be turned ON at completion of input	Bit

Functions

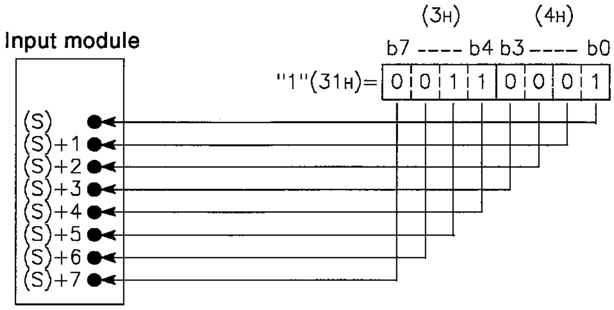
(1) Fetches ASCII data from the 8 points of input (X) designated by (S), converts it to hexadecimal values and stores the result at devices starting at that designated by (D1).



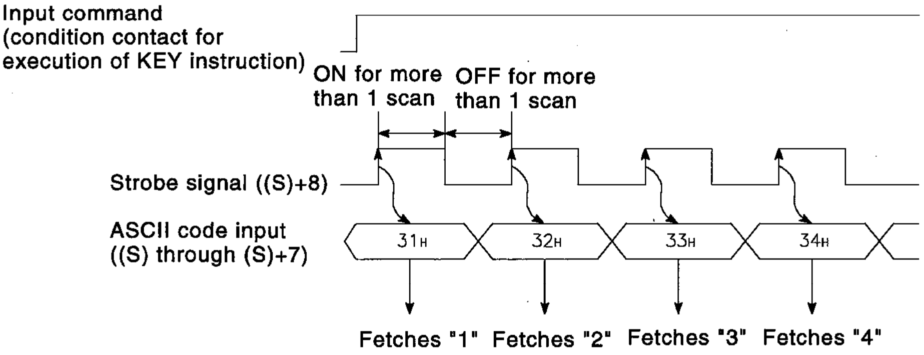
For example, in a case where the number of digits (n) has been set at 5, and the values "31", "33", "35", "37" and "39" have been input through X10 through X18 of the input module, the following will take place:



(2) Numerical input to input (X) designated by (S) undergoes bit development at (S) through (S)+7 and is input as the ASCII code corresponding to the numbers.
ASCII code which can be input is from 30H (0) through 39H (9), and from 41H (A) through 46H (F).

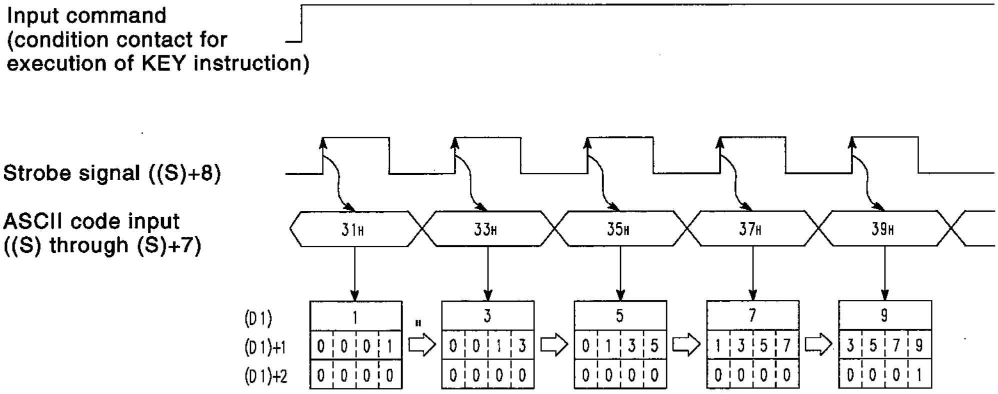


(3) After ASCII code is input to (S) through (S)+7, the strobe signal at (S)+8 goes ON to incorporate the designated numbers internally.
The strobe signal should be held at its ON or OFF state for more than one scan of the sequence program.
If this time is less than 1 scan, there will be cases when the data is correctly incorporated.



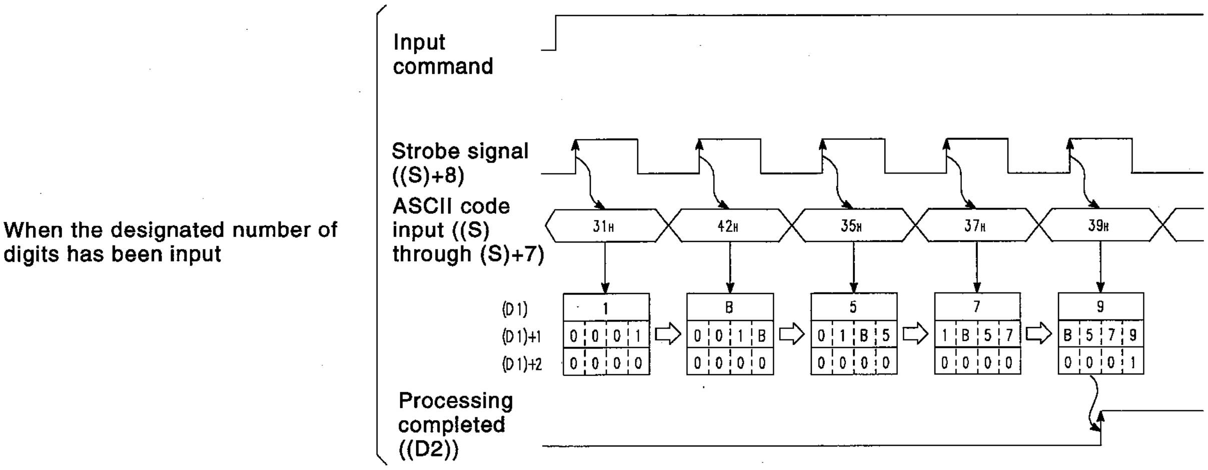
(4) The input command (contact for conditions for execution of KEY instruction) should always be left ON until the designated number of digits has been input.
If the input command is turned OFF, the KEY instruction cannot be executed.

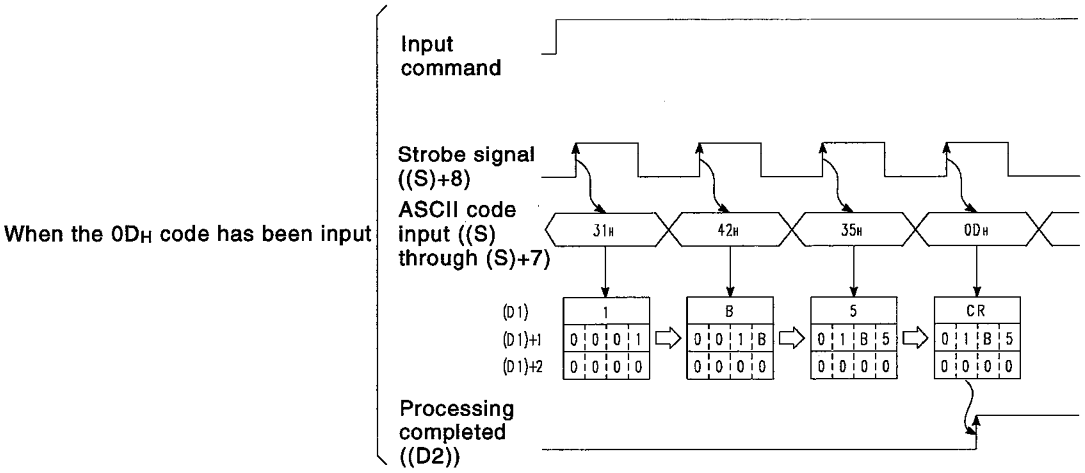
(5) The digits for the numbers actually fetched to (D1) will be stored at the device designated by (D1), and these will be converted to the ASCII codes input at (D1)+1 and (D1)+2, converted to hexadecimal BIN values, and stored.



(6) The number of digits that can be designated by (n) is from 1 to 8.

(7) The incorporation of input data internally is brought to an end when the number of digits designated by (n) has been input, or when the "00H" code is input, and the bit device designated by (D2) will go ON at such times.
For example, the operations at the location designated if n=5 will be as indicated below:





If input processing is to be performed a second time, it is necessary to clear the number of digits input and the input data stored at (D1), and to run a user program that will turn the bit device designated by (D2) OFF. If (D1) is not cleared and (D2) not turned OFF, the next input processing cannot be performed.

Operation Errors

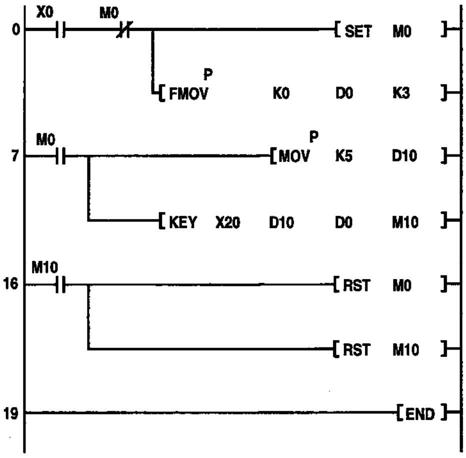
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The device designated by (S) is not an input (X) device. (Error code: 4100)
- The number of digits designated by n are outside the range of from 1 through 8. (Error code: 4100)

Program Example

(1) The following program incorporates data from the 5 or fewer digits from the connected numerical key pad connected to X20 through X28, and stores it starting from D0 when X0 goes ON.

[Ladder Mode]



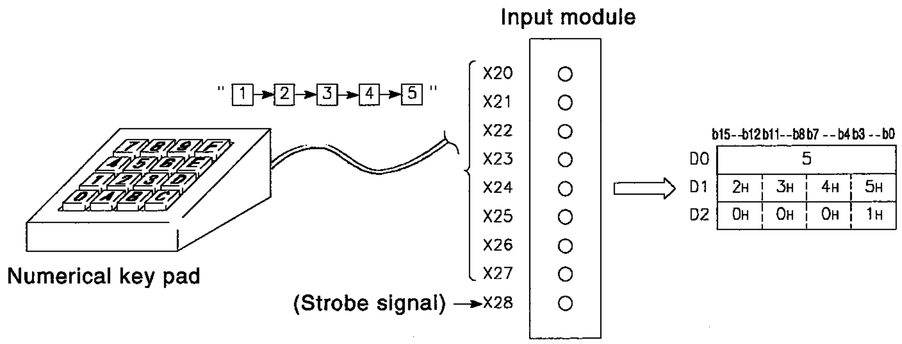
Clear input data from previous run

Set number of digits for input

Reset data input completion flag

[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ANI	M0
2	SET	M0
3	FMOVP	K0 D0 K3
7	LD	M0
8	MOVP	K5 D10
11	KEY	X20 D10 D0 M10
16	LD	M10
17	RST	M0
18	RST	M10
19	END	

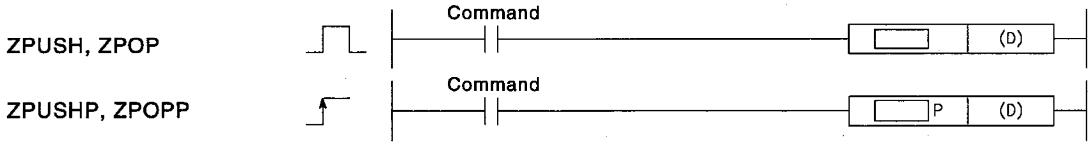


	b15--b12	b11--b8	b7--b4	b3--b0
D0	5			
D1	2H	3H	4H	5H
D2	0H	0H	0H	1H

7.18.7 Batch save or recovery of index register

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J◻\◻		Special Function Module U◻\G◻	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(D)	—	○		—					

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
(D)	First device number of devices to save index register	BIN 16 bits

Functions

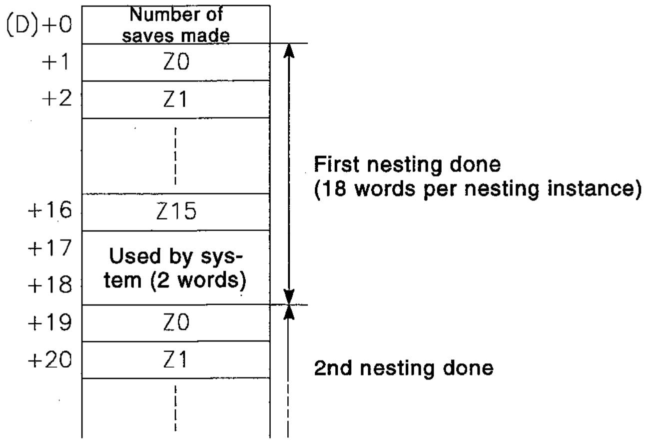
ZPUSH

(1) Saves the contents of index registers Z0 through Z15 to a location following the device designated by (D).

(2) The ZPOP instruction is used for data recovery. Nesting is possible within the ZPUSH to ZPOP cycle.

(3) If nesting has been done, each time the ZPUSH instruction is executed, the field used following (D) will be added to, so a field large enough to accommodate the number of times the instruction will be used should be maintained from the beginning.

(4) The composition of the field used following (D) is as shown below:



ZPOP

(1) The data saved to the field following the device designated by (D) is read to the index register Z0 through Z15.

Operation Errors

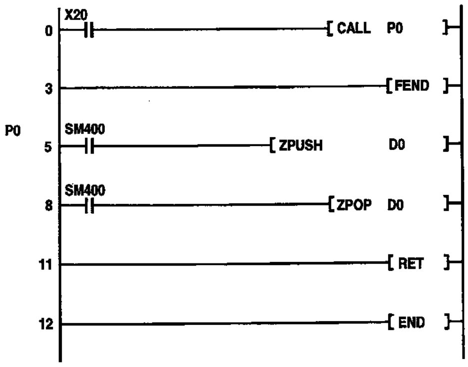
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range for the number of points to be used following (D) exceeds the relevant device range. (Error No.: 4101)
- The contents of (D)+0 (number of saves made) are 0. (Error No. 4100)

Program Example

(1) The following program saves the contents of the index register to the fields following D0 before calling the sub-routine following P0 that uses the index register.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X20
1	CALL	P0
2	FEND	
3		P0
4	LD	SM400
5	ZPUSH	D0
6	LD	SM400
8	ZPOP	D0
9	RET	
11	END	
12		

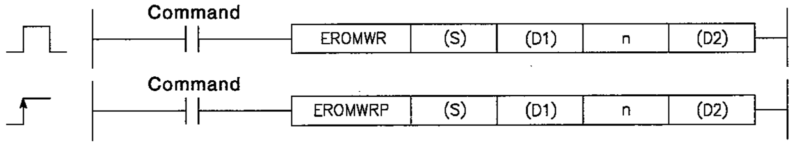
7.18.8 Batch write operation to EEPROM file register

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				U
(S)	—	o	—	—					—
(D1)	—	—	o	—					—
n	o	o	o	o					—
(D2)	o	—	—	—					—

[Instruction Symbol] [Execution Condition]

EROMWR

EROMWRP



Set Data

Set Data	Meaning	Data Type
(S)	First device number of devices where the write data is being stored	BIN 16 bits
(D1)	First device number of devices of write destination file register	
n	Number of words to write	
(D2)	Completion bit device	Bit

Functions

(1) Writes n-words of data from the device designated by (S) in respect to the file register of the E^2PROM drive designated by (D1).

(2) When the write operation has been completed, the bit device designated by (D2) is turned ON, and after 1 scan is automatically turned OFF again.

(3) The data write operation is conducted by END processing, with 64 words written during each process cycle.
For this reason, the write operation for the designated number of words will require n divided by 64 scans (fractions rounded up).
Further, note that scan time during processing is lengthened by approximately 10 ms.

(4) Do not update the data following (S) during write processing.
If the data following (S) is updated during processing, some data can be lost.

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The range designated by n exceeds the relevant device range of (S) or of (D1). (Error No.: 4101)
- The file register designated by (D1) does not exist, or is not an E^2PROM file register. (Error No.: 4101)

8. INSTRUCTIONS FOR DATA LINK

The QnACPU can be used to enact data links in MELSECNET/10 and MELSECNET systems. *

The instructions for data link are those instructions that enable the QnACPU to read data to the host station from other stations connected in a MELSECNET/10 or MELSECNET system.

(1) Types of instructions for data link
Instructions for data link include refresh instructions, QnA link dedicated instructions, and A-series compatible link instructions.

- Refresh instructions
  : Conduct refresh operations for designated network modules.
- QnA link dedicated instructions
  : New instructions for data link featured with QnACPU.
  The eight channels from channel 1 through channel 8 on the network module can be used to conduct communications.
  Fine controls can be implemented from the control data area.
- A-series compatible link instructions
  : Identical instructions to the AnACPU/AnUCPU dedicated instructions.

The instructions for data link that can be used with MELSECNET/10 and MELSECNET are determined.
Further, with MELSECNET/10, the instructions that can be used differ according to whether the CPU module at the object station is a QnACPU, ACPU, or remote I/O station.

Table 8.1 Instructions For Data Link

Instruction Name		Meaning	MELSECNET/10 Object Station			MELSECNET	Reference Section
			QnACPU	ACPU	Remote I/O Station		
Refresh instructions	ZCOM	Conducts refresh operations for designated network modules.	—	—	—	—	Section 8.1
QnA link dedicated instructions	READ	Reads QnACPU device data from object station of object network number	o	—	—	—	Section 8.2.1
	SREAD						Section 8.2.2
	WRITE	Writes QnACPU device data for object station of object network number	o	—	—	—	Section 8.2.3
	SWRITE						Section 8.2.4
	SEND	Writes data to object station (network module) of object network number	o	—	—	—	Section 8.2.5
	RECV	Reads data transmitted by the SEND instruction at the QnACPU	o	—	—	—	Section 8.2.6
	REQ	Reads/write clock data to the other station and conducts remote RUN/STOP.	o	—	—	—	Section 8.2.7
	ZNFR	Reads data from special function module at remote I/O station	—	—	o	—	Section 8.2.8
	ZNTO	Writes data to special function module at remote I/O station	—	—	o	—	Section 8.2.9

Instruction Name		Meaning	MELSECNET/10 Object Station			MELSECNET	Reference Section
			QnACPU	ACPU	Remote I/O Station		
A-series compatible link instructions	ZNRD	Reads QnACPU device data from object station of object network number	o	o	—	—	Section 8.3.1
		Reads local station device data (Valid only at master station)	—	—	—	o	Section 8.3.2
	ZNWT	Writes QnACPU device data for object station of object network number	o	o	—	—	Section 8.3.3
		Writes data to device at local station (Valid only at master station)	—	—	—	o	Section 8.3.4
	RFRP	Reads data from special function module at remote I/O station	—	—	—	o	Section 8.3.5
	RTOP	Writes data to special function module at remote I/O station	—	—	—	o	Section 8.3.6

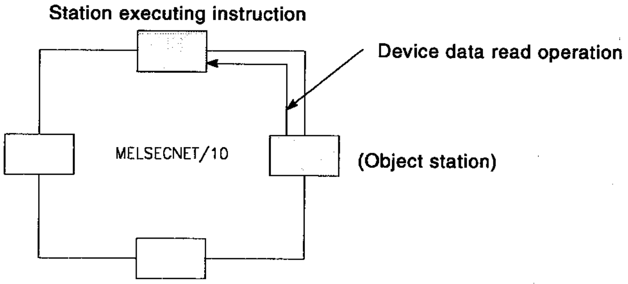
REMARK

* : The terms MELSECNET/10 and MELSECNET indicate the following
MELSECNET/10 .... The MELSECNET/10 Network System
MELSECNET..................... The MELSECNET (II,/B) Data Link System

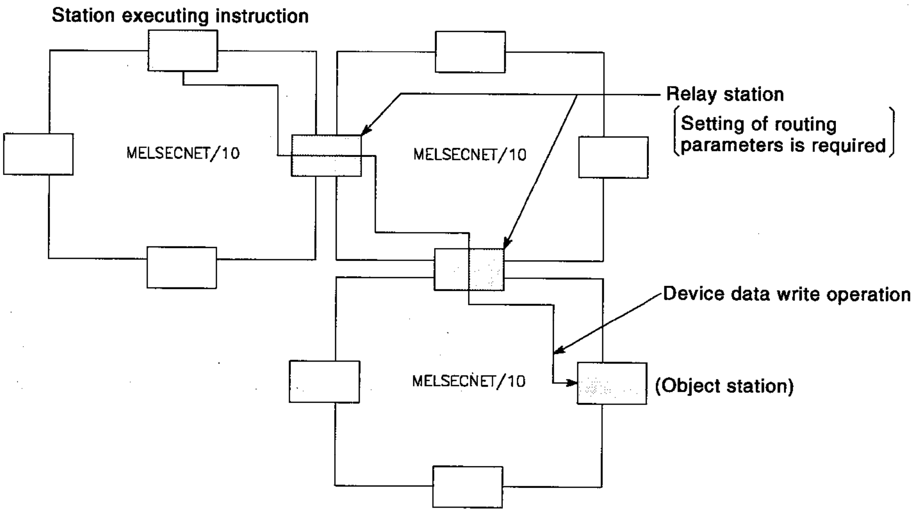
(2) Data read/write ranges

(a) With MELSECNET/10, a host station can perform read/write operations not only with stations connected to its own network, but also with stations connected to a designated network number.

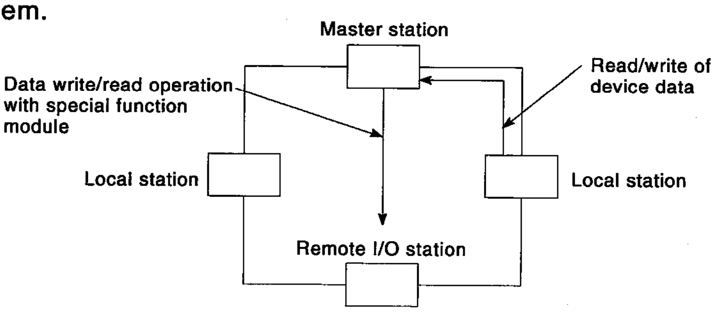
1) Read/Write operations with stations in the same network as the host station
Make the network number for the object station identical to the network number for the network module the host is connected to. This function is used to write or read device data from the host station to any other station at the object network number.



2) Read/Write operations with a station connected to a specified network number
The network number for the object station should be other than the network number used for the network module of the host station.
A station connected to the same network as the host station is used as a relay station to forward on read/write operations to a device at a station connected to a specified network No.



(b) Write and read operations to a remote station or local station can be conducted from the master station with the MELSECNET (II/B) system.

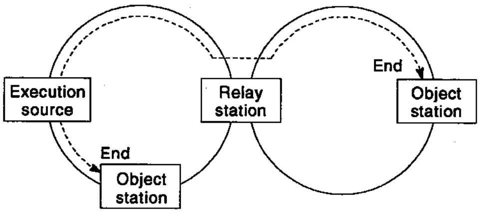


(3) QnA link dedicated instructions
The following are some points of consideration when using the dedicated QnA link instructions

(a) Simultaneous execution of instructions for data link
MELSECNET/10 network modules have eight areas for communications that use instructions for data link.
It is not possible to execute multiple instructions for data link with respect to one communications area at the same time with a MELSECNET/10 network module.
If the same communications area at the QnACPU is used to execute more than one instruction for data link, this should be done with programs that will be executed progressively using the completion device for the individual instructions for data link.

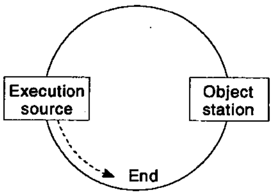
(b) Transmission completion confirmation
When conducting data transmissions with QnA dedicated link instructions, it is possible to select confirmation or no confirmation of transmission completion.
(When conducting read operations with QnA dedicated link instructions, only completion confirmation can be selected.)

- Transmission completion confirmation
  : This is the instruction completion when data is being written to a designated channel of a designated object station.

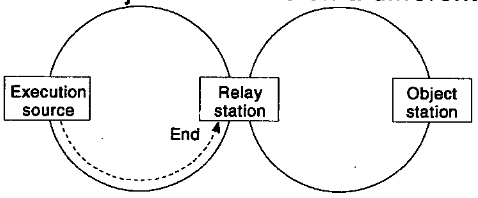


- No transmission completion confirmation
  : When the object station is on the same network as the host station, the instruction completion when data is transmitted from the host station.
  Further, if the object station is on another network, this is the instruction completion when the data has arrived at a relay station on the host's network.

[When object station is on the host's network]



[When the object station is on a different network]

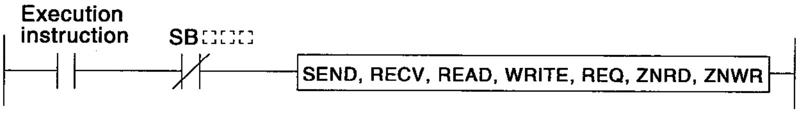


POINTS

(1) It is recommended that the transmission completion confirmation be set and the instruction executed in order to improve data integrity.

(2) If no confirmation has been selected, the transmitting station will experience a normal completion so long as the transmission itself ends normally, even if there are errors in the transmitted data.
Further, the "reception buffer full" error will be issued by the object station when the instruction is executed from more than one station with respect to the same station, even if the content of the transmitted data is correct.
However, the transmitting station will have a normal completion in this case.

(4) Network module channels
The network module has eight channels to execute instructions. Eight channels can be used at once, but the same channel cannot be used by multiple instructions.
Create a program performing interlocking with link special relay (SB) so that multiple instructions cannot be executed.

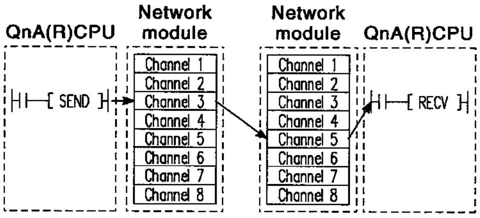
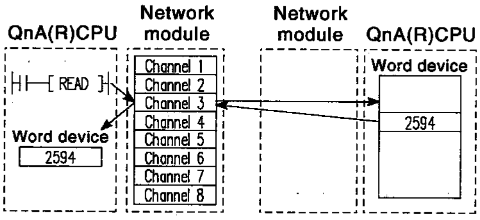
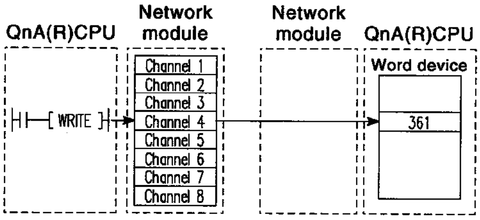
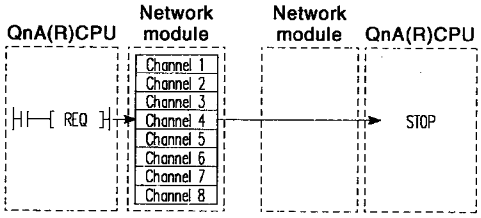


The interlock signals for each instruction is shown below:

Instruction	ZNRD*1	ZNWR*2	—	—	—	—	—	—
	SEND, RECV, READ, WRITE, and REQ Instructions							
	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8
1st	SB030	SB032	SB034	SB036	SB038	SB03A	SB03C	SB03E
2nd	SB230	SB232	SB234	SB236	SB238	SB23A	SB23C	SB23E
3rd	SB430	SB432	SB434	SB436	SB438	SB43A	SB43C	SB43E
4th	SB630	SB632	SB634	SB636	SB638	SB63A	SB63C	SB63E

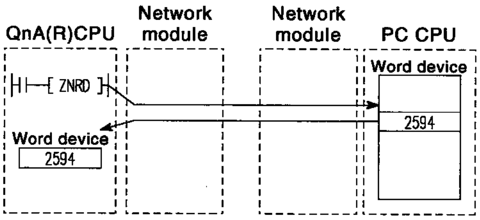
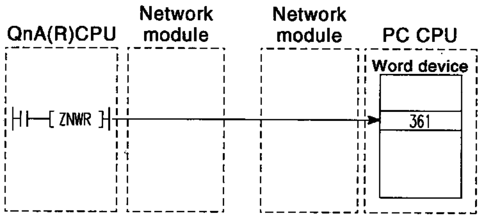
*1...... ZNRD always uses channel 1.
*2...... ZNWR always uses channel 2.

This section describes the instructions that can be used with MELSEC-NET/10. The overview of each instruction is shown below.

Instruction	Details	Instruction Execution Station (Host)	Target Station		
		Station Type	Station Type	PC CPU Type	
				QnA(R) CPU	Other than QnA(R) CPU
SEND RECV	Data is sent (SEND) and received (RECV) between the QnA(R)CPU stations.  	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	o	x
READ SREAD	Reads data from another station's word device. (With SREAD, device on target station can be turned on.)  	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	o	x
WRITE SWRITE	Writes data to another station's word device. (With SWRITE, device on target station can be turned on.)  	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	o	x
REQ	Perfrom "remote RUN/STOP" "clock data read and write" for other stations.  	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	o	x

* Channels 1 to 8 are common areas for SEND/RECV/READ/WRITE/REQ instructions.

There are no operation differences in the instruction fromat JP.□ and GP.□, and J.□ and G.□.

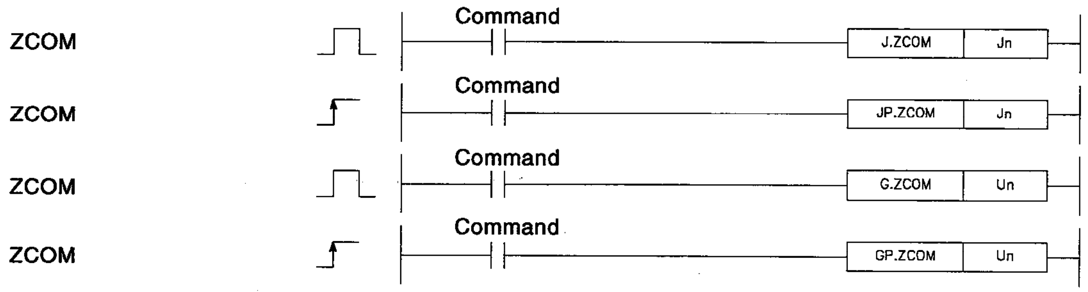
Instruction	Details	Instruction Execution Station (Host)	Target Station		
		Station Type	Station Type	PC CPU Type	
				QnA(R) CPU	Other than QnA(R) CPU
ZNRD	Read data from another station's word device.  	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	○	○
ZNWR	Write data to another station's word device.  	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	Control station Normal station Remote master station Multiple remote master station Parallel remote master station Multiple remote submaster station Parallel remote submaster station	○	○

8.1 Network Refresh Instruction

8.1.1 Network refresh

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
□	—								

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
Jn	Network number for host station	BIN 16 bits
Un	Head I/O number of host station network	

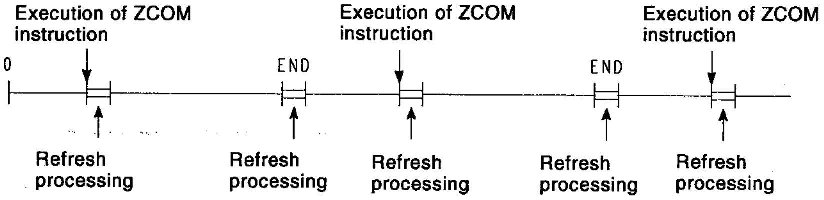
(1) The ZCOM instruction is used in the following cases.

(a) To speed up communications processing with remote I/O stations

(b) When it is desirable to stabilize data exchanges with another station that has different scan times during the execution of a data link

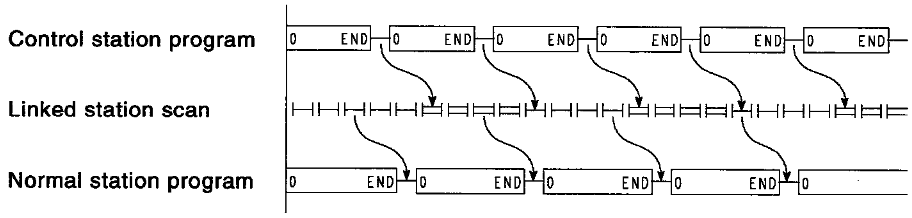
Functions

(1) When the ZCOM instruction is executed, the QnACPU temporarily suspends processing of the sequence program and conducts refresh processing of the network modules designated by Jn/Un.

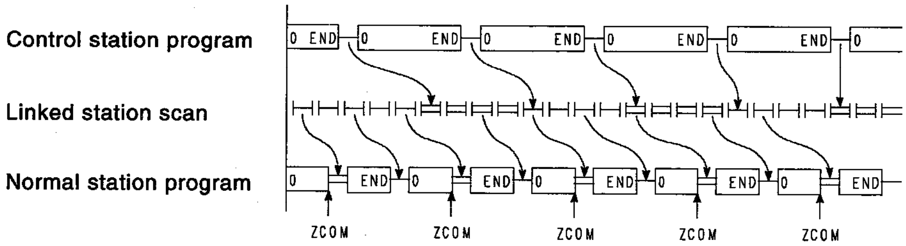


(2) In cases where the scan time for the host station's sequence program is longer than the scan time for the other station, the ZCOM instruction is used to ensure that the data from the other station has been properly incorporated.

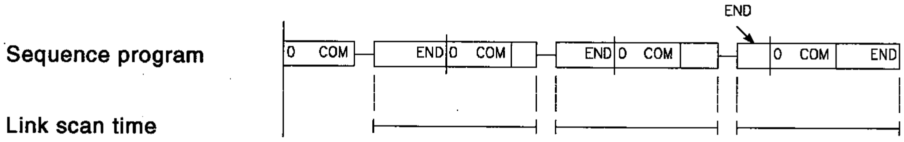
1) Example of data communications when the ZCOM instruction is not used



2) Example of data communications when ZCOM instruction has been used



(3) In cases where the link scan time is longer than the sequence program scan time, data communications will not be facilitated even if the ZCOM instruction is used.



(4) The ZCOM instruction can be used as often as desired in sequence programs.
However, note that each time a refresh operation is conducted will lengthen the sequence program scan time by the amount of time required for the refresh operation.

(5) The common data processing operations noted below are not conducted with the ZCOM operation.

(c) Exchange of signals between the QnACPU and the peripheral device

(d) Monitor processing at the other station

(e) Read processing of the buffer memory of special function modules other than the serial communications module.

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The designated network number is not connected to the host station. (Error code: 4102)
- The module for the designated I/O number is not a network unit or link unit. (Error code: 2111)

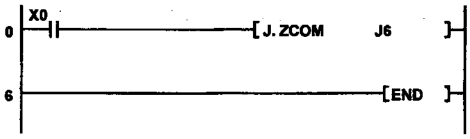
POINT

To conduct only common data processing, use the COM instruction (see Section 7.6.8)

Program Example

(1) The following program conducts a link refresh operation for the network module of network No. 6 while X0 is ON.

[Ladder Mode]

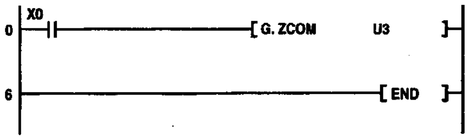


[List Mode]

Steps	Instruction	Device
0	LD	X0
1	J. ZCOM	J6
6	END	

(2) The following program conducts a link refresh operation for the network module mounted at I/O numbers X/Y30 through X/Y4F while X0 is ON.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	G. ZCOM	U3
6	END	

8.2 Instructions Dedicated to QnA Links

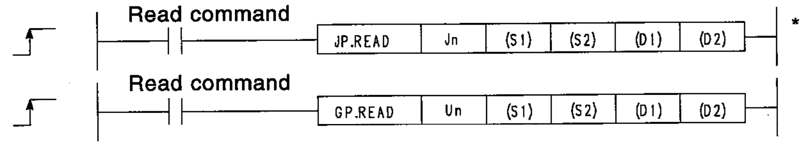
8.2.1 Reading word device data from another station

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—					
(S2)	—	o		—					
(D1)	—	o		—					
(D2)	o	o		—					

[Instruction Symbol] [Execution Condition]

READ

READ



* Cannot be used with CC-Link.

Set Data

[For MELSECNET/10, AJ71QC24]

Set Data	Meaning	Data Type
Jn	Network number for host station [*1]	BIN 16 bits
Un	Head I/O number for host station of network module [*2]	
(S1)	First device of host station storing control data	Device name
(S2)	First device of station where read data is being stored	
(D1)	First device of host station storing read data	
(D2)	Device turned ON for 1 scan on completion of instruction [*3]	Bit

REMARKS

1) *1 : The network number for the host station should be designated as follows.
   - 1 to 239 : Network No.
   - 254 : Network designated by setting for valid module for access from other stations.

2) *2 : The designation for the head I/O number for the host station network module can be from 0 to FEH.

3) *3 : Local device and file refistor for each program cannot be used as a device to use for the setting data.
(See QnACPU Programming Manual (Special Function Module) for details.)

POINTS

(1) The READ instruction can be executed only when the object station is a QnACPU.
(The READ instruction cannot be executed at an ACPU connected to MELSECNET/10.)
Only a station number for QnACPU should be set as the object station number.

(2) When reading the data from other station using the READ instruction, the parameter setting for the device must be identical between the host station and the object station.

[For CC-Link]

It can be used with the master module of the software version J or later.

Set Data	Meaning	Data Type
Un	Head I/O number for host station of master/host module *1	BIN 16 bits
(S1)	First device of host station storing control data	Device name
(S2)	First device of station where read data is being stored	
(D1)	First device of host station storing read data	
(D2)	Device turned ON for 1 scan on completion of instruction *2	Bit

REMARKS

*1 : The designation for the head I/O number for the host station master/host module can be from 0 to FEH.

*2 : Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnACPU Programming Manual (Special Function Module) for details)

Control Data

[For MELSECNET/10, AJ71QC24]

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Arrival confirmation made (bit 0=1 (fixed))	0001H 0081H	User
	Error completion type	Sets clock data setting status when error processing is completed. Do not set clock data : set bit 7 (b7) to 0 Set clock data : set bit 7 (b7) to 1 ((S1)+11 onward appended)		
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code *1	—	System
(S1)+2	Channel used by host station	Designates channel used by host station *2	1 to 8	User
(S1)+3	Dummy	Not used	0	—
(S1)+4	Network number of object station	Sets network number for station designated to read device data	1 to 239 254 *3	User
(S1)+5	Object station number	Sets station number for designated station	1 to 64	User
(S1)+6	Dummy	Not used	—	—
(S1)+7	Number of transmission retries	Set the number of repetitions to be made if read operation prompted by READ instruction is not completed within the monitoring time specified by (S1)+8.	1 to 15	User
	Results of retries	Number of repetitions conducted is stored.		System
(S1)+8	Arrival WDT time	Monitoring time for read operations prompted by READ instruction is set in units of seconds. If the read operation has not been completed within the time set, transmission will be retried for the number of times designated by (S1)+7.	1 to 32767 0 : 10 seconds fixed	User
(S1)+9	Receive data length	Set number of data blocks to be read	1 to 480*	User
(S1)+10	Dummy	Not used	—	—
(S1)+11	Clock set flag (set only on error)	Stores clock data enable/disable status When clock data is disabled : 0 When clock data is enabled : 0	—	System
(S1)+12	Clock data (set only on error)	Upper 8 bits : Month (1 to 12), Lower 8 bits : Year (0 to 99)	—	System
(S1)+13		Upper 8 bits : Hour (0 to 23), Lower 8 bits : Day of month (1 to 31)		
(S1)+14		Upper 8 bits : Seconds (0 to 59).Lower 8 bits : Minute (0 to 59)		
(S1)+15		Upper 8 bits : 00H, Lower 8 bits : Day of week (0 to 6)		
(S1)+16	Network number where error was detected	Stores network number of station where error was detected 1 to 239 : Network No.	—	System
(S1)+17	Station number where error was detected	Stores station number of station where error was detected However, does not store data if (S1)+1 completion status is "Channel in Use." 1 to 64 : Station number	—	System

POINT

*: If (Device number specified by (D1)) + (Receive data length) exceeds the device range specified by (D1), the data for the number of points that exceed the specified range are stored from the beginning of the next device.

REMARKS

1) *1 : See the following manual for information on error codes.
MELSECNET/10 for QnA/Q4AR Network System Reference Manual
2) *2 : See Chapter 8 for information on the channels of network modules.
3) *3 : Designated when the value 254 has been set by Jn.

[For CC-Link]

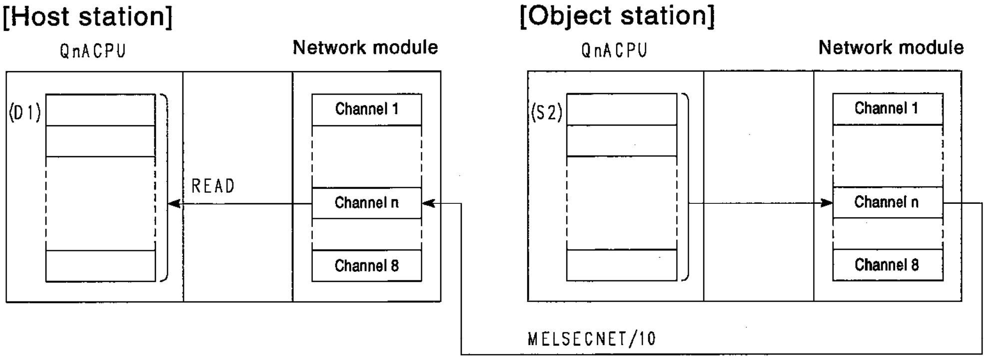
It can be used with the master module of the software version J or later.

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Arrival confirmation made (bit 0=1 (fixed))	1	User
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code	—	System
(S1)+2	Channel used by host station	Designates channel used by host station 1: Use the CH1 side 2: Use the CH2 side	1.2	User
(S1)+3	Dummy	Not used	—	—
(S1)+4	Network number of object station	Fixed to 0	0	User
(S1)+5	Object station number	Fixed to 0	0	User
(S1)+6	Master/host station number	0: Master station specified 1 to 64: Host station specified	0 to 64	—
(S1)+7	Number of transmission retries	Set the number of repetitions to be made if read operation prompted by READ instruction is not completed within the monitoring time specified by (S1)+8.	0 to 15	User
	Results of retries	Number of repetitions conducted is stored.	—	System
(S1)+8	Arrival WDT time	Monitoring time for read operations prompted by READ instruction is set in units of seconds. If the read operation has not been completed within the time set, transmission will be retried for the number of times designated by (S1)+7.	1 to 32767 0 : 10 seconds fixed (Default)	User
(S1)+9	Receive data length	Set number of data blocks to be read	1 to 480	User
(S1)+10	Dummy	Not used	—	—
(S1)+11	Dummy	Not used	—	—
(S1)+12	Dummy	Not used	—	—
(S1)+13				
(S1)+14				
(S1)+15				
(S1)+16	Network number where error was detected	Fixed to 0	—	System
(S1)+17	Station number where error was detected	Fixed to 0	—	System

Functions

[For MESECNET/10, AJ71QC24]

(1) Stores data of devices starting from the word device designated at (S2) of the station connected to MELSECNET/10 as specified by the control data network number and station number, in devices starting from the device designated at (D1) at the host station.
When the device data read operation for the object station has been completed, the completion device designated by (D2) goes ON.



(2) The device data read operation can be carried out with respect to a station connected in a MELSECNET/10 network whose number is designated, in addition to a station connected in the host station's network.

(3) Instructions for data link cannot be executed at more than one location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(4) The execution status of the READ instruction and whether this was completed normally or abnormally can be checked by the communications directive flag* for the channel used, the completion device (D), and the status display at completion device ((D2)+1).

(a) Communications directive flag
: Goes ON during execution of READ instruction, and OFF when OFF at the processing of the END instruction for the scan at completion of the read operation.

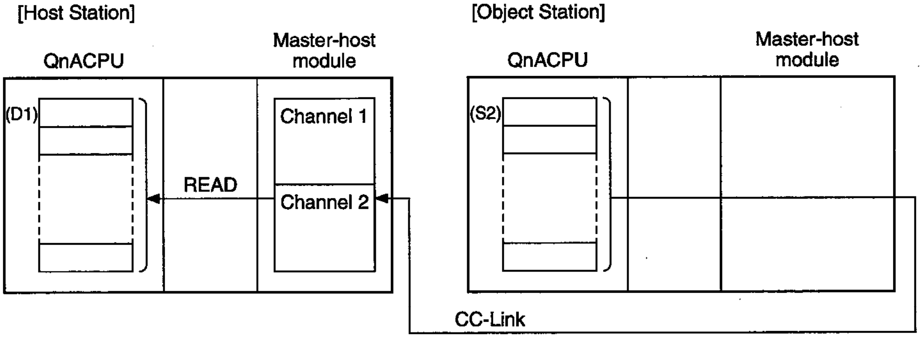
(b) Host station completion device
: Goes ON by END processing for the scan for completion of read operations conducted through the READ instruction, and OFF with the next END processing.

(c) Status display device at completion
: Turned ON and OFF according to the status when the READ instruction is completed.
- Normal completion : Stays OFF, with no changes
- Error completion : Goes ON with processing of the END instruction for the scan at completion of the read operation, and OFF at the processing of the next END instruction.

[For CC-Link]

It can be used with the master module of the software version J or later.

(1) Stores data of devices starting from the word device designated at (S2) of the station connected to the master/host station number as specified by the control data station number, in devices starting from the device designated at (D1) at the host station.
When the device data read operation for the object station has been completed, the completion device designated by (D2) goes ON..



(2) The device data read operation can be carried out with respect to a station connected in CC-Link.

(3) Instructions for data link cannot be executed at more than one location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

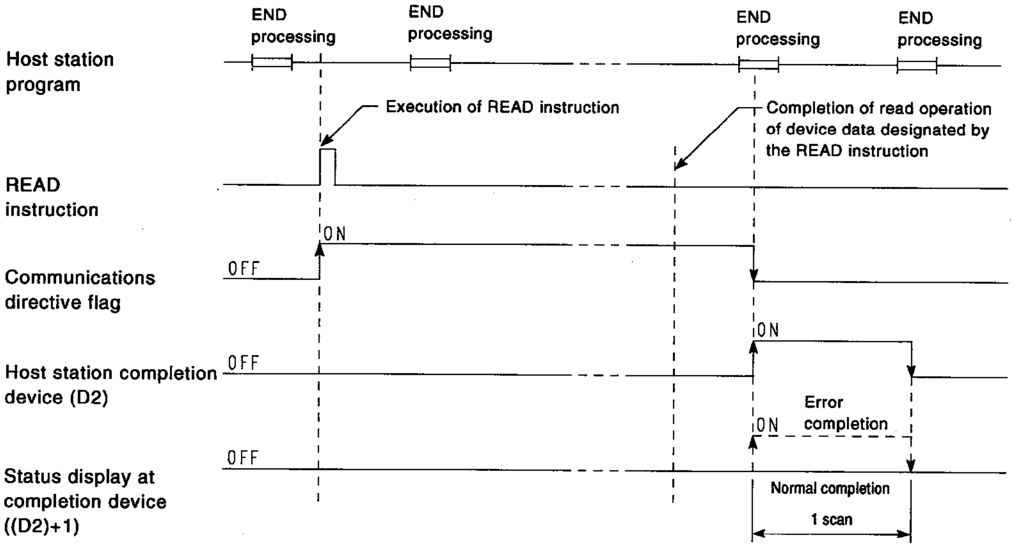
(4) The execution status of the READ instruction and whether this was completed normally or abnormally can be checked by the communications directive flag for the channel used, the completion device (D2), and the status display at completion device ((D2)+1).

(a) Communications directive flag
: Goes ON during execution of READ instruction, and OFF when OFF at the processing of the END instruction for the scan at completion of the read operation.

(b) Host station completion device
: Goes ON by END processing for the scan for completion of read operations conducted through the READ instruction, and OFF with the next END processing.

(c) Status display device at completion
: Turned ON and OFF according to the status when the READ instruction is completed.
- Normal completion : Stays OFF, with no changes
- Error completion : Goes ON with processing of the END instruction for the scan at completion of the READ instruction, and OFF at the processing of the next END processing.

[Host station operations during execution of READ instruction]



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Control data contents not within setting range (Error code: 4100)
- Network number specified by Jn is not connected to this station (Error code: 4102)
- Module for I/O number designated by Un is not a network module (Error code: 2111)

REMARK

* : The communications directive flags for the channels used are as indicated below:

Channel No.	1	2	3	4	5	6	7	8
Communications directive flag No.	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

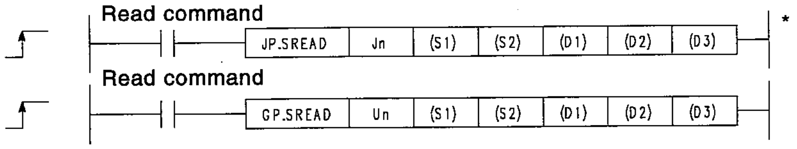
8.2.2 Reading word device data from another station

<table>
<tr><th rowspan="3">Set Data</th><th colspan="9">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System, User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th rowspan="2">Constant</th><th rowspan="2">Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th></tr>
<tr><td>(S1)</td><td>—</td><td colspan="2">o</td><td colspan="6">—</td></tr>
<tr><td>(S2)</td><td>—</td><td colspan="2">o</td><td colspan="6">—</td></tr>
<tr><td>(D1)</td><td>—</td><td colspan="2">o</td><td colspan="6">—</td></tr>
<tr><td>(D2)</td><td>o</td><td colspan="2">o</td><td colspan="6">—</td></tr>
<tr><td>(D3)</td><td>o</td><td colspan="2">o</td><td colspan="6">—</td></tr>
</table>

[Instruction Symbol] [Execution Condition]

SREAD

SREAD



Set Data

* Cannot be used with CC-Link.

[For MELSECNET/10, AJ71QC24]

Set Data	Meaning	Data Type
Jn	Network number for host station [1]	BIN 16 bits
Un	Head I/O number for host station of network module [2]	
(S1)	First device of host station storing control data	Device name
(S2)	First device of station where read data is being stored	
(D1)	First device of host station storing read data	
(D2)	Host station device that will turn ON for one scan when instruction has been completed [3]	Bit
(D3)	Object station device that will turn ON for one scan when instruction has been completed [3]	

REMARKS

1) *1 : The network number for the host station should be designated as follows.
- 1 to 239 : Network No.
- 254 : Network designated by setting for valid module for access from other stations.

2) *2 : The designation for the head I/O number for the host station network module can be from 0 to FE_H.

3) *3 : Local device and file registor for each program cannot be used as a device to use for the setting data.
(See QnACPU Programming Manual (Special Function Module) for details.)

[For CC-Link]

It can be used with the master module of the software version J or later.

Set Data	Meaning	Data Type
Un	Head I/O number for host station of master/host module [*1]	BIN 16 bits
(S1)	First device of host station storing control data	Device name
(S2)	First device of station where read data is being stored	
(D1)	First device of host station storing read data	
(D2)	Host station device that will turn ON for one scan when instruction has been completed	
(D3)	Object station device that will turn ON for one scan when instruction has been completed [*2]	Bit

REMARKS

*1 : The designation for the head I/O number for the host station master/ host module can be from 0 to FEH.

*2 : Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnA CPU Programming Manual (Special Function Module) for details)

POINTS

(1) The SREAD instruction can be executed only when the object station is a QnACPU station.
(The SREAD instruction cannot be executed at an ACPU connected to a MELSECNET/10 network.)
Only a station number for QnACPU should be set as the object station number.

(2) When reading the data from other station using the SREAD instruction, the parameter setting for the device must be identical between the host station and the object station.

Control Data

[For MELSECNET/10, AJ71QC24]

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Arrival confirmation made (bit 0=1; fixed)	0001H 0081H	User
	Error completion type	Sets clock data setting status when error processing is completed. Do not set clock data : set bit 7 (b7) to 0 Set clock data : set bit 7 (b7) to 1 ((S1)+11 onward appended)		
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code *1	—	System
(S1)+2	Channel used by host station	Designates channel the host station will use. *2	1 to 8	User
(S1)+3	Dummy	Not used	0	—
(S1)+4	Network number of object station	Sets network number for station designated to read device data	1 to 239 254 *3	User
(S1)+5	Object station number	Sets station number for designated station	1 to 64	User
(S1)+6	Dummy	Not used	—	—
(S1)+7	Number of transmission retries	Sets the number of retries for transmission in cases where a read operation made through the SREAD instruction has not been completed within the WDT time designated by (S1)+8.	0 to 15	User
	Results of retries	Number of repetitions conducted is stored.		System
(S1)+8	Arrival WDT time	The WDT time for read operations made through the SREAD instruction is set in units of seconds. If the read operation has not been completed within the time set, transmission will be retried for the number of times designated by (S1)+7.	1 to 32767 0 : 10 seconds fixed	User
(S1)+9	Receive data length	Set number of data blocks to be read	1 to 480	User
(S1)+10	Dummy	Not used	—	—
(S1)+11	Clock set flag (set only on error)	Stores clock data enable/disable status When clock data is disabled : 0 When clock data is enabled : 1	—	System
(S1)+12	Clock data (set only on error)	Upper 8 bits : Month (1 to 12), Lower 8 bits : Year (0 to 99)	—	System
(S1)+13		Upper 8 bits : Hour (0 to 23), Lower 8 bits : Day of month (1 to 31)		
(S1)+14		Upper 8 bits : Seconds (0 to 59), Lower 8 bits : Minute (0 to 59)		
(S1)+15		Upper 8 bits : 00H, Lower 8 bits : Day of week (0 to 6)		
(S1)+16	Network number where error was detected	Stores network number of station where error was detected 1 to 239 : Network No.	—	System
(S1)+17	Station number where error was detected	Stores station number of station where error was detected However, does not store data if (S1)+1 completion status is "Channel in Use." 1 to 64 : Station number	—	System

REMARKS

1) *1 : See the following manual for information on error codes.
MELSECNET/10 for QnA/Q4AR Network System Reference Manual
2) *2 : See Chapter 8 for information on the channels of network modules.
3) *3 : Designated when the value 254 has been set by Jn.

[For CC-Link]

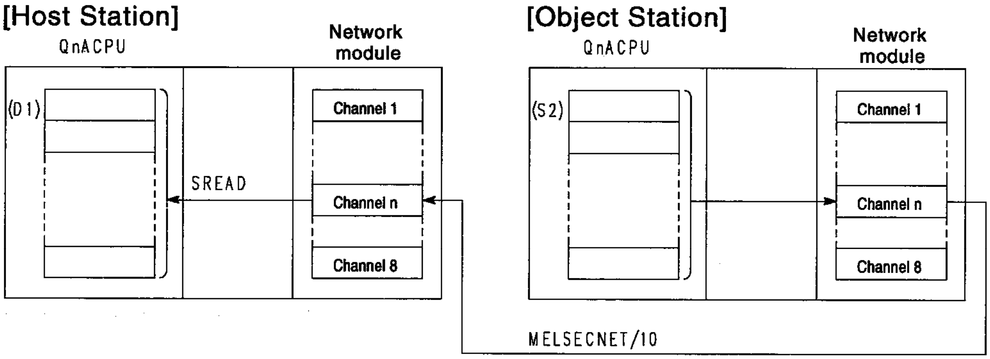
It can be used with the master module of the software version J or later.

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Arrival confirmation made (bit 0=1; fixed)	1	User
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code	—	System
(S1)+2	Channel used by host station	Designates channel the host station will use. 1: Use the CH1 side 2: Use the CH2 side	1.2	User
(S1)+3	Dummy	Not used	—	—
(S1)+4	Network number of object station	Fixed to 0	0	User
(S1)+5	Object station number	Fixed to 0	0	User
(S1)+6	Master/host station number	0: Master station specified 1 to 64: Host station specified	0 to 64	User
(S1)+7	Number of transmission retries	Sets the number of retries for transmission in cases where a read operation made through the SREAD instruction has not been completed within the WDT time designated by (S1)+8.	0 to 15	User
	Results of retries	Number of repetitions conducted is stored.	—	System
(S1)+8	Arrival WDT time	The WDT time for read operations made through the SREAD instruction is set in units of seconds. If the read operation has not been completed within the time set, transmission will be retried for the number of times designated by (S1)+7.	1 to 32767 0 : 10 seconds fixed (Default)	User
(S1)+9	Receive data length	Set number of data blocks to be read	1 to 480	User
(S1)+10	Dummy	Not used	—	—
(S1)+11	Dummy	Not used	—	—
(S1)+12	Dummy	Not used	—	—
(S1)+13				
(S1)+14				
(S1)+15				
(S1)+16	Network number where error was detected	Fixed to 0	—	System
(S1)+17	Station number where error was detected	Fixed to 0	—	System

Functions

[For MELSECNET/10, AJ71QC24]

(1) Stores data of devices starting from the word device designated at (S2) of the station connected to MELSECNET/10 as specified by the control data network number and station number, in devices starting from the device designated at (D1) at the host station.
When the device data read operation for the object station has been completed, the completion device designated by (D2) goes ON.
Also, when the transmission of the device data designated by (S2) has been completed, the device designated by (D3) goes ON at the object station.



(2) The device data read operation can be carried out with respect to a station connected in a MELSECNET/10 network whose number is designated, in addition to a station connected in the host station's network.

(3) Instructions for data link cannot be executed at more than one location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(4) The execution status of the SREAD instruction and whether this was completed normally or abnormally can be checked by the communications directive flag* for the channel used, the completion device ((D2) and (D3)), and the status display at completion device ((D2)+1).

(a) Communications directive flag
: Goes ON during execution of SREAD instruction, and OFF when OFF at the processing of the END instruction for the scan at completion of the read operation.

(b) Host station completion device
: Goes ON by END processing for the scan for completion of read operations conducted through the SREAD instruction, and OFF with the next END processing.

(c) Status display device at completion
: Turned ON and OFF according to the status when the SREAD instruction is completed.

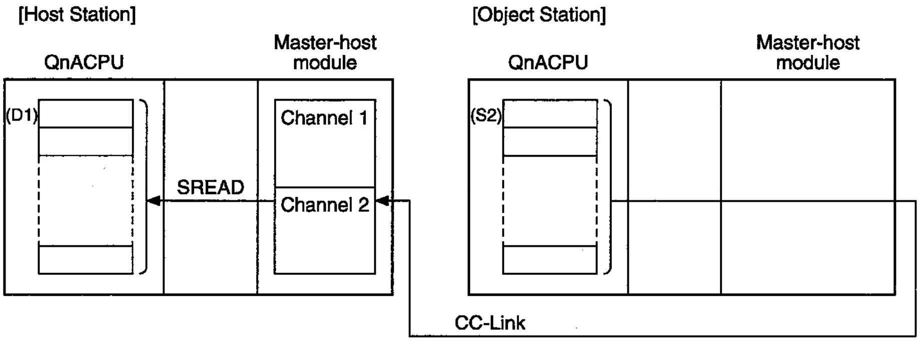
- Normal completion : Stays OFF, with no changes
- Error completion : Goes ON at END processing for the scan when SREAD instruction has been completed, and OFF at next END processing.

(d) Completion device for object station
: Goes ON at END processing for the scan when the transmission of the device data designated by the SREAD instruction has been completed, and OFF at the next END processing.

[For CC-Link]

It can be used with the master module of the software version J or later.

(1) Stores data of devices starting from the word device designated at (S2) of the station connected to master/host station number as specified by the control data station number, in devices starting from the device designated at (D1) at the host station.
When the device data read operation for the object station has been completed, the completion device designated by (D2) goes ON.
Also, when the transmission of the device data designated by (S2) has been completed, the device designated by (D3) goes ON at the object station.



(2) The device data read operation can be carried out with respect to a station connected in CC-Link.

(3) Instructions for data link cannot be executed at more than one location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(4) The execution status of the SREAD instruction and whether this was completed normally or abnormally can be checked by the communications directive flag* for the channel used, the completion device ((D2) and (D3)), and the status display at completion device ((D2)+1).

(a) Communications directive flag
: Goes ON during execution of SREAD instruction, and OFF when OFF at the processing of the END instruction for the scan at completion of the read operation.

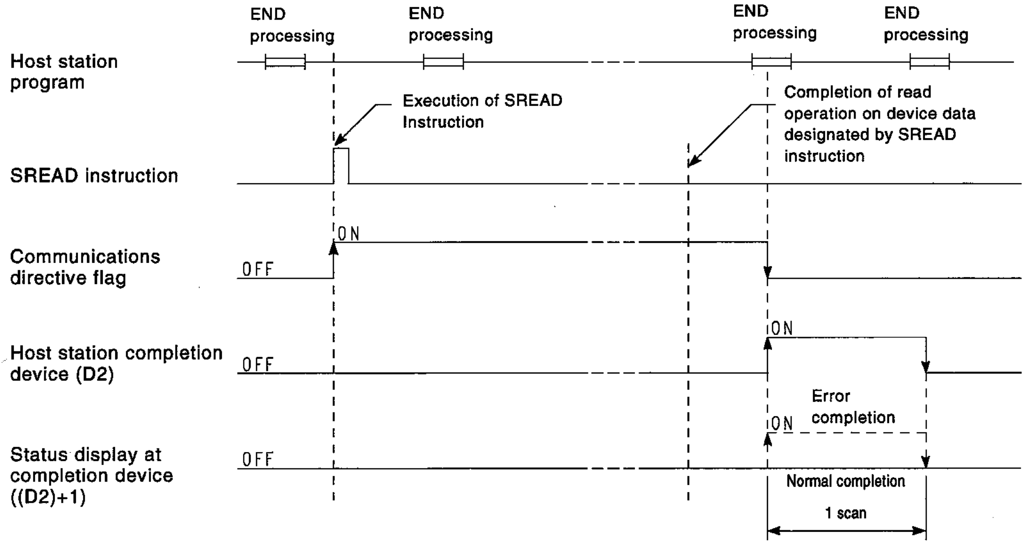
(b) Host station completion device
: Goes ON by END processing for the scan for completion of read operations conducted through the SREAD instruction, and OFF with the next END processing.

(c) Status display device at completion
: Turned ON and OFF according to the status when the SREAD instruction is completed.

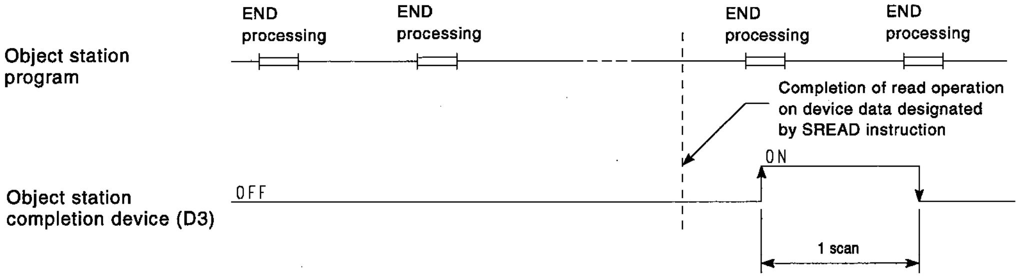
- Normal completion : Stays OFF, with no changes
- Error completion : Goes ON at END processing for the scan when SREAD instruction has been completed, and OFF at next END processing.

(d) Completed device of the target station
: This device is turned ON during the END processing of the scan that has completed transmission of the device data specified by a SREAD instruction, and turned OFF during the next END processing.

[Operation of host station when SREAD instruction is executed]



[Operation of object station when SREAD instruction is executed]



REMARK

* : The communications directive flags for the channels used are as indicated below.

Channel No.	1	2	3	4	5	6	7	8
Communications directive flag No.	SB30	SB32	SB34	SB36	SB38	SB40	SB42	SB44

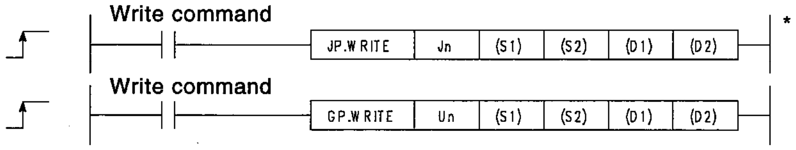
8.2.3 Device data write to station on MELSECNET/10 network

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—					
(S2)	—	o		—					
(D1)	—	o		—					
(D2)	o	o		—					

[Instruction Symbol] [Execution Condition]

WRITE

WRITE



* Cannot be used with CC-Link.

Set Data

[For MELSECNET/10, AJ71QC24]

Set Data	Meaning	Data Type
Jn	Network number for host station [*1]	BIN 16 bits
Un	Head I/O number for host station of network module [*2]	
(S1)	First device of host station storing control data	Device name
(S2)	First device at host station where the data to be written is being stored	
(D1)	First device at object station where the data will be stored when written	
(D2)	Device turned ON for 1 scan on completion of instruction [*3]	Bit

REMARKS

1) *1 : The network number for the host station should be designated as follows.
   - 1 to 239 : Network No.
   - 254 : Network designated by setting for valid module for access from other stations.
2) *2 : The designation for the head I/O number for the host station network module can be from 0 to FEH.
3) *3 : Local device and file registor for each program cannot be used as a device to use for the setting data.
   (See QnACPU Programming Manual (Special Function Module) for details.)

[For CC-Link]

It can be used with the master module of the software version J or later.

Set Data	Meaning	Data Type
Un	Head I/O number for host station of master/host module [*1]	BIN 16 bits
(S1)	First device of host station storing control data	Device name
(S2)	First device at host station where the data to be written is being stored	
(D1)	First device at object station where the data will be stored when written	
(D2)	Device turned ON for 1 scan on completion of instruction [*2]	Bit

REMARKS

*1 : The designation for the head I/O number for the host station master/host module can be from 0 to FEH.

*2 : Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnA CPU Programming Manual (Special Function Module) for details)

POINTS

(1) The WRITE instruction can be executed only if the object station is a QnACPU station.
(The WRITE instruction cannot be executed at an ACPU connected to the MELSECNET/10 network.)
Only a QnACPU station number should be set as the object station number.

(2) The designation "FFH (all stations at object network)" can be made by the WRITE instruction to an object station number for networks that are configured only with QnACPU stations.
The "FFH" designation cannot be made in respect to networks that are configured from a mix of QnACPU and ACPU stations.

(3) When reading the data from other station using the WRITE instruction, the parameter setting for the device must be identical between the host station and the object station.

Control Data

[For MELSECNET/10, AJ71QC24]

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Transmission completion confirmation : Set bit 0 to 0 No transmission completion confirmation : Set bit 0 to 1	0000H 0001H 0080H 0081H	User
	Error completion type	Sets clock data setting status when error processing is completed. Do not set clock data : set bit 7 (b7) to 0 Set clock data : set bit 7 (b7) to 1 ((S1)+10 onward appended)		
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code *1	—	System
(S1)+2	Channel used by host station	Designates channel the host station will use. *2	1 to 8	User
(S1)+3	Dummy	Not used	0	—
(S1)+4	Network number of object station	Designates network number for the object station	1 to 239 254 *3	User
(S1)+5	Object station number	Sets station number for designated station	1 to 64 : Station number 81H to 89H : Group designation FFH : All stations on object network	User
(S1)+6	Dummy	Not used	—	—
(S1)+7	Number of transmission retries	Valid when the execution type designated by (S1) is 1, and sets the number of transmission retries to be made when completion has not been made within the time designated by (S1)+8 for the WDT.	0 to 15	User
	Results of retries	Number of repetitions conducted is stored.		System
(S1)+8	Arrival WDT time	WDT time for the completion of the instruction is set in units of seconds. If operation has not been completed within the time set for the WDT, the transmission is retried for the number of times designated by (S1)+7.	1 to 32767 0 : 10 seconds fixed Valid only when execution type is set at 1	User
(S1)+9	Send data length	Sets the number of data block to be written.	1 to 480	User
(S1)+10	Dummy	Not used	—	—
(S1)+11	Clock set flag (set only on error)	Stores clock data enable/disable status When clock data is disabled : 0 When clock data is enabled : 1	—	System
(S1)+12	Clock data (set only on error)	Upper 8 bits : Month (1 to 12), Lower 8 bits : Year (0 to 99)	—	System
(S1)+13		Upper 8 bits : Hour (0 to 23), Lower 8 bits : Day of month (1 to 31)		
(S1)+14		Upper 8 bits : Seconds (0 to 59), Lower 8 bits : Minute (0 to 59)		
(S1)+15		Upper 8 bits : 00H, Lower 8 bits : Day of week (0 to 6)		
(S1)+16	Network number where error was detected	Stores network number of station where error was detected 1 to 239 : Network No.	—	System
(S1)+17	Station number where error was detected	Stores station number of station where error was detected However, does not store data if (S1)+1 completion status is "Channel in Use." 1 to 64 : Station number	—	System

REMARKS

1) *1 : See the following manual for information on error codes.
MELSECNET/10 for QnA/Q4AR Network System Reference Manual
2) *2 : See Chapter 8 for information on the channels of network modules.
3) *3 : Designated when the value 254 has been set by Jn.

[For CC-Link]

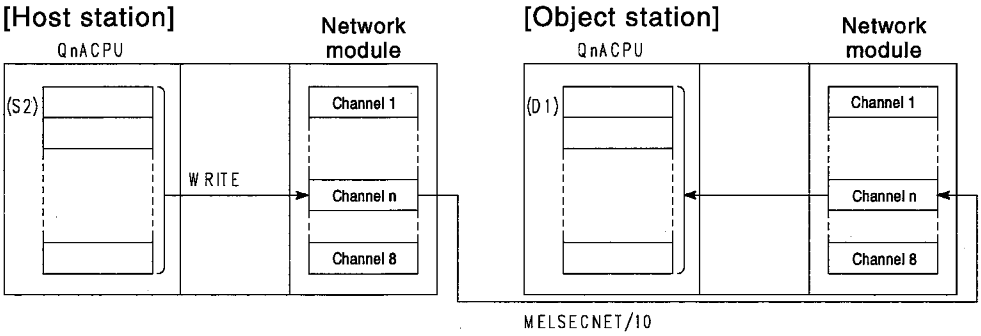
It can be used with the master module of the software version J or later.

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	0: No arrival confirmation 1 (Perform arrival confirmation)	0, 1	User
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code	—	System
(S1)+2	Channel used by host station	Designates channel the host station will use. 1: Use the CH1 side 2: Use the CH2 side	1.2	User
(S1)+3	Dummy	Not used	—	—
(S1)+4	Network number of object station	Fixed to 0	0	User
(S1)+5	Object station number	Fixed to 0	0	User
(S1)+6	Master/host station number	0: Master station specified 1 to 64: Host station specified	0 to 64	User
(S1)+7	Number of transmission retries	Valid when the execution type designated by (S1) is 1, and sets the number of transmission retries to be made when completion has not been made within the time designated by (S1)+8 for the WDT.	0 to 15	User
	Results of retries	Number of repetitions conducted is stored.	—	System
(S1)+8	Arrival WDT time	WDT time for the completion of the instruction is set in units of seconds. If operation has not been completed within the time set for the WDT, the transmission is retried for the number of times designated by (S1)+7.	1 to 32767 0:10 second fixed (Default) Valid only when execution type is set at 1	User
(S1)+9	Send data length	Sets the number of data block to be written.	1 to 480	User
(S1)+10	Dummy	Not used	—	—
(S1)+11	Dummy	Not used	—	—
(S1)+12	Dummy	Not used	—	—
(S1)+13				
(S1)+14				
(S1)+15				
(S1)+16	Network number where error was detected	Fixed to 0	—	System
(S1)+17	Station number where error was detected	Fixed to 0	—	System

Functions

[For MELSECNET/10, AJ71QC24]

(1) Stores the data at devices starting from that designated by the host station (S2) to word devices starting from that designated by (D2) of the station connected to the MELSECNET/10 network designated by the control data network number and station number.
When the write operation to the object station has been completed, the completion device designated by (D2) will go ON.



(2) Write operations of device data can be conducted with respect to stations connected to MELSECNET/10 designated network numbers in addition to stations connected to the host station's network.

(3) Instructions for data link cannot be executed at more than one location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(4) The fact that the execution of the WRITE instruction is underway, and both normal and error completions of this instruction, can be confirmed by the communications directive flag* of the channel in use, by the completion device (D2), and by the status display at completion device ((D2)+1) at the host station.

(a) Communications directive flag
: Goes ON at execution of the WRITE instruction, and OFF at the END processing of the scan when the write operation has been completed.

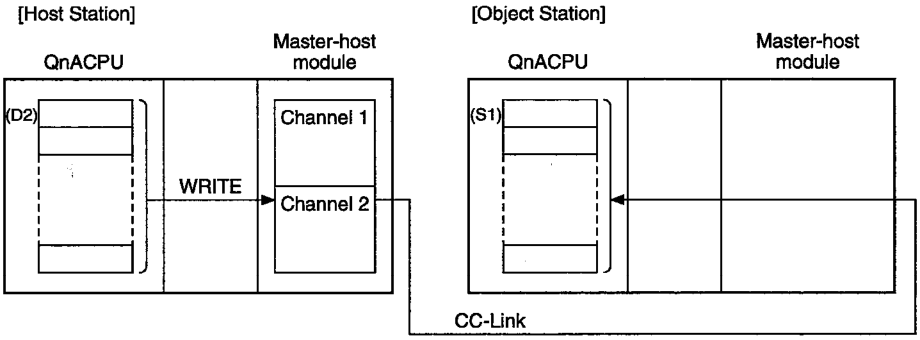
(b) Host station completion device
: Goes ON at the END processing of the scan when the WRITE instruction has been completed, and OFF at the next END processing.

(c) Status display device at completion
: Goes ON and OFF depending on the status at the completion of the WRITE instruction.
- Normal completion : Stays OFF, with no changes
- Error completion : Goes ON at the END processing of the scan when the WRITE instruction has been completed, and OFF at the next END processing.

[For CC-Link]

It can be used with the master module of the software version J or later.

(1) Stores the data at devices starting from that designated by the host station (S2) to word devices starting from that designated by (D1) of the station connected to the master/host station number designated by the control data station number.
When the write operation to the object station has been completed, the completion device designated by (D2) will go ON.



(2) Write operations of device data can be conducted with respect to stations connected to CC-Link.

(3) Instructions for data link cannot be executed at more than one location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

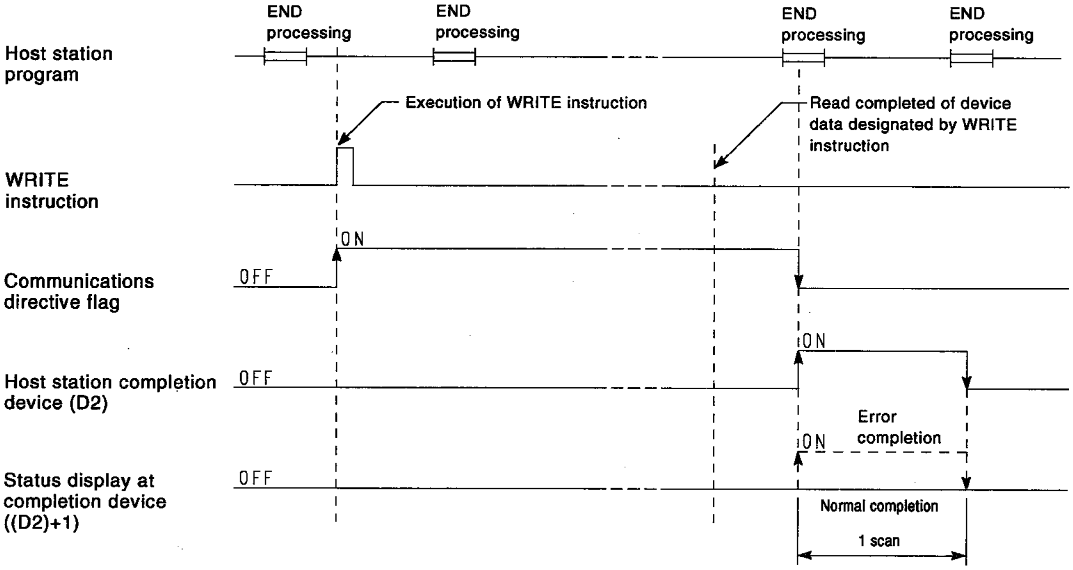
(4) The fact that the execution of the WRITE instruction is underway, and both normal and error completions of this instruction, can be confirmed by the communications directive flag of the channel in use, by the completion device (D2), and by the status display at completion device ((D2)+1) at the host station.

(a) Communications directive flag
: Goes ON at execution of the WRITE instruction, and OFF at the END processing of the scan when the write operation has been completed.

(b) Host station completion device
: Goes ON at the END processing of the scan when the WRITE instruction has been completed, and OFF at the next END processing.

(c) Status display device at completion
: Goes ON and OFF depending on the status at the completion of the WRITE instruction.
- Normal completion : Stays OFF, with no changes
- Error completion : Goes ON at the END processing of the scan when the WRITE instruction has been completed, and OFF at the next END processing.

[Host station operations during execution of WRITE instruction]



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Control data contents not within setting range (Error code: 4100)
- Network number specified by Jn is not connected to this station (Error code: 4102)
- Module for I/O number designated by Un is not a network module (Error code: 2111)

REMARK

* : The communications directive flags for the channels used are as indicated below:

Channel No.	1	2	3	4	5	6	7	8
Communications directive flag No.	SB30	SB32	SB34	SB36	SB38	SB3A	SB3B	SB3D

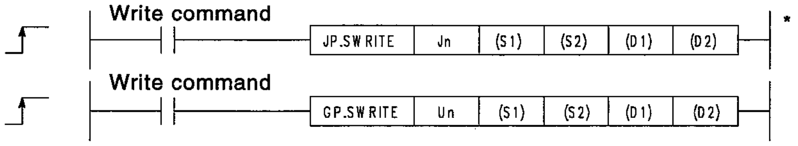
8.2.4 Writing device data to other stations

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—					
(S2)	—	o		—					
(D1)	—	o		—					
(D2)	o	o		—					
(D3)	o	o		—					

[Instruction Symbol] [Execution Condition]

SWRITE

SWRITE



Set Data

* Cannot be used with CC-Link.

[For MELSECNET/10, AJ71QC24]

Set Data	Meaning	Data Type
Jn	Network number for host station *1	BIN 16 bits
Un	Head I/O number for host station of network module *2	
(S1)	First device of host station storing control data	Device name
(S2)	First device at host station where the data to be written is being stored	
(D1)	First device at object station where the data will be stored when written	
(D2)	Host station device that will turn ON for one scan when instruction has been completed *3	Bit
(D3)	Object station device that will turn ON for one scan when instruction has been completed *3	

REMARKS

1) *1 : The network number for the host station should be designated as follows.
- 1 to 239 : Network No.
- 254 : Network designated by setting for valid module for access from other stations.

2) *2 : The designation for the head I/O number for the host station network module can be from 0 to FEH.

3) *3: Local device and file registor for each program cannot be used as a device to use for the setting data.
(See QnACPU Programming Manual (special function Module) for details.)

[For CC-Link]

It can be used with the master module of the software version J or later.

Set Data	Meaning	Data Type
Un	Head I/O number for host station of master/host module [1]	BIN 16 bits
(S1)	First device of host station storing control data	Device name
(S2)	First device at host station where the data to be written is being stored	
(D1)	First device at object station where the data will be stored when written	
(D2)	Host station device that will turn ON for one scan when instruction has been completed	Bit
(D3)	Object station device that will turn ON for one scan when instruction has been completed [2]	

REMARKS

*1 : The designation for the head I/O number for the host station master/host module can be from 0 to FEH.

*2 : Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnA CPU Programming Manual (Special Function Module) for details)

POINTS

(1) The SWRITE instruction can be executed only when the object station is a QnACPU station.
(The SWRITE instruction cannot be executed for ACPU stations connected to a MELSECNET/10 network.)
Only a station number for QnACPU should be set as the object station number.

(2) The designation "FFH (all stations at object network)" can be made by the SWRITE instruction to an object station number for networks that are configured only with QnACPU stations.
The "FFH" designation cannot be made in respect to networks that are configured from a mix of QnACPU and ACPU stations.

(3) When reading the data from other station using the SWRITE instruction, the parameter setting for the device must be identical between the host station and the object station.

Control Data

[For MELSECNET/10, AJ71QC24]

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	No transmission completion confirmation : Set bit 0 to 0 Transmission completion confirmation : Set bit 0 to 1	0000_H 0001_H 0080_H 0081_H	User
	Error completion type	Sets clock data setting status when error processing is completed. To not set clock data : Set bit 7 (b7) to 0 To set clock data : set bit 7 (b7) to 1 ((S1)+10 onward appended)		
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code *1	—	System
(S1)+2	Channel used by host station	Designates channel the host station will use. *2	1to 8	User
(S1)+3	Dummy	Not used	0	—
(S1)+4	Network number of object station	Designates network number for the object station	1 to 239 254 *3	User
(S1)+5	Object station number	Set station number for object station	1 to 64 : Station number 81_H to 89_H : Group designation FF_H : All stations on object network	User
(S1)+6	Dummy	Not used	—	—
(S1)+7	Number of transmission retries	Valid when the execution type designated by (S1) is 1, and sets the number of transmission retries to be made when completion has not been made within the time designated by (S1)+8 for the WDT.	0 to 15	User
	Results of retries	Number of repetitions conducted is stored.		System
(S1)+8	Arrival WDT time	WDT time for the completion of the instruction is set in units of seconds. If operation has not been completed within the time set for the WDT, the transmission is retried for the number of times designated by (S1)+7.	1 to 32767 0 : 10 seconds fixed Valid only when execution type is set at 1	User
(S1)+9	Send data length	Sets the number of data to be written.	1 to 480	User
(S1)+10	Dummy	Not used	—	—
(S1)+11	Clock set flag (set only on error)	Stores clock data enable/disable status When clock data is disabled : 0 When clock data is enabled : 1	—	System
(S1)+12	Clock data (set only on error)	Upper 8 bits : Month (1 to 12), Lower 8 bits : Year (0 to 99)	—	System
(S1)+13		Upper 8 bits : Hour (0 to 23), Lower 8 bits : Day of month (1 to 31)		
(S1)+14		Upper 8 bits : Seconds (0 to 59), Lower 8 bits : Minute (0 to 59)		
(S1)+15		Upper 8 bits : 00_H, Lower 8 bits : Day of week (0 to 6)		
(S1)+16	Network number where error was detected	Stores network number of station where error was detected 1 to 239 : Network No.	—	System
(S1)+17	Station number where error was detected	Stores station number of station where error was detected However, does not store data if (S1)+1 completion status is "Channel in Use." 1 to 64 : Station number	—	System

REMARKS

1) *1 : See the following manual for information on error codes.
MELSECNET/10 for QnA/Q4AR Network System Reference Manual
2) *2 : See Chapter 8 for information on the channels of network modules.
3) *3 : Designated when the value 254 has been set by Jn.

[For CC-Link]

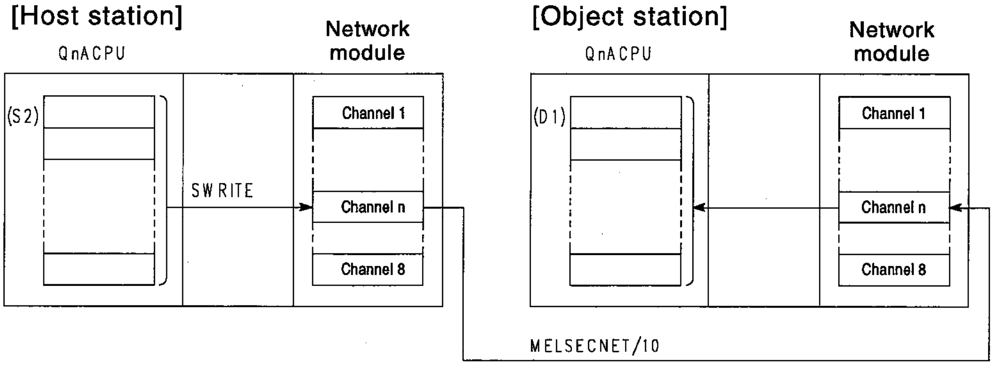
It can be used with the master module of the software version J or later.

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	0: No arrival confirmation 1: Perform arrival confirmation	0, 1	User
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code	—	System
(S1)+2	Channel used by host station	Designates channel the host station will use. 1: Use the CH1 side 2: Use the CH2 side	1.2	User
(S1)+3	Dummy	Not used	—	—
(S1)+4	Network number of object station	Fixed to 0	0	User
(S1)+5	Object station number	Fixed to 0	0	User
(S1)+6	Master/host station number	0: Master station specified 1 to 64: Host station specified	0 to 64	User
(S1)+7	Number of transmission retries	Valid when the execution type designated by (S1) is 1, and sets the number of transmission retries to be made when completion has not been made within the time designated by (S1)+8 for the WDT.	0 to 15	User
	Results of retries	Number of repetitions conducted is stored.	—	System
(S1)+8	Arrival WDT time	WDT time for the completion of the instruction is set in units of seconds. If operation has not been completed within the time set for the WDT, the transmission is retried for the number of times designated by (S1)+7.	1 to 32767 0:10 second fixed (Default) Valid only when execution type is set at 1	User
(S1)+9	Send data length	Sets the number of data block to be written.	1 to 480	User
(S1)+10	Dummy	Not used	—	—
(S1)+11	Dummy	Not used	—	—
(S1)+12	Dummy	Not used	—	—
(S1)+13				
(S1)+14				
(S1)+15				
(S1)+16	Network number where error was detected	Fixed to 0	—	System
(S1)+17	Station number where error was detected	Fixed to 0	—	System

Functions

[For MELSECNET/10, AJ71QC24]

(1) Stores the data starting from the device designated by the host station (S2) at word devices starting from that designated by (D1), of the station connected to the MELSECNET/10 network designated by the control data network number and station number.
When the write operation to the object station has been completed, the completion device designated by (D2) will go ON.
Also, when the write operation of the device data designated at the object station by (S2) has been completed, the device designated by (D3) will go ON.



(2) Write operations of device data can be conducted for stations connected to MELSECNET/10 designated network numbers in addition to stations connected to the host station's network.

(3) Instructions for data link cannot be executed at more than one location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(4) The fact that the execution of the SWRITE instruction is underway, and both normal and error completions of this instruction can be confirmed by the communications directive flag* of the channel in use, by the completion devices (D2) and (D3) for the host station and object station respectively, and by the status display at completion device ((D2)+1) at the host station.

(a) Communications directive flag
: Goes ON with the execution of the SWRITE instruction, and OFF at the END processing of the scan when the write operation has been completed.

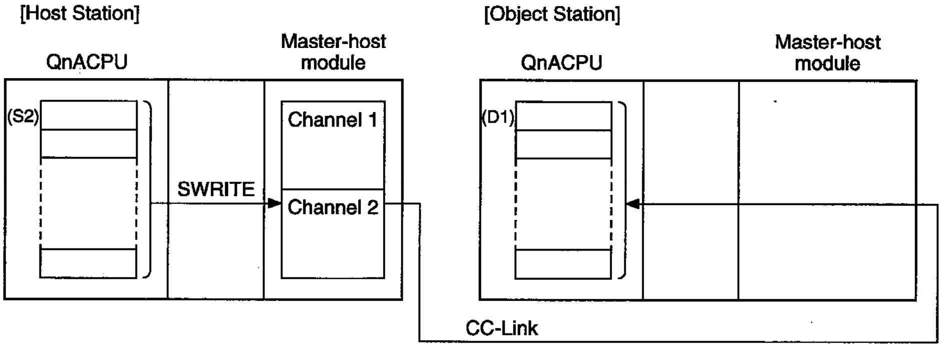
(b) Host station completion device
: Goes ON at the END processing of the scan when the SWRITE instruction has been completed, and OFF at the next END processing.

(c) Status display device at completion
: Goes ON and OFF depending on the status at the completion of the SWRITE instruction.
- Normal completion : Stays OFF, with no changes
- Error completion : Goes ON at the END processing of the scan when the WRITE instruction has been completed, and OFF at the next END processing.

[For CC-Link]

It can be used with the master module of the software version J or later.

(1) Stores the data starting from the device designated by the host station (S2) at word devices starting from that designated by (D1), of the station connected to the master/host station number designated by the control data station number.
When the write operation to the object station has been completed, the completion device designated by (D2) will go ON.
Also, when the write operation of the device data designated at the object station by (S2) has been completed, the device designated by (D3) will go ON.



(2) Write operations of device data can be conducted for stations connected to CC-Link.

(3) Instructions for data link cannot be executed at more than one location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(4) The fact that the execution of the SWRITE instruction is underway, and both normal and error completions of this instruction can be confirmed by the communications directive flag* of the channel in use, by the completion devices (D2) and (D3) for the host station and object station respectively, and by the status display at completion device ((D2)+1) at the host station.

(a) Communications directive flag
: Goes ON with the execution of the SWRITE instruction, and OFF at the END processing of the scan when the write operation has been completed.

(b) Host station completion device
: Goes ON at the END processing of the scan when the SWRITE instruction has been completed, and OFF at the next END processing.

(c) Status display device at completion
: Goes ON and OFF depending on the status at the completion of the SWRITE instruction.

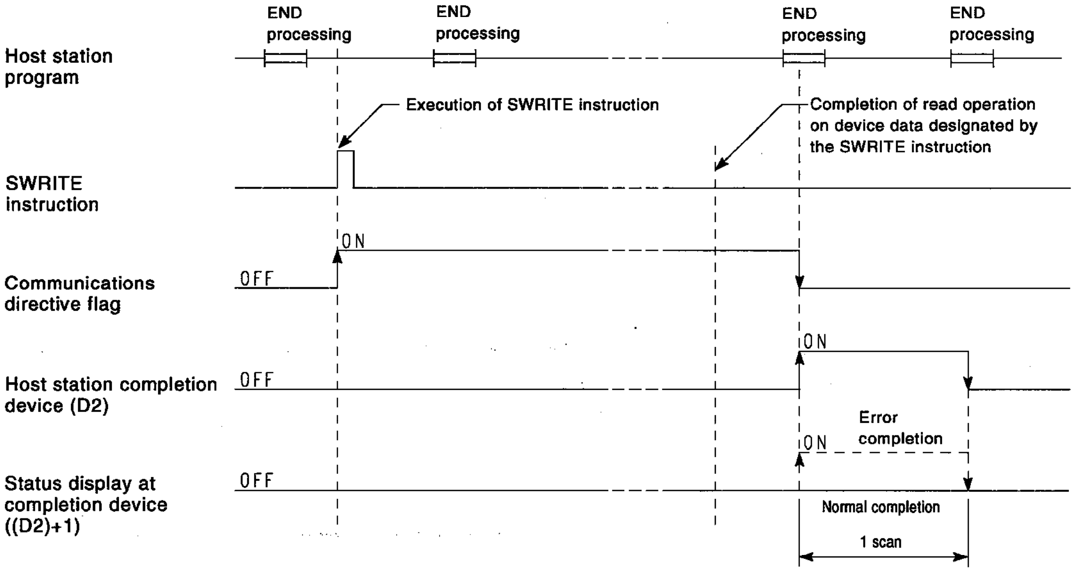
• Normal completion	: Stays OFF, with no changes
• Error completion	: Goes ON at the END processing of the scan when the WRITE instruction has been completed, and OFF at the next END processing.

REMARK

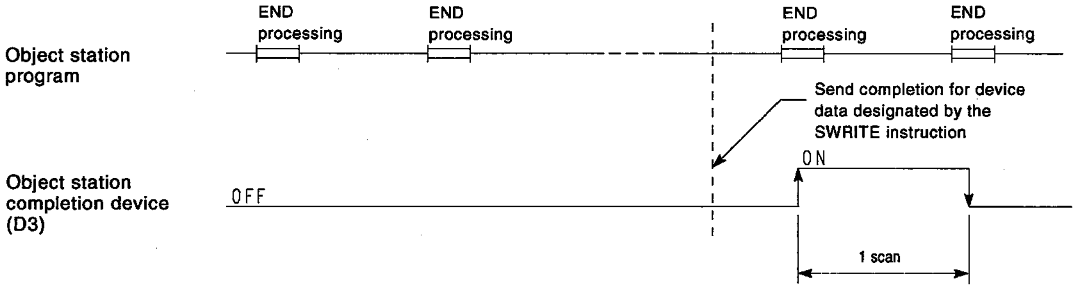
* : The communications directive flags for the channels used are as indicated below:

Channel No.	1	2	3	4	5	6	7	8
Communications directive flag No.	SB30	SB32	SB34	SB36	SB38	SB40	SB42	SB44

[Host station operations when SWRITE instruction is executed]



[Object station operations when SWRITE instruction is executed]



Operation Errors

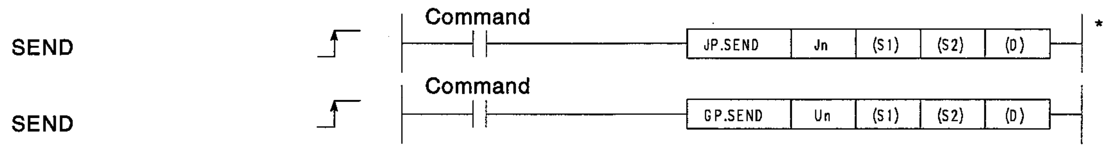
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Control data contents not within setting range (Error code: 4100)
- Network number specified by Jn is not connected to the host station (Error code: 4102)
- Module for I/O number designated by Un is not a network module (Error code: 2111)

8.2.5 Sending data to other stations

<table>
<tr><th rowspan="3">Set Data</th><th colspan="9">Usable Devices</th></tr>
<tr><th colspan="2">Internal Devices (System, User)</th><th rowspan="2">File Register</th><th colspan="2">MELSECNET/10 Direct J□\□</th><th rowspan="2">Special Function Module U□\G□</th><th rowspan="2">Index Register Zn</th><th rowspan="2">Constant</th><th rowspan="2">Other</th></tr>
<tr><th>Bit</th><th>Word</th><th>Bit</th><th>Word</th></tr>
<tr><td>(S1)</td><td>—</td><td colspan="2">o</td><td colspan="6">—</td></tr>
<tr><td>(S2)</td><td>—</td><td colspan="2">o</td><td colspan="6">—</td></tr>
<tr><td>(D)</td><td>o</td><td colspan="2">o</td><td colspan="6">—</td></tr>
</table>

[Instruction Symbol] [Execution Condition]



Set Data

* Cannot be used with CC-Link.

[For MELSECNET/10, AJ71QC24]

Set Data	Meaning	Data Type
Jn	Network number of host station [*1]	BIN 16 bits
Un	Head I/O number of host station link module [*2]	
(S1)	First device number of devices where control data is being stored	Device name
(S2)	First device number of devices where data to be sent is being stored	
(D)	Device that will go ON for 1 scan at send completion [*3]	Bit

REMARKS

1) *1 : The network number for the host station should be designated as follows.
   - 1 to 239 : Network No.
   - 254 : Network designated by setting for valid module for access from other stations.

2) *2 : The designation for the head I/O number for the host station network module can be from 0 to FEH.

3) *3 : Local device and file registor for each program cannot be used as a device to use for the setting data.
(See QnACPU Programming Manual (Special Function Module) for details.)

[For CC-Link]

It can be used with the master module of the software version J or later.

Set Data	Meaning	Data Type
Un	Head I/O number of host station link module [*1]	BIN 16 bits
(S1)	First device number of devices where control data is being stored	Device name
(S2)	First device number of devices where data to be sent is being stored	
(D)	Device that will go ON for 1 scan at send completion [*2]	Bit

REMARKS

*1 : The designation for the head I/O number for the host station master/host module can be from 0 to FEH.

*2 : Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnA CPU Programming Manual (Special Function Module) for details.)

Control Data

[For MELSECNET/10, AJ71QC24]

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	No transmission completion confirmation : Set bit 0 to 0 Transmission completion confirmation : Set bit 0 to 1	0000H 0001H 0080H 0081H	User
	Error completion type	Sets clock data setting status when error processing is completed. To not set clock data : set bit 7 (b7) to 0 To set clock data : set bit 7 (b7) to 1 ((S1)+11 onward appended)		
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code [1]	—	System
(S1)+2	Channel used by host station	Designates channel the host station will use. [2]	1to 8	User
(S1)+3	Channels used by object station	Designates channel for object station that will store data.	1 to 8	User
(S1)+4	Network number of object station	Designates network number for the object station	1 to 239 254 [3]	User
(S1)+5	Object station number	Designates station number of object station. (Range from 81H to 89H can be set only when execution type is 0.)	1 to 64 : Station number 81H to 89H : Group designation FFH : All stations on object network	User
(S1)+6	Dummy	Not used	—	—
(S1)+7	Number of transmission retries	Valid when execution type is set to 1, and sets number of retries for transmission when it has not been completed within the WDT time designated by (S1)+8.	0 to 15	User
	Results of retries	Number of repetitions conducted is stored.		System
(S1)+8	Arrival WDT time	WDT time for the completion of the instruction is set in units of seconds. If operation has not been completed within the time set for the WDT, the transmission is retried for the number of times designated by (S1)+7.	1 to 32767 0 : 10 seconds fixed Valid only when execution type is set at 1	User
(S1)+9	Send data length	Set number of data to send.	0 to 480	User
(S1)+10	Clock set flag (set only on error)	Stores clock data enable/disable status When clock data is disabled : 0 When clock data is enabled : 1	—	System
(S1)+11	Clock data (set only on error)	Upper 8 bits : Month (1 to 12) Lower 8 bits : Year (0 to 99)	—	System
(S1)+12		Upper 8 bits : Hour (0 to 23) Lower 8 bits : Day of month (1 to 31)		
(S1)+13		Upper 8 bits : Seconds (0 to 59) Lower 8 bits : Minute (0 to 59)		
(S1)+14		Upper 8 bits : 00H Lower 8 bits : Day of week (0 to 6)		
(S1)+16	Network number where error was detected	Stores network number of station where error was detected Network No.	—	System
(S1)+17	Station number where error was detected	Stores station number of station where error was detected However, does not store data if (S1)+1 completion status is "Channel in Use." 1 to 64 : Station number	—	System

REMARKS

1) *1 : See the following manual for information on error codes.
   - Error code: F000H to FFFFH
     MELSECNET/10 for QnA/Q4AR Network System Reference Manual
   - Error code: 4000H to 4FFFH
     Error Codes Returned to the Request Source in General Data Processing in Q2A(S1)/Q3A/Q4ACPU User's Manual
   - Error code: Other than those indicated above
     In this case, the error will be caused by the object station.
     See Troubleshooting in Q2A(S1)/Q3A/Q4ACPU User's Manual
2) *2 : See Chapter 8 for information on the channels of network modules.
3) *3 : Designated when the value 254 has been set by Jn.

[For CC-Link]

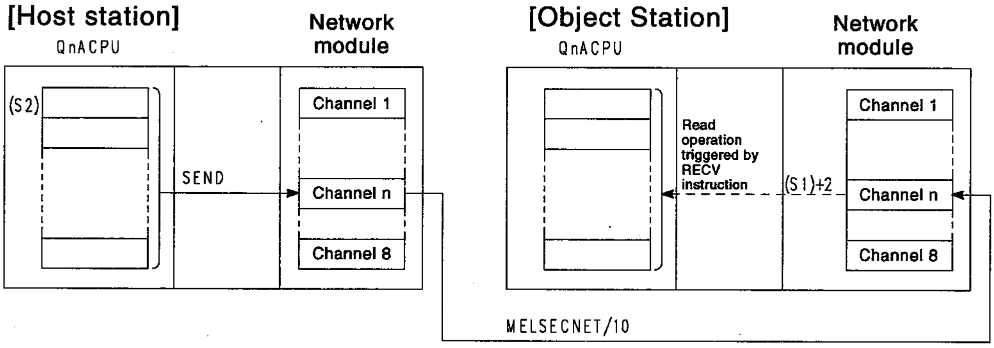
It can be used with the master module of the software version J or later.

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	0: No arrival confirmation 1 (Perform arrival confirmation)	0, 1	User
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code	—	System
(S1)+2	Channel used by host station	Designates channel the host station will use. 1: Use the CH1 side 2: Use the CH2 side	1.2	User
(S1)+3	Channel used by object station	Designates channel for object station that will store data. 1: Use the CH1 side 2: Use the CH2 side	1.2	User
(S1)+4	Network number of object station	Fixed to 0	0	User
(S1)+5	Object station number	Fixed to 0	0	User
(S1)+6	Master/host station number	0: Master station specified 1 to 64: Host station specified	0 to 64	User
(S1)+7	Number of transmission retries	Valid when the execution type is set to 1, and sets number of transmission retries when it has not been completed within the WDT time designated by (S1)+8.	0 to 15	User
	Results of retries	Number of repetitions conducted is stored.	—	System
(S1)+8	Arrival WDT time	WDT time for the completion of the instruction is set in units of seconds. If operation has not been completed within the time set for the WDT, the transmission is retried for the number of times designated by (S1)+7.	1 to 32767 0:10 second fixed (Default) Valid only when execution type is set at 1	User
(S1)+9	Send data length	Sets the number of data to send.	1 to 480	User
(S1)+10	Dummy	Not used	—	—
(S1)+11	Dummy	Not used	—	—
(S1)+12				
(S1)+13				
(S1)+14				
(S1)+16	Network number where error was detected	Fixed to 0	—	System
(S1)+17	Station number where error was detected	Fixed to 0	—	System

Functions

[For MELSECNET/10, AJ71QC24]

(1) Stores the data at devices starting from that designated by the host station (S2) to the designated channel of the station connected to the MELSECNET/10 network module designated by the control data network number and station number.
When the write operation to the object station has been completed, the completion device designated by (D) will go ON.



(2) Write operations of device data can be conducted with respect to stations connected to MELSECNET/10 designated network numbers in addition to stations connected to the host station's network.

(3) Instructions for data link cannot be executed simultaneously from more than 1 location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

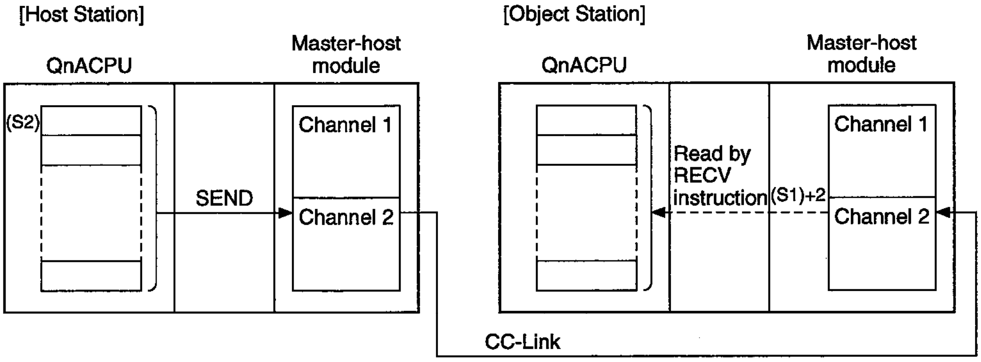
(4) The fact that the execution of the SEND instruction is underway, and both normal and error completions of this instruction, can be confirmed by the communications directive flag* of the channel in use, by the completion device (D), and by the completion status display device ((D)+1) at the host station.
(a) Communications directive flag
: Goes ON at execution of SEND instruction, and OFF at END processing of the scan when the write operation is completed.
(b) Host station completion device
: Goes ON at the END processing of the scan when the write operation triggered by the SEND instruction is completed, and OFF at the next END processing.
(c) Status display device at completion
: Goes ON and OFF depending on the status at the completion of the SEND instruction.
- Normal completion: Stays OFF with no change
- Error completion : Goes ON at the END processing of the scan when SEND instruction is completed, and OFF at next END processing.

[For CC-Link]

It can be used with the master module of the software version J or later.

(1) Stores the data at devices starting from that designated by the host station (S2) to the designated channel of the station connected to the master/host module designated by the control data station number. When the write operation to the object station has been completed, the completion device designated by (D) will go ON.

[Host station] [Object Station]



(2) Write operations of device data can be conducted with respect to stations connected to CC-Link.

(3) Instructions for data link cannot be executed simultaneously from more than 1 location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(4)

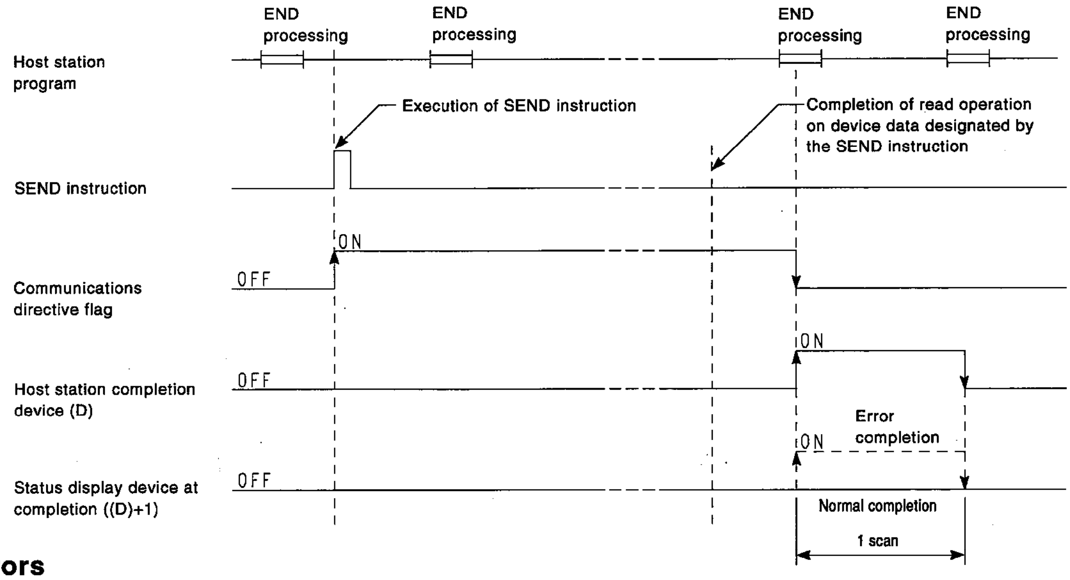
The fact that the execution of the SEND instruction is underway, and both normal and error completions of this instruction, can be confirmed by the communications directive flag* of the channel in use, by the completion device (D), and by the completion status display device ((D)+1) at the host station.

(a) Communications directive flag
: Goes ON at execution of SEND instruction, and OFF at END processing of the scan when the write operation is completed.

(b) Host station completion device
: Goes ON at the END processing of the scan when the write operation triggered by the SEND instruction is completed, and OFF at the next END processing.

(c) Status display device at completion
: Goes ON and OFF depending on the status at the completion of the SEND instruction.
- Normal completion: Stays OFF with no change
- Error completion : Goes ON at the END processing of the scan when SEND instruction is completed, and OFF at next END processing.

[Operations of host station during execution of SEND instruction]



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Contents of control data are not in the setting range (Error code: 4100)
- Network number specified by Jn is not connected to this station (Error code: 4102)
- Module for I/O number designated by Un is not a network module (Error code: 2111)

REMARK

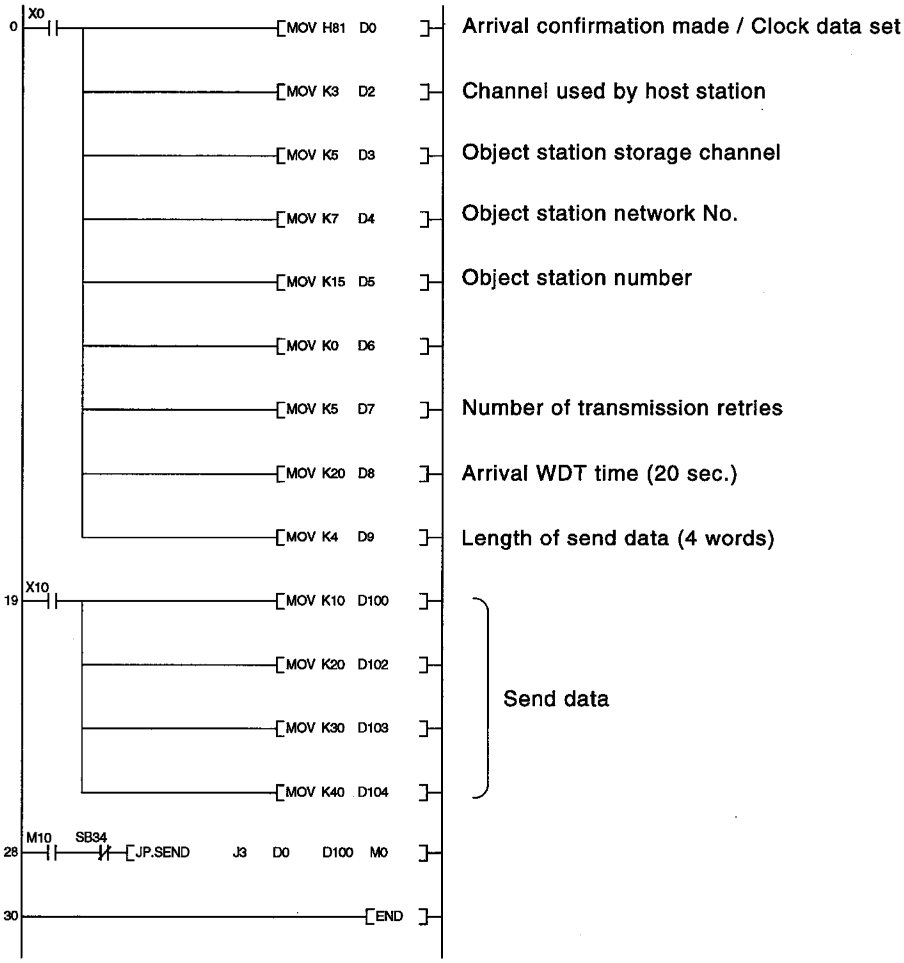
* : The communications directive flags for the channels used are as indicated below:

Channel No.	1	2	3	4	5	6	7	8
Communications directive flag No.	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

Program Example

(1) The following program uses channel 3 of the network module on network No. 7 to transmit data to channel 5 of station No. 15 on network No. 5 via the SEND instruction.

[Ladder Mode]



SB34 : Implement SEND instruction (channel 3)

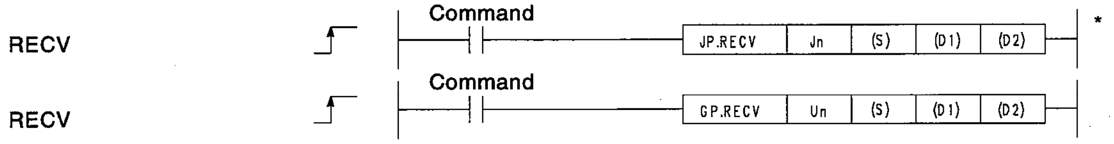
[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MOV	H81
		D0
3	MOV	K3
		D2
5	MOV	K5
		D3
7	MOV	K7
		D4
9	MOV	K15
		D5
11	MOV	K0
		D6
13	MOV	K5
		D7
15	MOV	K20
		D8
17	MOV	K4
		D9
19	LD	X10
20	MOV	K10
		D100
22	MOV	K20
		D102
24	MOV	K30
		D103
26	MOV	K40
		D104
28	LD	M10
29	ANI	SB34
30	JP.SEND	J3
		D0
		D100
		M0
38	END	

8.2.6 Receiving data from another station

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S)	—	o		—					
(D1)	—	o		—					
(D2)	o	o		—					

[Instruction Symbol] [Execution Condition]



* Cannot be used with CC-Link.

Set Data

[For MELSECNET/10, AJ71QC24]

Set Data	Meaning	Data Type
Jn	Network number of host station [*1]	BIN 16 bits
Un	Head I/O number of host station link module [*2]	
(S)	First device number of devices where control data is being stored	Device name
(D1)	First device number of devices to store send data	
(D2)	Device that will go ON for 1 scan at send completion [*3]	Bit

REMARKS

1) *1 : The network number for the host station should be designated as follows.
   - 1 to 239 : Network No.
   - 254 : Network designated by setting for valid module for access from other stations.

2) *2 : The designation for the head I/O number for the host station network module can be from 0 to FEH.

3) *3 : Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnA CPU Programming Manual (Special Function Module) for details.)

[For CC-Link]

It can be used with the master module of the software version J or later.

Set Data	Meaning	Data Type
Un	Head I/O number of host station link module [*1]	BIN 16 bits
(S)	First device number of devices where control data is being stored	Device name
(D1)	First device number of devices to store send data	
(D2)	Device that will go ON for 1 scan at send completion [*2]	Bit

REMARKS

*1 : The designation for the head I/O number for the host station master/host module can be from 0 to FEH.

*2 : Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnA CPU Programming Manual (Special Function Module) for details.)

Control Data

[For MELSECNET/10, AJ71QC24]

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Wait when no data (waits for a fixed period time until data is received) (bit 0 = 0 (fixed))	0000H 0080H	User
	Error completion type	Regulates whether clock data setting will be enabled or disabled at error completion. To not set clock data : Set bit 7 (b7) to 0 To set clock data : Set bit 7 (b7) to 1 ((S1)+11 onward appended)		
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code *1	—	System
(S1)+2	Channel used by host station	Designate channel where read data is being stored.*2	1 to 8	User
(S1)+3	Channels used by object station	Stores channel used by sending station	1 to 8	System
(S1)+4	Network number of object station	Stores network number of sending station	1 to 239 254 *3	System
(S1)+5	Object station station number	Stores station number of sending station	1 to 64	System
(S1)+6	Dummy	Not used	—	—
(S1)+7	Dummy	Not used	—	—
(S1)+8	Arrival WDT time	WDT time for the completion of the instruction is set in units of seconds. Error completion if reception has not been completed within WDT time	1 to 32767 0 : 10 seconds fixed Valid only when execution type is set at 1	User
(S1)+9	Allowable data length for reception	Stores number of data blocks received.	1 to 480	System
(S1)+10	Dummy	Not used	—	—
(S1)+11	Clock set flag (set only on error)	Stores clock data enable/disable status When clock data is disabled : 0 When clock data is enabled : 1	—	System
(S1)+12	Clock data (set only on error)	Upper 8 bits : Month (1 to 12), Lower 8 bits : Year (0 to 99)	—	System
(S1)+13		Upper 8 bits : Hour (0 to 23), Lower 8 bits : Day of month (1 to 31)		
(S1)+14		Upper 8 bits : Seconds (0 to 59), Lower 8 bits : Minute (0 to 59)		
(S1)+15		Upper 8 bits : 00H, Lower 8 bits : Day of week (1 to 6)		

REMARKS

1) *1 : See the following manual for information on error codes.
MELSECNET/10 for QnA/Q4AR Network System Reference Manual

2) *2 : See Chapter 8 for information on the channels of network modules.

[For CC-Link]

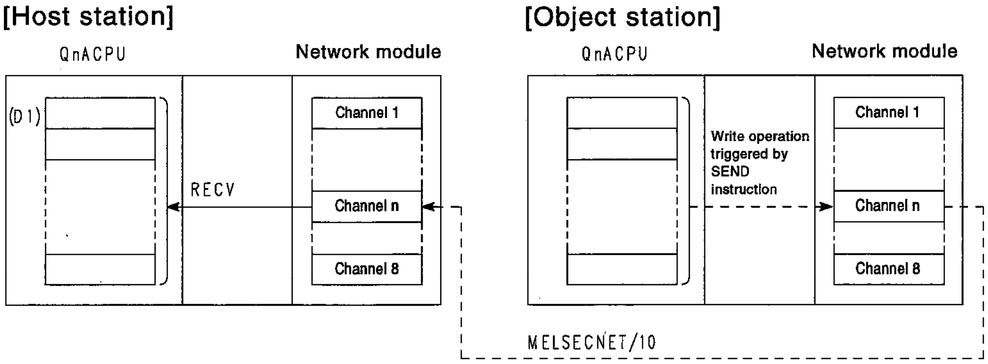
It can be used with the master module of the software version J or later.

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Fixed To 0 (No Arrival Confirmation)	0, 1	User
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code	—	System
(S1)+2	Channel used by host station	Designate channel where read data is being stored. 1: Use the CH1 side 2: Use the CH2 side	1.2	User
(S1)+3	Channels used by object station	Stores channel used by sending station 1: Use the CH1 side 2: Use the CH2 side	1.2	System
(S1)+4	Network number of object station	Fixed to 0	0	System
(S1)+5	Object station number	Fixed to 0	0	System
(S1)+6	Master/host station number	0: Master station specified 1 to 64: Host station specified	0 to 64	User
(S1)+7	Dummy	Not used	—	—
(S1)+8	Dummy	Not used	—	—
(S1)+9	Allowable data length for reception	Stores number of data blocks received.	1 to 480	System
(S1)+10	Dummy	Not used	—	—
(S1)+11	Dummy	Not used	—	—
(S1)+12	Dummy	Not used	—	—
(S1)+13				
(S1)+14				
(S1)+15				

Functions

[For MELSECNET/10, AJ71QC24]

(1) Stores the data sent via the SEND instruction from a station connected to the MELSECNET/10 network designated by the control data network number and station number following the device designated by (D1) from the network module of the host station. When the read operation on the device data from the network module has been completed, the completion device designated by (D2) will go ON.



(2) Instructions for data link cannot be executed simultaneously from more than 1 location with respect to the same channel. When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(3) The fact that the execution of the RECV instruction is underway, and both normal and error completions of this instruction, can be confirmed by the communications directive flag* of the channel in use, by the completion device (D2), and by the status display at completion device ((D2)+1) at the host station.

(a) Communications directive flag
: Goes ON at execution of RECV instruction, and OFF at the END processing of the scan when the read operation has been completed.

(b) Host station completion device
: Goes ON at the END processing of the scan when the read operation triggered by the RECV instruction has been completed, and OFF at the next END processing.

(c) Status display device at completion
: Goes ON and OFF depending on the status at completion of the RECV instruction.

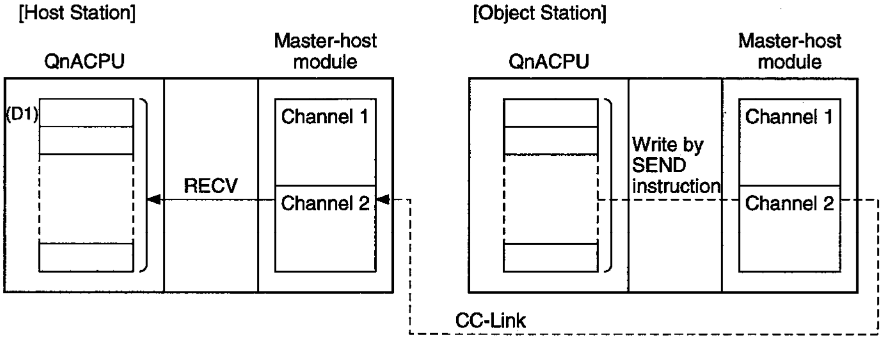
- Normal completion : Stays OFF with no change.
- Error completion : Goes ON at the END processing of the scan when the RECV instruction has been completed, and OFF at the next END processing.

[For CC-Link]

It can be used with the master module of the software version J or later.

(1) Stores the data sent via the SEND instruction from a station connected to the master/host station number designated by the control data station number following the device designated by (D1) from the network module of the host station.
When the read operation on the device data from the network module has been completed, the completion device designated by (D2) will go ON.

[Host station] [Object Station]



(2) Instructions for data link cannot be executed simultaneously from more than 1 location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(3) The fact that the execution of the RECV instruction is underway, and both normal and error completions of this instruction, can be confirmed by the communications directive flag of the channel in use, by the completion device (D2), and by the status display at completion device ((D2)+1) at the host station.

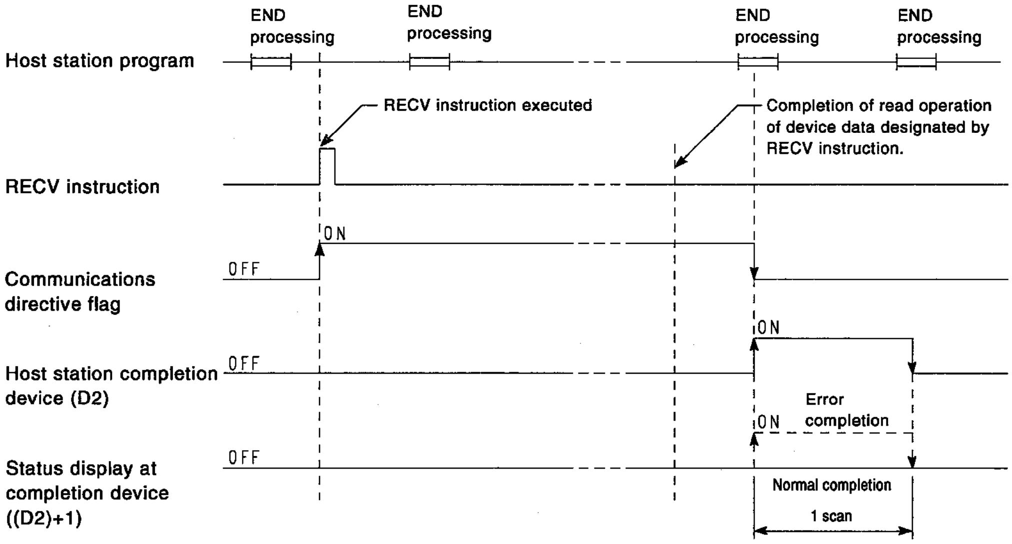
(a) Communications directive flag
: Goes ON at execution of RECV instruction, and OFF at the END processing of the scan when the read operation has been completed.

(b) Host station completion device
: Goes ON at the END processing of the scan when the read operation triggered by the RECV instruction has been completed, and OFF at the next END processing.

(c) Status display device at completion
: Goes ON and OFF depending on the status at completion of the RECV instruction.

• Normal completion	: Stays OFF with no change.
• Error completion	: Goes ON at the END processing of the scan when the RECV instruction has been completed, and OFF at the next END processing.

[Operations of the host station when the RECV instruction is executed]



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Contents of control data are not in the setting range. (Error code: 4100)
- Network number specified by Jn is not connected to this station (Error code: 4102)
- Module for I/O number designated by Un is not a network module (Error code: 2111)

REMARK

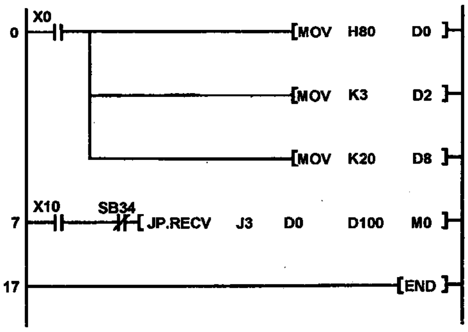
* : The communications directive flags for the channels used are as indicated below:

Channel No.	1	2	3	4	5	6	7	8
Communications directive flag No.	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

Program Example

(1) The following program reads data transmitted through the SEND instruction from channel 3 of the network module on network No. 3.

[Ladder Mode]



Sets clock data

Channel used by host station

Arrival WDT time (20 sec.)

SB34: Implements RECV instruction

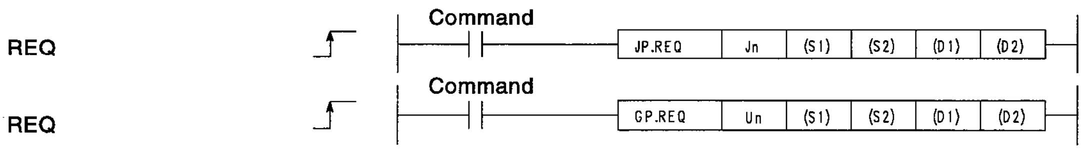
[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MOV	H80
		D0
3	MOV	K3
		D2
5	MOV	K20
		D8
7	LD	X10
8	ANI	SB34
9	JP.RECV	J3
		D0
		D100
		M0
17	END	

8.2.7 Transient requests from other stations (read/write clock data, remote RUN/STOP)

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	o		—					
(S2)	—	o		—					
(D1)	—	o		—					
(D2)	o	o		—					

[Instruction Symbol] [Execution Condition]



*

Set Data

* Cannot be used with CC-Link.

[For MELSECNET/10, AJ71QC24]

Set Data	Meaning	Data Type
Jn	Network number of host station [*1]	BIN 16 bits
Un	Head I/O number of host station link module [*2]	
(S1)	First device number of devices where control data is being stored	Device name
(S2)	First device number of devices where requested data is being stored	
(D1)	First device number of devices that will store response data	
(D2)	Device turned ON for 1 scan on completion of instruction [*3]	Bit

REMARKS

1) *1 : The network number for the host station should be designated as follows.
   - 1 to 239 : Network No.
   - 254 : Network designated by setting for valid module for access from other stations.

2) *2 : The designation for the head I/O number for the host station master/host module can be from 0 to FEH.

3) *3 : Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnA CPU Programming Manual (Special Function Module) for details.)

[For CC-Link]

It can be used with the master module of the software version J or later.

Set Data	Meaning	Data Type
Un	Head I/O number of host station link module [*1]	BIN 16 bits
(S1)	First device number of devices where control data is being stored	Device name
(S2)	First device number of devices where requested data is being stored	
(D1)	First device number of devices that will store response data	
(D2)	Device turned ON for 1 scan on completion ofinstruction [*2]	Bit

REMARKS

*1 : The designation for the head I/O number for the host station master/host module can be from 0 to FEH.

*2 : Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnA CPU Programming Manual (Special Function Module) for details.)

POINT

The REQ instruction can be executed only when the object station is a QnACPU station.
(The REQ instruction cannot be executed at an ACPU station connected to the MELSECNET/10 network.
Only a station number for QnACPU should be set as the object station number.

Control Data

[For MELSECNET/10, AJ71QC24]

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Arrival confirmation made (bit 0=1 (fixed))	0001H 0081H	User
	Error completion type	Sets clock data setting status when error processing is completed. Do not set clock data : set bit 7 (b7) to 0 Set clock data : set bit 7 (b7) to 1 ((S1)+11 onward appended)		
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code *1	—	System
(S1)+2	Channel used by host station	Designates channel the host station will use. *2	1 to 8	User
(S1)+3	Dummy	Not used	—	—
(S1)+4	Network number of object station	Designates network number for the object station	1 to 239 254 *3	User
(S1)+5	Object station station number	Designates station number of object station.	1 to 64	User
(S1)+6	Dummy	Not used	—	—
(S1)+7	Number of transmission retries	Sets the number of retries to attempt if the transmission has not been completed within the WDT time designated by (S1)+8.	1 to 15	User
	Results of retries	Number of repetitions conducted is stored.	—	System
(S1)+8	Arrival WDT time	WDT time for the completion of the instruction is set in units of seconds. If the read operation has not been completed within the time set, transmission will be retried for the number of times designated by (S1)+7.	1 to 32767 0 : 10 seconds fixed	User
(S1)+9	Request data length	Designates the length of the request data. When clock data is being read : 2 When clock data is being written : 7 During remote RUN/STOP : 4	2, 7, 4	User
(S1)+10	Response data length	Stores the length of the response data. When clock data is being written : 2 When clock data is being read : 6	2, 6	User
(S1)+11	Clock set flag (set only on error)	Stores clock data enable/disable status When clock data is disabled : 0 When clock data is enabled : 1	—	System
(S1)+12	Clock data (set only on error)	Upper 8 bits : Month (1 to 12), Lower 8 bits : Year (0 to 99)	—	System
(S1)+13		Upper 8 bits : Hour (0 to 23), Lower 8 bits : Day of month (1 to 31)		
(S1)+14		Upper 8 bits : Seconds (0 to 59), Lower 8 bits : Minute (0 to 59)		
(S1)+15		Upper 8 bits : 00H Lower 8 bits : Day of week (1 to 6)		
(S1)+16	Network number where error was detected	Stores network number of station where error was detected Will not be stored if the ((S1)+1) completion status is "Channel in use (F7C1H)". 1 to 239 : Network No.	—	System
(S1)+17	Station number where error was detected	Stores station number of station where error was detected Will not be stored if the ((S1)+1) completion status is "Channel in use (F7C1H)". 1 to 64 : Station number	—	System

REMARKS

1) *1 : See the following manual for information on error codes.
MELSECNET/10 for QnA/Q4AR Network System Reference Manual
2) *2 : See Chapter 8 for information on the channels of network modules.
3) *3 : Designated when the value 254 has been set by Jn.

[For CC-Link]

It can be used with the master module of the software version J or later.

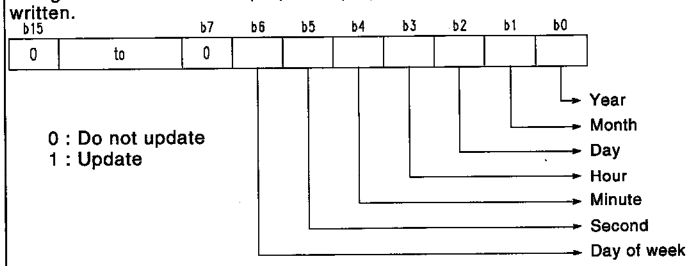
Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Arrival confirmation made (bit 0=1 (fixed))	1	User
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code	—	System
(S1)+2	Channel used by host station	Designates channel the host station will use. 1: Use the CH1 side 2: Use the CH2 side	1.2	User
(S1)+3	Dummy	Not used	—	—
(S1)+4	Network number of object station	Fixed to 0	0	User
(S1)+5	Object station number	Fixed to 0	0	User
(S1)+6	Master/host station number	0: Master station specified 1 to 64: Host station specified	0 to 64	User
(S1)+7	Number of transmission retries	Sets the number of retries to attempt if the transmission has not been completed within the WDT time designated by (S1)+8.	0 to 15	User
	Results of retries	Number of repetitions conducted is stored.	—	System
(S1)+8	Arrival WDT time	WDT time for the completion of the instruction is set in units of seconds. If the read operation has not been completed within the time set, transmission will be retried for the number of times designated by (S1)+7.	1 to 32767 0 : 10 seconds fixed (Default)	User
(S1)+9	Request data length	Designates the length of the request data. During remote RUN/STOP : 4	4	—
(S1)+10	Response data length	Stores the length of the response data. During remote RUN/STOP : 4	—	System
(S1)+11	Dummy	Not used	—	—
(S1)+12	Dummy	Not used	—	—
(S1)+13				
(S1)+14				
(S1)+15				
(S1)+16	Network number where error was detected	Fixed to 0	—	System
(S1)+17	Station number where error was detected	Fixed to 0	—	System

[Request data/Response data]

[For MELSECNET/10, AJ71QC24]

(1) During read/write operations on clock data

1) Request data

Device	Item	Meaning	Clock Data Read	Clock Data Write
(S2)	Request type	0001H : Clock data read 0011H : Clock data write	o	o
(S2)+1	Sub-request type	0002H : Clock data read 0001H : Clock data write	o	o
(S2)+2	Update pattern	Designate what item from (S2)+3 to (S2)+6 of the clock data is to be written.  0 : Do not update 1 : Update		o
(S2)+3	Month and year to update	Month and year (last two digits of year) stored in BCD code. b15 to b8: Month (01H to 12H) b7 to b0: Year (00H to 99H)		o
(S2)+4	Hour and day to update	Hour and day stored in BCD code b15 to b8: Hour (00H to 23H) b7 to b0: Day (01H to 31H)		o
(S2)+5	Minute and second to update	Second and minute stored in BCD code b15 to b8: Second (00H to 59H) b7 to b0: Minute (00H to 59H)		o
(S2)+6	Day of week to update	Day of week stored in BCD code b15 to b8: 00H b7 to b0: Day of week (00H to 06H) From 00H (Sunday) to 06H (Saturday)		o

2) Response data

Device	Item	Meaning	Clock Data Read	Clock Data Write
(D1)	Request type	0081H : Clock data read 0091H : Clock data write	o	o
(D1)+1	Sub-request type	0002H : Clock data read 0001H : Clock data write	o	o
(D1)+2	Month and year that were read	Month and year (last two digits of year) stored in BCD code b15 to b8 | b7 to b0 Month (01H to 12H) | Year (00H to 99H)	o	
(D1)+3	Hour and day that were read	Hour and day stored in BCD code b15 to b8 | b7 to b0 Hour (00H to 23H) | Day (01H to 31H)	o	
(D1)+4	Second and minute that were read	Second and minute stored in BCD code b15 to b8 | b7 to b0 Second (00H to 59H) | Minute (00H to 59H)	o	
(D1)+5	Day of the week that was read	Day of week stored in BCD code b15 to b8 | b7 to b0 00H | Day of week (00H to 06H) From 00H (Sunday) to 06H (Saturday)	o	

POINT

Clock data read/write operations are not possible when the system protect function has been engaged at the QnACPU of the object station (system protect switch SW5 is ON).

(2) During Remote RUN/STOP

1) Request data

Device	Item	Meaning	Remote RUN	Remote STOP
(S2)	Request type	0010H	o	o
(S2)+1	Sub-request type	0001H : Remote RUN 0002H : Remote STOP	o	o
(S2)+2	Mode	Designate whether or not to force a remote RUN operation. 0001H : Do not force run 0003H : Force run (set during remote STOP) (If the station which conducted the remote STOP operation cannot perform a remote RUN, the remote RUN operation can be forced from a different station.)	o	o
(S2)+3	Clear mode	Designate the QnACPU device memory status during remote RUN. 0000H : Do not clear (set during remote STOP) 0001H : Clear (except for latch range) 0002H : Clear (including latch range)	o	o

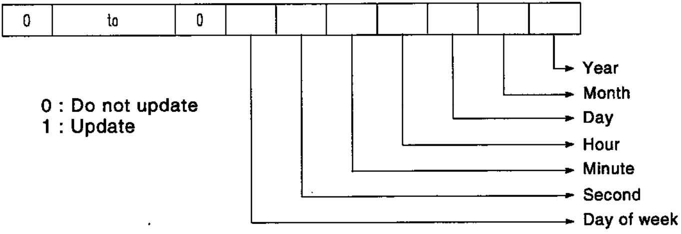
2) Response data..........None

[For CC-Link]

It can be used with the master module of the software version J or later.

(1) During read/write operations on clock data

1) Request data

Device	Item	Meaning	Clock Data Read	Clock Data Write
(S2)	Request type	0001H : Clock data read 0011H : Clock data write	o	o
(S2)+1	Sub-request type	0002H : Clock data read 0001H : Clock data write	o	o
(S2)+2	Update pattern	Designate what item from (S2)+3 to (S2)+6 of the clock data is to be written. 		o
(S2)+3	Month and day to update	Month and day stored in BCD code. b15 to b8: Day (01H to 31H) b7 to b0: Month (00H to 12H)		o
(S2)+4	Hour and minute to update	Hour and minute stored in BCD code. b15 to b8: Minute (00H to 60H) b7 to b0: Hour (00H to 23H)		o
(S2)+5	Day of week and second to update	Day of week and second stored in BCD code. b15 to b8: Day of week (00H to 06H) b7 to b0: Second (00H to 59H) From 00H (Sunday) to 06H (Saturday)		o

2) Response data

Device	Item	Meaning	Clock Data Read	Clock Data Write
(D1)	Month and year that were read	Month and year (the lower two digits year) stored in BCN code. b15 to b8: Month (01H to 12H) b7 to b0: Year (00H to 99H)	o	
(D1)+1	Hour and day that were read	Hour and day stored in BCN code. b15 to b8: Hour (00H to 23H) b7 to b0: Day (01H to 31H)	o	
(D1)+2	Second and minute that were read	Second and minute stored in BCD code. b15 to b8: Second (00H to 59H) b7 to b0: Minute (00H to 59H)	o	
(D1)+3	Day of the week that was read	Day of week stored in BCN code. b15 to b8: 00H b7 to b0: Day of week (00H to 06H) From 00H (Sunday) to 06H (Saturday)	o	

(2) Remote RUN/STOP

1) Request data

Device	Item	Meaning	Remote RUN	Remote STOP
(S2)	Request type	0010H	o	o
(S2)+1	Sub-request type	0001H : Remote RUN 0002H : Remote STOP	o	o
(S2)+2	Mode	Designate whether or not to force a remote RUN operation. 0001H : Do not force run 0003H : Force run (set during remote STOP) (If the station which conducted the remote STOP operation cannot perform a remote RUN, the remote RUN operation can be forced from a different station.)	o	o
(S2)+3	Clear mode	Designate the QnACPU device memory status during remote RUN. 0000H : Do not clear (set during remote STOP) 0001H : Clear (except for latch range) 0002H : Clear (including latch range)	o	o

POINTS

(1) The remote RUN/STOP function is enabled when the object station's QnACPU RUN/STOP key is set to RUN.

(2) Remote RUN/STOP operations are not possible when the system protect function has been engaged at the QnACPU of the object station (system protect switch SW5 is ON).

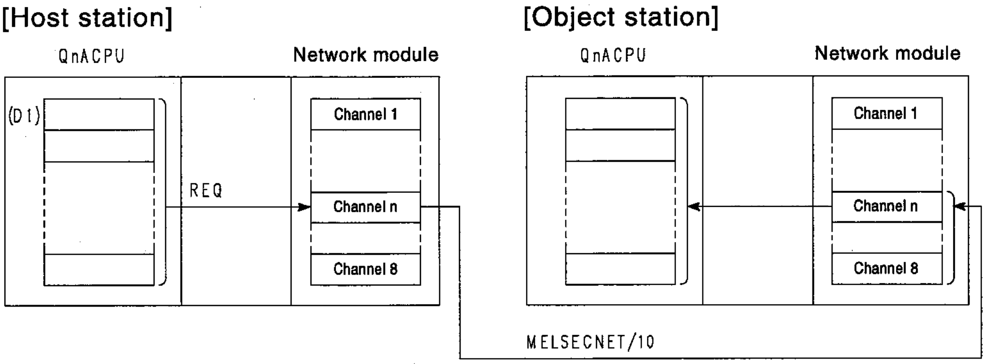
(3) If the object station has already been placed in the remote STOP/PAUSE mode from another station, RUN cannot be forced if the (S2)+1 mode is not set at "do not force run (0001H)".

(4) If the QnACPU of the object station that conducted the remote RUN/STOP operation is reset, the remote RUN/STOP information will be lost.

Functions

[For MELSECNET/10, AJ71QC24]

(1) Transmits the request data being stored at devices starting from that designated by (S2) of a station connected to the MELSECNET/10 network designated by the network number and station number in the control data, and conducts service requests.
The completion device designated by (D2) goes ON when the device data request from the object station is completed.



(2) Transmission operations of device data can be conducted with respect to stations connected to MELSECNET/10 designated network numbers in addition to stations connected to the host station's network.

(3) Instructions for data link cannot be executed simultaneously from more than 1 location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(4) The fact that the execution of the REQ instruction is underway, and both normal and error completions of this instruction, can be confirmed by the communications directive flag* of the channel in use, by the completion device (D2), and by the status display at completion device ((D2)+1) at the host station.

(a) Communications directive flag
: Goes ON at execution of REQ instruction, and OFF at END processing of scan when transmission has been completed.

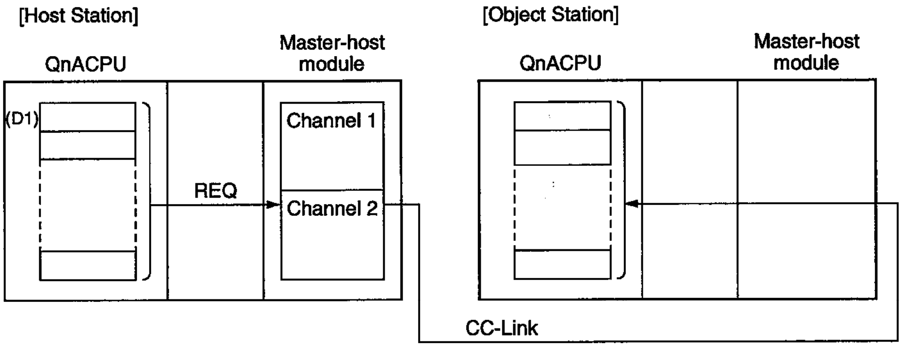
(b) Host station completion device
: Goes ON at END processing of scan when transmission triggered by REQ instruction has been completed, and OFF at next END processing.

(c) Status display device at completion
: Goes ON and OFF depending on the status at completion of the REQ instruction.
- Normal completion : Stays OFF, with no changes
- At error completion : Goes ON at END processing of scan when REQ instruction has been completed, and OFF at next END processing.

[For CC-Link]

It can be used with the master module of the software version J or later.

(1) Transmits the request data being stored at devices starting from that designated by (S2) of a station connected to the master/host number designated by station number in the control data, and conducts service requests.
The completion device designated by (D2) goes ON when the device data request from the object station is completed.



(2) Transmission operations of device data can be conducted with respect to stations connected to CC-Link.

(3) Instructions for data link cannot be executed simultaneously from more than 1 location with respect to the same channel.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

(4) The fact that the execution of the REQ instruction is underway, and both normal and error completions of this instruction, can be confirmed by the communications directive flag of the channel in use, by the completion device (D2), and by the status display at completion device ((D2)+1) at the host station.

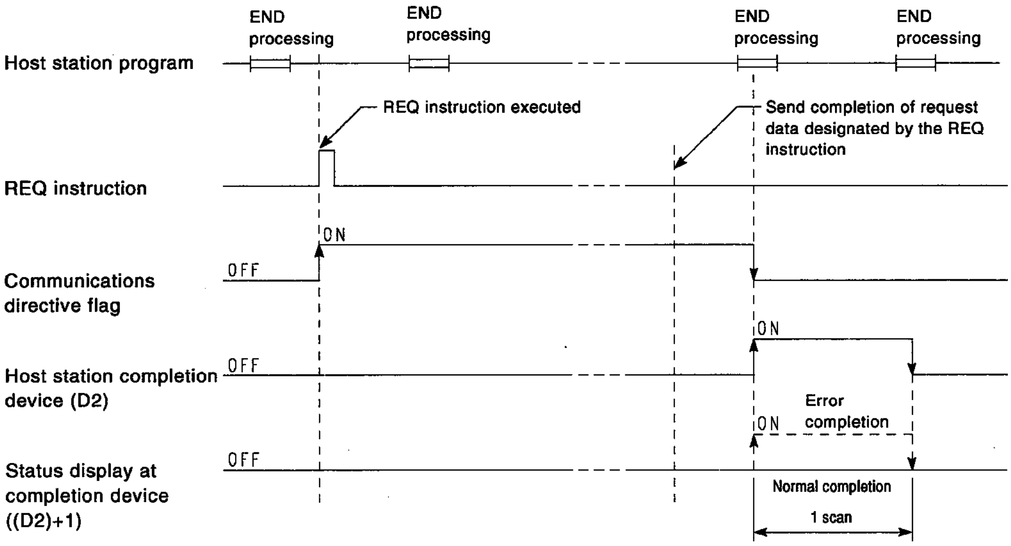
(a) Communications directive flag
: Goes ON at execution of REQ instruction, and OFF at END processing of scan when transmission has been completed.

(b) Host station completion device
: Goes ON at END processing of scan when transmission triggered by REQ instruction has been completed, and OFF at next END processing.

(c) Status display device at completion
: Goes ON and OFF depending on the status at completion of the REQ instruction.

- Normal completion : Stays OFF, with no changes
- At error completion : Goes ON at END processing of scan when REQ instruction has been completed, and OFF at next END processing.

[Operations of host station during the execution of the REQ instruction]



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Control data contents are not in the setting range (Error code: 4100)
- Network number specified by Jn is not connected to this station (Error code: 4102)
- The I/O number designated by Un is not a network module (Error code: 2111)

REMARK

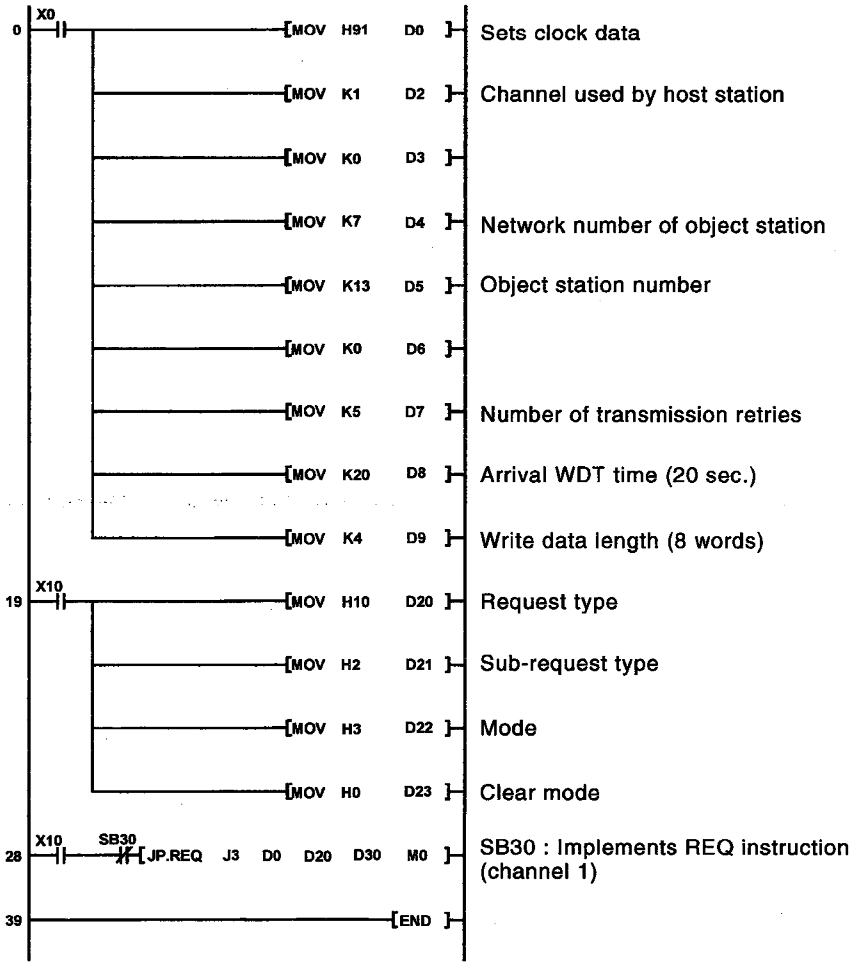
* : The communications directive flags for the channels used are as indicated below:

Channel No.	1	2	3	4	5	6	7	8
Communications directive flag No.	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

Program Example

(1) The following program performs a "STOP" operation on the QnACPU of station No. 13 at network No. 7.

[Ladder Mode]



```
 0  X0                 [MOV   H91   D0 ]  Sets clock data
                       [MOV   K1    D2 ]  Channel used by host station
                       [MOV   K0    D3 ]
                       [MOV   K7    D4 ]  Network number of object station
                       [MOV   K13   D5 ]  Object station number
                       [MOV   K0    D6 ]
                       [MOV   K5    D7 ]  Number of transmission retries
                       [MOV   K20   D8 ]  Arrival WDT time (20 sec.)
                       [MOV   K4    D9 ]  Write data length (8 words)
19  X10                [MOV   H10   D20]  Request type
                       [MOV   H2    D21]  Sub-request type
                       [MOV   H3    D22]  Mode
                       [MOV   H0    D23]  Clear mode
28  X10   SB30  [JP.REQ  J3  D0  D20  D30  M0 ]  SB30 : Implements REQ instruction
                                                  (channel 1)
39                                  [END ]
```

[List Mode]

```
Steps   Instruction   Device
  0     LD            X0
  1     MOV           H81
                      D0
  3     MOV           K1
                      D2
  5     MOV           K0
                      D3
  7     MOV           K7
                      D4
  9     MOV           K13
                      D5
 11     MOV           K0
                      D6
 13     MOV           K5
                      D7
 15     MOV           K20
                      D8
 17     LD            X10
 18     MOV           H10
                      D20
 20     MOV           H2
                      D21
 22     MOV           H3
                      D22
 24     MOV           H0
                      D23
 26     LD            X10
 27     ANI           SB30
 28     JP.REQ        J3
                      D0
                      D20
                      D30
                      M0
 37     END
```

8.2.8 Reading data from special function modules at remote I/O stations

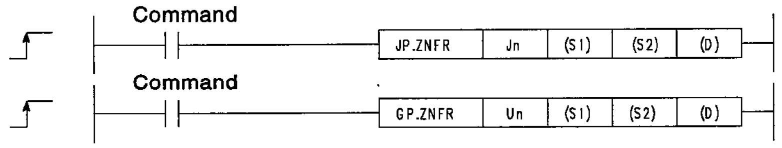
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	o	o	—					
(S2)	—	o*	—	—					
(D)	o	—	—	—					

* : Used only for link register

[Instruction Symbol] [Execution Condition]

ZNFR

ZNFR



Set Data

Set Data	Meaning	Data Type
Jn	Network number for host station [*1]	BIN 16 bits
Un	Head I/O number for host station of network unit [*2]	
(S1)	First device number of host station devices where control data is being stored	Device name
(S2)	First device number of host station link registers (W) where data will be stored after being read	
(D)	Device turned ON for 1 scan on completion of instruction [*3]	Bit

REMARKS

1) *1 : The network number for the host station should be designated as follows.
   - 1 to 239 : Network No.
   - 254 : Network designated by setting for valid module for access from other stations.

2) *2 : The designation for the head I/O number for the host station network unit can be from 0 to FEH.

3) *3 : Local device and file registor for each program cannot be used as a device to use for the setting data.
(See QnACPU Programming Manual (Special function Module) for details.)

Control Data

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Arrival confirmation made (bit 0=1 (fixed))	0001H 0081H	User
	Error completion type	Sets clock data setting status when error processing is completed. To not set clock data : Set bit 7 (b7) to 0 To set clock data : Set bit 7 (b7) to 1 ((S1)+11 onward appended)		
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code *1	—	System
(S1)+2	Dummy	Not used	—	—
(S1)+3	Buffer memory address	Designate first address in buffer memory	*2	User
(S1)+4	Dummy	Not used	—	—
(S1)+5	Object station number	Set station number for the remote I/O station that will read remote device data.	1 to 64	User
(S1)+6	Number in sequence	Designate which number in the series of special function modules installed at the object station this module is.	—	User
(S1)+7	Dummy	Not used	—	—
(S1)+8	Dummy	Not used	—	—
(S1)+9	Read data length	Set number of data to be read	1 to 256	User
(S1)+10	Clock data flag (set only during abnormal operation)	Stores clock data enable/disable status When clock data is disabled : 0 When clock data is enabled : 1	—	System
(S1)+11	Clock data (set only on error)	Upper 8 bits : Month (1 to 12), Lower 8 bits : Year (0 to 99)	—	System
(S1)+12		Upper 8 bits : Hour (0 to 23), Lower 8 bits : Day of month (1 to 31)		
(S1)+13		Upper 8 bits : Seconds (0 to 59), Lower 8 bits : Minute (0 to 59)		
(S1)+14		Upper 8 bits : 00H, Lower 8 bits : Day of week (1 to 6)		

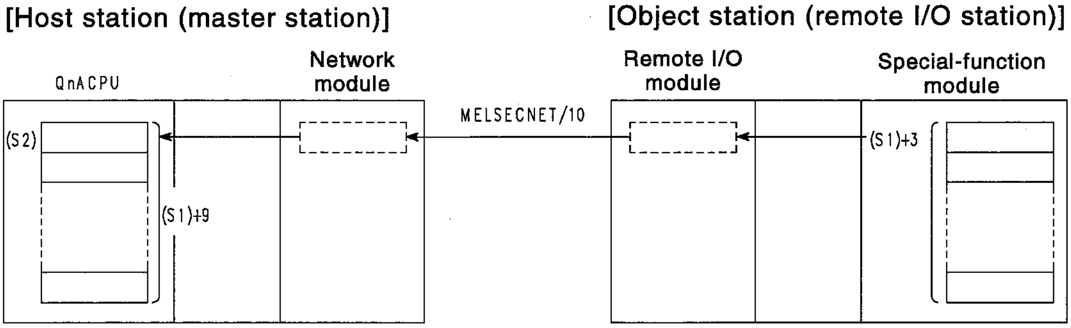
REMARKS

1) *1 : See the following manual for information on error codes.
MELSECNET/10 for QnA/Q4AR Network System Reference Manual

2) *2 : See the manual for the special function module where the read operation will be conducted.

Functions

(1) Stores the data from the buffer memory of the special function module at the MELSECNET/10 remote I/O station that has been designated by the control data at the object station to devices starting from that designated by (S2).
The completion device designated by (D) will go ON when the read operation on the data from the remote I/O station has been completed.



(2) Device data reading from a remote I/O station can be conducted only from a remote I/O station that is connected to the same network as is the remote station from the MELSECNET/10 remote station.

(3) The ZNFR instruction cannot be executed simultaneously from two locations from the same special function module.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that the executed ZNFR instruction will not be processed.

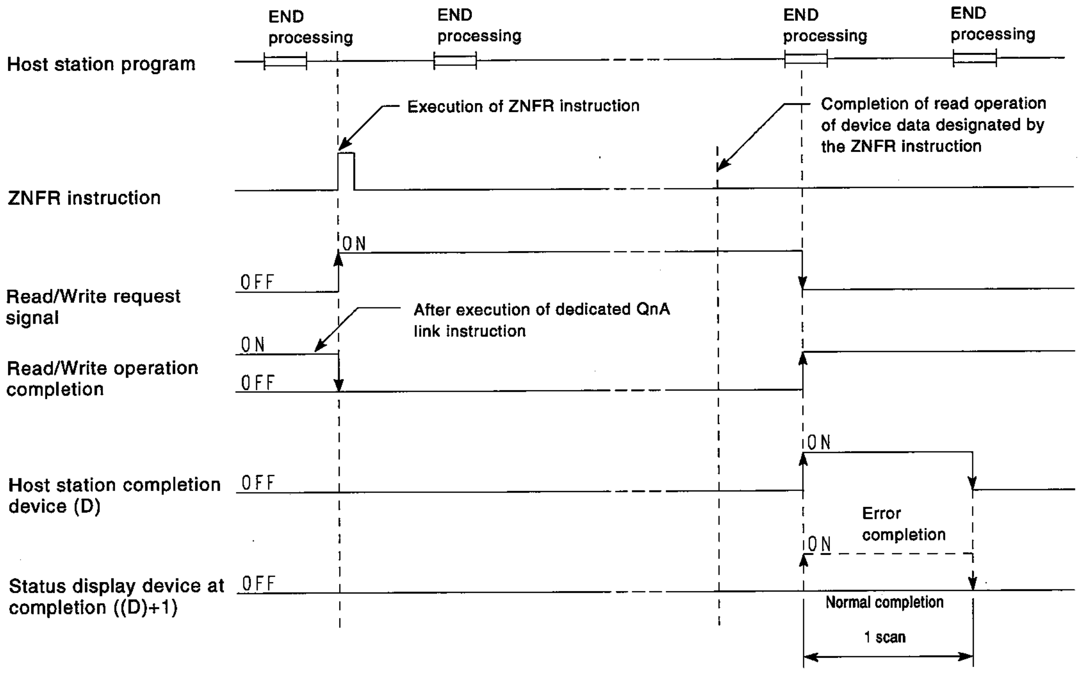
(4) The interlock signal broadcast during the execution of the ZNFR instruction contains read/write request signals, read/write completion signals, a completion device (D), and status at completion display device ((D)+1).

(a) Host station completion device
: Goes ON with the processing of the END signal for the scan at the completion of the ZNFR instruction, and OFF at the next END processing.

(b) Status display device at completion
: Turned ON and OFF according to the status at the completion of the ZNFR instruction.
- Normal completion : Stays OFF, with no changes
- Error completion : Goes ON with the processing of the END signal for the scan at the completion of the ZNFR instruction, and OFF at the next END processing.

(c) Cyclic transmission status at each station(SW74 to 77)
: The cyclic transmission status of all stations in the network is stored in the form of bit information and for a station where cyclic transmission is abnormal, the corresponding bit is set ON. Write a program so that ZNFR instruction is executed only at the stations where the bit is OFF (cyclic transmission status is normal) among the ZNFR execution object stations.

[Operations of the host station during the execution of the ZNFR instruction]



(5) The link registers (W) designated by (S2) are set at the network parameter "M←R (to master station from remote I/O station)," and are within the range allocated to the CPU by the link refresh parameters.

(6) In order to execute the ZNFR instruction, a link relay and link register that the OS can use are required.
The numbers of link relays and link registers to be used by the OS for each special function module are as follows:

- For M→R (from master station to remote I/O station)
.......Link relays : 4   Link registers : 4
- For M←R (to master station from remote I/O station)
.......Link relays : 4   Link registers : 4

Operation Errors

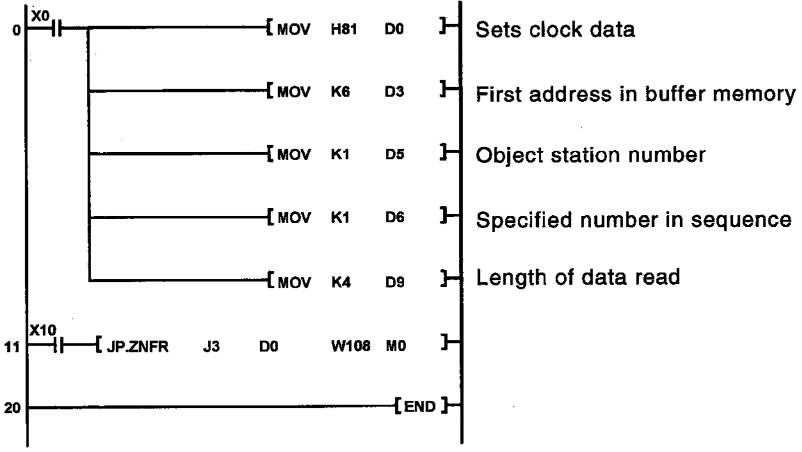
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Control data contents not within setting range (Error code: 4100)
- Network number specified by Jn is not connected to this station (Error code: 4102)
- Module for I/O number designated by Un is not a network module (Error code: 2111)

Program Example

(1) The following program reads addresses 6 through 9 of the buffer memory of the first special function module at station No. 1R1 of network No. 3 to W108 through 10B.

[Ladder Mode]



[List Mode]

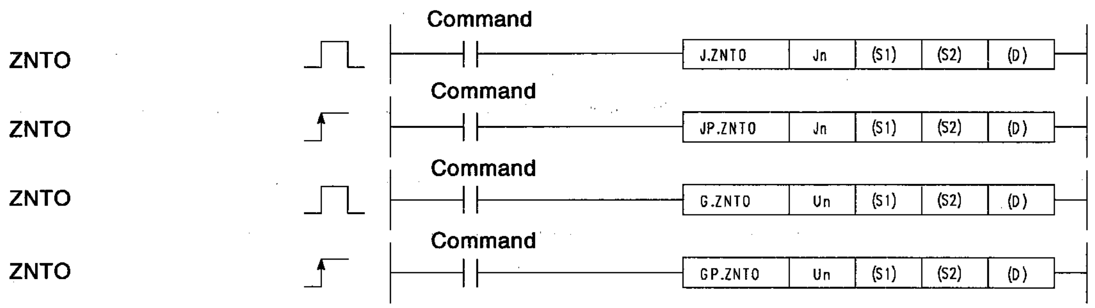
Steps	Instruction	Device
0	LD	X0
1	MOV	H81
		D0
3	MOV	K6
		D3
5	MOV	K1
		D5
7	MOV	K1
		D6
9	MOV	K4
		D8
11	LD	X10
12	JP.ZNRD	J3
		D0
		W108
		M0
20	END	

8.2.9 Writing data to special function module of remote I/O station

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	o	o	—					
(S2)	—	o *	—	—					
(D)	o	—	—	—					

*: Used only for link register

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
Jn	Network number for host station *1	BIN 16 bits
Un	Head I/O number for host station of network unit *2	
(S1)	First device number of host station devices where control data is being stored	Device name
(S2)	First device number of host station link registers (W) where data to be written is being stored	
(D)	Device turned ON for 1 scan on completion of instruction *3	Bit

REMARKS

1) *1 : The network number for the host station should be designated as follows.
- 1 to 239 : Network No.
- 254 : Network designated by setting for valid module for access from other stations.

2) *2 : The designation for the head I/O number for the host station network unit can be from 0 to FEH.

3) *3 : Local device and file registor for each program cannot be used as a device to use for the setting data.
(See QnACPU Programming Manual (Special function Module) for details.)

Control Data

Device	Item	Set Data	Setting Range	Set by
(S1)+0	Execution type	Response made (bit 0=1 (fixed))	0001_H 0081_H	User
	Error completion type	Confirm clock data setting status at error To not set clock data : Set bit 7 (b7) to 0 To set clock data : Set bit 7 (b7) to 1 ((S1)+11 onward appended)		
(S1)+1	Completion status	Status at completion of instruction is stored 0 : No errors (normal completion) Other than 0 : Error code *1	—	System
(S1)+2	Dummy	Not used	—	—
(S1)+3	Buffer memory address	Designate first address in buffer memory	*2	User
(S1)+4	Dummy	Not used	—	—
(S1)+5	Object station number	Designates station number of object station.	1 to 64	User
(S1)+6	Number of special function modules	Designate which number in sequence of special function modules installed at the object station this one is.	—	User
(S1)+7	Dummy	Not used	—	—
(S1)+8	Dummy	Not used	—	—
(S1)+9	Write data length	Designate number of data to write.	1 to 256	User
(S1)+10	Dummy	Not used	—	—
(S1)+11	Clock data flag (set only during abnormal operation)	Stores clock data enable/disable status When clock data is disabled : 0 When clock data is enabled : 1	—	System
(S1)+12	Clock data (set only on error)	Upper 8 bits : Month (1 to 12), Lower 8 bits: Year (0 to 99)	—	System
(S1)+13		Upper 8 bits : Hour (0 to 23), Lower 8 bits: Day of month (1 to 31)		
(S1)+14		Upper 8 bits : Seconds (0 to 59), Lower 8 bits: Minute (0 to 59)		
(S1)+15		Upper 8 bits : 00_H, Lower 8 bits: Day of week (1 to 6)		

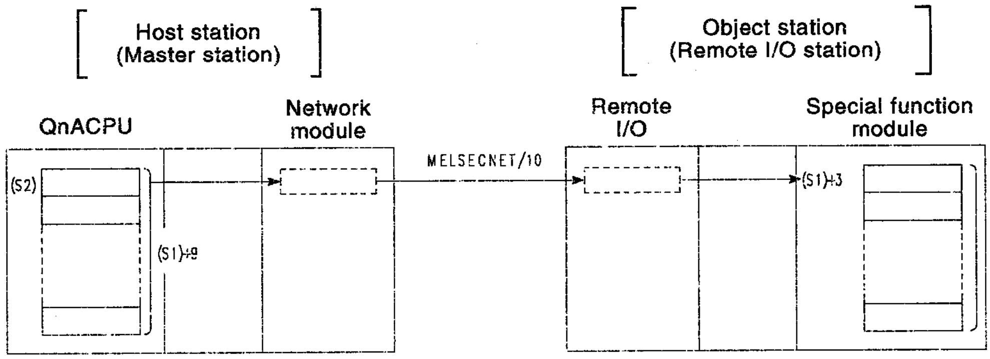
REMARKS

1) *1 : See the following manual for information on error codes.
MELSECNET/10 for QnA/Q4AR Network System Reference Manual

2) *2 : See the manual for the special function module where the write operation will be conducted.

Function

(1) Writes the data being stored at devices starting from that designated by (S2) to the buffer memory of the special function module at the MELSECNET/10 remote I/O station that has been designated by object station number in the control data.
The completion device designated by (D) goes ON when the data write to the remote I/O station has been completed.



(2) Write operations to remote I/O stations can be performed only when they are being done from a remote MELSECNET/10 master station to a remote I/O station connected on the same network.

(3) The ZNTO instruction cannot be conducted simultaneously to more than one location with respect to the same special function module. If conditions have been established for simultaneous execution of the instruction from more than one location, the ZNTO instruction will not be processed because a handshake is conducted automatically.

(4) The ZNTO execution in process interlock signal contains a read/write request signal, a read/write completion signal, the completion device (D), and the completion status device ((D)+1)

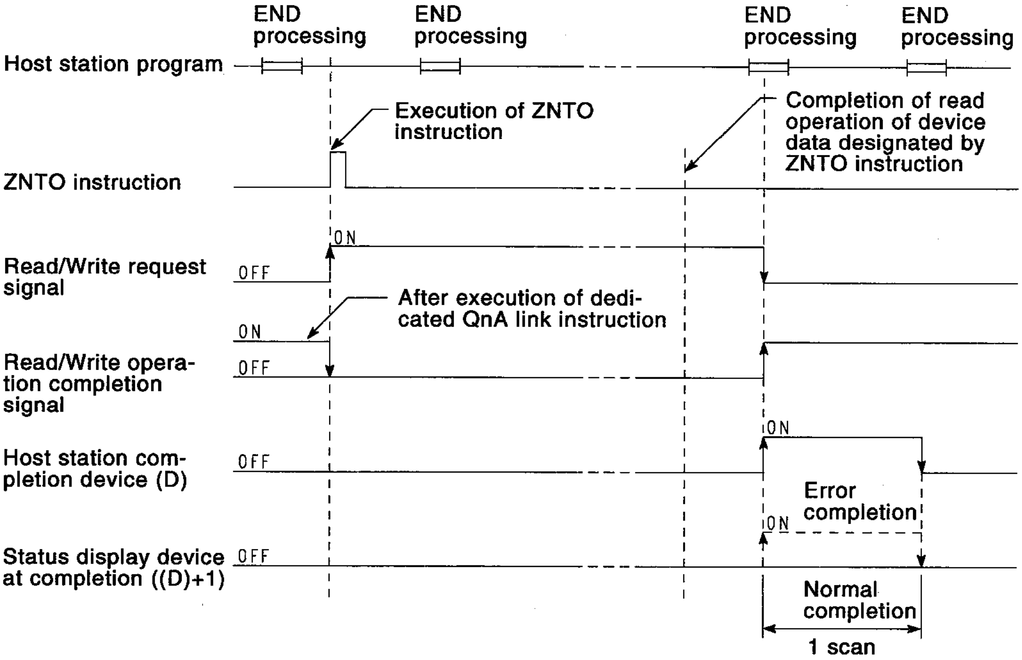
(a) Read/write request signal
: Goes ON with execution of QnA link dedicated instruction, and OFF with the processing of the END signal for the scan at the completion of a read/write operation.

(b) Read/write completion signal
: Goes OFF with execution of QnA dedicated link instruction, and ON with the processing of the END signal for the scan at the completion of a read/write operation.

(c) Host station completion device
: Goes ON at END completion of scan when the ZNTO instruction has been completed, and OFF the next END processing.

(d) Status display device at completion
: Goes ON and OFF depending on the status at the completion of the ZNTO instruction.
- Normal completion : Stays OFF, with no changes
- Error completion : Goes ON at END completion of scan when the ZNTO instruction has been completed, and OFF the next END processing.

[Host station operations during execution of ZNTO instruction]



(5) The link register (W) designated by (S2) is set at the link parameters as "Master to Remote" (from master station to remote I/O station), and falls within the range allocated to the CPU at the link refresh parameters.

(6) A link relay and link register for use by the OS is required to execute the ZNTO instruction.
The numbers of link relays and link registers to be used by the OS for each special function module are as follows:

- For M → R (from master station to remote I/O station)
  Link relays: 4, Link registers: 4
- For M ← R (to master station from remote I/O station)
  Link relays: 4, Link registers: 4

Operation Errors

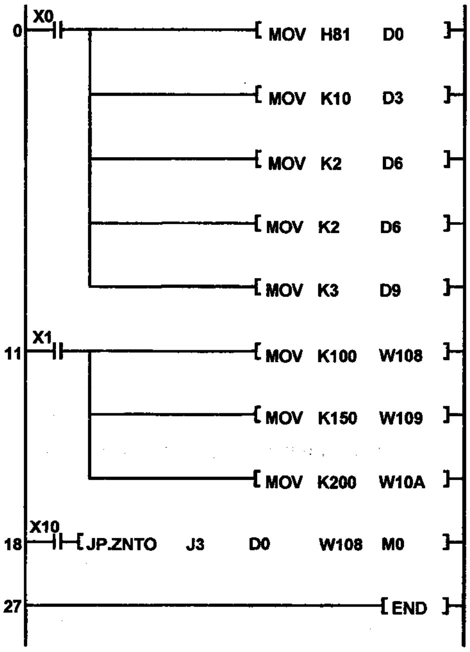
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Control data contents not within setting range (Error code: 4100)
- Network number specified by Jn is not connected to this station (Error code: 4102)
- Module for I/O number designated by Un is not a network module (Error code: 2111)

Program Example

(1) The following program writes the data at W108 through W10A to the buffer addresses 10 through 12 on the second special function module of station No. 1R2 on network No. 3.

[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	MOV	H81
		D0
3	MOV	K10
		D3
5	MOV	K2
		D5
7	MOV	K2
		D6
9	MOV	K3
		D9
11	LD	X1
12	MOV	K100
		W108
14	MOV	K150
		W109
16	MOV	K200
		W10A
18	LD	X10
19	JP.ZNTO	J3
		D0
		W108
		M0
27	END	

8.3 Instructions for A-Series Compatible Link

8.3.1 Reading device data from other stations (MELSECNET/10)

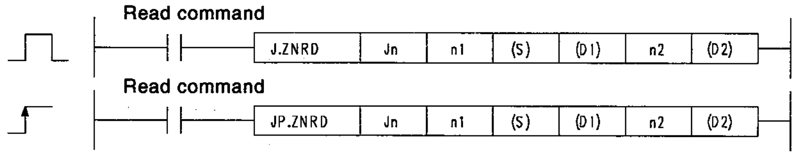
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n1	o	o	o	—				o	—
(S)	—	Δ*1	o	—				—	—
(D1)	—	o	Δ*2	—				—	—
n2	o	o	o	—				o	—
(D2)	o	o	o	—				—	—

*1: Can use only T, C, D, and W
*2: Can use only for Qn-A.

[Instruction Symbol] [Execution Condition]

ZNRD

ZNRD



Set Data

Set Data	Meaning	Data Type
Jn	Network number for host station	BIN 16 bits
n1	Object station number	
(S)	First device of station where read data is being stored	Device name
(D1)	First device of host station storing read data	
n2	Receive data length	
(D2)	Device turned ON for 1 scan on completion of instruction	Bit

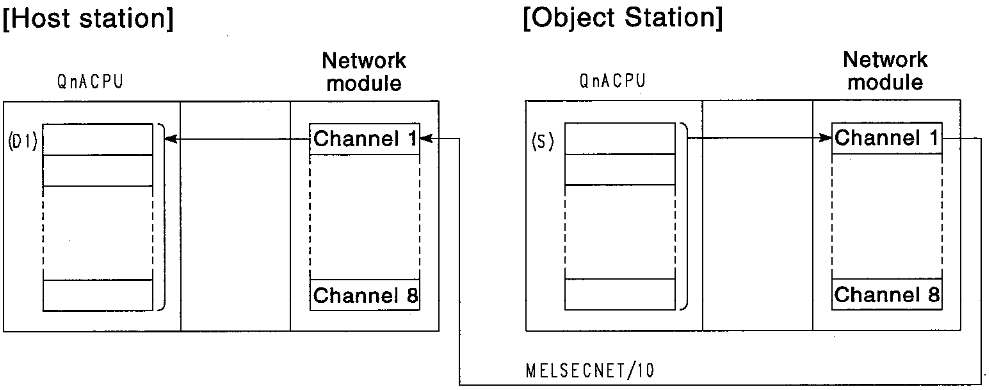
POINT

(1) When reading the data from other station using the ZNRD instruction, the parameter setting for the device must be identical between the host station and the object station.

(2) Local device and file registor for each program cannot be used as a device to use for the setting data.
(See QnACPU Programming Manual (Special function Module) for details.)

Functions

(1) Reads the data at devices starting from that designated by (S) of the network number in the MELSECNET/10 data link system that was designated by Jn and n1 to devices starting from that designated by (D1). The completion device designated by (D2) goes ON for one scan when the read operation of the device data from the object station has been completed.



(2) Device data read operations from an object station can be conducted only from a station that is connected to the same network on the MELSECNET/10 system that the host station is connected to. Note that the routing parameter must be set.

(3) Network numbers from 1 through 239 can be designated by Jn. Designation of 0 (J0) is equivalent to the designation of MELSECNET. (See Section 8.3.2).

(4) A station number from 1 to 64 can be designated at n1.

(5) The data length (number of points) that can be designated by n2 is between 1 and 230.

(6) Read operations from other stations triggered by the ZNRD instruction can be conducted with AnUCPU stations as well as with QnACPU stations.

(7) Instructions for data link cannot be executed simultaneously from more than 1 location with respect to the same channel. When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently processing will not be possible for instructions used for data links.

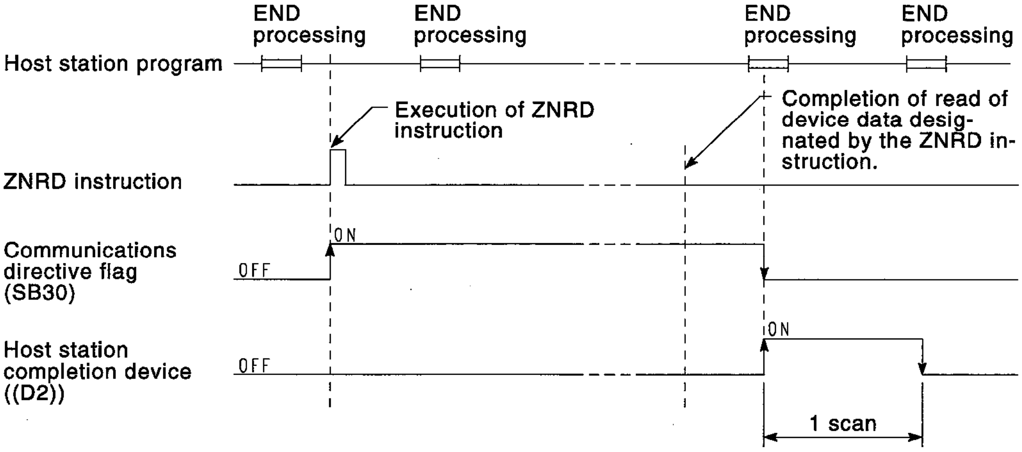
(8) Both the host station and the object station use channel 1 of the network module for the ZNRD instruction. QnA dedicated instructions using multiple ZNRD instructions and channel 1 cannot be executed simultaneously with respect to a designated network module. Establish an interlock based on the read/write request signal and completion device so that multiple instructions for data link will not be executed simultaneously.

(9) Confirmation that the execution of a ZNRD instruction is underway, and that it has been completed, can be made by use of the ZNRD directive flag (SB30) and completion device (D2).

(a) ZNRD directive flag
: Goes ON at execution of ZNRD instruction, and OFF at END processing of the scan when the read operation has been completed.

(b) Host station completion device
: Goes ON at END processing of the scan when the ZNRD instruction has been completed, and OFF at the next END processing.

[Operations of the host station during the execution of the ZNRD instruction]



(10) Normal/error completion of the ZNRD instruction can be confirmed with the ZNRD instruction processing completion register (SW31).
For normal completion, SW31 is at 0.
For error completion, the appropriate error code will be stored at SW31.*

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The n2 points from the device designated by (S1) exceeds the final device number of the relevant device range. (Error code: 4101)
- Network number designated by Jn does not exist. (Error code: 4102)
- Station number designated by n1 does not exist. (Error code: 4102)
- The reception data length designated by n2 is not in the 1 through 230 range. (Error code: 4100)

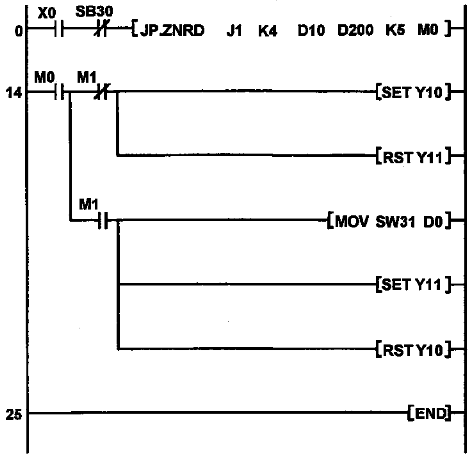
REMARK

* : See the following manual for further information on error codes at error completion:
MELSECNET/10 for QnA Network System Reference Manual

Program Example

(1) The following program reads the data from D10 through D14 at station No.4 to D200 through D204 of station No.1 while X0 is ON.

[Ladder Mode]



Normal completion display

Error code

Error completion display

[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ANI	SB30
2	JP.ZNRD	J1
		K4
		D10
		D200
		K5
		M0
14	LD	M0
15	MPS	
16	ANI	M1
17	SET	Y10
18	RST	Y11
19	MPP	
20	AND	M1
21	MOV	SW31
		D0
23	SET	Y11
24	RST	Y10
25	END	

8.3.2 Reading device data from local stations (MELSECNET)

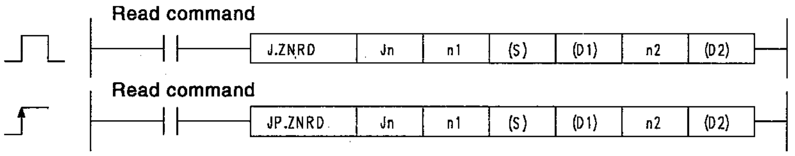
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n1	o	o	o	—				o	—
(S)	—	Δ *	—	—				—	—
(D1)	—	o	o	—				—	—
n2	—	o	o	—				o	—
(D2)	o	o	o	—				—	—

*: Can use only T, C, D, and W

[Instruction Symbol] [Execution Condition]

ZNRD

ZNRD



Set Data

Set Data	Meaning	Data Type
Jn	Host station network number (fixed at 0 (J0))	BIN 16 bits
n1	Number of object local station	
(S)	First device at local station that will store the read data	Device name
(D1)	First device of host station storing read data	
n2	Receive data length	
(D2)	Device turned ON for 1 scan on completion of instruction	Bit

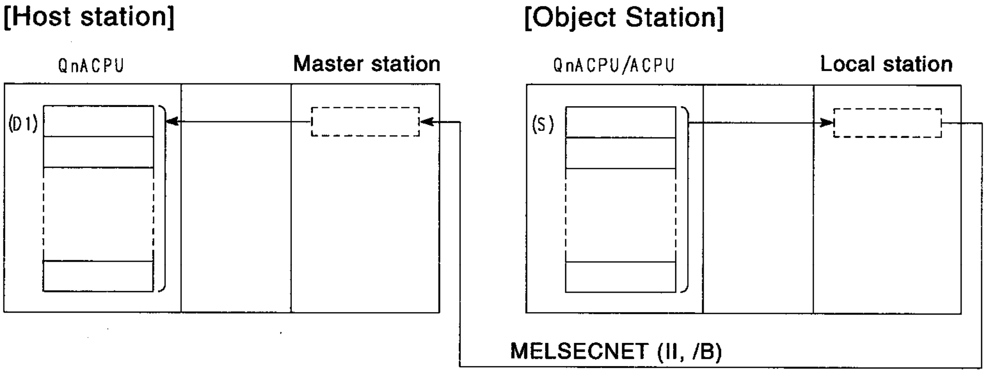
POINT

(1) When reading the data from other station using the ZNRD instruction, the parameter setting for the device must be identical between the host station and the object station.

(2) Local device and file registor for each program cannot be used as a device to use for the setting data.
(See QnACPU Programming Manual (Special function Module) for details.)

Functions

(1) Reads the data two points from the word device designated by (S) of the local station designated by n1 in the MELSECNET data link system and stores it to devices starting from that designated by (D1) at the host station.
The completion device designated by (D2) goes ON for one scan on the completion of the read operation of the device data from the local station.



(2) Device data read operations from local stations can be conducted only at a MELSECNET master station.

(3) In MELSECNET systems, the object network number (Jn) is fixed at 0(J0).
If numbers 1 through 239 are used for Jn, they must be designated from MELSECNET/10. (See Section 8.3.1.)

(4) A station number from 1 to 64 can be designated at n1.
In the case of MELSECNET/B, a station number of from 1 to 31 can be designated at n1.

(5) The device range that can be designated by (S) is indicated below.

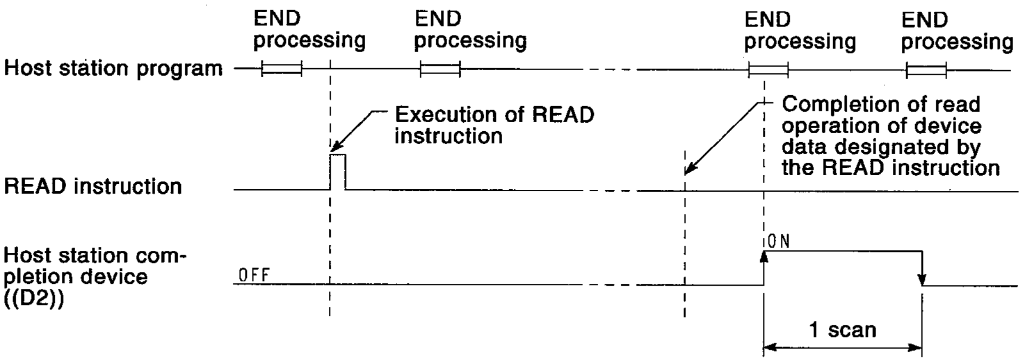
	MELSECNET Mode	MELSECNET II Mode
T	T0 to T255	T0 to T2047
C	C0 to C255	C0 to C1023
D	D0 to D1023	D0 to D6143
W	W0 to W3FF	W0 to WFFF

(6) A receive data length (number of points) of from 1 to 32 can be designated at n2.

(7) The ZNRD and ZNWR instructions cannot be executed simultaneously.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently executed ZNRD or ZNWR instructions will not be processed.
To execute multiple ZNRD and ZNWR instructions, use the completion device and execute them in succession.

(8) The completion device (D2) goes ON with the END instruction processing of the scan when ZNRD is completed, and goes OFF at the next END processing.

[Operations of the host station during the execution of the ZNRD instruction]



Operation Errors

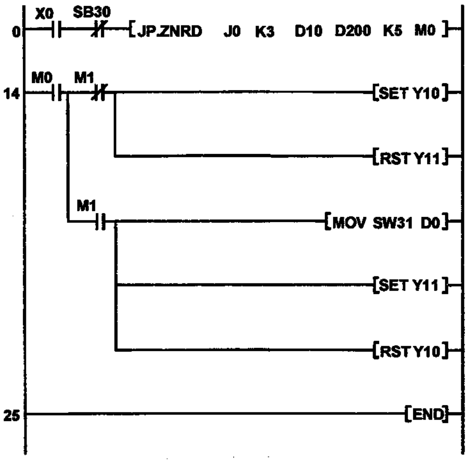
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The n2 points from the device designated by (S1) exceeds the final device number of the relevant device range. (Error code: 4101)
- Network number designated by Jn does not exist. (Error code: 4102)
- Station number designated by n1 does not exist. (Error code: 4102)
- The length of the received data designated by n2 is outside the range of from 1 to 32. (Error code: 4100)

Program Example

(1) The following program reads the data from D10 through D14 of local station No.3 to D200 through D204 of the master station while X0 at the master station is ON.

[Ladder Mode]



Normal completion display

Error code

Error completion display

[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ANI	SB30
2	JP.ZNRD	J0 K3 D10 D200 K5 M0
14	LD	M0
15	MPS	
16	ANI	M1
17	SET	Y10
18	RST	Y11
19	MPP	
20	AND	M1
21	MOV	SW31 D0
23	SET	Y11
24	RST	Y10
25	END	

8.3.3 Writing device data to other stations (MELSECNET/10)

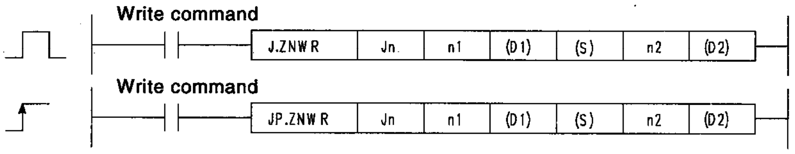
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n1	o	o	o	—				o	—
(D1)	—	Δ *1	o	—				—	—
(S)	—	o	Δ *2	—				—	—
n2	—	o	o	—				o	—
(D2)	o	o	o	—				—	—

*1: Can only use T, C, D, and W
*2: Can use only for Qn-A.

[Instruction Symbol] [Execution Condition]

ZNWR

ZNWR



Set Data

Set Data	Meaning	Data Type
Jn	Network number for host station	BIN 16 bits
n1	Object station number	
(D1)	First device at object station where data will be written	Device name
(S)	First device at host station where the data to be written is being stored	
n2	Number of points to write	
(D2)	Device turned ON for 1 scan on completion of instruction	Bit

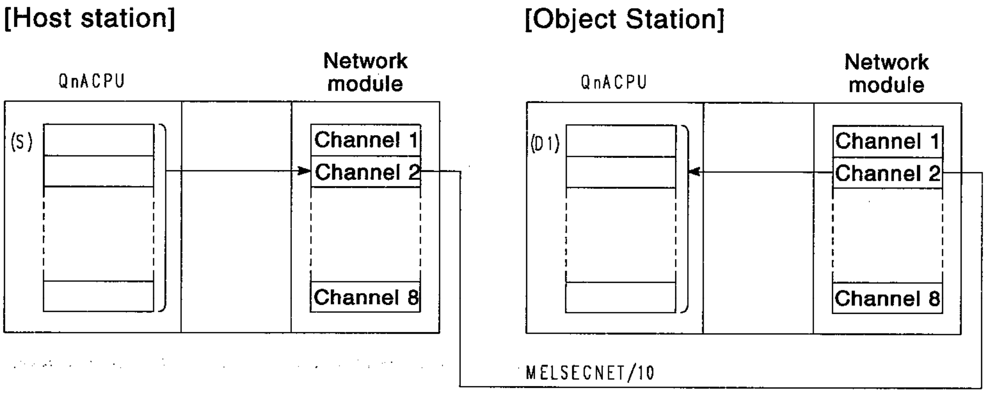
POINT

(1) When reading the data from other station using the ZNWR instruction, the parameter setting for the device must be identical between the host station and the object station.

(2) Local device and file registor for each program cannot be used as a device to use for the setting data.
(See QnACPU Programming Manual (Special function Module) for details.)

Functions

(1) Reads the data n2 points from the word device designated by (S) of the host station in the MELSECNET/10 data link system and stores it to devices starting from that designated by (D1) on the connecting station on the network number designated by Jn and n1.
When the write operation to the object station has been completed, the completion device designated by (D2) will go ON.



(2) Write operations for host station device data can be conducted only with a station connecting to the network on the MELSECNET/10 system that the host station is connected to.
Note that the routing parameter must be set.

(3) Network numbers from 1 through 239 can be designated by Jn.
Designation of 0 (J0) is equivalent to the designation of MELSECNET. (See Section 8.3.4.)

(4) A station number from 1 to 64 can be designated at n1.

(5) The send data length designated by n can be from 1 through 230.

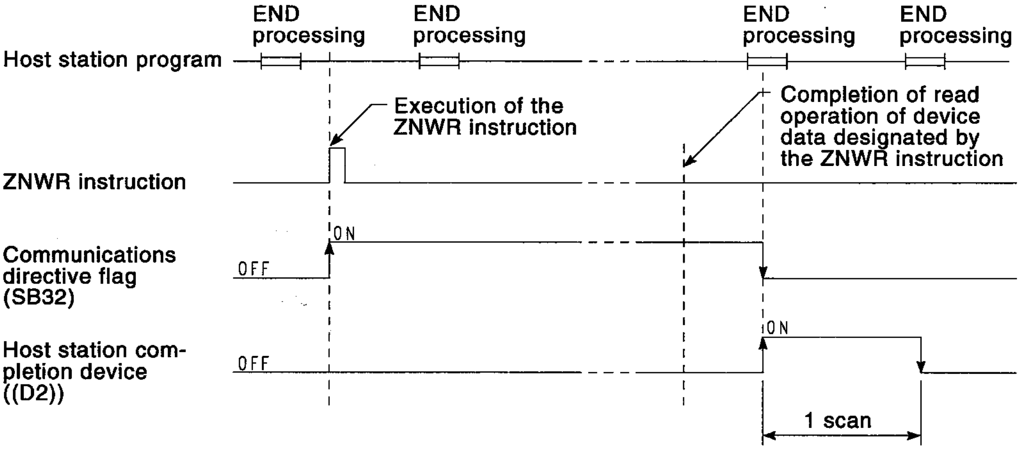
(6) Write operations to other stations triggered by the ZNWR instruction can be conducted with respect to AnUCPU object stations as well as QnACPU stations.

(7) Confirmation that the execution of a ZNWR instruction is underway, and that it has been completed, can be made by use of the ZNWR directive flag (SB32) and completion device (D2).

(a) ZNRW directive flag
: Goes ON with the execution of the ZNRW instruction, and OFF at the END processing of the scan when the write operation has been completed.

(b) Host station completion device
: Goes ON with the END processing of the scan when the ZNWR instruction is completed, and OFF at the next END processing.

[Operations of the host station during the execution of the ZNWR instruction]



(8) Normal/error completion of the ZNWR instruction can be confirmed with the ZNWR instruction processing completion register (SW33).
For normal completion, SW33 is at 0.
For error completion, the appropriate error code will be stored at SW33.*

Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The n2 points from the device designated by (S1) exceeds the final device number of the relevant device range. (Error code: 4101)
- Network number designated by Jn does not exist. (Error code: 4102)
- Station number designated by n1 does not exist. (Error code: 4102)
- The reception data length designated by n2 is not in the 1 through 230 range. (Error code: 4100)

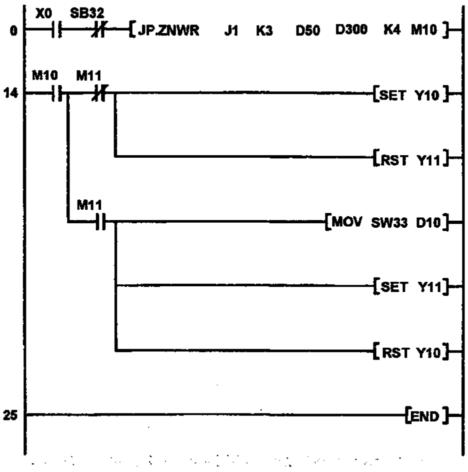
REMARK

* : See the following manual for further information on error codes at error completion:
MELSECNET/10 for QnA Network System Reference Manual

Program Example

(1) The following program writes the data at D300 through D304 of station No.2 to D50 through D53 of station No.3 when X0 of station No. 2 is ON.

[Ladder Mode]



Normal completion display

Error Code

Error completion display

[List Mode]

Steps	Instruction	Device
0	LD	X0
1	ANI	SB32
2	JP.ZNWR	J1
		K3
		D50
		D300
		K4
		M10
14	LD	M10
15	MPS	
16	ANI	M11
17	SET	Y10
18	RST	Y11
19	MPP	
20	AND	M11
21	MOV	SW33
		D10
23	SET	Y11
24	RST	Y10
25	END	

8.3.4 Writing data to devices at local stations (MELSECNET)

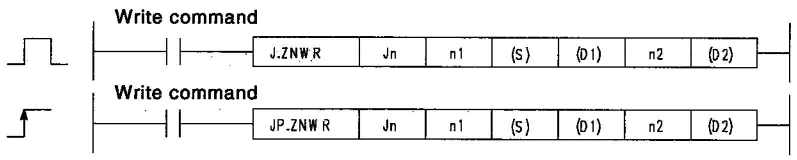
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n1	o	o	o	—				o	—
(S)	—	Δ *	—	—				—	—
(D1)	—	o	o	—				—	—
n2	—	o	o	—				o	—
(D2)	o	o	o	—				—	—

*: Can use only T, C, D, and W

[Instruction Symbol] [Execution Condition]

ZNWR

ZNWR



Set Data

Set Data	Meaning	Data Type
Jn	Object network number (fixed at 0 (J0))	BIN 16 bits
n1	Number of object local station	
(S)	First device at local station where the data is written.	Device name
(D1)	First device of host station where the data to be written is stored.	
n2	Receive data length	
(D2)	Device turned ON for 1 scan on completion of instruction	Bit

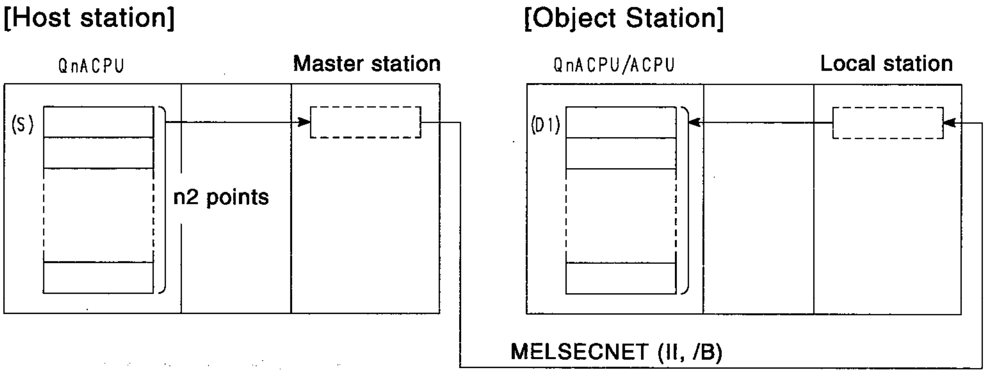
POINT

(1) When reading the data from other station using the ZNWR instruction, the parameter setting for the device must be identical between the host station and the object station.

(2) Local device and file registor for each program cannot be used as a device to use for the setting data.
(See QnACPU Programming Manual (Special function Module) for details.)

Functions

(1) Reads the data n2 points from the word device designated by (S) in the MELSECNET data link system and stores it to devices starting from that designated by (D1) of the local station designated by n1. The completion device designated by (D2) goes ON for one scan on the completion of the write operation of the device data to the local station.



(2) Device data write operations to local stations can be conducted only by a MELSECNET master station.

(3) In MELSECNET systems, the object network number (Jn) is fixed at 0(J0).
If Jn has been set at from 1 to 239, this is taken as a MELSECNET/10 designation. (See Section 8.3.3.)

(4) A station number from 1 to 64 can be designated at n1.
In the case of MELSECNET/B, a station number of from 1 to 31 can be designated at n1.

(5) The device range that can be designated by (S) is indicated below.

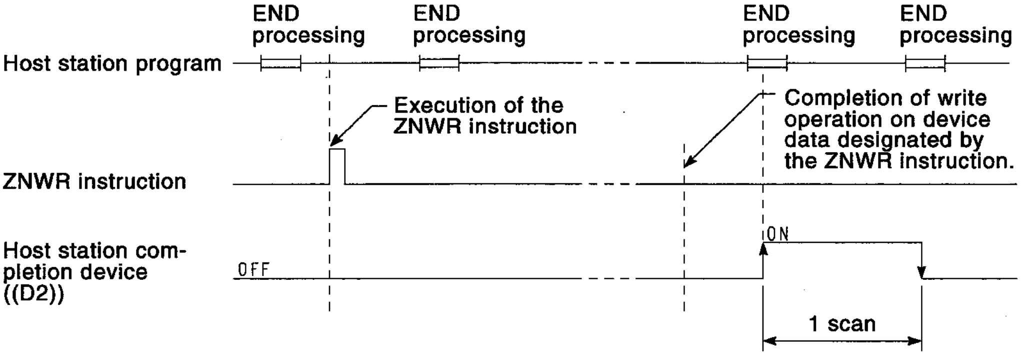
	MELSECNET Mode	MELSECNET II Mode
T	T0 to T255	T0 to T2047
C	C0 to C255	C0 to C1023
D	D0 to D1023	D0 to D6143
W	W0 to W3FF	W0 to WFFF

(6) The send data length (number of points) designated by n2 can be from 1 to 32.

(7) The ZNRD and ZNWR instructions cannot be executed simultaneously.
Engage an interlock with the completion device so that the ZNRD and ZNWR instructions will not be executed at the same time.
When executing the ZNRD and ZNWR instructions more than once, use a completion device so they can be executed successively.

(8) Completion device (D2) goes ON at the END processing of the scan when the ZNWR instruction has been completed, and goes OFF at the next END processing.

[Operations of the host station during the execution of the ZNWR instruction]



Operation Errors

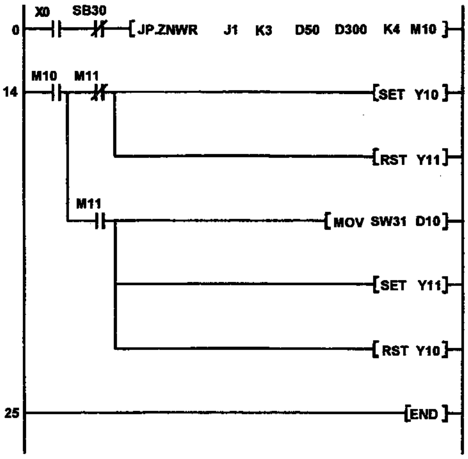
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The area 2 points from the device designated by (S) exceeds the number of the final device in the relevant device range. (Error code: 4101)
- Network number designated by Jn does not exist. (Error code: 4102)
- Station number designated by n1 does not exist. (Error code: 4102)
- The send data length designated by n2 is outside the range of from 1 through 32. (Error code: 4100)

Program Example

(1) The following program writes the data at D300 through D303 of the master station to D50 through D53 of the local station when X0 at the master station is ON.

[Ladder Mode]



Normal completion display

Error code

Error completion display

[List Mode]

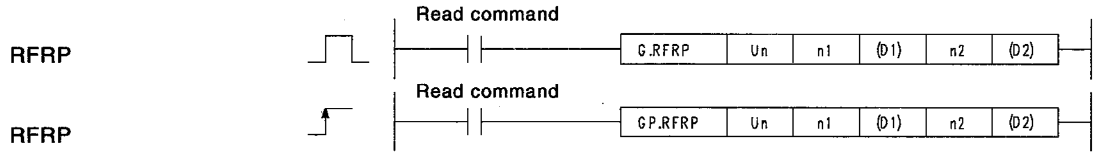
Steps	Instruction	Device
0	LD	X0
1	ANI	SB30
2	JP.ZNWR	J1
		K3
		D50
		D300
		K4
		M10
14	LD	M10
15	MPS	
16	ANI	M11
17	SET	Y10
18	RST	Y11
19	MPP	
20	AND	M1
21	MOV	SW31
		D10
23	SET	Y11
24	RST	Y10
25	END	

8.3.5 Reading data from a remote I/O station special function module (MELSECNET)

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n1	o	o	o	—				o	—
(D1)	—	Δ *	—	—				—	—
n2	o	o	o	—				o	—
(D2)	o	o	o	—				—	—

*: Can only use link register

[Instruction Symbol] [Execution Condition]

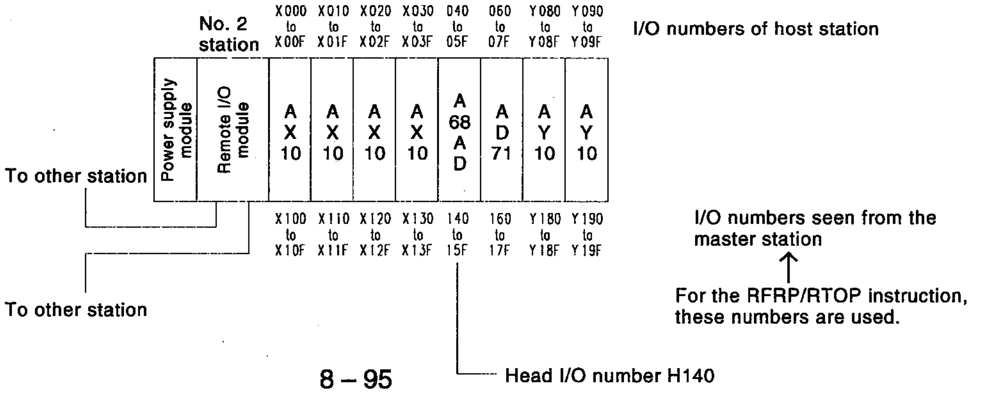


Set Data

Set Data	Meaning	Data Type
Un	Head I/O number of master station allocated to special function module that will perform data read operation	BIN 16 bits
n1	First number of buffer memory at special function module where data to be read is being stored	
(D1)	First device number of link registers of host station that will store read data	Device name
n2	Receive data length	BIN 16 bits
(D2)	Device turned ON for 1 scan on completion of instruction	Bit

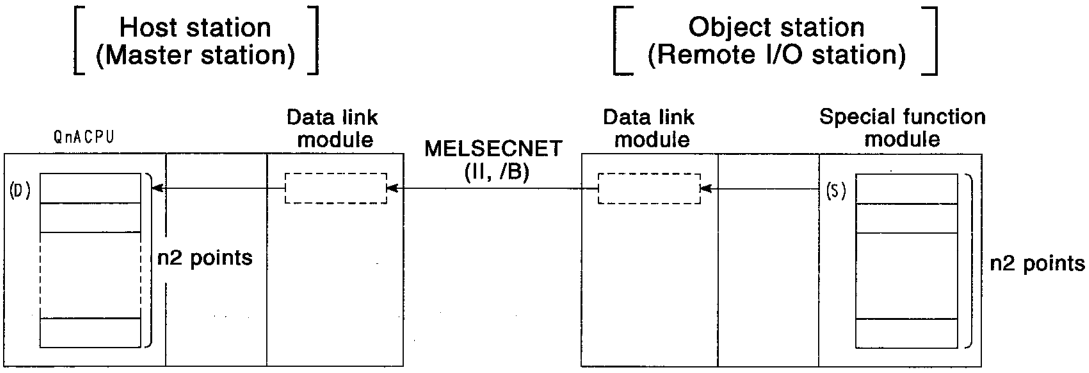
REMARKS

- Designate Un by the head I/O number of a special function module seen from the master station.
  Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnACPU Programming Manual (Special function Module) for details.)
  [Example]
  Assume that inputs and outputs of remote I/O station No. 2 are allocated to X100 to X17F and Y140 to Y190, respectively.



Functions

(1) Stores the data n2 points from the buffer memory designated by n1 of the remote I/O station special function module allocated to the I/O number designated by Un in the MELSECNET data link system to devices starting from that designated by (D1) of the host station (master station).
The completion device designated by (D2) goes ON for one scan on the completion of the read operation of the device data from the remote I/O station.

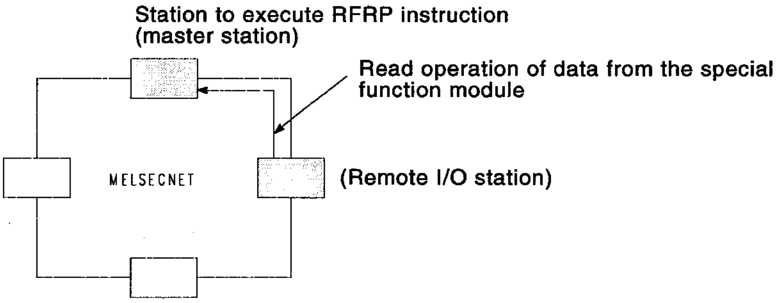


POINT

Even when conducting only a read operation from the special function module using the RFRP instruction, it is necessary to set the "master station to remote I/O station" link registers that the system will use at the link parameters.

(2) The ON/OFF control of the RFRP instruction execution flag (Y_nE) and completion flag ($X_{(n+1)}E$) is automatically performed by internal processing, and there is no need for the user to perform these controls.

(3) Device data read operations from remote I/O stations can be performed only at the MELSECNET master station.
Further, execution of the RFRP instruction can be performed only by special function modules that occupy 32 I/O points.



(4) If an error occurs in a special function module and the RFRP instruction is executed with $X_{(n+1)}D$ ON, YnD is automatically turned ON and an error reset is performed.
However, communications processing cannot be conducted during error reset processing.

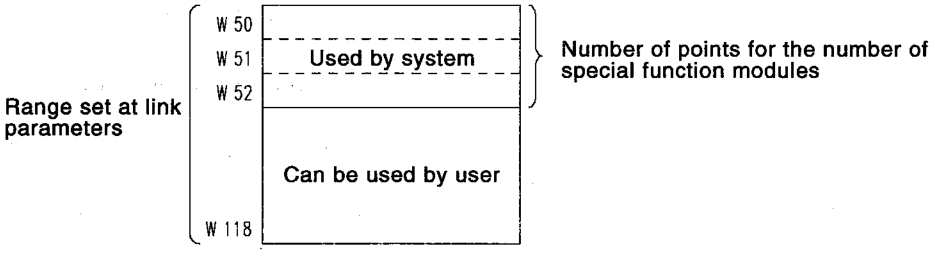
(5) Head I/O numbers for special function modules designated by Un are set as the upper three digits when the I/O numbers are expressed as four digits.
For example, in the case of X/Y0200, the setting would be 20.

(6) For additional information concerning the special function module buffer memory designated by n1, see the manual for the special function module in use.

(7) The send data length (number of points) can be set from 1 to 16 at n2.

(8) The link register that can be designated by (D1) is the range set at the link parameters as "from remote I/O station to master station".

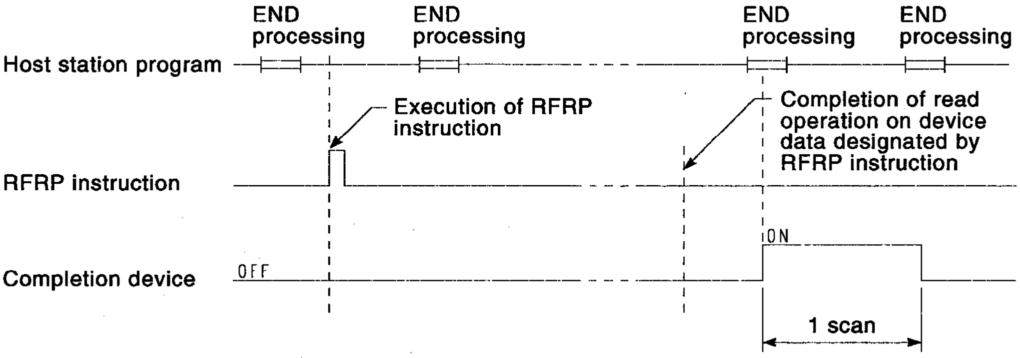
(9) An area starting from the first link register of those set at the parameter "master station → Remote I/O station", and corresponding to the number of special function modules mounted in the relevant remote I/O station, is used by the system.
For example, in a case where the setting for "master station → remote I/O stations" in the link parameters was W50 through W118, and there were three special function modules mounted at the remote I/O station in question, W50 through W52 would be used by the system.



(10) The RFRP and RTOP instructions cannot be executed simultaneously for the same special function module.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently executed RFRP or RTOP instructions will not be processed.
To execute both RFRP and RTOP instructions for the same special function module, use the completion device and execute them in succession.

(11) The completion device (D) goes ON with the END instruction processing of the scan when RFRP is completed, and goes OFF at the next END processing.

[Operations of host station during execution of the RFRP instruction]



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- The I/O number designated by n1 is not that of a remote I/O station (Error code: 4102)
- The I/O number designated by n1 is not the head I/O number of a special function module (Error code: 4102)
- The area 2 points from the device designated by (S) exceeds the number of the final device in the relevant device range. (Error code: 4101)
- The module designated by Un does not exist (Error code: 2413)
- The number of points designated by n2 is outside the 1 to 16 range (Error code: 4100)

8.3.6 Writing data to special function modules of remote I/O stations (MELSECNET)

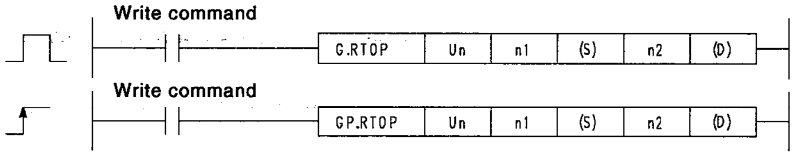
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n1	o	o	o	—				o	—
(S)	—	Δ *	—	—				—	—
n2	o	o	o	—				—	—
(D)	o	o	o	—				o	—

*: Can only use link register

[Instruction Symbol] [Execution Condition]

RTOP

RTOP

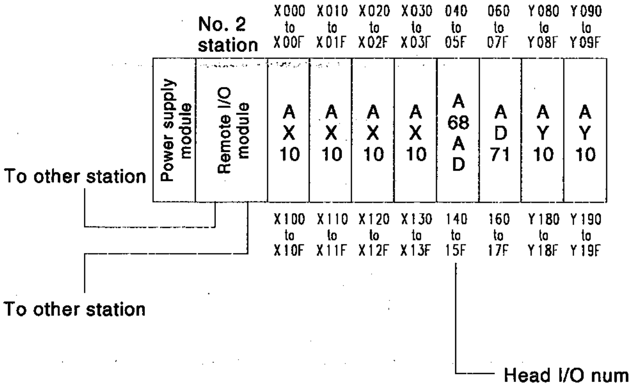


Set Data

Set Data	Meaning	Data Type
Un	Head I/O number for master station allocated to special function module that will conduct data write operation	BIN 16 bits
n1	First number of buffer memory of special function module that will store write data	
(S)	First device number of host station link registers where write data is being stored	Device
n2	Send data length	BIN 16 bits
(D)	Device turned ON for 1 scan on completion of instruction	Bit

REMARKS

- Designate Un by the head I/O number of a special function module seen from the master station.
  Local device and file registor for each program cannot be used as a device to use for the setting data. (See QnACPU Programming Manual (Special function Module) for details.)
  [Example]
  Assume that inputs and outputs of remote I/O station No. 2 are allocated to X100 to X17F and Y140 to Y190, respectively.



I/O numbers of host station

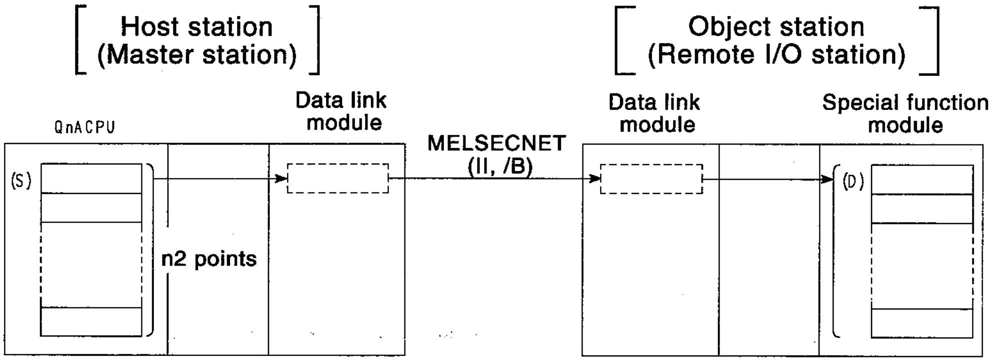
I/O numbers seen from the master station

For the RFRP/RTOP instruction, these numbers are used.

Head I/O number H140

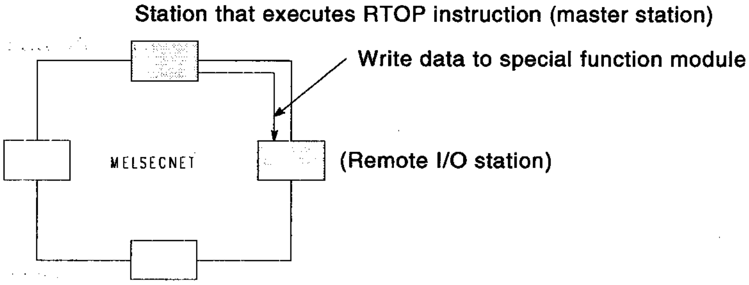
Functions

(1) Stores the data n2 points from the host station (master station) in the MELSECNET data link system designated by (S) following the buffer memory designated by (D) of the special function module of the remote I/O station allocated to the I/O number designated by Un.
When the device write operation to the remote I/O station has been completed, the completion device designated by (D) goes ON for one scan.



(2) Automatic ON/OFF controls of the RTOP instruction execution flag (YnF), and completion flag ($X_{(n+1)}$F) are conducted by internal processing for the RTOP instruction, so users do not need to perform such processing.

(3) Write operations of device data from remote I/O stations can be conducted only by the MELSECNET master station.
Further, execution of the RTOP instruction can be performed only by special function modules that occupy 32 I/O points.



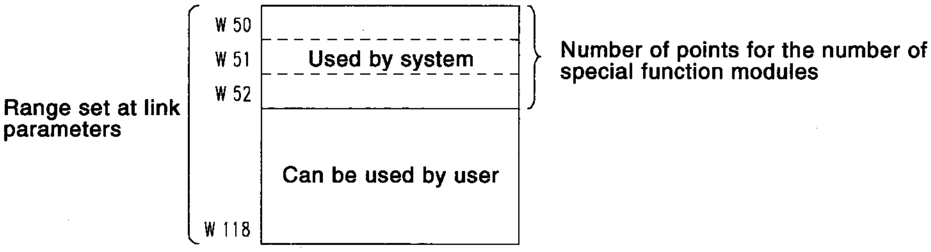
(4) If the RTOP instruction is executed when an error has occurred at the special function module and $X_{(n+1)}$D has gone ON, Y_nD will automatically go ON and an error reset operation will be performed.
However, communications processing cannot be conducted during error reset processing.

(5) Head I/O numbers for special function modules designated by Un are set as the upper three digits when the I/O numbers are expressed as four digits.
For example, in the case of X/Y0200, the setting would be 20.

(6) For additional information concerning the special function module buffer memory designated by n1, see the manual for the special function module in use.

(7) The send data length (number of points) can be set from 1 to 16 at n2.

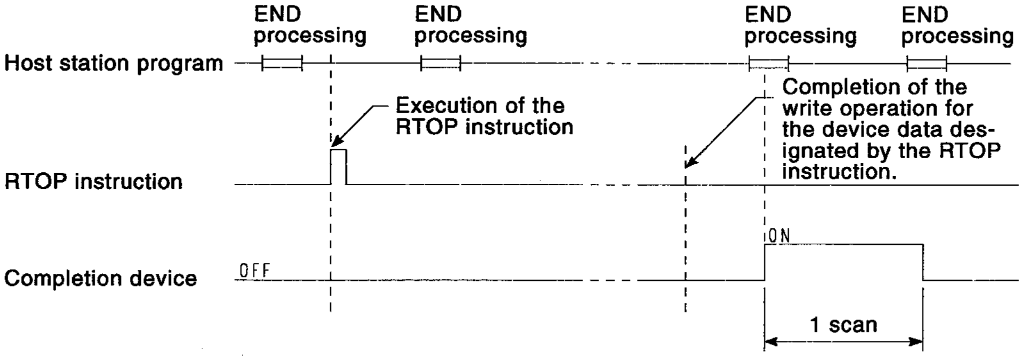
(8) Link registers that can be designated with (S) are the ranges set in the link parameters "from master station to remote I/O station" selection.
However, the area encompassing the number of special function modules mounted at the relevant remote I/O station from the link register beginning as set in the " from master station to remote I/O station" selection in the link parameters is reserved for system use.
For example, in a case where the setting for "master station → remote I/O stations" in the link parameters was W50 through W118, and there were three special function modules mounted at the remote I/O station in question, W50 through W52 would be used by the system.



(9) The RFRP and RTOP instructions cannot be executed simultaneously for the same special function module.
When conditions have been established for simultaneous execution at two or more locations, a handshake is automatically conducted, so that subsequently executed RFRP or RTOP instructions will not be processed.
If it is desired to execute multiple RFRP/RTOP instructions for the same special function module, use a completion device so they will be executed successively.

(10) The completion device (D) goes ON at the END processing of the scan when the RTOP instruction has been completed, and OFF at the next END processing.

[Operations of host station during the execution of the RTOP instruction]



Operation Errors

(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

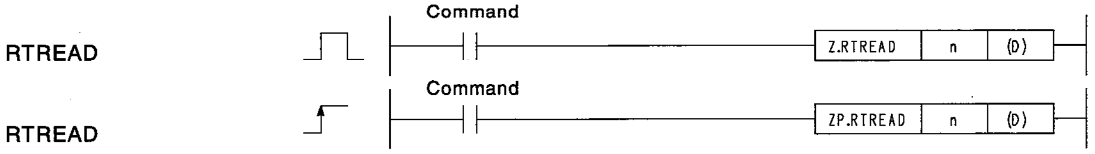
- The I/O number designated by n1 is not that of a remote I/O station (Error code: 4102)
- The I/O number designated by n1 is not the head I/O number of a special function module (Error code: 4102)
- The area 2 points from the device designated by (S) exceeds the number of the final device in the relevant device range. (Error code: 4101)
- The module designated by Un does not exist (Error code: 2413)
- The number of points designated by n2 is outside the 1 to 16 range (Error code: 4100)

8.4 Routing Information Read/Write

8.4.1 Reading routing information

Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n	—	o		—				o	—
(D)	—	o		—				—	—

[Instruction Symbol] [Execution Condition]



Set Data

Set Data	Meaning	Data Type
n	Transmission destination network number (1 to 239)	BIN 16 bits
(D)	First device number of devices which will store read data	Device name

Function

(1) Reads data from transmission destination network number designated by n, using routing information set at the routing parameters, and stores starting from (D).

(2) If no data for the transmission destination network designated by n has been set at the routing parameters, stores 0 starting from (D).

(3) The contents of the data stored starting from (D) is as indicated below.

		(Individual data ranges)
(D)+0	Relay network number	(1 to 239)
+1	Relay station number	(1 to 64)
+2	Routing station number	(1 to 64)

Operation Errors

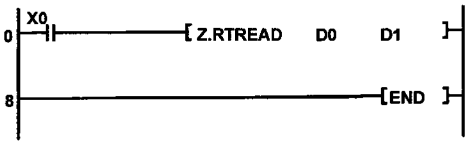
(1) In the following case an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Data designated by n is not in the range of from 1 to 239. (Error code: 4100)

Program Example

(1) The following program reads the routing information for the network number designated by D0 when X0 goes ON.

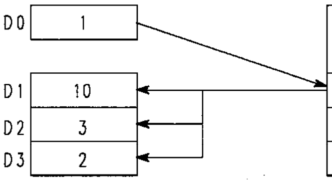
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	Z.RTREAD	D0 D1
8	END	

[Operation]



[Contents of routing parameter settings]

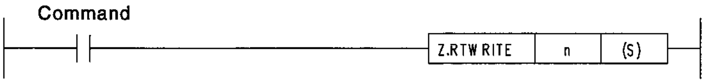
Transmission destination network number	Relay network number	Relay station number	Routing station number
1	10	3	2
2	10	2	2
3	10	1	2

8.4.2 Registering routing information

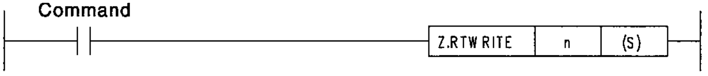
Set Data	Usable Devices								
	Internal Devices (System, User)		File Register	MELSECNET/10 Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
n	—	o		—				o	—
(S)	—	o		—				—	—

[Instruction Symbol] [Execution Condition]

RTWRITE



RTWRITE



Set Data

Set Data	Meaning	Data Type
n	Transmission destination network number (1 to 239)	BIN 16 bits
(S)	First device number of devices where data to write is being stored	Device name

Functions

(1) Registers routing data starting from (S) in the area for the destination network number designated by n in the routing parameters.

(2) If data for the destination network designated by n is set in the routing parameters, it will be used to update the data starting from (S).

(3) If all data starting from (S) is 0, the data for the destination network designated by n will be deleted from the routing parameters.

Operation Errors

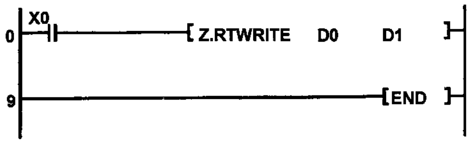
(1) In the following cases an operation error occurs, the error flag (SM0) turns ON, and an error code is stored at SD0.

- Data designated by n is not in the range of from 1 to 239. (Error code: 4100)
- The data starting from (S) exceeds individual setting ranges. (Error code: 4100)

Program Example

(1) The following program writes the routing information designated by D1 through D3 to the network module of the network number designated by D0 when X0 is ON.

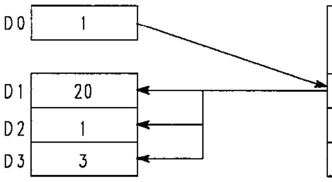
[Ladder Mode]



[List Mode]

Steps	Instruction	Device
0	LD	X0
1	Z.RTWRITE	D0 D1
9	END	

[Operation]



[Contents of routing parameter settings]

Transmission destination network number	Relay network number	Relay station number	Routing station number
1	20	1	3
2	10	2	2
3	10	1	2

MEMO

9. ERROR CODES

If an error occurs when the power to the programmable controller is turned ON, when the programmable controller is set to the RUN state, or during RUN, the QnACPU self diagnostic function indicates the error (LED indication, message on the LED display), and stores the error information at special relays, SM, and special registers, SD.
The following information deals with the contents of errors that might occur at the QnACPU, and the corrective measures that should be taken to deal with these errors.

9.1 How to Read Error Codes

When errors occur, it is possible to read the error codes, error messages, and other information at peripheral devices capable of GPP functions. Details on the operation of such peripheral devices can be found in the SW[ ]IVD-GPPQ Operating Manual (Online) of the peripheral device.

9.2 Error Code List

The following information deals with error codes and the meanings, causes, and corrective measures of error messages.

Error Code (SD0) *1	Error Messages	Common Information (SD5 to 15) *1	Individual Information (SD16 to 26) *1	LED Status		Operating States of CPU	Diagnostic Timing
				RUN	ERROR		
1000	MAIN CPU DOWN	—	—	Off	Flicker	Stop	Continual
1010	END NOT EXECUTE	—	—	Off	Flicker	Stop	When an END instruction is executed.
1011							
1012							
1101	RAM ERROR	—	—	Off	Flicker	Stop	At power ON/At reset
1102							
1103							
1104							
1200	OPE. CIRCUIT ERR.	—	—	Off	Flicker	Stop	At power ON/At reset
1201							
1202							
1203							When END instruction executed
1204							
1205							
1206							When instruction executed
1300	FUSE BREAK OFF	—	—	Off/On	Flicker/On	Stop/ Continue *2	When an END instruction is executed.
1310	I/O INT ERROR	Unit/module No.	—	Off	Flicker	Stop	During interrupt
1401	SP. UNIT DOWN	Unit/module No.	Program error location	Off	Flicker	Stop	At power ON/At reset
1402							During execution of FROM/TO instruction set
1411	CONTROL-BUS ERR.	Unit/module No.	Program error location	Off	Flicker	Stop	At power ON/At reset
1412							During execution of FROM/ TO instruction set
1421	SYS. UNIT DOWN *3	—	—	Light off	Light flashing	Stop	Always
1500	AC DOWN	—	—	On	Off	Continue	Continual
1510	DUALL DOWN 5VDC *4	—	—	Light on	Light on	Continue	Always
1520	DOWN 5VDC *5	—	—	Lihgt off	Light flashing	Stop	Always
1530	DOWN 24VDC *3	—	—	Light on	Light on	Continue	Always

*1 Characters in parentheses ( ) indicate the special register numbers where individual information is being stored.
*2 The CPU operation status when an error occurs can be set at the parameters. (LED display will change accordingly.)
*3 This can only be detected in resundant systems. Detection is possible in either the control system or the standby system.
*4 This can only be detected in the resundant system control system.
*5 This can be detected in either a sta dalone system or a redundant system. However, in a redundant sysytem it can only be detected in the contorol system.

Error Code (SD0) *1	Error Contents and Cause	Corrective Action
1000	Run-away or failure of main CPU (1) Malfunctioning due to noise or other reason (2) Hardware fault	(1) Measure noise level. (2) Reset and establish the RUN status again. If the same error is displayed again, this suggests a CPU hardware error. Contact your nearest Mitsubishi representative.
1010	Entire program was executed without the execution of an END instruction.	(1) Measure.noise level. (2) Reset and establish the RUN status again. If the same error is displayed again, this suggests a CPU hardware error. Contact your nearest Mitsubishi representative.
1011	(1) When the END instruction is executed it is read as another instruction code, e.g. due to noise.	
1012	(2) The END instruction has been changed to another instruction code somehow.	
1101	Error in internal RAM where CPU sequence program is stored.	This suggests a CPU hardware error. Contact your nearest Mitsubishi representative.
1102	Error in RAM used as work area in CPU	
1103	Internal CPU device error	
1104	Address RAM error in CPU	
1200	Operation circuit which performs CPU internal index qualification operation does not operate normally.	
1201	Internal CPU hardware (logic) does not operate normally.	
1202	The operation circuit that executes sequence processing in the CPU does not operate normally.	
1203	The operation circuit that conducts index qualification in the CPU is not operating properly.	
1204	The hardware (logic) in the CPU us not operating properly.	
1205	The operation circuit that conducts sequence processing in the CPU in not operating properly.	
1206	The DSP operation circuit in the CPU is not operating properly.	
1300	(QnA, A4AR) There is an output module with a blown fuse.	(1) Check the blown fuse indicator LEDs of the output modules, and change the fuse at the module whose LED is lit. (2) Read the error common information at a peripheral device, and change the fuse at the output module corresponding to the numerical value (module number) read. Alternatively, monitor the special registers SD1300 to SD1331 at a peripheral device, and change the fuse at the output module whose bit has a value of "1".
	(Q2AS) (1) There is an output module with a fuse blown. (2) External power supply for output load is turned off or disconnected.	(1) Check the ERR LED of output module and replace the module that is turned on. (2) The moudule with a blown fuse can be checked with a peripheral device. Monitor the special registers D9100 to D9107 and check if there if a bit "1," which corresponds to the module with a blown fuse. (3) Check whether the external power supply for output load is On or Off.
1310	An interruption has occurred although there is no interrupt module.	One of the individual modules is experiencing hardware problems, so check the modules. Contact your nearest Mitsubishi representative and explain the problem with the defective module.
1401	When parameter I/O allocation was being made, there was no return signal from the special function module during initial communications stage. When error is generated, the initial I/O number of the special function module that corresponds to the common information is stored.	The special function module that was being accessed is experiencing hardware problems. Contact your nearest Mitsubishi representative.
1402	The special function module was accessed during the execution of a FROM/TO instruction set, but there was no response. When an error is generated, the program error location corresponding to the individual information is stored.	
1411	When parameter I/O allocation was being made, the special function module could not be accessed during the initial communications stage. When error is generated, the initial I/O number of the special function module that corresponds to the common information is stored.	A special function module, the CPU module, or the base unit is experiencing problems. Contact your nearest Mitsubishi representative.
1412	The FROM/TO instruction set could not be executed, due to control pulse error with special function module. When an error is generated, the program error location corresponding to the individual information is stored.	
1421	System management module AS92R hardware fault.	This is a hardware fault, to explain the sysmbol to the nearest service center, dealer, or branch office and consult with them.
1500	A momentary power interruption of the power supply occurred. The power supply went off.	Check the power supply.
1510	The 5VDC supplied voltage of one of two power modules in the redundant system extension base has dropped below 80percent of the rated voltage.	Check the supply voltage of the power module. If the voltage is abnoemal then replace the power module.
1520	The 5VDC supply voltage of the extension base power module has dropped below 80 percent of therated voltage.	
1530	The 24VDC power voltage supplied to the system management module AS92R has dropped below 80 percent of the rated voltage.	Check the power supply.

*1 Characters in parentheses ( ) indicate the special register numbers where individual information is being stored.

Error Code List (Continued)

Error Code (SD0)	Error Messages	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		Operating States of CPU	Diagnostic Timing
				RUN	ERROR		
1600	BATTERY ERROR	Drive Name	——	On	Off	Continue	Continual
1601				BAT.ALM LED On			
1602							
2000	UNIT VERIFY ERR.	Unit/module No.	——	Off/On	Flicker/On	Stop/ Continue *2	When an END instruction is executed.
2100	SP. UNIT LAY ERR.	Unit/module No.	——	Off	Flicker	Stop	At power ON/At reset
2101							
2102							
2103							
2104							
2105	SP. UNIT LAY ERR.	Unit/module No.	——	Off	Flicker	Stop	At power ON/At reset
2106							
2107							
2108							
2109*6							
2110	SP. UNIT ERROR	Unit/module No.	Program error location	Off/On	Flicker/On	Stop/ Continue *2	During execution of instruction set
2111							
2112							
2113		FFFFH (fixed)					
2200	MISSING PARA.	Drive Name	——	Off	Flicker	Stop	At power ON/At reset
2210	BOOT ERROR	Drive Name	——	Off	Flicker	Stop	At power ON/At reset
2300	ICM. OPE. ERROR	Drive Name	——	Off/On	Flicker/On	Stop/ Continue *2	When memory card is inserted or removed
2301							
2302							

*2 The CPU operation status when an error occurs can be set at the parameters. (LED display will change accordingly.)
*6 This can only be detected in the redundant system standby system.

Error Code (SD0)	Error Contents and Cause	Corrective Action
1600	(1) Voltage in the CPU main unit battery has dropped below stipulated level. (2) The lead connector of the CPU main unit battery has not been installed.	(1) Change the battery. (2) If the battery is for internal RAM or for the back-up power function, install a lead connector.
1601	Voltage of the battery on memory card 1 has dropped below stipulated level.	Change the battery.
1602	(QnA, Q4AR) Voltage of the battery on memory card 2 has dropped below stipulated level.	
2000	I/O module information when power was turned ON was wrong. · While operation is in progress, an I/O module (or a special function module) has begun to come loose, or has come loose.	Read the error common information at the peripheral device, and check and/or change the module that corresponds to the numerical value (module number) there. Alternatively, monitor the special registers SD1400 to SD1431 at a peripheral device, and change the fuse at the output module whose bit has a value of "1".
2100	In parameter I/O allocation settings, a special function module was allocated to a location reserved for an I/O module. Or, the opposite has happened.	Reset the parameter I/O allocation setting to conform with the actual status of the special function modules.
2101	13 or more special function modules (not counting the AI61) capable of sending an interrupt to the CPU module have been installed.	Keep the number of special function modules that can initiate an interrupt (with the exception of the AI61 module) to 12 or fewer.
2102	7 or more computer link modules (excludes A(1S)J71QC24) have been installed.	Keep the number of computer link modules (excludes A(1S)J71QU24) installed to 6 or fewer.
2103	Two or more AI61 interrupt modules have been installed.	Install only 1 AI61 module.
2104	At the MELSECNET/MINI auto refresh parameter settings, the module allocation that was set is different from the actual module models at the station numbers in the link system.	Reset the parameter MELSECNET/MINI auto refresh unit module allocation setting so that it conforms to the station number of the module that is actually linked.
2105	There are too many special function modules that can use dedicated instructions allocated (number of modules installed). (The total of the figures indicated below is above 1344.) (Number of AD59 modules installed x 5) (Number of AD57(S1)/AD58 modules installed x 8) (Number of AJ71C24(S3/S6/S8) modules installed x 10) (Number of AJ71UC24 modules installed x 10) (Number of AJ71C21(S1) modules installed x 29) (Number of AJ71PT32-S3/AJ71T32-S3 modules installed x 125) (Number of AJ71QC24(R2,R4) modules installed x 29) (Number of AJ71ID1(2)-R4 modules installed x 18) (Number of AD75 modules installed x 12) Total > 1344	Reduce the number of special function modules installed. * : When the expansion mode is used.
2106	(1) 5 or more AJ71QLP21 or AJ71QBR11 modules have been installed. (2) 3 or more AJ71AP21/R21 or AJ71AT21B modules have been installed. (3) A total of 5 or more AJ71QLP21, AJ71QBR11, AJ71AP21/R21, or AJ71AT21 modules has been installed. (4) Identical network numbers or station numbers exist in the MELSECNET/10 network system. (5) There are 2 or more master stations or local stations in the MELSECNET (II) or MELSECNET/B data link system at the same time.	(1) Keep at 4 or fewer (2) Keep at 2 or fewer (3) Keep total at 4 or fewer (4) Check network number and station number. (5) Check station number.
2107	Head X/Y set at the parameter I/O allocation settings is also the head X/Y for some other module.	Reset the parameter I/O allocation setting to conform with the actual status of the special function modules.
2108	AJ71LP21 or AJ71BR11 for use with the AnUCPU network module has been installed.	Change network module to AJ71QLP21 or AJ71QBR11.
2109[*6]	The control system and standby system module configurations are diffierent when a redundant system is in the backup mode.	Check the redundant systems module configuration.
2110	(1) The location designated by the FROM/TO instruction set is not a special function module. (2) The special function module being accessed is faulty.	(1) Read error individual information, then check and edit the FROM/TO instruction set that corresponds to the numerical value there (program error location). (2) The special function module that is being accessed has a hardware error. Consult the nearest service center, agent or our branch office and describe the symptom.
2111	The location designated by link direct device (J□\□) is not a network module.	
2112	The location designated by a special function module dedicated instruction is not a special function module. Alternatively, it is not the relevant special function module.	Read error individual information, then check and edit the special function module dedicated instruction that corresponds to the numerical value there (program error location).
2113	No special function module data for simulation purposes has been set in the simulation data.	Read error individual information, then check and edit the special function module simulation data that corresponds to the numerical value there (program error location).
2200	There is no parameter file at the drive designated by DIP switches as a valid drive switch.	Check and correct the setting of the parameter enabled drive switch. Put a parameter file in the drive designated by the parameter enabled drive switch.
2210	There is no boot file in the drive designated by the parameter enabled drive switch even though the DIP switch boot switch is ON.	Check and correct the setting of the parameter enabled drive switch. Put a boot file in the drive designated by the parameter enabled drive switch.
2300	(1) The memory card has been removed without placing the memory card in/out switch OFF. (2) The card insert switch is turned ON although a memory card is not actually installed.	(1) Remove memory card after placing the memory card in/out switch OFF. (2) Turn on the card insert switch after inserting a memory card.
2301	(1) The memory card has not been formatted. (2) Memory card format status is incorrect.	(1) Format memory card. (2) Reformat memory card.
2302	A memory card that cannot be used with the QnACPU has been installed.	Check memory card.

Error Code List (Continued)

Error Code (SD0)	Error Messages	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		Operating States of CPU	Diagnostic Timing
				RUN	ERROR		
2400	FILE SET ERROR	File name	Parameter number	Off	Flicker	Stop	At power ON/At reset
2401							
2410	FILE OPE. ERROR	File name	Program error location	Off/On	Flicker/On	Stop/ Continue *2	When instruction is executed
2411							
2412							
2413							
2500	CAN'T EXE. PRG.	File name	——	Off	Flicker	Stop	At power ON/At reset
2501							
2502							
2503							
2504							
3000	PARAMETER ERROR	File name	Parameter number	Off	Flicker	Stop	At power ON/Reset/ STOP → RUN
3001							
3003							
3004							
3100	LINK PARA. ERROR	File name	Parameter number	Off	Flicker	Stop	At power ON/Reset/ STOP → RUN
3101							
3102							
3103							
3104							
3105							
3107							
3200	SFG PARA. ERROR	File name	Parameter number	Off	Flicker	Stop	STOP → RUN
3201							
3202							
3203							
4000	INSTRCT CODE ERR.	Program error location	——	Off	Flicker	Stop	At power ON/Reset/ STOP → RUN
4001							
4002							
4003							
4004							

*2 The CPU operation status when an error occurs can be set at the parameters. (LED display will change accordingly.)

Error Code (SD0)	Error Contents and Cause	Corrective Action
2400	The file designated at the PC file settings in the parameters cannot be found.	Read the error individual information at the peripheral device, check to be sure that the parameter drive name and file name correspond to the numerical values there (parameter number), and correct. Create the designated file.
	Ethernet parameter that was added for QnACPU with the function version "B," has been set to QnACPU without the function version "B."	Change to QnACPU with the function version "B." Delete the Ethernet parameter.
2401	The file designated at the parameter PC RAS settings fault history area has not been created.	Read the error individual information at the peripheral device, check to be sure that the parameter drive name and file name correspond to the numerical values there (parameter number), and correct. Check the space remaining in the memory card.
2410	The file designated by the sequence program cannot be found.	Read the error individual information at the peripheral device, check to be sure that the program corresponds to the numerical values there (program location), and correct. Create the designated file.
2411	The file is a file that cannot be designated by the sequence program (comment file, etc.).	Read the error individual information at the peripheral device, check to be sure that the program corresponds to the numerical values there (program location), and correct.
2412	The SFC program file is one that cannot be designated by the sequence program.	Read the error individual information at the peripheral device, check to be sure that the program corresponds to the numerical values there (program location), and correct.
2413	No data has been written to the file designated by the sequence program.	Read the error individual information at the peripheral device, check to be sure that the program corresponds to the numerical values there (program location), and correct. Check to ensure that the designated file has not been write protected.
2500	There is a program file that uses a device that is outside the device allocation range that has been designated at the parameter device settings.	Read the error common information at the peripheral device, check to be sure that the parameter device allocation setting and the program file device allocation correspond to the numerical values there (file name), and correct if necessary.
2501	There are multiple program files although "none" has been set at the parameter program settings.	Edit the parameter program setting to "yes". Alternatively, delete unneeded programs.
2502	The program file is not a QnACPU program file. Alternatively, the file contents are not those of a sequence program.	Check whether the program version is ***.QPG, and check the file contents to be sure they are for a sequence program.
2503	There are no program files at all.	Check program configuration.
2504	Two or more SFC normal programs or control programs have been designated.	Check parameters and program configuration.
3000	The parameter settings for timer time limit setting, the RUN-PAUSE contact, the common pointer number, the general data processing, number of vacant slots, or system interrupt settings are outside the range that can be used by the CPU.	(1) Read the error detailed information at the peripheral device, check the parameter items corresponding to the numerical values (parameter numbers) there, and correct when necessary. (2) If the error is still generated following the correction of the parameter settings, it is likely that there is a memory error, either in the internal CPU RAM or on the memory card. Contact your nearest Mitsubishi representative.
3001	Parameter contents have been destroyed.	
3003	The number of devices set at the parameter device settings is outside the range that the CPU can use.	
3004	The parameter file is not applicable for QnACPU. Alternatively, the contents of the file are not parameters.	Check whether the parameter file version is ***.QPA, and check the file contents to be sure they are parameters.
3100	Network parameters have not been written although the QnACPU is the control station, or the master station.	(1) Write after correcting network parameters. (2) If the error persists after corrections have been made, contact your nearest Mitsubishi representative.
3101	There is an error in the refresh parameters.	
3102	An error was discovered when the network parameter check was made at the network module.	
3103	AJ71QE71 does not exist in the position of I/O number set by the parameter. I/O number designation is overlapping. Numbers of the parameter and loaded AJ71QE71 are different. Ethernet (parameter + dedicated instruction) is set to more than 5.	
3104	Ethernet and MELSECNET/10 use the same network number. Network number, station number and group number set by the parameter is out of range. I/O number is out of range of CPU used.	
3105	The contents of the parameter peculiar to Ethernet are abnormal.	
3107	The contents of the CC-Link parameter are abnormal.	
3200	Parameter contents are incorrect.	Write after correcting parameters.
3201	Contents of SFC block attributes information are incorrect.	
3202	The number of step relays designated in the parameters is less than the number used by the program.	
3203	The execution type set for an SFC program in the parameters is other than scan execution type.	
4000	An instruction code that cannot be decoded has been included in the program.	Read common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4001	Program includes a dedicated instruction for SFC program although it is not an SFC program.	
4002	The extension instruction designated by the program has an incorrect instruction name.	
4003	The extension instruction designated by the program has an incorrect number of devices.	
4004	The extension instruction designated by the program designates a device which cannot be used.	

Error Code List (Continued)

Error Code (SD0)	Error Messages	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		Operating States of CPU	Diagnostic Timing
				RUN	ERROR		
4010	MISSING END INS.	Program error location	—	Off	Flicker	Stop	At power ON/Reset/ STOP → RUN
4020	CAN'T SET (P)	Program error location	—	Off	Flicker	Stop	At power ON/Reset/ STOP → RUN
4021							
4030	CAN'T SET (I)	Program error location	—	Off	Flicker	Stop	At power ON/Reset/ STOP → RUN
4100	OPERATION ERROR	Program error location	—	Off/On	Flicker/On	Stop/ Continue [*2]	When instruction is executed
4101							
4102							
4103							
4107							
4108							
4200	FOR NEXT ERROR	Program error location	—	Off	Flicker	Stop	When instruction is executed
4201							
4202							
4203							
4210	CAN'T EXECUTE (P)	Program error location	—	Off	Flicker	Stop	When instruction is executed
4211							
4212							
4213							
4220	CAN'T EXECUTE (I)	Program error location	—	Off	Flicker	Stop	When instruction is executed
4221							
4223							
4230	INST. FORMAT ERR.	Program error location	—	Off	Flicker	Stop	When instruction is executed
4231							
4235							
4300	EXTEND INST. ERR.	Program error location	—	Off/On	Flicker/On	Stop/ Continue [*2]	When instruction is executed
4301							
4305							
4400	SFCP. CODE ERROR	Program error location	—	Off	Flicker	Stop	STOP → RUN
4410	CAN'T SET (BL)	Program error location	—	Off	Flicker	Stop	STOP → RUN
4411							
4420	CAN'T SET (S)	Program error location	—	Off	Flicker	Stop	STOP → RUN
4421							
4422							
4500	SFCP. FORMAT ERR.	Program error location	—	Off	Flicker	Stop	STOP → RUN
4501							
4502							
4503							
4504							

*2 The CPU operation status when an error occurs can be set at the parameters. (LED display will change accordingly.)

Error Code (SD0)	Error Contents and Cause	Corrective Action
4010	There is no END (FEND) instruction in the program.	Read the error common information at a peripheral device, check the files corresponding to the numerical values there (program error location), and correct the problem.
4020	The total number of internal file pointers used by the program exceeds the number of internal file pointers set at the parameters.	Read the common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4021	Numbers for the common pointers used by individual files are overlapping.	
4030	Numbers for the allocation pointers used by individual files are overlapping.	Read the common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4100	Data which cannot be dealt with by the instruction is included.	Read the common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4101	The number of uses set for data which can be dealt with by the instruction exceeds the usable range. Alternatively, the storage data or constants for the devices designated by the instruction exceeds the usable range.	
4102	The network number or station number designated by a network dedicated instruction is incorrect.	
4103	The configuration of the PID dedicated instruction is incorrect.	
4107	Numbers of execution to the CC-Link instruction are beyond 64.	Set the numbers of execution to the CC-Link instruction to 64 or less.
4108	The CC-Link parameter is not set when the CC-Link instruction is executed.	Execute the CC-Link instruction after setting the CC-Link parameter.
4200	No NEXT instruction was executed following the execution of a FOR instruction. Alternatively, there are fewer NEXT instructions than FOR instructions.	Read the common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4201	A NEXT instruction was executed although no FOR instruction has been executed. Alternatively, there are more NEXT instructions than FOR instructions.	
4202	More than 16 nesting levels have been inserted.	Keep nesting levels at 16 or under.
4203	A BREAK instruction was executed although no FOR instruction was executed.	Read the common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4210	CALL instruction was executed, but no destination pointer was found.	Read the common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4211	There was no RET instruction in the executed sub-routine program.	
4212	The RET instruction was before the FEND instruction in the main program.	
4213	More than 16 nesting levels have been inserted.	Keep nesting levels at 16 or under.
4220	Interrupt input was generated, but no corresponding interrupt pointer was found.	Read the common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4221	There was no IRET instruction in the executed interrupt program.	
4223	The IRET instruction was before the FEND instruction in the main program.	
4230	There is a one-to-one relationship between the number of CHK and CHKEND instructions.	Read the common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4231	There is a one-to-one relationship between the number of IX and IXEND instructions.	
4235	The configuration of the check conditions for the CHK instruction is incorrect. Alternatively, a CHK instruction has been used inside a low-speed program.	
4300	The designation of a MELSECNET/MINI-S3 master module control instruction was wrong.	Read the common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4301	The designation of an AD57/AD58 control instruction was wrong.	
4305	The designation of the CC-Link instruction was wrong.	
4400	No SFCP or SFCPEND instruction in SFC program.	Read common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4410	The block number designated by the SFC program exceeds the maximum setting value.	
4411	Block number designations overlap in SFC program.	
4420	Step number designated in SFC program exceeds step No. 511.	
4421	Total number of steps in all SFC programs exceed maximum value.	Reduce total number of steps to below the maximum.
4422	Step number designations overlap in SFC program.	Read common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4500	The number of BLOCK and BEND instructions in SFC program is not in a one-to-one relationship.	
4501	The configuration of the STEP* to TRAN* to TSET to SEND instructions in the SFC program is incorrect.	
4502	There was no STEPI* instruction in SFC program block.	
4503	The step designated by the TSET instruction in the SFC program does not exist.	
4504	The step designated by the TAND instruction in the SFC program does not exist.	

Error Code List (Continued)

Error Code (SD0)	Error Messages	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		Operating States of CPU	Diagnostic Timing
				RUN	ERROR		
4600	SFCP. OPE. ERROR	Program error location	——	Off/On	Flicker/On	Stop/ Continue *2	When instruction is executed
4601							
4602							
4610	SFCP. EXE. ERROR	Program error location	——	On	On	Continue	STOP → RUN
4611							
4620	BLOCK EXE. ERROR	Program error location	——	Off	Flicker	Stop	When instruction is executed
4621							
4630	STEP EXE. ERROR	Program error location	——	Off	Flicker	Stop	When instruction is executed
4631							
4632							
4633							
5000	WDT ERROR	Time (value set)	Time (value actually measured)	Off	Flicker	Stop	Continual
5001							
5010	PRG. TIME OVER	Time (value set)	Time (value actually measured)	On	On	Continue	Continual
5011							
6000	PRG. VERIFY ERR.	File name	——	Light off	Light flashing	Continue	Always
6010	MODE VERIFY ERR.	——	——	Light on	Light on	Continue	Always
6100	TRK. MEMORY ERR.	——	——	Light on	Light on	Continue	When power on/ reset/STOP→RUN
6101							When END instruction executed
6200	CONTROL EXE.*4	Cause of switch	——	Light on	Light off	Continue	Always
6210	CONTROL WAIT. *6	Cause of switch	——	Light on	Light off	Continue	Always
6220	CAN'T EXE CHANGE *4	Cause of switch	——	Light on	Light on	Continue	Always
6221							
6222							
9000	F**** *6	Program error location	Annunciator number	On USER LED On	Off	Continue	When instruction is executed
9010	<CHK> ERR ***-*** *7	Program error location	Failure No.	On USER LED On	Off	Continue	When instruction is executed

*2 The CPU operation status when an error occurs can be set at the parameters. (LED display will change accordingly.)
*3 Can only be detected in a redundant system. Can be detected either in the control system or the standby system.
*4 Can only be detected in the control system of a redundant system.
*5 Can only be detected in the standby system of a redundant system.
*6 **** indicates detected annunciator number.
*7 *** indicates detected contact and coil number.

Error Code (SD0)	Error Contents and Cause	Corrective Action
4600	The SFC program includes data that cannot be dealt with.	Read common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4601	Exceeds device range that can be designated by the SFC program.	
4602	The start instruction in the block controls of the SFC program is preceeded by the end instruction.	
4610	Active step information at resumptive start of SFC program is not correct.	The program is automatically subjected to an initial start.
4611	Keyswitch was reset during RUN when resumptive start was designated for SFC program.	
4620	Startup was executed at a block in the SFC program that was already started up.	Read common error information at a peripheral device, check error step corresponding to its numerical value (program error location), and correct the problem.
4621	Startup was attempted at a block that does not exist in the SFC program.	
4630	Startup was executed at a block in the SFC program that was already started up.	
4631	Startup was attempted at a block that does not exist in the SFC program.	
4632	There were too many simultaneous active steps in blocks that can be designated by the SFC program.	
4633	There were too many simultaneous active steps in all blocks that can be designated.	
5000	Program scan time for initial execution type program goes over the initial execution WDT time set in the parameter PC RAS settings.	Read the error individual information at a peripheral device, check the numerical value (time) there, and shorten scan time if necessary.
5001	Program scan time goes over the WDT value set in the parameter PC RAS settings.	
5010	The low-speed execution type program run time that is set in the parameter PC RAS setting goes over the margin time of constant scan.	Review and change the constant scan time and low-speed execution type program run time set for parameters to secure sufficient length of margin time of constant scan.
5011	Low-speed scan type program scan time goes over the low-speed execution WDT set in the parameter PC RAS settings.	Read the error individual information at a peripheral device, check the numerical value (time) there, and shorten scan time if necessary.
6000	The control system and standby system programs and parameters in the redundant system are not the same.	Match up the control system and standby system programs and parameters.
6010	The operation status of the control system and standby system in the redundant system are not the same.	Match up the control system and standby system operating statuses.
6100	CPU module tracking memory error was detected during initial	Because this is a CPU module hardware error, explain the symptoms to the nearest service center, dealer, or branch office and consult with them. Replace the CPU module in the order of redundant system CPU → control system CPU.
6101	The CPU module detected an error during the handshake for tracking.	Check the condition of the other system.
6200	The standby system in a redundant system switched to the control system.	Check the control system condition.
6210	The control system in a redundant system switched to the standby system.	Check the control system condition.
6220	Standby system in a redundant system could not switch from the control system to the standby system because of an error status, etc.	Check the standby system condition.
6221	Switch could not be conducted because of a bus switching module error.	Because this is a bus switching module hardware error, explain the symptoms to the nearest service center, dealer, or branch office and consult with them.
6222	Switching could not be conducted because a remote I/O network redundant master station was installed in the standby station during initial.	Check the remote I/O network setting.
9000	Annunciator F went ON	Read the error individual information at a peripheral device, and check the program corresponding to the numerical value (annunciator number).
9010	Error detected by the CHK instruction.	Read the error individual information at a peripheral device, and check the program corresponding to the numerical value (error number) there.

9.3 Resetting an error

A QnACPU allows error resetting operation only for errors which permit the CPU to run continuously.

The error resetting procedure is indicated below.

1) Eliminate the cause of the error.
2) Store the error code to be reset to special register SD500.
3) Turn ON special relay SM50.
4) The error is reset.

When the CPU restarts after resetting the error, special relays, special registers, LEDs, and LED display unit related to the error are restored to the state before the occurrence of the error.
After the resetting of an error, if the same error occurs again, it is registered to the trouble history again.

If more than one annunciator is detected, error resetting operation resets only the F number that was detected first.

POINT

Lower 2 digits of the error code stored to SD50 are disregarded for error resetting operation.

(Example)

After the occurrence of errors of error code numbers 2100 and 2111, if error code 2100 is stored to SD50 and the error is reset, the error or error code 2111 is also reset.

APPENDICES

APPENDIX 1 OPERATION PROCESSING TIME

1.1 Definition

(1) At the QnACPU, operation processing time is the total of the following:
- Total of each instruction processing time
- END processing time
- I/O refresh time

(2) Instruction processing time
This is the total of processing time of each instruction shown in Appendix 1.2

(3) END processing time
The END processing time is the total of the following:
- END instruction time indicated in Appendix 1.2
- MELSECNET related refresh time
- Processing time of communications with peripheral devices.
- Communication time with such as a serial communication module

(4) I/O refresh time
I/O refresh time = (No. of input points/16) * N1 + (No. of output points/16) * N2

	N1 (μs)	N2 (μs)
Q2ASCPU(S1) Q2ACPU	5.2	5.0
Q3ACPU	4.8	4.65
Q2ASHCPU(S1) Q4ACPU Q4ARCPU	4.34	4.26

1.2 Operation Processing Time

The processing times for the individual instructions are shown in the table on the following pages.
Operation processing times can vary substantially depending on the nature of the sources and destinations of the instructions, and the values contained in the following tables should therefore be taken as a set of general guidelines to processing times rather than as being strictly accurate.

(1) Sequence instructions

Instruction	Conditions (Device)	Processing Time (μs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
LD LDI AND ANI OR ORI		0.20	0.15	0.075		
LDP LDF ANDP ANDF ORP ORF		6.6	5.0	2.5		
ANB ORB MPS MRD MPP		0.20	0.15	0.075		
INV	When not executed	2.4	1.8	0.9		
	When executed					
MEP MEF	When not executed	2.0	1.5	0.75		
	When executed					
EGP EGF	When not changed	0.6	0.3	0.15		
	When changed					

Instruction	Conditions (Device)				Processing Time (µs)				
					Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
OUT					0.40	0.30	0.15		
					0.40	0.30	0.15		
	F	When OFF			7.0	5.3	2.7		
		When ON	When displayed		167	126	63		
			Display completed		166	125	62		
	T	When not executed			1.6	1.2	0.6		
		When executed	After time up		1.6	1.2	0.6		
			When added	K	1.6	1.2	0.6		
				D	1.6	1.2	0.6		
	C	When not executed			1.6	1.2	0.6		
		When executed	After time up		1.6	1.2	0.6		
			When added	K	1.6	1.2	0.6		
				D	1.6	1.2	0.6		
OUTH	T	When not executed			1.6	1.2	0.6		
		When executed	After time up		1.6	1.2	0.6		
			When added	K	1.6	1.2	0.6		
				D	1.6	1.2	0.6		
SET	When not executed				0.40	0.30	0.15		
	When executed		When not changed (ON ON)		0.40	0.30	0.15		
			When changed (OFF ON)		0.40	0.30	0.15		
	F	When not executed			1.2	0.90	0.45		
		When executed	When displayed		277	208	104		
			Display completed		1.2	0.90	0.45		
RST	When not executed				0.40	0.30	0.15		
	When executed		When not changed (OFF OFF)		0.40	0.30	0.15		
			When changed (ON OFF)		0.40	0.30	0.15		
	SM	When not executed			0.40	0.30	0.15		
		When executed			0.40	0.30	0.15		
	F	When not executed			1.2	0.90	0.45		
		When executed	When displayed		148	112	56		
			Display completed		1.2	0.90	0.45		
	T, C	When not executed			1.4	1.1	0.6		
		When executed							
	D	When not executed			0.60	0.45	0.23		
		When executed							
	Z	When not executed			1.2	0.90	0.45		
		When executed			10.8	8.1	4.1		
	R	When not executed			1.0	0.75	0.38		
		When executed							
PLS PLF					2.6	2.0	0.98		
FF	Y	When not executed			1.2	0.90	0.45		
		When executed							
DELTA DELTAP	DYO	When not executed			1.2	0.90	0.45		
		When executed			17.8	13.4	6.7		

Instruction	Conditions (Device)		Processing Time (μs)				
			Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
SFT SFTP	When not executed		1.2	0.90	0.45		
	When executed		4.2	3.2	1.6		
MC			0.60	0.45	0.23		
MCR			0.20	0.15	0.075		
FEND END	Performs error check		1643	1236	618		
	No error check performed · Battery check · Blown fuse check · I/O module verification		1106	832	416		
NOP			0.2	0.15	0.075		
NOPLF PAGE			0.2	0.15	0.075		
LD=	When continuity established		3.8	2.9	1.5		
	When no continuity		3.6	2.7	1.4		
AND=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	2.8	2.1	1.1		
		When no continuity	3.2	2.4	1.2		
OR=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.8	2.9	1.5		
		When no continuity	2.8	2.1	1.1		
LD<>	When continuity established		4.4	3.3	1.7		
	When no continuity		3.6	2.7	1.4		
AND<>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	2.8	2.1	1.1		
		When no continuity	3.2	2.4	1.2		
OR<>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.8	2.9	1.5		
		When no continuity	2.8	2.1	1.1		
LD>	When continuity established		4.4	3.3	1.7		
	When no continuity		3.6	2.7	1.4		
AND>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	2.8	2.1	1.1		
		When no continuity	3.2	2.4	1.2		
OR>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.8	2.9	1.5		
		When no continuity	2.8	2.1	1.1		
LD<=	When continuity established		4.4	3.3	1.7		
	When no continuity		3.6	2.7	1.4		
AND<=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	2.8	2.1	1.1		
		When no continuity	3.2	2.4	1.2		
OR<=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.8	2.9	1.5		
		When no continuity	2.8	2.1	1.1		

Instruction	Conditions (Device)		Processing Time (μs)				
			Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
LD<	When continuity established		4.4	3.3	1.7		
	When no continuity		3.6	2.7	1.4		
AND<	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	2.8	2.1	1.1		
		When no continuity	3.2	2.4	1.2		
OR<	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.8	2.9	1.5		
		When no continuity	2.8	2.1	1.1		
LD<=	When continuity established		4.4	3.3	1.7		
	When no continuity		3.6	2.7	1.4		
AMD>=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	2.8	2.1	1.1		
		When no continuity	3.2	2.4	1.2		
OR>=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.8	2.9	1.5		
		When no continuity	2.8	2.1	1.1		
LDD=	When continuity established		5.0	3.8	1.9		
	When no continuity		4.2	3.2	1.6		
ANDD=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.4	2.6	1.3		
		When no continuity	3.8	2.9	1.5		
ORD=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	4.4	3.3	1.7		
		When no continuity	3.4	2.6	1.3		
LDD<>	When continuity established		5.0	3.8	1.9		
	When no continuity		4.2	3.2	1.6		
ANDD<>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.4	2.6	1.3		
		When no continuity	3.8	2.9	1.5		
ORD<>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	4.4	3.3	1.7		
		When no continuity	3.4	2.6	1.3		
LDD>	When continuity established		3.8	2.9	1.5		
	When no continuity		4.2	3.2	1.6		
ANDD>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	2.8	2.1	1.1		
		When no continuity	3.8	2.9	1.5		
ORD>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.8	2.9	1.5		
		When no continuity	3.4	2.6	1.3		
LDD>	When continuity established		4.4	3.3	1.7		
	When no continuity		3.6	2.7	1.4		
ANDD<=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.4	2.6	1.3		
		When no continuity	3.2	2.4	1.2		
ORD<=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	4.4	3.3	1.7		
		When no continuity	2.8	2.1	1.1		

Instruction	Conditions (Device)		Processing Time (μs)				
			Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
LDD<	When continuity established		3.8	2.9	1.5		
	When no continuity		4.2	3.2	1.6		
ANDD<	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	2.8	2.1	1.1		
		When no continuity	3.8	2.9	1.5		
ORD<	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.8	2.9	1.5		
		When no continuity	3.4	2.6	1.3		
LDD>=	When continuity established		4.4	3.3	1.7		
	When no continuity		3.6	2.7	1.4		
ANDD>=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	3.4	2.6	1.3		
		When no continuity	3.2	2.4	1.2		
ORD>=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	4.4	3.3	1.7		
		When no continuity	2.8	2.1	1.1		
LDE=	When continuity established		235	177	89		35
	When no continuity		231	174	87		87
ANDE=	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	234	176	88		34
		When no continuity	230	172	86		86
ORE=	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	234	176	88		35
		When no continuity	230	172	86		86
LDE<>	When continuity established		231	174	87		35
	When no continuity		234	176	88		88
ANDE<>	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	230	172	86		34
		When no continuity	234	176	88		88
ORE<>	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	231	174	87		35
		When no continuity	234	176	88		88
LDE>	When continuity established		231	174	87		35
	When no continuity		234	176	88		88
ANDE>	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	230	172	86		34
		When no continuity	234	176	88		88
ANDE>	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	231	174	87		34
		When no continuity	234	176	88		88
LDE<=	When continuity established		235	177	89		34
	When no continuity		231	174	87		87
ANDE<=	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	234	176	88		34
		When no continuity	230	172	86		86
ORE<=	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	234	176	88		34
		When no continuity	230	172	86		86

Instruction	Conditions (Device)		Processing Time (μs)				
			Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
LDE<	When continuity established		231	174	87		35
	When no continuity		234	176	88		88
AMDE<	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	230	172	86		34
		When no continuity	234	176	88		88
ORE<	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	231	174	87		34
		When no continuity	234	176	88		88
LDE>=	When continuity established		235	177	89		35
	When no continuity		231	174	87		87
ANDE>=	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	234	176	88		34
		When no continuity	231	174	87		87
ORE>=	When not executed		1.4	1.1	0.55		0.55
	When executed	When continuity established	234	176	88		34
		When no continuity	230	172	86		86
LD$=	When continuity established		97	73	37		
	When no continuity		81	61	31		
AND$=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	96	72	36		
		When no continuity	81	61	31		
OR$=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	97	73	37		
		When no continuity	80	60	30		
LD$<>	When continuity established		83	62	31		
	When no continuity		97	73	37		
AND$<>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	80	60	30		
		When no continuity	96	72	36		
OR$<>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	81	61	31		
		When no continuity	96	72	36		
LD$>	When continuity established		83	62	31		
	When no continuity		97	73	37		
AND$>	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	80	60	30		
		When no continuity	96	72	36		
OR$>	When continuity established		1.4	1.1	0.55		
	When no continuity	When continuity established	81	61	31		
		When no continuity	96	72	36		
LD$<=	When continuity established		97	73	37		
	When no continuity		81	61	31		
AND$<=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	96	72	36		
		When no continuity	81	61	31		
OR$<=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	97	73	37		
		When no continuity	80	60	30		

Instruction	Conditions (Device)		Processing Time (μs)				
			Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
LD$<	When continuity established		81	61	31		
	When no continuity		97	73	37		
AND$<	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	80	60	30		
		When no continuity	96	72	36		
OR$<	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	81	61	31		
		When no continuity	96	72	36		
LD$>=	When continuity established		97	73	37		
	When no continuity		81	61	31		
AND$>=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	96	72	36		
		When no continuity	81	61	31		
OR$>=	When not executed		1.4	1.1	0.55		
	When executed	When continuity established	97	73	37		
		When no continuity	80	60	30		
BKCMP= (S1) (S2) (D) n	n = 1		120	90	45		
BKCMP=P (S1) (S2) (D) n	n = 96		367	276	138		
BKCMP<> (S1) (S2) (D) n	n = 1		123	92	46		
BKCMP<>P (S1) (S2) (D) n	n = 96		346	260	130		
BKCMP> (S1) (S2) (D) n	n = 1		123	92	46		
BKCMP>P (S1) (S2) (D) n	n = 96		366	275	138		
BKCMP>= (S1) (S2) (D) n	n = 1		121	91	46		
BKCMP>=P (S1) (S2) (D) n	n = 96		386	290	145		
BKCMP< (S1) (S2) (D) n	n = 1		121	91	96		
BKCMP<P (S1) (S2) (D) n	n = 96		366	275	138		
BKCMP<= (S1) (S2) (D) n	n = 1		121	91	46		
BKCMP<=P (S1) (S2) (D) n	n = 96		348	261	131		

Instruction	Conditions (Device)	Processing Time (µs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
+ (S) (D) +P (S) (D)		2.4	1.8	0.9		
+ (S1) (S2) (D) +P (S1) (S2) (D)		2.7	2.0	1.0		
− (S) (D) −P (S) (D)		2.4	1.8	0.9		
− (S1) (S2) (D) −P (S1) (S2) (D)		2.6	2.0	1.0		
D+ (S) (D) D+P (S) (D)		2.8	2.1	1.1		
D+ (S1) (S2) (D) D+P (S1) (S2) (D)		3.2	2.4	1.2		
D− (S) (D) D−P (S) (D)		2.8	2.1	1.1		
D− (S1) (S2) (D) D−P (S1) (S2) (D)		3.2	2.4	1.2		
* (S1) (S2) (D) *P (S1) (S2) (D)		2.8	2.1	1.1		
/ (S1) (S2) (D) /P (S1) (S2) (D)		6.8	5.1	2.6		
D* (S1) (S2) (D) D*P (S1) (S2) (D)		20	15	7.5		
D/ (S1) (S2) (D) D/P (S1) (S2) (D)		36	27	13.5		
B+ (S) (D) B+P (S) (D)		5.5	4.1	2.1		

Instruction	Conditions (Device)	Processing Time (μs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
B+ (S1) (S2) (D) B+P (S1) (S2) (D)		13	9.6	4.8		
B- (S) (D) B-P (S) (D)		5.2	3.9	2.0		
B- (S1) (S2) (D) B-P (S1) (S2) (D)		13	9.4	4.7		
DB+ (S) (D) DB+P (S) (D)		29	22	11		
DB+ (S1) (S2) (D) DB+P (S1) (S2) (D)		32	24	12		
DB- (S) (D) DB-P (S) (D)		29	22	11		
DB- (S1) (S2) (D) DB-P (S1) (S2) (D)		32	24	12		
B* (S1) (S2) (D) B*P (S1) (S2) (D)		9.4	7.1	3.6		
B/ (S1) (S2) (D) B/P (S1) (S2) (D)		9.4	7.1	3.6		
DB* (S1) (S2) (D) DB*P (S1) (S2) (D)		62	46	23		
BD/ (S1) (S2) (D) DB/P (S1) (S2) (D)		69	52	26		
E+ (S) (D)	(S)=0, (D)=0	54	40	20		35
E+P (S) (D)	(S)=2^{127}, (D)=2^{127}	524	394	197		35
E+ (S1) (S2) (D)	(S1)=0, (S2)=0	54	40	20		35
E+P (S1) (S2) (D)	(S1)=2^{127}, (S2)=2^{127}	524	394	197		35

Instruction	Conditions (Device)	Processing Time (μs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
E- (S) (D)	(S)=0, (D)=0	54	40	20		35
E-P (S) (D)	(S)=2^{127}, (D)=2^{127}	515	387	194		35
E- (S1) (S2) (D)	(S1)=0, (S2)=0	55	41	21		35
E-P (S1) (S2) (D)	(S1)=2^{127}, (S2)=2^{127}	520	391	146		36
E* (S1) (S2) (D)	(S1)=0, (S2)=0	55	41	21		35
E*P (S1) (S2) (D)	(S1)=2^{126}, (S2)=2^{127}	567	426	218		36
E/ (S1) (S2) (D)	(S1)=0, (S2)=1	149	112	56		37
E/P (S1) (S2) (D)	(S1)=2^{127}, (S2)=-2^{126}	1109	834	417		38
$+ (S) (D) $+P (S) (D)		179	134	67		
$+ (S1) (S2) (D) $+P (S1) (S2) (D)		206	155	78		
INC INCP		1.9	1.4	0.7		
DINC DINCP		2.3	1.7	0.9		
DEC DECP		1.9	1.4	0.7		
DDEC DDECP		2.3	1.7	0.9		
BCD BCDP		2.7	2.0	1.0		
DBCD DBCDP		7.9	5.9	3.0		
BIN BINP		2.7	2.0	1.0		

Instruction	Conditions (Device)	Processing Time (µs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
DBIN DBINP		4.8	3.6	1.8		
INT	(S)=0	20	15	7.5		
INTP	(S)=32766.5	54	40	20		
DINT	(S)=0	20	15	7.5		
DINTP	(S)=1234567890.3	59	44	22		
FLT	(S)=0	27	20	10		
FLTP	(S)=7FFFH	55	41	21		
DFLT	(S)=0	28	21	11		
DFLTP	(S)=7FFFFFFFH	56	42	21		
DBL DBLP		12	8.6	4.3		
WORD WORDP		12	9.0	4.5		
GRY GRYP		12	9.0	4.5		
DGRY DGRYP		14	10	5.0		
GBIN GBINP		46	34	17		
DGBIN DGBINP		83	62	31		
NEG NEGP		9.3	7	3.5		
DNEG DNEGP		11	8.2	4.1		

Instruction	Conditions (Device)	Processing Time (µs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
ENEG ENEGP		9.8	7.4	3.7		
BKBCD (S) (D) n	n=1	102	76	38		
BKBCDP (S) (D) n	n=96	272	204	102		
BKBIN (S) (D) n	n=1	102	76	38		
BKBINP (S) (D) n	n=96	272	204	102		
MOV	(S)=K4X0, (D)=D1	0.7	0.5	0.3		
MOVP	(S)=K4X0, (D)=J1\W1	392 391	305 299	176 165		*
DMOV	(S)=K8X0, (D)=D1	2.4	1.8	0.9		
DMOVP	(S)=K8X0, (D)=J1\W1	406 395	313 301	183 167		
EMOV EMOVP		12	8.6	4.3		
$MOV $MOVP		100	75	38		
CML CMLP		2.0	1.5	0.8		
DCML DCMLP		2.4	1.8	0.9		
BMOV (S) (D) n	n=1	43	32	16		
BMOVP (S) (D) n	n=96	81	61	31		
FMOV (S) (D) n	n=1	18	13	6.5		
FMOVP (S) (D) n	n=96	36	27	14		
XCH XCHP DXCH DXCHP		3.1	2.3	1.2		

* The upper row indicates the processing time when A38B/A1S38B and the extension base are used. The lower row indicates the processing time when A38HB/A1S38HB is used.

Instruction	Conditions (Device)	Processing Time (µs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
BXCH (D1) (D2) n	n=1	77	58	29		
BXCHP (D1) (D2) n	n=96	213	160	80		
SWAP SWAPP		9.2	6.9	3.5		
CJ		7.8	5.8	2.9		
SCJ		7.8	5.8	2.9		
JMP		8.0	6.0	3.0		
GOEND		2.0	1.5	0.75		
DI		2.3	1.7	0.9		
EI		3.1	2.3	1.2		
IMASK		8.1	6.5	3.3		
IRET		4.0	3.0	1.5		
RFS	n=1	31.3	23.4	11.7		
RFSP	n=96	97.6	72.8	36.4		
UDCNT1		42.6	31.8	15.9		
UDCNT2		44.6	33.3	16.7		
TTMR		25.9	19.3	9.7		
STMR		41.7	31.1	15.6		
ROTC		66.1	49.3	24.7		
RAMP		45.4	33.9	17.0		
SPD		48.9	36.5	18.3		
PLSY		26.9	20.1	10.1		
PWM		32.8	24.5	12.3		
MTR		29.2	21.8	10.9		

Instruction	Conditions (Device)	Processing Time (µs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
WAND (S) (D) WANDP (S) (D)		2.4	1.8	0.9		
WAND (S1) (S2) (D) WANDP (S1) (S2) (D)		9.5	7.1	3.6		
DAND (S) (D) DANDP (S) (D)		3.0	2.3	1.2		
DAND (S1) (S2) (D) DAND (S1) (S2) (D)		19	14	7.0		
BKAND (S1) (S2) (D) n BKANDP (S1) (S2) (D) n	n=1	89	67	34		
	n=96	184	138	69		
WOR (S) (D) WORP (S) (D)		2.4	1.8	0.9		
WOR (S1) (S2) (D) WORP (S1) (S2) (D)		9.5	7.1	3.6		
DOR (S) (D) DORP (S) (D)		3.0	2.3	1.2		
DOR (S1) (S2) (D) DORP (S1) (S2) (D)		19	14	7.0		
BKOR (S1) (S2) (D) n BKORP (S1) (S2) (D) n	n=1	89	67	34		
	n=96	184	138	69		
WXOR (S) (D) WXORP (S) (D)		2.4	1.8	0.9		
WXOR (S1) (S2) (D) WXORP (S1) (S2) (D)		17.2	7.1	3.6		
DXOR (S) (D) DXOR (S) (D)		3.0	2.3	1.2		

Instruction	Conditions (Device)	Processing Time (μs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
DXOR (S1) (S2) (D) DXORP (S1) (S2) (D)		19	14	7.0		
BKXOR (S1) (S2) (D) n BKXORP (S1) (S2) (D) n	n=1	89	67	34		
	n=96	184	138	69		
WXNR (S) (D) WXNRP (S) (D)		2.4	1.8	0.9		
WXNR (S1) (S2) (D) WXNRP (S1) (S2) (D)		9.5	7.1	3.6		
DNXR (S) (D) DNXRP (S) (D)		3.0	2.3	1.2		
DNXR (S1) (S2) (D) DNXRP (S1) (S2) (D)		24	18	9		
BKNXOR (S1) (S2) (D) n BKNXORP (S1) (S2) (D) n	n=1	89	67	34		
	n=96	184	138	69		
ROR (D) n RORP (D) n	n=1	5.0	3.8	1.9		
	n=15	5.0	3.8	1.9		
RCR (D) n RCRP (D) n	n=1	4.0	3.0	1.5		
	n=15	4.0	3.0	1.5		
ROL (D) n ROLP (D) n	n=1	5.0	3.8	1.9		
	n=15	5.0	3.8	1.9		
RCL (D) n RCLP (D) n	n=1	4.0	3.0	1.5		
	n=15	4.0	3.0	1.5		
DROR (D) n DRORP (D) n	n=1	9.8	7.4	3.7		
	n=31	10	7.8	3.9		
DRCR (D) n DRCRP (D) n	n=1	11	8.1	4.1		
	n=31	11	8.3	4.2		

Instruction	Conditions (Device)	Processing Time (μs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
DROL (D) n	n=1	9.8	7.4	3.7		
DROLP (D) n	n=31	10	7.8	3.9		
DRCL (D) n	n=1	11	8.1	4.1		
DRCLP (D) n	n=31	11	8.3	4.2		
SFR (D) n	n=1	4.4	3.3	1.7		
SFRP (D) n	n=15	5.0	3.8	1.9		
SFL (D) n	n=1	4.4	3.3	1.7		
SFLP (D) n	n=15	5.0	3.8	1.9		
BSFLR (D) n	n=1	51	38	19		
BSFLRP (D) n	n=96	60	45	23		
BSFL (D) n	n=1	49	37	19		
BSFLP (D) n	n=96	58	44	22		
DSFR (D) n	n=1	3.6	2.6	1.3		
DSFRP (D) n	n=96	63	47	24		
DSFL (D) n	n=1	3.6	2.6	1.3		
DSFLP (D) n	n=96	65	49	25		
BSET (D) n	n=1	20	15	7.5		
BSETP (D) n	n=15	20	15	7.5		
BRST (D) n	n=1	20	15	7.5		
BRSTP (D) n	n=15	20	15	7.5		
TEST (S1) (S2) (D) TESTP (S1) (S2) (D)		21	16	8.0		
DTEST (S1) (S2) (D) DTESTP (S1) (S2) (D)		24	18	9.0		
BKRST (S) n	n=1	45	34	17		
BKRSTP (S) n	n=96	49	37	19		

Instruction	Conditions (Device)		Processing Time (μs)				
			Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
SER (S1) (S2) (D) n	n=1	All match	58	44	22		
		None match	57	43	21		
SERP (S1) (S2) (D) n	n=96	All match	293	220	110		
		None match	340	256	128		
DSER (S1) (S2) (D) n	n=1	All match	61	46	23		
		None match	58	44	22		
DSERP (S1) (S2) (D) n	n=96	All match	354	266	133		
		None match	354	266	133		
SUM SUMP	(S)=0 (S)=FFFF		9.8	7.4	3.7		
DSUM	(S)=0		12	9.0	4.5		
DSUMP	(S)=FFFFFFFFH		31	23	12		
DECO (S) (D) n	n=2		48	36	18		
DECOP (S) (D) n	n=8		62	47	24		
ENCO (S) (D) n	n=2	M1=ON	52	39	20		
		M4=ON	52	39	20		
ENCOP (S) (D) n	n=8	M1=ON	65	49	25		
		M256=ON	65	49	25		
SEG SEGP			3.2	2.4	1.2		
DIS (S) (D) n	n=1		46	34	17		
DISP (S) (D) n	n=4		51	38	19		
UNI (S) (D) n	n=1		53	40	20		
UNIP (S) (D) n	n=4		57	43	22		
NDIS (S1) (D) (S2) NDISP (S1) (D) (S2)			104	78	39		
NUNI (S1) (D) (S2) NUNIP (S1) (D) (S2)			105	79	40		
WTOB (S) (D) n	n=1		125	94	47		
WTOBP (S) (D) n	n=96		257	193	97		

Instruction	Conditions (Device)	Processing Time (μs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
BTOW (S) (D) n BTOWP (S) (D) n	n=1	121	91	46		
	n=96	233	175	88		
MAX (S) (D) n MAXP (S) (D) n	n=1	43	32	16		
	n=96	318	239	120		
MIN (S) (D) n MINP (S) (D) n	n=1	43	32	16		
	n=96	436	326	163		
DMAX (S) (D) n DMAXP (S) (D) n	n=1	71	53	27		
	n=96	427	321	161		
DMIN (S) (D) n DMINP (S) (D) n	n=1	71	53	27		
	n=96	268	201	101		
SORT (S1) n (S2) (D1) (D2)	n=1	43	32	16		
	n=96	40*	30*	15*		
DSORT (S1) n (S2) (D1) (D2)	n=1	44	33	17		
	n=96	43*	32*	16*		
FOR n	n=0	5.2	3.9	2.0		
NEXT		8.0	6.0	3.0		
BREAK BREAKP		26	19	9.5		
CALL Pn CALLP Pn	Internal file pointer	5.1	3.8	1.9		
	Common pointer	85	64	32		
CALL Pn (S1) to (S5) CALLP Pn (S1) to (S5)		348	261	131		
RET	Return to original program	7.5	5.6	2.8		
	Return to other program	51	38	19		
FCALL Pn FCALLP Pn	Internal file pointer	8.8	6.6	3.3		
	Common pointer	48	36	18		

* : Indicates extension of scan time to completion of instruction

Instruction	Conditions (Device)	Processing Time (µs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
FCALL Pn (S1) to (S5) FCALLP Pn (S1) to (S5)		338	254	127		
ECALL * Pn ECALLP * Pn * : Program Name		187	140	70		
ECALL * Pn (S1) to (S5) ECALLP * Pn (S1) to (S5) * : Program Name		515	387	144		
EFCALL * Pn EFCALLP * Pn * : Program Name		188	141	71		
EFCALL * Pn (S1) to (S5) EFCALLP * Pn (S1) to (S5) * : Program Name		516	388	194		
COM		137	103	52		
IX		31	23	12		
IXEND		12	8.9	4.5		
IXDEV + IXSET	Number of contacts 1	127	95	46		
	Number of contacts 14	238	179	85		
FIFW	Number of data points 0	27	20	10		
FIFWP	Number of data points 96	27	20	10		
FIFR	Number of data points 1	34	25	13		
FIFRP	Number of data points 96	79	59	30		
FPOP	Number of data points 1	46	34	17		
FPOPP	Number of data points 96	46	34	17		
FINS	Number of data points 0	48	36	18		
FINSP	Number of data points 96	96	72	36		

Instruction	Conditions (Device)		Processing Time (µs)				
			Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
FDEL	Number of data points 1		47	35	18		
FDELP	Number of data points 96		97	73	37		
FROM n1 n2 (D) n3	n3=1		253	217	160 *		
			252	210	154		
FROMP n1 n2 (D) n3	n3=1000		4514	4286	4150		
			2855	2127	2038		
DFRO n1 n2 (D) n3	n3=1		260	221	165		
			257	214	156		
DFROP n1 n2 (D) n3	n3=500		4543	4271	4082		
			2883	2129	2064		
TO n1 n2 (D) n3	n3=1		254	217	162		
			276	211	154		
TOP n1 n2 (D) n3	n3=1000		4500	4319	4188		
			2878	2155	2043		
DTO n1 n2 (D) n3	n3=1		260	221	165		
			257	216	157		
DTOP n1 n2 (D) n3	n3=500		4471	4315	4198		
			2819	2172	2062		
PR	SM7010N	Variable 1 character	83	62	31		
		Variable 32 characters	123	92	46		
	SM7010FF		54	40	20		
PRC			400	301	151		
LED	When displayed		223	167	84		
	Display completed		79	59	30		
LEDC	When displayed		559	420	210		
	Display completed		413	310	155		
LEDR	No display → no display		18	13	6.5		
	LED instruction execution → no display		205	154	77		
CHKST			15	11	5.5		
CHK	Contact 1 no error		61	46	23		
	Contact 150 no error		4232	3182	1591		
	Contact 1 error		224	168	84		
CHKCIR	Step 10		15	11	5.5		
SLT	All internal devices		2399	1804	902		
	File register 8K points		7254	5454	2727		
	SLT execution completion		15	11	5.5		
SLTR			1.1	0.8	0.4		
STRA	Start		47	35	18		
	STRA execution completion		15	11	5.5		
STRAR			1.1	0.8	0.4		
PTRA			15	11	5.5		
PTRAR			15	11	5.5		

* The upper row indicates the processing time when A38B/A1S38B and the extension base are used. The lower row indicates the processing time when A38HB/A1S38HB is used.

Instruction	Conditions (Device)	Processing Time (μs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
PTRAEXE	When operating	1.6	1.2	0.6		
PTRAEXEP	Trace in progress	169	127	64		
BINDA	(S)=1	40	30	15		
BINDAP	(S)=-32768	60	45	23		
DBINDA	(S)=1	63	47	24		
DBINDAP	(S)=-2147483648	217	163	82		
BINHA	(S)=1	46	34	17		
BINHAP	(S)=FFFFH	48	36	18		
DBINHA	(S)=1	59	44	22		
DBINHAP	(S)=FFFFFFFFH	62	46	23		
BCDDA	(S)=1	58	43	22		
BCDDAP	(S)=9999	54	40	20		
DBCDDA	(S)=1	61	46	23		
DBCDDAP	(S)=99999999	75	56	28		
DABIN	(S)=1	133	100	50		
DABINP	(S)=-32768	145	109	55		
DDABIN	(S)=1	241	181	91		
DDABINP	(S)=-2147483648	268	201	101		
HABIN	(S)=1	32	24	12		
HABINP	(S)=FFFFH	38	28	14		
DHABIN	(S)=1	54	40	20		
DHABINP	(S)=FFFFFFFFH	63	47	24		
DABCD	(S)=1	36	27	14		
DABCDP	(S)=9999	42	31	16		
DDABCD	(S)=1	63	47	24		
DDABCDP	(S)=99999999	75	56	28		

Instruction	Conditions (Device)	Processing Time (μs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
COMRD COMRDP		36	27	14		
LEN	1 character	48	36	18		
LENP	96 characters	229	172	86		
STR STRP		132	99	50		
DSTR DSTRP		285	214	107		
VAL VALP		258	194	97		
DVAL DVALP		402	302	151		
ESTR ESTRP		1337	1005	503		
EVAL	Decimal point format all 2-digit exponent	242	182	91		
EVALP	Exponent format all 6-digit exponent	306	230	115		
ASC (S) (D) n	n=1	164	123	62		
ASCP (S) (D) n	n=96	780	586	293		
HEX (S) (D) n	n=1	161	121	61		
HEXP (S) (D) n	n=96	826	621	311		
RIGHT (S) (D) n	n=1	131	98	49		
RIGHTP (S) (D) n	n=96	354	266	133		
LEFT (S) (D) n	n=1	129	97	49		
LEFTP (S) (D) n	n=96	354	266	133		
MIDR MIDRP		141	106	53		

Instruction	Conditions (Device)		Processing Time (μs)				
			Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
MIDW MIDWP			341	256	128		
INSTR INSTRP	No match		156	117	59		
	Match	First	141	106	53		
		Final	155	116	58		
EMOD EMODP			1313	987	494		
EREXP EREXPP			4423	3325	1663		
SIN SINP			4921	3700	1850		35
COS COSP			6462	4858	2429		35
TAN TANP			6515	4898	2449		37
ASIN ASINP			890	669	335		44
ACOS ACOSP			801	602	301		44
ATAN ATANP			7818	5878	2939		39
RAD RADP			465	349	175		31
DEG DEGP			492	369	185		31
SQR SQRP			4520	3398	1699		31

Instruction	Conditions (Device)	Processing Time (μs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
EXP	(S)=-10	5871	4414	2207		37
EXPP	(S)=1	5950	4474	2237		37
LOG	(S)=1	1191	896	448		37
LOGP	(S)=10	6839	5142	2571		37
RND RNDP		10	7.5	3.8		
SRND SRNDP		8.8	6.6	3.3		
BSQR	(S)=0	16	12	6.0		
BSQRP	(S)=9999	97	73	37		
BDSQR	(S)=0	17	13	6.5		
BDSQRP	(S)=99999999	88	66	33		
BSIN BSINP		30	22	11		
BCOS BCOSP		32	24	12		
BTAN BTANP		30	22	11		
BASIN BASINP		52	39	20		
BACOS BACOSP		53	40	20		
BATAN BATANP		56	42	21		
LIMIT LIMITP		24	18	9.0		

Instruction	Conditions (Device)	Processing Time (µs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
DLIMIT DLIMITP		28	21	11		
BAND BANDP		24	18	9.0		
DBAND DBANDP		28	21	11		
ZONE ZONEP		24	18	9.0		
DZONE DZONEP		28	21	11		
RSET RSETP		19	14	7.0		
QDRSET QDRSETP		322	242	121		
QCDSET QCDSETP		218	164	82		
DATERD DATERDP		36	27	14		
DATEWR DATEWRP		42	31	16		
DATE+ DATE+P	No digit increase	60	45	23		
	Digit increase	60	45	23		
DATE- DATE-P	No digit increase	59	44	22		
	Digit increase	60	45	23		
SECOND SECONDP		27	20	10		

Instruction	Conditions (Device)	Processing Time (µs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
HOUR HOURP		31	23	12		
MSG	1 character	7.2	5.4	2.7		
	32 characters	7.4	5.6	2.8		
PKEY	Initial time	51	38	19		
	No acceptance	48	36	18		
PSTOP PSTOPP		122	92	46		
POFF POFFP		120	90	45		
PSCAN PSCANP		122	92	46		
PLOW PLOWP		124	93	47		
WDT WDTP		12	8.7	4.4		
DUTY		1.6	1.2	0.6		
ZRRDB ZRRDBP		19	14	6.9		
ZRWRB ZRWRBP		21	16	7.8		
ADRSET ADRSETP		13	9.3	4.7		
KEY		43.4	32.4	16.2		
ZPUSH ZPUSHP		27.6	20.6	10.3		
ZPOP ZPOPP		12.7	9.5	4.8		
EROMWR EROMWRP		62.6	46.7	23.4		
ZCOM		4296.6	3206.4	1603.2		
READ		770.6	575.1	287.6		
SREAD		858.9	641.0	320.5		
WRITE		791.9	591.0	295.5		

Instruction	Conditions (Device)	Processing Time (μs)				
		Q2A, Q2AS	Q3A	Q2ASH	Q4A	Q4AR
SWRITE		848.6	633.3	316.6		
SEND		575.7	429.6	214.8		
RECV		375.9	280.5	140.3		
REQ		527.4	393.6	196.8		
ZNFR		982.1	732.9	366.5		
ZNTO		989.3	738.3	369.2		
ZNRD	MELSECNET/10	598.6	446.7	223.4		
	MELSECNET(II)	649.2	484.5	242.3		
ZNWR	MELSECNET/10	614.3	458.4	229.2		
	MELSECNET(II)	665.6	496.7	248.4		
RFRP		590.9	441.0	220.5		
RTOP		588.8	439.4	219.7		

APPENDIX 2 COMPARISON OF QnACPU WITH AnNCPU, AnACPU, AND AnUCPU

2.1 Usable Devices

Table 2.1 Device Comparison

Device Name		QnACPU	AnUCPU	AnACPU	AnNCPU
No. of inputs/outputs		— Q2A : 512 points Q2A-S1 : 1024 points Q3A : 2048 points Q4A : 4096 points	— A2U : 512 points A2U-S1 : 1024 points A3U : 2048 points A4U : 4096 points	— A2A : 512 points A2A-S1 : 1024 points A3A : 2048 points —	A1N : 256 points A2N : 512 points A2N-S1 : 1024 points A3N : 2048 points —
Internal relays		8192 points*1	Total 8192 points	Total 8192 points	Total 2048 points
Latch relays		8192 points*1			
Step relays	Sequence program	—			
	SFC	8192 points	—	—	—
Annunciators		2048 points *1	2048 points	2048 points	256 points
Edge triggered relays		2048 points *1	—	—	—
Link relays		8192 points*1	8192 points	4096 points	1024 points
Special relays for link		2048 points	56 points	56 points	56 points
Timers		2048 points *1	Total 2048 points	Total 2048 points	Total 256 points
Retentive timers		0 points *1			
Counters		1024 points *1	1024 points	1024 points	256 points
Data registers		12288 points *1	8192 points	6144 points	1024 points
Link registers		8192 points*1	8192 points	4096 points	1024 points
Special registers for link		2048 points	56 points	56 points	56 points
Function inputs		5 points (FX0 to FX4)	—	—	—
Function outputs		5 points (FY0 to FY4)	—	—	—
Special Relays		2048 points	256 points	256 points	256 points
Function registers		5 points (FD0 to FD4)	—	—	—
Special registers		2048 points	256 points	256 points	256 points
Link direct devices		Designated by J□\□	—	—	—
Special direct devices		Designated by U□\G□	—	—	—
Index registers	Z	16 points (Z0 to Z15)	7 points (Z, Z1 to Z6)	7 points (Z, Z1 to Z6)	1 point (Z)
	V*2	—	7 points (V, V1 to V6)	7 points (V, V1 to V6)	1 point (V)
File registers		32767 points/blocks (R0 to R32767)	8192 points/blocks (R0 to R8191)	8192 points/blocks (R0 to R8191)	8192 points/blocks (R0 to R8191)
Accumulators*3		—	2 points	2 points	2 points
Nesting		15 points	8 points	8 points	8 points
Pointers		4096 points	256 points	256 points	256 points
Interrupt pointers		48 points	32 points	32 points	32 points
SFC blocks		320 points	—	—	—
SFC transition devices		512 points	—	—	—
Decimal constants		K-2147483648 to K2147483647	K-2147483648 to K2147483647	K-2147483648 to k2147483647	K-2147483648 to k2147483647
Hexadecimal constants		H0 to HFFFFFFFF	H0 to HFFFFFFFF	H0 to HFFFFFFFF	H0 to HFFFFFFFF
Real number constants		E±1.17549-38 to E±3.40282+38	—	—	—
Character strings		"QnACPU", "ABCD"	—	—	—

*1 :The number of device points can be changed at the parameters
*2 :QnACPU uses V as an edge relay
*3 :Instructions that used accumulators with the AnNCPU, AnACPU, and AnUCPU have different formats with the QnACPU.

2.2 I/O Control Mode

Table 2.2 I/O control Mode

I/O Control Mode			QnACPU	AnUCPU	AnACPU	AnNCPU
Refresh mode			o	o	o	o[*2]
	Direct input/output method	Partial refresh instructions	o	o	o	o
		Dedicated instruction[*1]	—	o	o	—
		Direct access input	o	—	—	—
		Direct access output	o	—	—	—
Direct mode			—	—	—	o[*2]

Symbol in table....o : Usable, — : Unusable

*1....................The DOUT, DSET, and SRST instructions are direct output dedicated instructions.
There are no dedicated instructions for direct input.

*2....................Switching between the refresh mode and direct mode is conducted with an AnNCPU DIP switch

2.3 Data That Can Be Used by Instructions

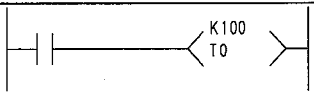
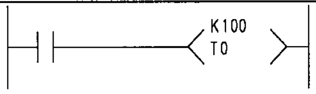
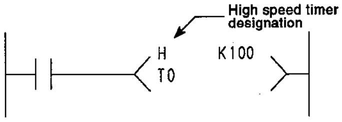
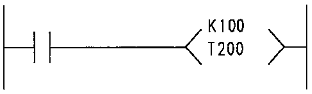
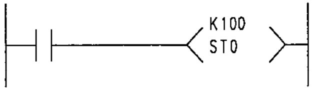
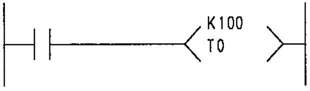
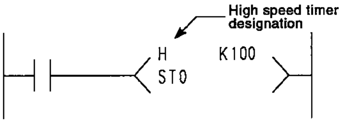
Table 2.3 Data That Can Be Used by Instructions

Set Data		QnACPU	AnUCPU	AnACPU	AnNCPU
Bit data	Bit device	o	o	o	o
	Word device	o (Bit designation required)	—	—	—
Word data	Bit device	o (Digit designation required)	o (Digit designation required)	o (Digit designation required)	o (Digit designation required)
	Word device	o	o	o	o
Double word data	Bit device	o (Digit designation required)	o (Digit designation required)	o (Digit designation required)	o (Digit designation required)
	Word device	o	o	o	o
Real number data		o	o	o	—
Character-string data		o	—	—	—

Symbols in table...o : Usable, — : Unusable

2.4 Timer Comparison

Table 2.4 Timer Comparison

Functions		QnACPU	AnUCPU	AnACPU	AnNCPU
Low speed timer	Measurement unit	• 100 ms (default value) Can be changed within the range of from 10 to 100 ms (parameter setting)	• Fixed at 100 ms		
	Designation method				
High speed timer	Measurement unit	• 10 ms (default value) Can be changed within the range of from 1 to 100 ms (parameter setting)	• Fixed at 10 ms		
	Designation method	 * High speed timer setting: Conducted by sequence program	 * High speed timer setting: Conducted at parameters		
Retentive timer	Measurement unit	• Same measurement unit as low speed timer	• Fixed at 10 ms		
	Designation method				
High speed retentive timer	Measurement unit	• Same measurement unit as high speed timer	• None		
	Designation method	 * High speed timer setting: Conducted by sequence program			
Setting range for set values		• 1 to 32767	• 1 to 32767		
Processing for set value 0		• Momentarily ON	• No maximum (does not time out)		
Index qualification	Contact	• Enabled (only Z0 and Z1 are usable)	• Capable		• Incapable
	Coil	• Enabled (only Z0 and Z1 are usable)	• Incapable		• Incapable
	Set value	• Incapable	• Incapable		• Incapable
	Present value	• Enabled (Z0 to Z15 are usable)	• Capable		• Capable
Update processing for present value		• When OUT Tn instruction is executed	• When END processing is done		
Contact ON/OFF processing					

(1) Cautions on using timers

A QnACPU updates the present value of timers and turns ON/OFF the contacts of them at the execution of OUT T□ instruction.

Therefore, if "Present value ≥ Set value" when the timer coil goes ON, the contact of that timer goes ON.

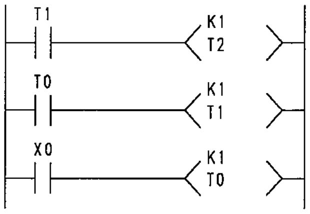
When creating a program in which the operation of the timer contact triggers the operation of other timer, create the program according to the operation order of the timers - create the program for the timer that operates later first.

In the following cases, all timers go ON at the same scan if the program is created in the order the timers operate,

- With high speed timers, if the set value is smaller than a scan time.
- With slow timers, if "1" is set.

Example

- For timers T0 to T2, the program is created in the order the timer operates later.

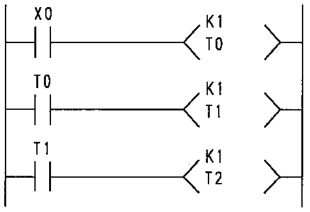


T2 timer starts counting from the next scan after the timer T1 contact is turned ON.

T1 timer starts counting from the next scan after the timer T0 contact is turned ON.

T0 timer starts counting if X0 is turned ON.

- For timers T0 to T2, the program is created in the order of timer operation.

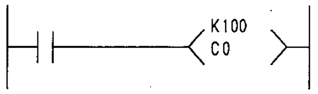
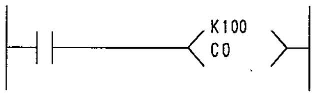


T0 timer starts counting if X0 is turned ON.

T1 and T2 timer contacts are turned ON if the contact of T0 is turned ON.

2.5 Comparison of Counters

Table 2.5 Comparison of Counters

Functions		QnACPU	AnUCPU	AnACPU	AnNCPU
Designation method		 K100 C0	 K100 C0		
Index qualification	Contact	• Enabled (only Z0 and Z1 are usable)	• Capable		• Incapable
	Coil	• Enabled (only Z0 and Z1 are usable)	• Capable		• Incapable
	Set value	• Incapable	• Incapable		• Incapable
	Present value	• Enabled (Z0 to Z15 are usable)	• Capable		• Capable
Update processing for present value		• When OUT Tn instruction is executed	• When END processing is done		
Contact ON/OFF processing					

2.6 Comparison of Display Instructions

Table 2.6 Comparison of Display Instructions

Instruction	QnACPU	AnUCPU	AnACPU	AnNCPU
PR	• When SM701 is OFF : Output continued until 00H encountered • When SM701 is ON : 16 characters output	• When M9049 is OFF : Output continued until 00H encountered • When M9049 is ON : 16 characters output		
PRC	• When SM701 is OFF : 32 character comment output • When SM701 is ON : Upper 16 characters output	16-character comment output		

2.7 Instructions Whose Designation Format Has Changed (Except Dedicated Instructions for AnACPU and AnUCPU)

Because the QnACPU does not have accumulators (A0, A1), the format of AnUCPU, AnACPU and AnNCPU instructions that used accumulators has been changed.

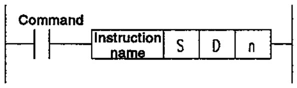
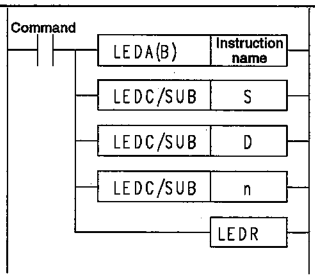
Table 2.7 Instructions Whose Expression Has Changed

Functions	QnACPU		AnUCPU/AnACPU/AnNCPU	
	Instruction Format	Remark	Instruction Format	Remark
16 bit rotation to right	ROR D n	• D : Rotation data	ROR n	• Rotation data is set at A0
	RCR D n	• D : Rotation data • The carry flag uses SM700	RCR n	• Rotation data is set at A0 • Carry flag uses M9012
16-bit rotation to left	ROL D n	• D : Rotation data	ROL n	• Rotation data is set at A0
	RCL D n	• D : Rotation data • The carry flag uses SM700	RCL n	• Rotation data is set at A0 • Carry flag uses M9012
32-bit rotation to right	DROR D n	• D : Rotation data	DROR n	• Rotation data set at A0 and A1
	DRCR D n	• D : Rotation data • The carry flag uses SM700	DRCR n	• Rotation data set at A0 and A1 • Carry flag uses M9012
32-bit rotation to left	DROL D n	• D : Rotation data	DROL n	• Rotation data set at A0 and A1
	DRCL D n	• D : Rotation data • The carry flag uses SM700	DRCL n	• Rotation data set at A0 and A1 • Carry flag uses M9012
16-bit data search	SER S1 S2 D n	• Search results are stored at the D and D+1 devices	SER S1 S2 n	• Search results stored at A0 and A1
32-bit data search	DSER S1 S2 D n	• Search results are stored at the D and D+1 devices	DSER S1 S2 n	• Search results stored at A0 and A1
16-bit data bit check	SUM S D	• Check results are stored at the D device	SUM S	• Check results stored at A0
16-bit data bit check	DSUM S D	• Check results are stored at the D device	DSUM S	• Check results stored at A0
Partial refresh	RFS D n	• Added dedicated instruction	SEG D n	• Only when M9052 is ON
8 character ASCII conversion	SMOV Character strings D		ASC Caracter strings D	
Carry flag set	SET SM700	• No dedicated instruction	STC	
Carry flag reset	RST SM700	• No dedicated instruction	CLC	
Jump to END instruction	GOEND	• Added dedicated instruction	CJ P255	• P255 : END instruction designation
CHK instruction	CHKST CHK	• Added CHKST instruction	CJ Pn P254 CHK	

2.8 AnACPU and AnUCPU Dedicated Instructions

(1) Method of expression of dedicated instructions
Dedicated instructions based on the LEDA, LEDB, LEDC, SUB and LEDR instructions with AnACPU or AnUCPU have for the QnACPU been changed to formats identical to the basic instructions and the application instructions.

Table 2.8 Method of Expression of Dedicated Instructions

QnACPU	AnUCPU/AnACPU
	 * : S, D, and n indicate data used by instructions.

(2) Dedicated instructions whose names have been changed
Dedicated instructions for the AnUCPU or AnACPU which have the same instruction name as is used for basic instructions and application instructions have undergone name changes in the QnACPU.

Table 2.9 Dedicated Instructions whose Names Have Been Changed

Functions	QnACPU	AnUCPU/AnACPU
Floating decimal point addition	E+	ADD
Floating decimal point subtraction	E-	SUB
Floating decimal point multiplication	E*	MUL
Floating decimal point division	E/	DIV
Data dissociation	NDIS	DIS
Data association	NUNI	UNI
Updating check patterns	CHKCIR, CHKEND	CHK, CHKEND

2.9 Instructions Which Can Be Programmed Only in the General Purpose Mode

The following instructions can be used only when programming in the QnACPU general purpose mode:
(After program creation, program is compiled and converted to actual sequence program.)

- CHKCIR to CHKEND (updates check pattern of check instruction)
- IX to IXEND (index qualification for entire ladder)
- IXDEV, IXSET

APPENDIX 3 SPECIAL RELAY LIST

Special relays, SM, are internal relays whose applications are fixed in the programmable controller.
For this reason, they cannot be used by sequence programs in the same way as the normal internal relays.
However, they can be turned ON or OFF as needed in order to control the QnACPU.

The headings in the table that follows have the following meanings.

Item	Function of Item
Number	• Indicates the number of the special relay.
Name	• Indicates the name of the special relay.
Meaning	• Indicates the nature of the special relay.
Explanation	• Contains detailed information about the nature of the special relay.
Set by (When set)	• Indicates whether the relay is set by the system or user, and, if it is set by the system, when setting is performed. <Set by> S : Set by system U : Set by user (in sequence program or test operation at a peripheral device) S/U : Set by both system and user <When set> → indicated only if setting is done by system. Each END : Set during each END processing Initial : Set only during initial processing (when power supply is turned ON, or when going from STOP to RUN) Status change : Set only when there is a change in status Error : Set when error is generated Instruction execution : Set when instruction is executed Request : Set only when there is a user request (through SM, etc.)
Corresponding ACPU M9[ ][ ][ ]	• Indicates special relay M9[ ][ ][ ]/ corresponding to the ACPU. (Change and notation when there has been a change in contents) • Items indicated as "New" have been newly added for QnACPU

For details on the following items, see these manuals:
- Networks → MELSECNET/10 Network System Reference Manual for QnA
- SFC → QnACPU Programming Manual (SFC)

Special Relay List

(1) Diagnostic Information

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9[ ][ ][ ]
SM0	Diagnostic errors	OFF : No error ON : Error	• ON if diagnosis results show error occurrence (Includes external diagnosis) • Stays ON subsequently even if normal operations restored	S (Error)	New
SM1	Self-diagnostic error	OFF : No self-diagnosis errors ON : Self-diagnosis	• Comes ON when an error occurs as a result of self -diagnosis. • Stays ON subsequently even if normal operations restored	S (Error)	M9008
SM5	Error common information	OFF : No error common information ON : Error common information	• When SM0 is ON, ON if there is error common information	S (Error)	New
SM16	Error individual information	OFF : No error individual information ON : Error individual information	• When SM0 is ON, ON if there is error individual information	S (Error)	New
SM50	Error reset	OFF → ON: Error reset	• Conducts error reset operation • See Chapter 5 for further information	U	New
SM51	Battery low latch	OFF : Normal ON : Battery low	• ON if battery voltage at CPU or memory card drops below rated value. Stays ON subsequently even after normal operation is restored • Synchronous with BAT. ALARM LED	S (Error)	M9007
SM52	Battery low	OFF : Normal ON : Battery low	• Same as SM51, but goes OFF subsequently when battery voltage returns to normal.	S (Error)	M9006
SM53	AC DOWN detection	OFF : AC DOWN not detected ON : AC DOWN detected	• Comes ON when there is a momentary power interruption not exceeding 20 ms; reset by turning the power OFF then ON again.	S (Error)	M9005
SM54	MINI link errors	OFF : Normal ON : Error	• Goes ON if MINI (S3) link error is detected at even one of the installed AJ71PT32 (S3) modules. Stays ON subsequently even after normal operation is restored.	S (Error)	M9004
SM56	Operation Errors	OFF : Normal ON : Operation error	• ON when operation error is generated. • Stays ON subsequently even if normal operations restored	S (Error)	M9011
SM60	Blown fuse detection	OFF : Normal ON : Module with blown fuse	• Comes ON even if there is only one output module with a blown fuse, and remains ON even after return to normal. • Blown fuse state is checked even for remote I/O station output modules.	S (Error)	M9000
SM61	I/O module Verification error	OFF : Normal ON : Error	• Comes ON if there is a discrepancy between the actual I/O modules and the registered information when the power is turned on. • I/O module verification is also conducted for remote I/O station modules.	S (Error)	M9002
SM62	Annunciator detection	OFF : Not detected ON : Detected	• Goes ON if even one annunciator F goes ON.	S (Instruction execution)	M9009
SM80	CHK detection	OFF : Not detected ON : Detected	• Goes ON if error is detected by CHK instruction. • Stays ON subsequently even after normal operation is restored.	S (Instruction execution)	New
SM90	Startup of watchdog timer for step transition (Enabled only when SFC program exists)	OFF : Not started (watchdog timer reset) ON : Started (watchdog timer started)	Corresponds to SD90 • Goes ON when measurement of step transition watchdog timer is commenced. • Resets watchdog timer when it goes OFF.	U	M9108
SM91			Corresponds to SD91		M9109
SM92			Corresponds to SD92		M9110
SM93			Corresponds to SD93		M9111
SM94			Corresponds to SD94		M9112
SM95			Corresponds to SD95		M9113
SM96			Corresponds to SD96		M9114
SM97			Corresponds to SD97		New
SM98			Corresponds to SD98		New
SM99			Corresponds to SD99		New

Special Relay List

(2) System information

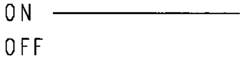
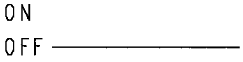
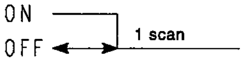
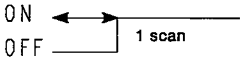
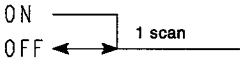
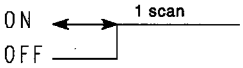
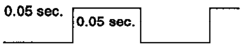
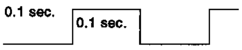
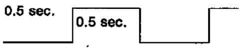
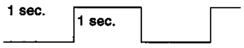
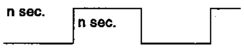
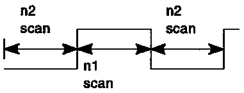
Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9[ ][ ][ ]
SM202	LED off command	OFF → ON : LED off	• At change from OFF to ON, the LEDs corresponding to the individual bits at SD202 go off	U	New
SM203	STOP contact	STOP state	• Goes ON at STOP state	S (Status change)	M9042
SM204	PAUSE contact	PAUSE state	• Goes ON at PAUSE state	S (Status change)	M9041
SM205	STEP-RUN contact	STEP-RUN state	• Goes ON at STEP-RUN state	S (Status change)	M9054
SM206	PAUSE enable coil	OFF : PAUSE disabled ON : PAUSE enabled	• PAUSE state is entered if this relay is ON when the remote PAUSE contact goes ON	U	M9040
SM210	Clock data set request	OFF : Ignored ON : Set request	• When this relay goes from OFF to ON, clock data being stored from SD210 through SD213 after execution of END instruction for changed scan is written to the clock device.	U	M9025
SM211	Clock data error	OFF : No error ON : Error	• ON when error is generated in clock data (SD210 through SD213) value, and OFF if no error is detected.	S (Request)	M9026
SM212	Clock data display	OFF : Ignored ON : Display	• Displays clock data as month, day, hour, minute, and second at the LED display at front of CPU. (Enabled only for Q3ACPU and Q4ACPU)	U	M9027
SM213	Clock data read request	OFF : Ignored ON : Read request	• When this relay is ON, clock data is read to SD210 through SD213 as BCD values.	U	M9028
SM250	Max. loaded I/O read	OFF : Ignored ON : Read	• When this relay goes from OFF to ON, maximum loaded I/O number is read to SD250.	U	New
SM251	I/O change flag	OFF : No replacement ON : Replacement	(QnA, Q4AR) • After the head I/O number of the I/O module being replaced is set in SD251 is set, on-line I/O module replacement is enabled when this relay is ON. (Only one module can be replaced at each setting.) • To replace an I/O module in the RUN state, use the program or a peripheral device to turn this relay ON; to replace an I/O module in the STOP state, turn this relay ON in the test mode of a peripheral device. • Do not switch between RUN and STOP states until I/O module replacement is completed.	U (END)	M9094
SM252	I/O change OK	OFF : Replacement prohibited ON : Replacement enabled	(QnA, Q4AR) • Goes ON when I/O replacement is OK.	S (Send)	New
SM255	MELSECNET/10 module 1 information	OFF : Operative network ON : Standby network	• Goes ON for standby network (If no designation has been made concerning active or standby, active is assumed.)	S (Initial)	New
SM256		OFF : Reads ON : Does not read	• For refresh from link to CPU (B, W, etc.) indicate whether to read from the link module.	U	New
SM257		OFF : Writes ON : Does not write	• For refresh from CPU to link (B, W, etc.), designate whether to write to the link module.	U	New
SM260	MELSECNET/10 module 2 information	OFF : Operative network ON : Standby network	• Goes ON for standby network (If no designation has been made concerning active or standby, active is assumed.)	S (Initial)	New
SM261		OFF : Reads ON : Does not read	• For refresh from link to CPU (B, W, etc.) indicate whether to read from the link module.	U	New
SM262		OFF : Writes ON : Does not write	• For refresh from CPU to link (B, W, etc.), designate whether to write to the link module.	U	New
SM265	MELSECNET/10 module 3 information	OFF : Operative network ON : Standby network	• Goes ON for standby network (If no designation has been made concerning active or standby, active is assumed.)	S (Initial)	New
SM266		OFF : Reads ON : Does not read	• For refresh from link to CPU (B, W, etc.) indicate whether to read from the link module.	U	New
SM267		OFF : Writes ON : Does not write	• For refresh from CPU to link (B, W, etc.), designate whether to write to the link module.	U	New
SM270	MELSECNET/10 module 4 information	OFF : Operative network ON : Standby network	• Goes ON for standby network (If no designation has been made concerning active or standby, active is assumed.)	S (Initial)	New
SM271		OFF : Reads ON : Does not read	• For refresh from link to CPU (B, W, etc.) indicate whether to read from the link module.	U	New
SM272		OFF : Writes ON : Does not write	• For refresh from CPU to link (B, W, etc.), designate whether to write to the link module.	U	New

Special Relay List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9[ ][ ][ ]
SM320	Presence /absence of SFC program	OFF : SFC program absent ON : SFC program present	• ON if SFC program is correctly registered, and OFF if not registered. • Goes OFF if SFC dedicated instruction is not correct.	S (Initial)	M9100
SM321	Start/stop SFC program	OFF : SFC program stop ON : SFC program start	• Initial value is set at the same value as SM320. (Goes ON automatically if SFC program is present.) • SFC program will not execute if this goes OFF prior to SFC program processing. • Subsequently, starts SFC program when this goes from OFF to ON. • Subsequently, stops SFC program when this goes from ON to OFF.	S (Initial) U	M9101 format change
SM322	SFC program start state	OFF : Initial start ON : Restart	• Initial value is set at ON or OFF depending on parameters. • When OFF, all execution states are cleared from time SFC program was stopped; starts from the initial step of block where the start request was made. • When ON, starts from execution block and execution step active at time SFC program was stopped. (ON is enabled only when resumptive start has been designated at parameters.) • SM902 is not automatically designated for latch.	S (Initial) U	M9102 format change
SM323	Presence /absence of continuous transition for entire block	OFF : Continuous transition not effective ON : Continuous transition effective	• When OFF, transition occurs at one scan/one step, for all blocks. • When ON, transition occurs continuously for all blocks in one scan. • In designation of individual blocks, priority is given to the continuous transition bit of the block. (Designation is checked when block starts.)	U	M9103
SM324	Continuous transition prevention flag	OFF : When transition is executed ON : When no transition	• When continuous transition is effective, goes ON when continuous transition is not being executed; goes OFF when continuous transition is being executed. • Normally ON when continuous transition is not effective.	S (Instruction execution)	M9104
SM325	Output mode at block stop	OFF : OFF ON : Preserves	When block stops, selects active step operation output. • All coil outputs go OFF when OFF. • Coil outputs are preserved when ON.	S (Initial) U	M9196
SM326	SFC device clear mode	OFF : Clear device ON : Preserves device	Selects the device status when the stopped CPU is run after the sequence program or SFC program has been modified when the SFC program exists.	U	New
SM327	Output during end step execution	OFF : OFF ON : Preserves	Selects the output action of the step being held when a block is ended by executing the end step. • All coil outputs go OFF when OFF. • Coil outputs are preserved when ON.	S (Initial) U	New

Special Relay List

(3) System clocks/counters

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9[ ][ ][ ]
SM400	Always ON		• Normally is ON	S (Every END processing)	M9036
SM401	Always OFF		• Normally is OFF	S (Every END processing)	M9037
SM402	ON for 1 scan only after RUN		• After RUN, ON for one scan only. • This connection can be used for scan execution type programs only.	S (Every END processing)	M9038
SM403	After RUN, OFF for 1 scan only		• After RUN, OFF for 1 scan only • This connection can be used for scan execution type programs only.	S (Every END processing)	M9039
SM404	Low speed execution type program ON for 1 scan only after RUN		• After RUN, ON for one scan only. • This contact can be used for low-speed execution type programs only.	S (Every END processing)	New
SM405	Low speed execution type program After RUN, OFF for 1 scan only		• After RUN, OFF for 1 scan only • This contact can be used for low-speed execution type programs only.	S (Every END processing)	New
SM410	0.1 second clock		• Repeatedly changes between ON and OFF at each designated time interval. • Operation continues even during STOP. • When power supply is turned OFF, or reset is performed, goes from OFF to start.	S (Status change)	M9030
SM411	0.2 second clock				M9031
SM412	1 second clock				M9032
SM413	2 second clock				M9033
SM414	2n second clock		• Goes between ON and OFF in accordance with the number of seconds designated by SD414.		M9034 format change
SM420	User timing clock No.0		• Relay repeats ON/OFF switching at fixed scan intervals. • When power supply is turned ON, or reset is performed, goes from OFF to start. • The ON/OFF intervals are set with the DUTY instruction. DUTY n1 n2 SM420	S (Every END processing)	M9020
SM421	User timing clock No.1				M9021
SM422	User timing clock No.2				M9022
SM423	User timing clock No.3				M9023
SM424	User timing clock No.4				M9024
SM430	User timing clock No.5		• For use with SM420 through SM424 low speed programs.	S (Every END processing)	New
SM431	User timing clock No.6				
SM432	User timing clock No.7				
SM433	User timing clock No.8				
SM434	User timing clock No.9				

Special Relay List

(4) Scan information

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9[ ][ ][ ]
SM510	Low speed program execution flag	OFF : Completed or not executed ON : Execution under way.	• Goes ON when low-speed execution type program is executed.	S (Every END processing)	New
SM551	Reads module service interval	OFF : Ignored ON : Read	• When this goes from OFF to ON, the module service interval designated by SD550 is read to SD551 through 552.	U	New

(5) Memory cards

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9[ ][ ][ ]
SM600	Memory card A usable flags	OFF : Unusable ON : Use enabled	• ON when memory card A is ready for use by user	S (Initial)	New
SM601	Memory card A protect flag	OFF : No protect ON : Protect	• Goes ON when memory card A protect switch is ON	S (Initial)	New
SM602	Drive 1 flag	OFF : No drive 1 ON : Drive 1 present	• Goes ON when drive 1 (card 1 RAM area) is present	S (Initial)	New
SM603	Drive 2 flag	OFF : No drive 2 ON : Drive 2 present	• Goes ON when drive 2 (card 1 ROM area) is present	S (Initial)	New
SM604	Memory card A in-use flag	OFF : Not in use ON : In use	• Goes ON when memory card A is in use	S (Initial)	New
SM605	Memory card A remove/insert prohibit flag	OFF : Remove/insert enabled ON : Remove/insert prohibited	• Goes ON when memory card A cannot be inserted or removed	U	New
SM620	Memory card B usable flags	OFF : Unusable ON : Use enabled	(QnA, Q4AR) • ON when memory card B is ready for use by user	S (Initial)	New
SM621	Memory card B protect flag	OFF : No protect ON : Protect	(QnA, Q4AR) • Goes ON when memory card B protect switch is ON	S (Initial)	New
SM622	Drive 3 flag	OFF : No drive 3 ON : Drive 3 present	(QnA, Q4AR) • Goes ON when drive 3 (card 2 RAM area) is present	S (Initial)	New
SM623	Drive 4 flag	OFF : No drive 4 ON : Drive 4 present	(QnA, Q4AR) • Goes ON when drive 4 (card 2 ROM area) is present	S (Initial)	New
SM624	Memory card B in-use flag	OFF : Not in use ON : In use	(QnA, Q4AR) • Goes ON when memory card B is in use	S (Initial)	New
SM625	Memory card B remove/insert prohibit flag	OFF : Remove/insert enabled ON : Remove/insert prohibited	(QnA, Q4AR) • Goes ON when memory card B cannot be inserted or removed	U	New
SM640	File register use	OFF : File register not in use ON : File register in use	• Goes ON when file register is in use	S (Status change)	New
SM650	Comment use	OFF : Comment not used ON : Comment in use	• Goes ON when comment file is in use	S (Status change)	New
SM660	Boot operation	OFF : Internal memory execution ON : Boot operation in progress	• Goes ON while boot operation is in process • Goes OFF if boot designation switch is OFF	S (Status change)	New
SM672	Memory card A file register access range flag	OFF : Within access range ON : Outside access range	• Goes ON when access is made to area outside the range of file register R of memory card A (Set within END processing.) • Reset at user program	S/U	New
SM673	Memory card B file register access range flag	OFF : Within access range ON : Outside access range	• Goes ON when access is made outside the range of file registers, R. of memory card B. (Set within END processing.) • Reset at user program	S/U	New

Special Relay List

(6) Instruction-Related Special Relays

Number	**Name**	**Meaning**	**Explanation**	**Set by (When Set)**	**Corresponding ACPU M9[ ][ ][ ]**
SM700	Carry flag	OFF : Carry OFF ON : Carry ON	• Carry flag used in application instruction	S (Instruction execution)	M9012
SM701	Number of output characters selection	OFF : 16 characters output ON : Outputs until NUL	• When SM701 is OFF, 16 characters of ASCII code are output. • When SM701 is ON, output conducted until NUL (00_H) code is encountered.	U	M9049
SM702	Search method	OFF : Search next ON : 2-part search	• Designates method to be used by search instruction. • Data must be arranged for 2-part search.	U	New
SM703	Sort order	OFF : Ascending order ON : Descending order	• The sort instruction is used to designate whether data should be sorted in ascending order or in descending order.	U	New
SM704	Block comparison	OFF : Non-match found ON : All match	• Goes ON when all data conditions have been met for the BKCMP instruction.	S (Instruction execution)	New
SM707	Selection of real number instruction processing type	OFF: Speed oriented ON: Accuracy oriented	(Q4AR) • When SM707 is OFF, real number instructions are processed at high speed. • When it is ON, real number instructions are processed with high accuracy.	U	New
SM710	CHK instruction priority ranking flag	OFF : Conditions priority ON : Pattern priority	• Remains as originally set when OFF. • CHK priorities updated when ON.	S (Instruction execution)	New
SM711	Divided transmission status	OFF : Other than during divided processing ON : During divided processing	• In processing of AD57(S1), goes ON when screen is split for transfer, and goes OFF when split processing is completed.	S (Instruction execution)	M9065
SM712	Transmission processing selection	OFF : Batch transmission ON : Divided transmission	• In processing of AD57(S1), goes ON when canvas screen is divided for transfer.	S (Instruction execution)	M9066
SM714	Communication request registration area BUSY signal	OFF : Communication request to remote terminal module enabled ON : Communication request to remote terminal module disabled	• Used to determine whether communications requests to remote terminal modules connected to the AJ71PT32-S3 can be executed or not.	S (Instruction execution)	M9081
SM715	EI flag	0 : During DI 1 : During EI	ON when EI instruction is being executed.	S (Instruction execution)	New
SM736	PKEY instruction execution in progress flag	OFF : Instruction not executed ON : Instruction execution	• ON when PKEY instruction is being executed. Goes OFF when CR is input, or when input character string reaches 32 characters.	S (Instruction execution)	New
SM737	Keyboard input reception flag for PKEY instruction	OFF : Keyboard input reception enabled ON : Keyboard input reception disabled	• Goes ON when keyboard input is being conducted. Goes when keyboard input has been stored at the CPU.	S (Instruction execution)	New
SM738	MSG instruction reception flag	OFF : Instruction not executed ON : Instruction execution	• Goes ON when MSG instruction is executed.	S (Instruction execution)	New
SM774	PID bumpless processing	OFF : Forces match ON : Does not force match	• In manual mode, designates whether or not to force the SV value to match the PV value.	U	New
SM775	Selection of link refresh processing during COM instruction execution	OFF : Performs link refresh ON : No link refresh performed	• Select whether or not to perform link refresh processing in cases where only general data processing will be conducted during the execution of the COM instruction.	U	New

(7) Debug

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9[ ][ ][ ]
SM800	Sampling trace preparation	OFF : Not prepared ON : Ready	• Goes ON when sampling trace is ready	S (Status change)	New
SM801	Sampling trace start	OFF : Suspend ON : Start	• Sampling trace started when this goes ON • Suspended when OFF (Related special M all OFF)	U	M9047
SM802	Sampling trace execution in progress	OFF : Suspend ON : Start	• Goes ON during execution of sampling trace	S (Status change)	M9046
SM803	Sampling trace trigger	OFF → ON : Start	• Sampling trace trigger goes ON when this goes from OFF to ON (Identical to STRA instruction execution state)	U	M9044
SM804	After sampling trace trigger	OFF : Not after trigger ON : After trigger	• Goes ON after sampling trace trigger	S (Status change)	New
SM805	Sampling trace completed	OFF : Not completed ON : End	• Goes ON at completion of sampling trace	S (Status change)	9043
SM806	Status latch preparation	OFF : Not prepared ON : Ready	• Goes ON when status latch is ready	S (Status change)	New
SM807	Status latch command	OFF → ON : Latch	• Runs status latch command	U	New
SM808	Status latch completion	OFF : Latch not completed ON : Latch completed	• Comes ON when status latch is completed.	S (Status change)	9055
SM809	Status latch clear	OFF → ON : Clear	• Enable next status latch	U	New
SM810	Program trace preparation	OFF : Not ready ON : Ready	• Goes ON when program trace is ready	S (Status change)	New
SM811	Start program trace	OFF : Suspend ON : Start	• Program trace started when this goes ON • Suspended when OFF (Related special M all OFF)	U	New
SM812	Program trace execution underway	OFF : Suspend ON : Start	• ON when program trace execution is underway	S (Status change)	New
SM813	Program trace trigger	OFF → ON : Start	• Program trace trigger goes ON when this goes from OFF to ON (Identical to PTRA instruction execution status)	U	New
SM814	After program trace trigger	OFF : Not after trigger ON : After trigger	• Goes ON after program trace trigger	S (Status change)	New
SM815	Program trace completion	OFF : Not completed ON : End	• Goes ON at completion of program trace	S (Status change)	New
SM820	Step trace preparation	OFF : Not prepared ON : Ready	• Goes ON after program trace registration, at ready.	S (Status change)	
SM821	Step trace starts	OFF : Suspend ON : Start	• When this goes ON, step trace is started • Suspended when OFF (Related special M all OFF)	U	M9182 format change
SM822	Step trace execution underway	OFF : Suspend ON : Start	• Goes ON when step trace execution is underway • Goes OFF at completion or suspension	S (Status change)	M9181
SM823	After step trace trigger	OFF : Not after trigger ON : Is after first trigger	• Goes ON if even 1 block within the step trace being executed is triggered. • Goes OFF when step trace is commenced.	S (Status change)	
SM824	Step trace After trigger	OFF : Is not after all triggers ON : Is after all triggers	• Goes ON if all blocks within the step trace being executed are triggered. • Goes OFF when step trace is commenced.	S (Status change)	
SM825	Step trace completed	OFF : Not completed ON : End	• Goes ON at step trace completion. • Goes OFF when step trace is commenced.	S (Status change)	M9180
SM826	Sampling trace error	OFF : Normal ON : Errors	• Goes ON if error occurs during execution of sampling trace.	S (Status change)	New
SM827	Status latch error	OFF : Normal ON : Errors	• Goes ON if error occurs during execution of status latch.	S (Status change)	New
SM828	Program trace error	OFF : Normal ON : Errors	• Goes ON if error occurs during execution of program trace.	S (Status change)	New

(8) Latch area

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9[ ][ ][ ]
SM900	Power cut file	OFF : No power cut file ON : Power cut file present	• Goes ON if a file is present during access when power is interrupted.	S/U (Status change)	New
SM910	RKEY registration flag	OFF : Keyboard input not registered ON : Keyboard input registered	• Goes ON at registration of keyboard input. • OFF if keyboard input is not registered.	S (Instruction execution)	New

(9) A to QnA conversion correspondences

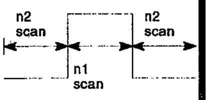
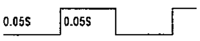
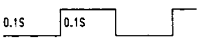
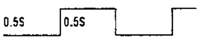
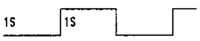
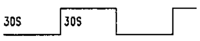
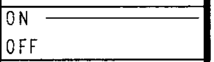
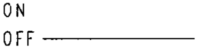
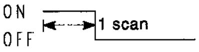
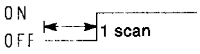
The special relays obtained when ACPU special relays M9000 through M9255 are subjected to A - QnA conversion are described here.

These special relays are all set by the system, and cannot be turned ON or OFF with user programs.

Users who wish to turn these relays ON or OFF should edit the programs at the special relays for QnA.

For details on the ACPU special relays, see the user's manuals for the individual CPUs and the MELSECNET and MELSECNET/B data link system reference manuals.

Special Relay List

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning
M9000	SM1000	SM60	Fuse blown	OFF : Normal ON : Fuse blown module with blown fuse present
M9002	SM1002	SM61	I/O module verification error	OFF : Normal ON : Error
M9004	SM1004	SM54	MINI link error	OFF : Normal ON : Error
M9005	SM1005	SM53	AC DOWN detection	OFF : AC DOWN not detected ON : AC DOWN detected
M9006	SM1006	SM52	Battery low	OFF : Normal ON : Battery low
M9007	SM1007	SM51	Battery low latch	OFF : Normal ON : Battery low
M9008	SM1008	SM1	Self-diagnostic error	OFF : No error ON : Error
M9009	SM1009	SM62	Annunciator detection	OFF : No F number detected ON : F number detected
M9011	SM1011	SM56	Operation error flag	OFF : No error ON : Error
M9012	SM1012	SM700	Carry flag	OFF : Carry OFF ON : Carry ON
M9016	SM1016	—	Data memory clear flag	OFF : Ignored ON : Output cleared
M9017	SM1017	—	Data memory clear flag	OFF : Ignored ON : Output cleared
M9020	SM1020	SM420	User timing clock No.0	
M9021	SM1021	SM421	User timing clock No.1	
M9022	SM1022	SM422	User timing clock No.2	
M9023	SM1023	SM423	User timing clock No.3	
M9024	SM1024	SM424	User timing clock No.4	
M9025	SM1025	SM210	Clock data set request	OFF : Ignored ON : Set request present used
M9026	SM1026	SM211	Clock data error	OFF : No error ON : Error
M9027	SM1027	SM212	Clock data display	OFF : Ignored ON : Display
M9028	SM1028	SM213	Clock data read request	OFF : Ignored ON : Read request
M9029	SM1029	—	Batch processing of data communications requests	OFF : Batch processing not conducted ON : Batch processing conducted
M9030	SM1030	SM410	0.1 second clock	
M9031	SM1031	SM411	0.2 second clock	
M9032	SM1032	SM412	1 second clock	
M9033	SM1033	SM413	2 second clock	
M9034	SM1034	SM414	1 minute clock	
M9036	SM1036	SM400	Always ON	
M9037	SM1037	SM401	Always OFF	
M9038	SM1038	SM402	ON for 1 scan only after RUN	
M9039	SM1039	SM403	RUN flag (After RUN, OFF for 1 scan only)	
M9040	SM1040	SM206	PAUSE enable coil	OFF : PAUSE disabled ON : PAUSE enabled
M9041	SM1041	SM204	PAUSE status contact	OFF : PAUSE not in effect ON : PAUSE in effect
M9042	SM1040	SM203	STOP status contact	OFF : STOP not in effect ON : STOP in effect
M9043	SM1043	SM805	Sampling Trace completed	OFF : Sampling trace in progress ON : Sampling trace completed
M9044	SM1044	SM803	Sampling trace	0 → 1 STRA Same as execution 1 → 0 STRAR Same as execution
M9045	SM1045	—	Watchdog timer (WDT) reset	OFF : Does not reset WDT ON : Resets WDT

Special Relay List (Continued)

ACPU Special Relay	Special Relay after Conver-sion	Special Relay for Modifica-tion	Name	Meaning
M9046	SM1046	SM802	Sampling trace	OFF : Trace not in progress ON : Trace in progress
M9047	SM1047	SM801	Sampling trace preparations	OFF : Sampling trace suspended ON : Sampling trace started
M9049	SM1049	SM701	Selection of number of characters output	OFF : Output until NULL code encountered ON : 16 characters output
M9051	SM1051	—	CHG instruction execution disable	OFF : Enabled ON : Disable
M9052	SM1052	—	SEG instruction switch	OFF : 7SEG segment display ON : I/O partial refresh
M9054	SM1054	SM205	STEP RUN flag	OFF : STEP RUN not in effect ON : STEP RUN in effect
M9055	SM1055	SM808	Status latch completion flag	OFF : Not completed ON : Completed
M9056	SM1056	—	Main side P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested
M9057	SM1057	—	Sub side P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested
M9058	SM1058	—	Main program P, I set completion	Momentarily ON at P, I set completion
M9059	SM1059	—	Sub program P, I set completion	Momentarily ON at P, I set completion
M9060	SM1060	—	Sub program 2 P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested
M9061	SM1061	—	Sub program 3 P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested
M9065	SM1065	SM711	Divided processing execution detection	OFF : Divided processing not underway ON : During divided processing
M9066	SM1066	SM712	Divided processing request flag	OFF : Batch processing ON : Divided processing
M9070	SM1070	—	A8UPU/A8PUJ required search time	OFF : Read time not shortened ON : Read time shortened
M9081	SM1081	SM714	Communication request registration area BUSY signal	OFF : Empty spaces in communication request registration area ON : No empty spaces in communication request registration area

ACPU Special Relay	Special Relay after Conver-sion	Special Relay for Modifica-tion	Name	Meaning
M9084	SM1084	—	Error check	OFF : Error check executed ON : No error check
M9091	SM1091	—	Instruction error flag	OFF : No error ON : Error
M9094	SM1094	SM251	I/O change flag	OFF : Replacement ON : No replacement
M9100	SM1100	SM320	Presence/ absence of SFC program	OFF : SFC programs not used ON : SFC programs used
M9101	SM1101	SM321	Start/stop SFC program	OFF : SFC programs stop ON : SFC programs start
M9102	SM1102	SM322	SFC program start state	OFF : Initial Start ON : Continue
M9103	SM1103	SM323	Presence/ absence of continuous transition	OFF : Continuous transition not effective ON : Continuous transition effective
M9104	SM1104	SM324	Continuous transition suspension flag	OFF : When transition is completed ON : When no transition
M9108	SM1108	SM90	Step transition watchdog timer start (equivalent of D9108)	OFF : Watchdog timer reset ON : Watchdog timer reset start
M9109	SM1109	SM91	Step transition watchdog timer start (equivalent of D9109)	
M9110	SM1110	SM92	Step transition watchdog timer start (equivalent of D9110)	
M9111	SM1111	SM93	Step transition watchdog timer start (equivalent of D9111)	
M9112	SM1112	SM94	Step transition watchdog timer start (equivalent of D9112)	
M9113	SM1113	SM95	Step transition watchdog timer start (equivalent of D9113)	
M9114	SM1114	SM96	Step transition watchdog timer start (equivalent of D9114)	
M9180	SM1180	SM825	Active step sampling trace completion flag	OFF : Trace started ON : Trace completed
M9181	SM1181	SM822	Active step sampling trace execution flag	OFF : Trace not being executed ON : Trace execution under way
M9182	SM1182	SM821	Active step sampling trace permission	OFF : Trace disable/suspend ON : Trace enable
M9196	SM1196	SM325	Operation output at block stop	OFF : Coil output OFF ON : Coil output ON

Special Relay List (Continued)

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning
M9197 • M9198	SM1197 • SM1198	—	Switch between blown fuse and I/O verification error display	Display is changed depending on combination of M9197 ON/OFF state and M9198 ON/OFF state.
M9199	SM1199	—	On-line recovery of sampling trace status latch data	OFF : Does not perform data recovery ON : Performs data recovery
M9200	SM1200	—	ZNRD instruction (LRDP instruction for ACPU) reception (for master station)	OFF : Not accepted ON : Accepted
M9201	SM1201	—	ZNRD instruction (LRDP instruction for ACPU) completion (for master station)	OFF : Not completed ON : End
M9202	SM1202	—	ZNWR instruction (LWTP instruction for ACPU) reception (for master station)	OFF : Not accepted ON : Accepted
M9203	SM1203	—	ZNWR instruction (LWTP instruction for ACPU) completion (for master station)	OFF : Not completed ON : End
M9204	SM1204	—	ZNWR instruction (LWTP instruction for ACPU) reception (for local station)	OFF : Not completed ON : End
M9205	SM1205	—	ZNRD instruction (LRDP instruction for ACPU) reception (for local station)	OFF : Not completed ON : End
M9206	SM1206	—	Host station link parameter error	OFF : Normal ON : Abnormal
M9207	SM1207	—	Link parameter check results	OFF : YES ON : NO
M9208	SM1208	—	Sets master station B and W transmission range (for lower link master stations only).	OFF : Transmits to tier 2 and tier 3 ON : Transmits to tier 2 only
M9209	SM1209	—	Link parameter check command (for lower link master stations only).	OFF : Executing the check function ON : Check non-execution
M9210	SM1210	—	Link card error (for master station)	OFF : Normal ON : Abnormal
M9211	SM1211	—	Link module error (for local station use)	OFF : Normal ON : Abnormal
M9224	SM1224	—	Link state	OFF : Online ON : Offline, station-to-station test, or self-loopback test
M9225	SM1225	—	Forward loop error	OFF : Normal ON : Abnormal
M9226	SM1226	—	Reverse loop error	OFF : Normal ON : Abnormal

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning
M9227	SM1227	—	Loop test state	OFF : Not being executed ON : Forward or reverse loop test execution underway
M9232	SM1232	—	Local station operation state	OFF : RUN or STEP RUN state ON : STOP or PAUSE state
M9233	SM1233	—	Local station error detect state	OFF : No errors ON : Error detection
M9235	SM1235	—	Local station, remote I/O station parameter error detect state	OFF : No errors ON : Error detection
M9236	SM1236	—	Local station, remote I/O station initial communications state	OFF : No communications ON : Communications underway
M9237	SM1237	—	Local station, remote I/O station error	OFF : Normal ON : Abnormal
M9238	SM1238	—	Local station, remote I/O station forward or reverse loop error	OFF : Normal ON : Abnormal
M9240	SM1240	—	Link state	OFF : Online ON : Offline, station-to-station test, or self-loopback test
M9241	SM1241	—	Forward loop line error	OFF : Normal ON : Abnormal
M9242	SM1242	—	Reverse loop line error	OFF : Normal ON : Abnormal
M9243	SM1243	—	Loopback implementation	OFF : Loopback not being conducted ON : Loopback implementation
M9246	SM1246	—	Data not received	OFF : Reception ON : No reception
M9247	SM1247	—	Data not received	OFF : Reception ON : No reception
M9250	SM1250	—	Parameters not received	OFF : Reception ON : No reception
M9251	SM1251	—	Link relay	OFF : Normal ON : Abort
M9252	SM1252	—	Loop test state	OFF : Not being executed ON : Forward or reverse loop test execution underway
M9253	SM1253	—	Master station operation state	OFF : RUN or STEP RUN state ON : STOP or PAUSE state
M9254	SM1254	—	Local station other than host station operation state	OFF : RUN or STEP RUN state ON : STOP or PAUSE state
M9255	SM1255	—	Local station other than host station error	OFF : Normal ON : Abnormal

Special Relay List

(10) For redundant systems (Host system CPU information *1) for Q4AR only
SM1510 to SM1599 are only valid for redundant systems. All off for standalone systems.

Number	Name	Meaning	Explanation	Set by (When Set)	ACPU M9[ ][ ][ ]
SM1500	Hold mode	OFF : No-hold ON : Hold	• Specifies whether or not to hold the output value when a range over occurs for the S.IN instruction range check.	U	New
SM1501	Hold mode	OFF : No-hold ON : Hold	• Specifies whether or not the output value is held when a range over occurs for the S.OUT instruction range check.	U	New
SM1510	Operation mode	OFF : Redundant system backup mode, independent system ON : Redundant system separate mode	• Turns on when the operating mode is redundant system separate.	S (Each END)	New
SM1511	Start mode when power supply is on	OFF : System A fixed mode ON : Previous control system latch mode	• Turns on when the start mode for a redundant system when the power is turned on is the previous control system latch mode.	S (Initial)	New
SM1512	Start mode when CPU is started	OFF : Initial start ON : Hot start	• Turns on when the CPU operation mode is hot start when the redundant system is started up.	S (Initial)	New
SM1513	Operation status when CPU is started	OFF : Initial start ON : Hot start	• Turns on when the CPU operation mode is hot start when the redundant system is actually start up.	S (Initial)	New
SM1514	Operation mode when CPU is switched	OFF : Initial start ON : Hot start	• Turns on when the operation is hot start when the CPU operation is switched for a redundant system.	S (Initial)	New
SM1515	Output hold mode	OFF : Output reset ON : Output hold	• Turns on when the output mode during a stop error is output hold.	S (Each END)	New
SM1516	Operation system status	OFF : Control system ON : Standby system	• Turns on when the CPU operation system status is the standby system.	S (Status change)	New
SM1517	CPU startup status	OFF : Power supply on startup ON : Operation system switch starup	• Turns on when the CPU is started up by the operation system switch.	S (Status change)	New
SM1518	Tracking execution mode	OFF : Batch operation mode ON : Carryover mode	• When turned off when the tracking memory is in use during END, standby is executed until execution is possible. • When turned on when the tracking memory is being used -during END, this is repeatedly executed until the next END	U	New

*1 Stores the whole system CPU information.

Special Relay List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	ACPU D9[ ][ ]
SM1520	Data tracking transmission link specification	OFF : No trigger ON : Trigger	• Specified the blocks to trigger when the data is transmitted by the data tracking instruction S. TRUCK. SM1520 Block 1 SM1521 Block 2 SM1522 Block 3 SM1523 Block 4 SM1524 Block 5 SM1525 Block 6 SM1526 Block 7 SM1527 Block 8 SM1528 Block 9 SM1529 Block 10 SM1530 Block 11 SM1531 Block 12 SM1532 Block 13 SM1533 Block 14 SM1534 Block 15 SM1535 Block 16 SM1536 Block 17 SM1537 Block 18 SM1538 Block 19 SM1539 Block 20 SM1540 Block 21 SM1541 Block 22 SM1542 Block 23 SM1543 Block 24 SM1544 Block 25 SM1545 Block 26 SM1546 Block 27 SM1547 Block 28 SM1548 Block 29 SM1549 Block 30 SM1550 Block 31 SM1551 Block 32 SM1552 Block 33 SM1553 Block 34 SM1554 Block 35 SM1555 Block 36 SM1556 Block 37 SM1557 Block 38 SM1558 Block 39 SM1559 Block 40 SM1560 Block 41 SM1561 Block 42 SM1562 Block 43 SM1563 Block 44 SM1564 Block 45 SM1565 Block 46 SM1566 Block 47 SM1567 Block 48 SM1568 Block 49 SM1569 Block 50 SM1570 Block 51 SM1571 Block 52 SM1572 Block 53 SM1573 Block 54 SM1574 Block 55 SM1575 Block 56 SM1576 Block 57 SM1577 Block 58 SM1578 Block 59 SM1579 Block 60 SM1580 Block 61 SM1581 Block 62 SM1582 Block 63 SM1583 Block 64	U	New
SM1521					
SM1522					
SM1523					
SM1524					
SM1525					
SM1526					
SM1527					
SM1528					
SM1529					
SM1530					
SM1531					
SM1532					
SM1533					
SM1534					
SM1535					
SM1536					
SM1537					
SM1538					
SM1539					
SM1540					
SM1541					
SM1542					
SM1543					
SM1544					
SM1545					
SM1546					
SM1547					
SM1548					
SM1549					
SM1550					
SM1551					
SM1552					
SM1553					
SM1554					
SM1555					
SM1556					
SM1557					
SM1558					
SM1559					
SM1560					
SM1561					
SM1562					
SM1563					
SM1564					
SM1565					
SM1566					
SM1567					
SM1568					
SM1569					
SM1570					
SM1571					
SM1572					
SM1573					
SM1574					
SM1575					
SM1576					
SM1577					
SM1578					
SM1579					
SM1580					
SM1581					
SM1582					
SM1583					
SM1590	Switching status from the network module	OFF : Normal ON : Switching unsuccessful	• Turns on when could not be executed normally when the network module detects a network error and issues a switching request to the host system CPU.	S (Error ocurrs)	New

Special Relay List

(11) For redundant system (Other system CPU information *1) for Q4AR only
SM1600 to SM1650 only valid for the CPU redundant system backup mode, so they cannot be refreshed during the separate mode. Either the backup mode or the separate mode is valid for the SM4651 to SM1699. SM1600 to SM1699 are all turned off for stand-alone system.

Number	Name	Meaning	Explanation	Set by (When Set)	For full system SM[ ][ ]*2
SM1600	Diagnosis error	OFF : No error ON : Error	• Turns on if a error occurs in the diagnosis results.(Including external diagnosis) • Remains on even if returns to normal thereafter.	S (Each END)	SM0
SM1601	Self diagnosis error	OFF : No self diagnosis error ON : Self diagnosis error	• Turns on when an error occurs in the self-diagnosis results. • Remains on even if returns to normal thereafter.	S (Each END)	SM1
SM1605	Error common information	OFF : No error common information ON : Error common information	• Turns on when there is error common information and the SM1600 is on.	S (Each END)	SM5
SM1616	Error individual information	OFF : No error individual information ON : Error individual information	• Turns on when there is error individual information and the SM1600 is on.	S (Each END)	SM16
SM1653	STOP contact	STOP status	• Turns on when in the STOP status.	S (Each END)	SM203
SM1654	PAUSE contact	PAUSE status	• Turns on when in the PAUSE status.	S (Each END)	SM204
SM1655	STEP-RUN contact	STEP-RUN status	• Turns on when in the STEP-RUN status.	S (Each END)	SM205

*1 Stores other system CPU diagnostic information and system information.

*2 This shows the special relay(SM[ ][ ]) for the host system CPU.

Special Relay List

(12) For redundant system (tracking) for Q4AR only
Either the backup mode or the second mode is valid for SM1700 to SM1799. All is turned off for standalone system.

Number	Name	Meaning	Explanation	Set by (When Set)	for ACPU M9[ ][ ][ ]
SM1700	Tracking execution flag	OFF : Execution not possible ON : Execution possible	• Turns on when tracking is executed normally.	S (status change	New
SM1712	Transmission trigger end flag	OFF : Transmission uncompleted ON : Transmission end	• One scan turns on when the corresponding data transmission has been completed. SM1712 Block 1 SM1713 Block 2 SM1714 Block 3 SM1715 Block 4 SM1716 Block 5 SM1717 Block 6 SM1718 Block 7 SM1719 Block 8 SM1720 Block 9 SM1721 Block 10 SM1722 Block 11 SM1723 Block 12 SM1724 Block 13 SM1725 Block 14 SM1726 Block 15 SM1727 Block 16 SM1728 Block 17 SM1729 Block 18 SM1730 Block 19 SM1731 Block 20 SM1732 Block 21 SM1733 Block 22 SM1734 Block 23 SM1735 Block 24 SM1736 Block 25 SM1737 Block 26 SM1738 Block 27 SM1739 Block 28 SM1740 Block 29 SM1741 Block 30 SM1742 Block 31 SM1743 Block 32 SM1744 Block 33 SM1745 Block 34 SM1746 Block 35 SM1747 Block 36 SM1748 Block 37 SM1749 Block 38 SM1750 Block 39 SM1751 Block 40 SM1752 Block 41 SM1753 Block 42 SM1754 Block 43 SM1755 Block 44 SM1756 Block 45 SM1757 Block 46 SM1758 Block 47 SM1759 Block 48 SM1760 Block 49 SM1761 Block 50 SM1762 Block 51 SM1763 Block 52 SM1764 Block 53 SM1765 Block 54 SM1766 Block 55 SM1767 Block 56 SM1768 Block 57 SM1769 Block 58 SM1770 Block 59 SM1771 Block 60 SM1772 Block 61 SM1773 Block 62 SM1774 Block 63 SM1775 Block 64	S (status change)	New
SM1713					
SM1714					
SM1715					
SM1716					
SM1717					
SM1718					
SM1719					
SM1720					
SM1721					
SM1722					
SM1723					
SM1724					
SM1725					
SM1726					
SM1727					
SM1728					
SM1729					
SM1730					
SM1731					
SM1732					
SM1733					
SM1734					
SM1735					
SM1736					
SM1737					
SM1738					
SM1739					
SM1740					
SM1741					
SM1742					
SM1743					
SM1744					
SM1745					
SM1746					
SM1747					
SM1748					
SM1749					
SM1750					
SM1751					
SM1752					
SM1753					
SM1754					
SM1755					
SM1756					
SM1757					
SM1758					
SM1759					
SM1760					
SM1761					
SM1762					
SM1763					
SM1764					
SM1765					
SM1766					
SM1767					
SM1768					
SM1769					
SM1770					
SM1771					
SM1772					
SM1773					
SM1774					
SM1775					

APPENDIX 4 SPECIAL REGISTER LIST

The special registers, SD, are internal registers with fixed applications in the programmable controller.
For this reason, it is not possible to use these registers in sequence programs in the same way that normal registers are used.
However, it is possible to write data to them in order to conduct QnACPU controls.
Data stored in the special registers are stored as BIN values if no special designation has been made to the contrary.

The headings in the table that follows have the following meanings.

Item	Function of Item
Number	• Indicates special register number
Name	• Indicates name of special register
Meaning	• Indicates contents of special register
Explanation	• Discusses contents of special register in more detail
Explanation Set by (When set)	• Indicates whether the relay is set by the system or user, and, if it is set by the system, when setting is performed. <Set by> S : Set by system U : Set by user (sequence programs or test operations from peripheral devices) S/U: Set by both system and user <When set>→Indicated only for registers set by system Each END : Set during each END processing Initial : Set only during initial processing (when power supply is turned ON, or when going from STOP to RUN) Status change : Set only when there is a change in status Error : Set when error occurs Instruction execution : Set when instruction is executed Request : Set only when there is a user request (through SM, etc.)
Corresponding ACPU D9[ ][ ][ ]	• Indicates corresponding special register in ACPU (D9[ ][ ][ ]) (Change and notation when there has been a change in contents) • Items indicated as "New" have been newly added for QnACPU

For details on the following items, see these manuals:
- Networks →MELSECNET/10 Network System Reference Manual for QnA
- SFC →QnACPU Programming Manual (SFC)

Special Register List

(1) Diagnostic Information

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD0	Diagnostic errors	Diagnosis error code	• Error codes for errors found by diagnosis are stored as BIN data. • Contents identical to latest fault history information.	S (Error)	D9008 format change
SD1	Clock time for diagnosis error occurrence	Clock time for diagnosis error occurrence	• Year (last two digits) and month that SD0 data was updated is stored as BCD 2-digit code. B15 to B8 B7 to B0 Year (0 to 99) | Month (1 to 12) (Example) : October, 1995 H9510	S (Error)	New
SD2			• The day and hour that SD0 was updated is stored as BCD 2-digit code. B15 to B8 B7 to B0 Day (1 to 31) | Hour (0 to 23) (Example) : 10 p.m. on 25th H2510		
SD3			• The minute and second that SD0 data was updated is stored as BCD 2-digit code. B15 to B8 B7 to B0 Minutes (0 to 59) | Seconds (0 to 59) (Example) : 35 min. 48 sec. (past the hour) H3548		
SD4	Error information categories	Error information category code	Category codes which help indicate what type of information is being stored in the common information areas (SD5 through SD15) and the individual information areas (SD16 through SD26) are stored here. B15 to B8 B7 to B0 Individual information category codes | Common information category codes • The common information category codes store the following codes: 0 : No error 1 : Unit/module No. 2 : File name/Drive name 3 : Time (value set) 4 : Program error location 5 : Switch cause (for Q4AR only) • The individual information category codes store the following codes: 0 : No error 1 : (Open) 2 : File name/Drive name 3 : Time (value actually measured) 4 : Program error location 5 : Parameter number 6 : Annunciator number 7 : Check instruction malfunction number	S (Error)	New
SD5	Error common information	Error common information	• Common information corresponding to the error codes (SD0) is stored here. • The following four types of information are stored here: (1) Slot No. Number | Meaning SD5 | Slot No. SD6 | I/O No. SD7 | (Vacant) SD8 SD9 SD10 SD11 SD12 SD13 SD14 SD15	S (Error)	New
SD6					
SD7					
SD8					
SD9					
SD10					
SD11					
SD12					
SD13					
SD14					
SD15					

Special Register List (Continued)

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When Set)</th><th>For ACPU D9[ ][ ][ ]</th></tr>
<tr><td>SD15</td><td>Error common information</td><td>Error common information</td><td>

- (2) File name/Drive name

<table>
<tr><th>Number</th><th colspan="2">Meaning</th></tr>
<tr><td>SD5</td><td colspan="2">Drive</td></tr>
<tr><td>SD6</td><td colspan="2" rowspan="4">File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD7</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td><td>Extension*</td><td>$2E_H$ (.)</td></tr>
<tr><td>SD11</td><td colspan="2">(ASCII code: 3 characters)</td></tr>
<tr><td>SD12</td><td colspan="2" rowspan="4">(Vacant)</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>

(Example)
File name=
ABCDEFGH. IJK

<table>
<tr><th>B15 to B8</th><th>B7 to B0</th></tr>
<tr><td>42_H (B)</td><td>41_H (A)</td></tr>
<tr><td>44_H (D)</td><td>43_H (C)</td></tr>
<tr><td>46_H (F)</td><td>45_H (E)</td></tr>
<tr><td>48_H (H)</td><td>47_H (G)</td></tr>
<tr><td>49_H (I)</td><td>$2D_H$ (.)</td></tr>
<tr><td>48_H (K)</td><td>$4A_H$ (J)</td></tr>
</table>

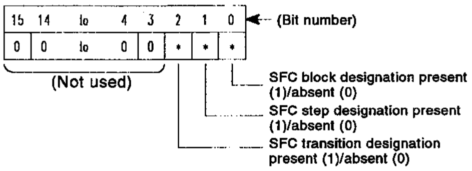
(3) Time (value set)

<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5</td><td>Time : 1 µs units (0 to 999 µs)</td></tr>
<tr><td>SD6</td><td>Time : 1 ms units (0 to 65535 ms)</td></tr>
<tr><td>SD7</td><td rowspan="9">(Vacant)</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td></tr>
<tr><td>SD11</td></tr>
<tr><td>SD12</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>

(4) Program error location

<table>
<tr><th>Number</th><th colspan="2">Meaning</th></tr>
<tr><td>SD5</td><td colspan="2" rowspan="4">File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD6</td></tr>
<tr><td>SD7</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td><td>Extension*</td><td>$2E_H$ (.)</td></tr>
<tr><td>SD10</td><td colspan="2">(ASCII code: 3 characters)</td></tr>
<tr><td>SD11</td><td colspan="2">Pattern*</td></tr>
<tr><td>SD12</td><td colspan="2">Block No.</td></tr>
<tr><td>SD13</td><td colspan="2">Step No./transition No.</td></tr>
<tr><td>SD14</td><td colspan="2" rowspan="2">Sequence step No. (L)
Sequence step No. (H)</td></tr>
<tr><td>SD15</td></tr>
</table>

* Contents of pattern data



(5) Switch cause

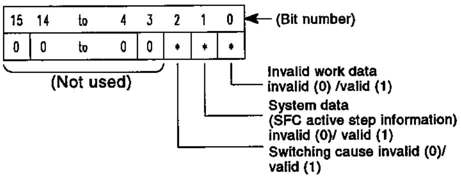
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5</td><td>Switch cause
(0: autoimatic switch/ 1: manual switch)</td></tr>
<tr><td>SD6</td><td>Switch direction (0:standby system to control system/ 1: control system to standby system)</td></tr>
<tr><td>SD7</td><td>Tracking flag *</td></tr>
<tr><td>SD8</td><td rowspan="8">(Vacant)</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td></tr>
<tr><td>SD11</td></tr>
<tr><td>SD12</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>

* Tracking flag contents
Shows whether or not the tracking data is valid.

</td><td>S (Error)</td><td>New</td></tr>
</table>

* : Refer to REMARK in APP-49.

Special Register List (Continued)

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When Set)</th><th>Corresponding ACPU D9[ ][ ][ ]</th></tr>
<tr><td>SD15</td><td></td><td></td><td>
</td><td></td><td></td></tr>
<tr><td>SD16</td><td rowspan="11">Error individual information</td><td rowspan="11">Error individual information</td><td rowspan="11">

- Individual information corresponding to error codes (SD0) is stored here.

(1) File name/Drive name

<table>
<tr><th>Number</th><th colspan="2">Meaning</th></tr>
<tr><td>SD16</td><td colspan="2">Drive</td></tr>
<tr><td>SD17</td><td colspan="2" rowspan="4">File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD18</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td></tr>
<tr><td>SD21</td><td>Extension*</td><td>$2E_H$ (.)</td></tr>
<tr><td>SD22</td><td colspan="2">(ASCII code: 3 characters)</td></tr>
<tr><td>SD23</td><td colspan="2" rowspan="4">(Vacant)</td></tr>
<tr><td>SD24</td></tr>
<tr><td>SD25</td></tr>
<tr><td>SD26</td></tr>
</table>

(Example)
File name=
ABCDEFGH. IJK

<table>
<tr><th>B15 to B8</th><th>B7 to B0</th></tr>
<tr><td>42_H (B)</td><td>41_H (A)</td></tr>
<tr><td>44_H (D)</td><td>43_H (C)</td></tr>
<tr><td>46_H (F)</td><td>45_H (E)</td></tr>
<tr><td>48_H (H)</td><td>47_H (G)</td></tr>
<tr><td>49_H (I)</td><td>$2D_H$ (.)</td></tr>
<tr><td>$4B_H$ (K)</td><td>$4A_H$ (J)</td></tr>
</table>

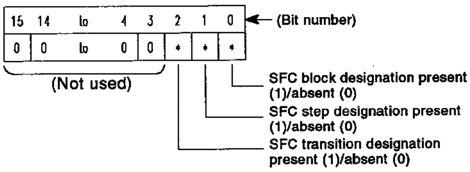
(2) Time (value actually measured)

<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>Time : 1 μs units (0 to 999 μs)</td></tr>
<tr><td>SD17</td><td>Time : 1 ms units (0 to 65535 ms)</td></tr>
<tr><td>SD18</td><td rowspan="9">(Vacant)</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td></tr>
<tr><td>SD21</td></tr>
<tr><td>SD22</td></tr>
<tr><td>SD23</td></tr>
<tr><td>SD24</td></tr>
<tr><td>SD25</td></tr>
<tr><td>SD26</td></tr>
</table>

(3) Program error location

<table>
<tr><th>Number</th><th colspan="2">Meaning</th></tr>
<tr><td>SD16</td><td colspan="2" rowspan="4">File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD17</td></tr>
<tr><td>SD18</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td><td>Extension*</td><td>$2E_H$ (.)</td></tr>
<tr><td>SD21</td><td colspan="2">(ASCII code: 3 characters)</td></tr>
<tr><td>SD22</td><td colspan="2">Pattern*</td></tr>
<tr><td>SD23</td><td colspan="2">Block No.</td></tr>
<tr><td>SD24</td><td colspan="2">Step No./transition No.</td></tr>
<tr><td>SD25</td><td colspan="2" rowspan="2">Sequence step No. (L)
Sequence step No. (H)</td></tr>
<tr><td>SD26</td></tr>
</table>

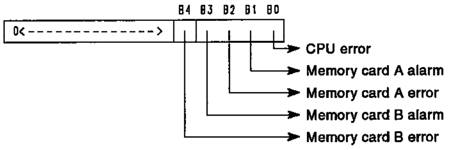
* Contents of pattern data



</td><td rowspan="11">S (Error)</td><td rowspan="11">New</td></tr>
<tr><td>SD17</td></tr>
<tr><td>SD18</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td></tr>
<tr><td>SD21</td></tr>
<tr><td>SD22</td></tr>
<tr><td>SD23</td></tr>
<tr><td>SD24</td></tr>
<tr><td>SD25</td></tr>
<tr><td>SD26</td></tr>
</table>

* : Refer to REMARK in APP-54.

Special Register List (Continued)

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When Set)</th><th>Corresponding ACPU D9[ ][ ]</th></tr>
<tr><td>SD16
SD17
SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>Error individual information</td><td>Error individual information</td><td>(Continued)
(4) Parameter number
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>Parameter number*1</td></tr>
<tr><td>SD17</td><td rowspan="10">(Vacant)</td></tr>
<tr><td>SD18</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td></tr>
<tr><td>SD21</td></tr>
<tr><td>SD22</td></tr>
<tr><td>SD23</td></tr>
<tr><td>SD24</td></tr>
<tr><td>SD25</td></tr>
<tr><td>SD26</td></tr>
</table>
(5) Annunciator number / CHK instruction malfunction number
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>No.</td></tr>
<tr><td>SD17</td><td rowspan="10">(Vacant)</td></tr>
<tr><td>SD18</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td></tr>
<tr><td>SD21</td></tr>
<tr><td>SD22</td></tr>
<tr><td>SD23</td></tr>
<tr><td>SD24</td></tr>
<tr><td>SD25</td></tr>
<tr><td>SD26</td></tr>
</table>
</td><td>S (Error)</td><td>New</td></tr>
<tr><td>SD50</td><td>Error reset</td><td>Error number that performs error reset</td><td>• Stores error number that performs error reset</td><td>U</td><td>New</td></tr>
<tr><td>SD51</td><td>Battery low latch</td><td>Bit pattern indicating where battery voltage drop occurred</td><td>• All corresponding bits go ON when battery voltage drops.
• Subsequently, these remain ON even after battery voltage has been returned to normal.

 </td><td>S (Error)</td><td>New</td></tr>
<tr><td>SD52</td><td>Battery low</td><td>Bit pattern indicating where battery voltage drop occurred</td><td>• Same configuration as SD51 above
• Subsequently, goes OFF when battery voltage is restored to normal.</td><td>S (Error)</td><td>New</td></tr>
<tr><td>SD53</td><td>AC DOWN detection</td><td>Number of times for AC DOWN</td><td>• 1 is added to the stored value each time the input voltage becomes 80 % or less of the rating while the CPU module is operating, and the value is stored in BIN code.</td><td>S (Error)</td><td>D9005</td></tr>
</table>

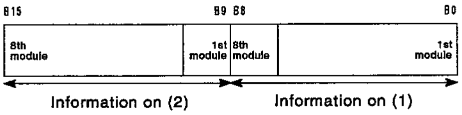
*1 : For details of parameters, refer to User's Manual of the CPU you are using.

REMARK

1) Extensions are shown below.

SD10	SD11		Extension name	File type
Higher 8 bits	Lower 8 bits	Higher 8 bits		
51H	50H	41H	QPA	Parameters
51H	50H	47H	QPG	Sequence program/SFC program
51H	43H	44H	QCD	Device comment
51H	44H	49H	QDI	Device initial value
51H	44H	52H	QDR	File register
51H	44H	53H	QDS	Simulation data
51H	44H	4CH	QDL	Local device
51H	54H	53H	QTS	Sampling trace data
51H	54H	4CH	QTL	Status latch data
51H	54H	50H	QTP	Program trace data
51H	54H	52H	QTR	SFC trace file
51H	46H	44H	QFD	Trouble history data

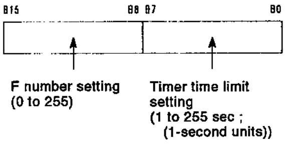
Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD54	MINI link errors	Error detection state	(1) The relevant station bit goes ON when any of the Installed MINI (-S3) X(n+0)/X(n+20), X(n+6)/(n+26), X(n+7)/(n+27) or X(n+8)/X(n+28) goes ON. (2) Goes ON when communications between the installed MINI (-S3) and the CPU are not possible.  	S (Error)	D9004 format change
SD60	Blown fuse number	Number of module with blown fuse	• Value stored here is the lowest station I/O number of the module with the blown fuse.	S (Error)	D9000
SD61	I/O module verification error number	I/O module verification error module number	• The lowest I/O number of the module where the I/O module verification number took place.	S (Error)	D9002
SD62	Annunciator number	Annunciator number	• The first annunciator number to be detected is stored here.	S (Instruction execution)	D9009
SD63	Number of annunciators	Number of annunciators	• Stores the number of annunciators searched.	S (Instruction execution)	D9124
SD64	Table of detected annunciator numbers	Annunciator detection number	When F goes ON due to OUT F or SET F , the F numbers which go progressively ON from SD64 through SD79 are registered. F numbers turned OFF by RST F are deleted from SD64 to SD79, and are shifted to the data register following the data register where the deleted F numbers had been stored. Execution of the LED R instruction shifts the contents of SD64 to SD79 up by one. (This can also be done by using the INDICATOR RESET switch on the front of the CPU of the Q3A/Q4ACPU.) After 16 annunciators have been detected, detection of the 17th will not be stored from SD64 through SD79.	S (Instruction execution)	D9125
SD65					D9126
SD66					D9127
SD67					D9128
SD68					D9129
SD69					D9130
SD70					D9131
SD71					D9132
SD72					New
SD73					New
SD74					New
SD75					New
SD76					New
SD77					New
SD78					New
SD79					New
SD80	CHK number	CHK number	• Error codes detected by the CHK instruction are stored as BCD code.	S (Instruction execution)	New
SD90	Step transition watchdog timer setting value (Enabled only when SFC program exists)	F number for timer set value and time over error	Corresponds to SM90	U	D9108
SD91			Corresponds to SM91		D9109
SD92			Corresponds to SM92		D9110
SD93			Corresponds to SM93		D9111
SD94			Corresponds to SM94		D9112
SD95			Corresponds to SM95		D9113
SD96			Corresponds to SM96		D9114
SD97			Corresponds to SM97		New
SD98			Corresponds to SM98		New
SD99			Corresponds to SM99		New

		SET F50	SET F25	SET F99	RST F25	SET F15	SET F70	SET F65	SET F38	SET F110	SET F151	SET F210	LEDR	
SD62	0	50	50	50	50	50	50	50	50	50	50	50	99	...(Number detected)
SD63	0	1	2	3	2	3	4	5	6	7	8	9	8	...(Number of annunciators detected)
SD64	0	50	50	50	50	50	50	50	50	50	50	50	99	
SD65	0	0	25	25	99	99	99	99	99	99	99	99	15	
SD66	0	0	0	99	0	15	15	15	15	15	15	15	70	
SD67	0	0	0	0	0	0	70	70	70	70	70	70	65	
SD68	0	0	0	0	0	0	0	65	65	65	65	65	38	
SD69	0	0	0	0	0	0	0	0	38	38	38	38	110	
SD70	0	0	0	0	0	0	0	0	0	110	110	110	151	
SD71	0	0	0	0	0	0	0	0	0	0	151	151	210	(Number detected)
SD72	0	0	0	0	0	0	0	0	0	0	0	210	0	
SD73	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD74	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD75	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD76	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD77	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD78	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD79	0	0	0	0	0	0	0	0	0	0	0	0	0	

SD90 to SD99:

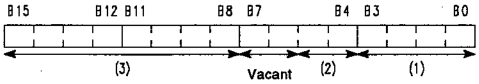
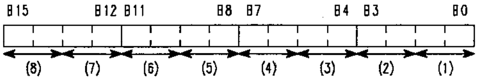
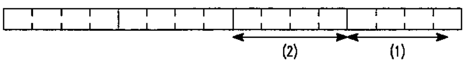
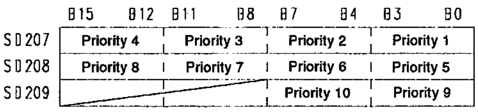
- F numbers which go ON at step transition watchdog timer set value and watchdog timer over errors.



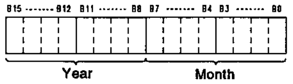
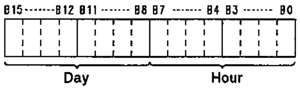
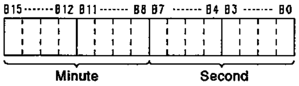
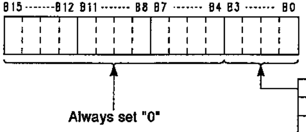
- Timer is started by turning SM90 through SM99 ON during active step, and if the transition conditions for the relevant steps are not met within the timer limits, the designated annunciator (F) will go ON.

Special Register List

(2) System information

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD200	State of switch	State of CPU switch	• The CPU switch state is stored in the following format:  (1) : CPU key State of switch — 0 : RUN, 1 : STOP, 2 : L.CLR (2) : Memory cards switch — B4 corresponds to card A, and B5 corresponds to card B OFF at 0; ON at 1 (3) : DIP switch — B8 through B12 correspond to SW1 through SW5 of system setting switch 1. B14 and B15 correspond to SW1 and SW2 of system setting switch 2, respectively. OFF at 0; ON at 1	S (Every END processing)	New
SD201	LED status	State of CPU-LED	• Information concerning which of the following states the LEDs on the CPU are in is stored in the following bit patterns: • 0 is off, 1 is on, and 2 is flicker  (1) : RUN (2) : ERROR (3) : USER (4) : BAT.ALARM (5) : BOOT (6) : CARD A (Memory card) (7) : CARD B (Memory card) (8) : Vacant	S (Status change)	New
SD202	LED off	Bit pattern of LED that is turned off	• Stores bit patterns of LEDs turned off (Only USER and BOOT enabled) • Turned off at 1, not turned off at 0	U	New
SD203	Operating state of CPU	Operating state of CPU	• The CPU operating state is stored as indicated in the following figure:  (1) : Operating state of CPU — 0 : RUN, 1 : STEP-RUN, 2 : STOP, 3 : PAUSE (2) : STOP/PAUSE cause — 0 : Key switch, 1 : Remote contact, 2 : Peripheral, computer link, or operation from some other remote source, 3 : Internal program instruction, 4 : Errors Note : Priority is earliest first	S (Every END processing)	D9015 format change
SD207	LED display priority ranking	Priorities 1 to 4	• When error is generated, the LED display (flicker) is made according to the error number setting priorities. • The setting areas for priorities are as follows:  Default Value SD207=H4321 SD208=H8765 SD209=H00A9 • No display is made if "0" is set. However, even if "0" has been set, information concerning CPU operation stop (including parameter settings) errors will be indicated by the LEDs without conditions. See Section 7.9.5 REMARK for the priority order.	U	D9038
SD208		Priorities 5 to 8			D3039 format change
SD209		Priorities 9 to 10			New

Special Register List (Continued)

Number	Name	Meaning		Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD210	Clock data	Clock data (year, month)		• The year (last two digits) and month are stored as BCD code at SD210 as shown below:  Example : July 1993 H9307	S/U (Request)	D9025
SD211	Clock data	Clock data (day, hour)		• The day and hour are stored as BCD code at SD211 as shown below:  Example : 31st, 10 a.m. H3110		D9026
SD212	Clock data	Clock data (minute, second)		• The minutes and seconds (after the hour) are stored as BCD code at SD212 as shown below:  Example : 35 min., 48 sec. (after the hour) H3548		D9027
SD213	Clock data	Clock data (day of week)		• The day of the week is stored as BCD code at SD213 as shown below:  Example : Friday H0005 Day of week 0 Sunday 1 Monday 2 Tuesday 3 Wednesday 4 Thursday 5 Friday 6 Saturday		D9028
SD220	LED display data	Display indicator data		• LED display ASCII data (16 characters) stored here. B15 to B8 / B7 to B0 SD220: 15th character from the right / 16th character from the right SD221: 13th character from the right / 14th character from the right SD222: 11th character from the right / 12th character from the right SD223: 9th character from the right / 10th character from the right SD224: 7th character from the right / 8th character from the right SD225: 5th character from the right / 6th character from the right SD226: 3rd character from the right / 4th character from the right SD227: 1st character from the right / 2nd character from the right	S (When changed)	New
SD221						
SD222						
SD223						
SD224						
SD225						
SD226						
SD227						
SD250	Loaded maximum I/O	Loaded maximum I/O No.		• When SM250 goes from OFF to ON, the upper 2 digits of the final I/O number plus 1 of the modules loaded are stored as BIN values.	S (Request END)	New
SD251	Head I/O No. for replacement	Head I/O number for module replacement		• Stores the upper two digits of the first I/O number of an I/O module that is removed/replaced in the online status. (default value : 100H)	U	D9094
SD254	NET/10 information	Number of modules installed		• Indicates the number of modules installed on NET/10.	S (Initial)	New
SD255		Information from 1st module	I/O No.	• NET/10 I/O number of first module installed		
SD256			Network No.	• NET/10 network number of first module installed		
SD257			Group number	• NET/10 group number of first module installed		
SD258			Station No.	• NET/10 station number of first module installed		
SD259			Standby information	• In the case of standby stations, the module number of the standby station is stored. (1 to 4)		
SD260 to SD264		Information from 2nd module		• Configuration is identical to that for the first module.		
SD265 to SD269		Information from 3rd module		• Configuration is identical to that for the first module.		
SD270 to SD274		Information from 4th module		• Configuration is identical to that for the first module.		

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD290	Device allocation (Same as parameter contents)	Number of points allocated for X	• Stores the number of points currently set for X devices.	S (Initial)	New
SD291		Number of points allocated for Y	• Stores the number of points currently set for Y devices		
SD292		Number of points allocated for M	• Stores the number of points currently set for M devices		
SD293		Number of points allocated for L	• Stores the number of points currently set for L devices		
SD294		Number of points allocated for B	• Stores the number of points currently set for B devices		
SD295		Number of points allocated for F	• Stores the number of points currently set for F devices		
SD296		Number of points allocated for SB	• Stores the number of points currently set for SB devices		
SD297		Number of points allocated for V	• Stores the number of points currently set for V devices		
SD298		Number of points allocated for S	• Stores the number of points currently set for S devices		
SD299		Number of points allocated for T	• Stores the number of points currently set for T device		
SD300		Number of points allocated for ST	• Stores the number of points currently set for ST devices		
SD301		Number of points allocated for C	• Stores the number of points currently set for C devices		
SD302		Number of points allocated for D	• Stores the number of points currently set for D devices		
SD303		Number of points allocated for W	• Stores the number of points currently set for W devices		
SD304		Number of points allocated for SW	• Stores the number of points currently set for SW devices		
SD392	Software version	Internal system software version	• Stores the internal system software version in ASCII code. The software version is stored in the higher byte position. The data in the lower byte position is indefinite. Higher byte | Lower byte For version "A", for example, "41H" is stored. Note: The internal system software version may differ from the version indicated by the version symbol printed on the case.		D9060

Special Register List

(3) System clocks/counters

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD412	1 second counter	Number of counts in 1-second units	• Following programmable controller CPU RUN, 1 is added each second • Count repeats from 0 to 32767 to -32768 to 0	S (Status change)	D9022
SD414	2n second clock setting	2n second clock units	• Stores value n of 2n second clock (Default is 30) • Setting can be made between 1 and 32767		
SD420	Scan counter	Number of counts in each scan	• Incremented by 1 for each scan execution after the PC CPU is set to RUN. * • Count repeats from 0 to 32767 to -32768 to 0	S (Every END processing)	New
SD430	Low speed scan counter	Number of counts in each scan	• Incremented by 1 for each scan execution after the PC CPU is set to RUN. • Count repeats from 0 to 32767 to -32768 to 0 • Used only for low speed execution type programs	S (Every END processing)	New

* : Not counted by the scan in an initial execution type program.

(4) Scan information

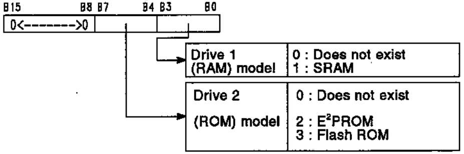
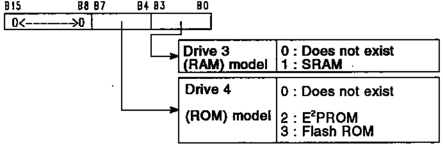
Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD500	Execution program No.	Execution type of program being executed	• Program number of program currently being executed is stored as BIN value.	S (Status change)	New
SD510	Low speed program No.	File name of low speed execution in progress	• Program number of low speed program currently being executed is stored as BIN value. • Enabled only when SM510 is ON.	S (Every END processing)	New
SD520	Current scan time	Current scan time (in 1 ms units)	• Stores current scan time (in 1 ms units) Range from 0 to 65535	S (Every END processing)	D9017 format change
SD521		Current scan time (in 1 µs units)	• Stores current scan time (in 1 µs units) Range from 00000 to 900 (Example) A current scan of 23.6 ms would be stored as follows: D520=23 D521=600		New
SD522	Initial scan time	Initial scan time (in 1 ms units)	• Stores scan time for first scan (in 1 ms units) • Range from 0 to 65535	S (First END processing)	New
SD523		Initial scan time (in 100 µs units)	• Stores scan time for first scan (in 1 µs units) • Range of 000 to 900		
SD524	Minimum scan time	Minimum scan time (in 1 ms units)	• Stores minimum value of scan time (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	D9018 format change
SD525		Minimum scan time (in 100 µs units)	• Stores minimum value of scan time (in 100 µs units) • Range of 000 to 900		New
SD526	Maximum scan time	Maximum scan time (in 1 ms units)	• Stores maximum value of scan time, excepting the first scan. (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	D9019 format change
SD527		Maximum scan time (in 100 µs units)	• Stores maximum value of scan time, excepting the first scan. (in 100 µs units) • Range of 000 to 900		New
SD528	For low speed execution type programs current scan time	Current scan time (in 1 ms units)	• Stores current scan time for low speed execution type program (in 1 ms units)	S (Every END processing)	New
SD529		Current scan time (in 100 µs units)	• Stores current scan time for low speed execution type program (in 100 µs units) • Range of 000 to 900		
SD532	Minimum scan time for low speed execution type programs	Minimum scan time (in 1 ms units)	• Stores minimum value of scan time for low speed execution type program (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	New
SD533		Minimum scan time (in 100 µs units)	• Stores minimum value of scan time for low speed execution type program (in 100 µs units) • Range of 000 to 900		
SD534	Maximum scan time for low speed execution type programs	Maximum scan time (in 1 ms units)	• Stores the maximum scan time for all except low speed execution type program's first scan. (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	New
SD535		Maximum scan time (in 100 µs units)	• Stores the maximum scan time for all except low speed execution type program's first scan. (in 100 µs units) • Range of 000 to 900		
SD540	END processing time	END processing time (in 1 ms units)	• Stores time from completion of scan program to start of next scan. (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	New
SD541		END processing time (in 100 µs units)	• Stores time from completion of scan program to start of next scan. (in 100 µs units) • Range of 000 to 900		
SD542	Constant scan wait time	Constant scan wait time (in 1 ms units)	• Stores wait time when constant scan time has been set. (in 1 ms units) • Range from 0 to 65535	S (First END processing)	New
SD543		Constant scan wait time (in 100 µs units)	• Stores wait time when constant scan time has been set. (in 100 µs units) • Range of 000 to 900		
SD544	Cumulative execution time for low speed execution type programs	Cumulative execution time for low speed execution type programs (in 1 ms units)	• Stores cumulative execution time for low speed execution type programs. (in 1 ms units) • Range from 0 to 65535 • Cleared to 0 following 1 low speed scan	S (Every END processing)	New
SD545		Cumulative execution time for low speed execution type programs (in 100 µs units)	• Stores cumulative execution time for low speed execution type programs. (in 100 µs units) • Range of 000 to 900 • Cleared to 0 following 1 low speed scan		

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD546	Execution time for low speed execution type programs	Execution time for low speed execution type programs (in 1 ms units)	• Stores low speed program execution time during 1 scan (in 1 ms units) • Range from 0 to 65535 • Stores each scan	S (Every END processing)	New
SD547		Execution time for low speed execution type programs (in 100 µs units)	• Stores low speed program execution time during 1 scan (in 100 µs units) • Range of 000 to 900 • Stores each scan		
SD548	Scan program execution time	Scan program execution time (in 1 ms units)	• Stores execution time for scan execution type program during 1 scan (in 1 ms units) • Range from 0 to 65535 • Stores each scan	S (Every END processing)	New
SD549		Scan program execution time (in 100 µs units)	• Stores execution time for scan execution type program during 1 scan (in 100 µs units) • Range of 000 to 900 • Stores each scan		
SD550	Service interval measurement module	Unit/module No.	• Sets I/O number for module that measures service interval	U	New
SD551	Service interval time	Module service interval (in 1 ms units)	• When SM551 is ON, stores service interval for module designated by SD550. (in 1 ms units) • Range from 0 to 65535	S (Request)	New
SD552		Module service interval (in 100 µs units)	• When SM551 is ON, stores service interval for module designated by SD550. (in 1 µs units) • Range from 000 to 999		

Special Register List

(5) Memory card

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD600	Memory card A models	Memory card A models	• Indicates memory card A model installed  	S (Initial)	New
SD602	Drive 1 (RAM) capacity	Drive 1 capacity	• Drive 1 capacity is stored in 1 K byte units	S (Initial)	New
SD603	Drive 2 (ROM) capacity	Drive 2 capacity	• Drive 2 capacity is stored in 1 K byte units	S (Initial)	New
SD604	Memory card A use conditions	Memory card A use conditions	• The use conditions for memory card A are stored as bit patterns (In use when ON) • The significance of these bit patterns is indicated below: B0 : Boot operation (QBT) B1 : Parameters (QPA) B2 : Device comments (QCD) B3 : Device initial value (QDI) B4 : File R (QDR) B5 : Sampling trace (QTS) B6 : Status latch (QTL) B7 : Program trace (QTP) B8 : Simulation data (QDS) B9 : CPU fault history (QFD) B10 : SFC trace (QTS) B11 : Local device (QDL) B12 : B13 : B14 : B15 :	S (Status change)	New
SD620	Memory card B models	Memory card B models	(QnA, Q4AR) • Indicates memory card B models installed  	S (Initial)	New
SD622	Drive 3 (RAM) capacity	Drive 3 capacity	(QnA, Q4AR) • Drive 3 capacity is stored in 1 K byte units	S (Initial)	New
SD623	Drive 4 (ROM) capacity	Drive 4 capacity	(QnA, Q4AR) • Drive 4 capacity is stored in 1 K byte units	S (Initial)	New
SD624	Memory card B use conditions	Memory card B use conditions	(QnA, Q4AR) • The use conditions for memory card B are stored as bit patterns (In use when ON) • The significance of these bit patterns is indicated below: B0 : Boot operation (QBT) B1 : Parameters (QPA) B2 : Device comments (QCD) B3 : Device initial value (QDI) B4 : File R (QDR) B5 : Sampling trace (QTS) B6 : Status latch (QTL) B7 : Program trace (QTP) B8 : Simulation data (QDS) B9 : CPU fault history (QFD) B10 : SFC trace (QTS) B11 : Local device (QDL) B12 : B13 : B14 : B15 :	S (Status change)	New
SD640	File register drive	Drive number:	• Stores drive number being used by file register	S (Status change)	New
SD641 SD642 SD643 SD644 SD645 SD646	File register file name	File register file name	• Stores file register file name (with extension) selected at parameters or by use of QDRSET instruction as ASCII code. B15 to B8 / B7 to B0 SD641: Second character / First character SD642: Fourth character / Third character SD643: Sixth character / Fifth character SD644: Eighth character / Seventh character SD645: First character of extension / 2EH(.) SD646: Third character of extension / Second character of extension	S (Status change)	New

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ]
SD647	File register capacity	File register capacity	• Stores the data capacity of the currently selected file register in 1 K word units.	S (Status change)	New
SD648	File register block number	File register block number	• Stores the currently selected file register block number.	S (Status change)	D9035
SD650	Comment drive	Comment drive number	• Stores the comment drive number selected at the parameters or by the QCDSET instruction.	S (Status change)	New
SD651 SD652 SD653 SD654 SD655 SD656	Comment file name	Comment file name	• Stores the comment file name (with extension) selected at the parameters or by the QCDSET instruction in ASCII code. B15 to B8 B7 to B0 SD651: Second character / First character SD652: Fourth character / Third character SD653: Sixth character / Fifth character SD654: Eighth character / Seventh character SD655: First character of extension / 2EH(.) SD656: Third character of extension / Second character of extension	S (Status change)	New
SD660	Boot operation designation file	Boot designation file drive number	• Stores the drive number where the boot designation file (*.QBT) is being stored.	S (Initial)	New
SD661 SD662 SD663 SD664 SD665 SD666		File name of boot designation file	• Stores the file name of the boot designation file (*.QBT). B15 to B8 B7 to B0 SD661: Second character / First character SD662: Fourth character / Third character SD663: Sixth character / Fifth character SD664: Eighth character / Seventh character SD665: First character of extension / 2EH(.) SD666: Third character of extension / Second character of extension	S (Initial)	New

(6) Instruction-Related Registers

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ]
SD705 SD706	Mask pattern	Mask pattern	(QnA, Q4AR) • During block operations, turning SM705 ON makes it possible to use the mask pattern being stored at SD705 (or at SD705 and SD706 if double words are being used) to operate on all data in the block with the masked values.	U	New
SD714	Number of vacant communication request registration areas	0 to 32	• Stores the number of vacant blocks in the communications request area for remote terminal modules connected to the AJ71PT32-S3.	S (During execution)	M9081
SD715 SD716 SD717	IMASK instruction mask pattern	Mask pattern	• Patterns masked by use of the IMASK instruction are stored in the following manner: B15 to B0 SD715: I15 I14 I13 I12 I11 I10 I9 I8 I7 I6 I5 I4 I3 I2 I1 I0 SD716: I31 I30 I29 I28 I27 I26 I25 I24 I23 I22 I21 I20 I19 I18 I17 I16 SD717: I47 I46 I45 I44 I43 I42 I41 I40 I39 I38 I37 I36 I35 I34 I33 I32	S (During execution)	New
SD718 SD719	Accumulator	Accumulator	• For use as replacement for accumulators used in A-series programs.	S/U	New
SD736	PKEY input	PKEY input	• SD that temporarily stores keyboard data input by means of the PKEY instruction.	S (During execution)	New

Special Register List (Continued)

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When Set)</th><th>Corresponding ACPU D9[ ][ ]</th></tr>
<tr><td>SD738</td><td rowspan="32">Message storage</td><td rowspan="32">Message storage</td><td rowspan="32">• Stores the message designated by the MSG instruction.
<table>
<tr><th></th><th>B15 to B8</th><th>B7 to B0</th></tr>
<tr><td>SD738</td><td>2nd character</td><td>1st character</td></tr>
<tr><td>SD739</td><td>4th character</td><td>3rd character</td></tr>
<tr><td>SD740</td><td>6th character</td><td>5th character</td></tr>
<tr><td>SD741</td><td>8th character</td><td>7th character</td></tr>
<tr><td>SD742</td><td>10th character</td><td>9th character</td></tr>
<tr><td>SD743</td><td>12th character</td><td>11th character</td></tr>
<tr><td>SD744</td><td>14th character</td><td>13th character</td></tr>
<tr><td>SD745</td><td>16th character</td><td>15th character</td></tr>
<tr><td>SD746</td><td>18th character</td><td>17th character</td></tr>
<tr><td>SD747</td><td>20th character</td><td>19th character</td></tr>
<tr><td>SD748</td><td>22nd character</td><td>21st character</td></tr>
<tr><td>SD749</td><td>24th character</td><td>23rd character</td></tr>
<tr><td>SD750</td><td>26th character</td><td>25th character</td></tr>
<tr><td>SD751</td><td>28th character</td><td>27th character</td></tr>
<tr><td>SD752</td><td>30th character</td><td>29th character</td></tr>
<tr><td>SD753</td><td>32nd character</td><td>31st character</td></tr>
<tr><td>SD754</td><td>34th character</td><td>33rd character</td></tr>
<tr><td>SD755</td><td>36th character</td><td>35th character</td></tr>
<tr><td>SD756</td><td>38th character</td><td>37th character</td></tr>
<tr><td>SD757</td><td>40th character</td><td>39th character</td></tr>
<tr><td>SD758</td><td>42nd character</td><td>41st character</td></tr>
<tr><td>SD759</td><td>44th character</td><td>43rd character</td></tr>
<tr><td>SD760</td><td>46th character</td><td>45th character</td></tr>
<tr><td>SD761</td><td>48th character</td><td>47th character</td></tr>
<tr><td>SD762</td><td>50th character</td><td>49th character</td></tr>
<tr><td>SD763</td><td>52nd character</td><td>51st character</td></tr>
<tr><td>SD764</td><td>54th character</td><td>53rd character</td></tr>
<tr><td>SD765</td><td>56th character</td><td>55th character</td></tr>
<tr><td>SD766</td><td>58th character</td><td>57th character</td></tr>
<tr><td>SD767</td><td>60th character</td><td>59th character</td></tr>
<tr><td>SD768</td><td>62nd character</td><td>61st character</td></tr>
<tr><td>SD769</td><td>64th character</td><td>63rd character</td></tr>
</table></td><td rowspan="32">S (During execution)</td><td rowspan="32">New</td></tr>
<tr><td>SD739</td></tr>
<tr><td>SD740</td></tr>
<tr><td>SD741</td></tr>
<tr><td>SD742</td></tr>
<tr><td>SD743</td></tr>
<tr><td>SD744</td></tr>
<tr><td>SD745</td></tr>
<tr><td>SD746</td></tr>
<tr><td>SD747</td></tr>
<tr><td>SD748</td></tr>
<tr><td>SD749</td></tr>
<tr><td>SD750</td></tr>
<tr><td>SD751</td></tr>
<tr><td>SD752</td></tr>
<tr><td>SD753</td></tr>
<tr><td>SD754</td></tr>
<tr><td>SD755</td></tr>
<tr><td>SD756</td></tr>
<tr><td>SD757</td></tr>
<tr><td>SD758</td></tr>
<tr><td>SD759</td></tr>
<tr><td>SD760</td></tr>
<tr><td>SD761</td></tr>
<tr><td>SD762</td></tr>
<tr><td>SD763</td></tr>
<tr><td>SD764</td></tr>
<tr><td>SD765</td></tr>
<tr><td>SD766</td></tr>
<tr><td>SD767</td></tr>
<tr><td>SD768</td></tr>
<tr><td>SD769</td></tr>
</table>

Special Register List

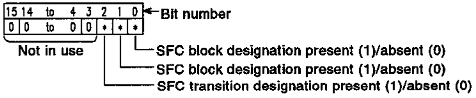
(7) Debug

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When Set)</th><th>Corresponding ACPU D9[ ][ ][ ]</th></tr>
<tr><td>SD806
SD807
SD808
SD809
SD810
SD811</td><td>Status latch file name</td><td>Status latch file name</td><td>
- Stores file name (with extension) from point in time when status latch was conducted as ASCII code.

<table>
<tr><th></th><th>B15 to B8</th><th>B7 to B0</th></tr>
<tr><td>SD806</td><td>Second character</td><td>First character</td></tr>
<tr><td>SD807</td><td>Fourth character</td><td>Third character</td></tr>
<tr><td>SD808</td><td>Sixth character</td><td>Fifth character</td></tr>
<tr><td>SD809</td><td>Eighth character</td><td>Seventh character</td></tr>
<tr><td>SD810</td><td>First character of extension</td><td>2EH(.)</td></tr>
<tr><td>SD811</td><td>Third character of extension</td><td>Second character of extension</td></tr>
</table>
</td><td>S (During execution)</td><td>New</td></tr>
<tr><td>SD812
SD813
SD814
SD815
SD816</td><td>Status latch step</td><td>Status latch step</td><td>
- Stores step number from point in time when status latch was conducted

<table>
<tr><td>SD812</td><td>Pattern*</td></tr>
<tr><td>SD813</td><td>Block No.</td></tr>
<tr><td>SD814</td><td>Step No./transition No.</td></tr>
<tr><td>SD815</td><td>Sequence step No. (L)</td></tr>
<tr><td>SD816</td><td>Sequence step No.(H)</td></tr>
</table>

* Contents of pattern data



</td><td>S (During execution)</td><td>D9055 format change</td></tr>
</table>

Special Register List

(8) Latch area

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When Set)</th><th>Corresponding ACPU D9[ ][ ]</th></tr>
<tr><td>SD900</td><td>Drive where power was interrupted</td><td>Access file drive number during power loss</td><td>• Stores drive number if file was being accessed during power loss.</td><td>S (Status change)</td><td>New</td></tr>
<tr><td>SD901
SD902
SD903
SD904
SD905
SD906</td><td>File name active during power loss</td><td>Access file name during power loss</td><td>• Stores file name (with extension) in ASCII code if file was being accessed during power loss.
<table>
<tr><th></th><th>B15 to B8</th><th>B7 to B0</th></tr>
<tr><td>SD901</td><td>2nd character</td><td>1st character</td></tr>
<tr><td>SD902</td><td>4th character</td><td>3rd character</td></tr>
<tr><td>SD903</td><td>6th character</td><td>5th character</td></tr>
<tr><td>SD904</td><td>8th character</td><td>7th character</td></tr>
<tr><td>SD905</td><td>1st character of extension</td><td>2E_H(.)</td></tr>
<tr><td>SD906</td><td>3rd character of extension</td><td>2nd character of extension</td></tr>
</table></td><td>S (Status change)</td><td>New</td></tr>
<tr><td>SD910
SD911
SD912
SD913
SD914
SD915
SD916
SD917
SD918
SD919
SD920
SD921
SD922
SD923
SD924
SD925</td><td>RKEY input</td><td>RKEY input</td><td>• Stored in sequence that PU key code was entered.
<table>
<tr><th></th><th>B15 to B8</th><th>B7 to B0</th></tr>
<tr><td>SD910</td><td>2nd character</td><td>1st character</td></tr>
<tr><td>SD911</td><td>4th character</td><td>3rd character</td></tr>
<tr><td>SD912</td><td>6th character</td><td>5th character</td></tr>
<tr><td>SD913</td><td>8th character</td><td>7th character</td></tr>
<tr><td>SD914</td><td>10th character</td><td>9th character</td></tr>
<tr><td>SD915</td><td>12th character</td><td>11th character</td></tr>
<tr><td>SD916</td><td>14th character</td><td>13th character</td></tr>
<tr><td>SD917</td><td>16th character</td><td>15th character</td></tr>
<tr><td>SD918</td><td>18th character</td><td>17th character</td></tr>
<tr><td>SD919</td><td>20th character</td><td>19th character</td></tr>
<tr><td>SD920</td><td>22nd character</td><td>21st character</td></tr>
<tr><td>SD921</td><td>24th character</td><td>23rd character</td></tr>
<tr><td>SD922</td><td>26th character</td><td>25th character</td></tr>
<tr><td>SD923</td><td>28th character</td><td>27th character</td></tr>
<tr><td>SD924</td><td>30th character</td><td>29th character</td></tr>
<tr><td>SD925</td><td>32nd character</td><td>31st character</td></tr>
</table></td><td>S (During execution)</td><td>New</td></tr>
</table>

(9) A to QnA conversion correspondences

ACPU special registers D9000 to D9255 correspond to the special registers SD1000 to SD1255 after A-series to the QnA-series conversion.

These special registers are all set by the system, and users cannot use them to set program data.

Users who need to set data with these registers should edit the special registers for the QnA.

However, before conversion users could set data at special registers D9200 to D9255 only, and after conversion users can also set data at registers 1200 to 1255.

For more detailed information concerning the contents of the ACPU special registers, see the individual CPU users manual, and the MELSECNET and MELSECNET/B data link system reference manual.

REMARK

Supplemental explanation on "Special Register for Modification" column

(1) For the device numbers for which a special register for modification is specified, modify it to the special register for QnACPU.

(2) For the device numbers for which ⊟ is specified, special register after conversion can be used.

(3) Device numbers for which ☒ is specified do not function for QnACPU.

Special Register List

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9000	SD1000	SD60	Fuse blown	Number of module with blown fuse
D9002	SD1002	SD61	I/O module verification error	I/O module verification error module number
D9004	SD1004	SD54	MINI link errors	Stores setting status made at parameters (modules 1 to 8)
D9005	SD1005	SD53	AC DOWN counter	Number of times for AC DOWN
D9008	SD1008	SD0	Self-diagnostic error	Self-diagnostic error number
D9009	SD1009	SD62	Annunciator detection	F number at which external failure has occurred
D9010	SD1010	—	Error step	Step number at which operation error has occurred.
D9011	SD1011	—	Error step	Step number at which operation error has occurred.
D9014	SD1014	—	I/O control mode	I/O control mode number
D9015	SD1015	SD203	Operating state of CPU	Operating state of CPU
D9016	SD1016	—	Program number	Stores sequence program under execution as BIN value
D9017	SD1017	SD520	Scan time	Minimum scan time (10 ms units)
D9018	SD1018	SD524	Scan time	Scan time (10 ms units)
D9019	SD1019	SD526	Scan time	Maximum scan time (10 ms units)
D9020	SD1020	—	Constant scan	Constant scan time (User sets in 10 ms units)
D9021	SD1021	—	Scan time	Scan time (in 1 ms units)
D9022	SD1022	SD412	1 second counter	Count in units of 1 ms.
D9025	SD1025	SD210	Clock data	Clock data (year, month)
D9026	SD1026	SD211	Clock data	Clock data (day, hour)
D9027	SD1027	SD212	Clock data	Clock data (minute, second)
D9028	SD1028	SD213	Clock data	Clock data (day of week)
D9035	SD1035	SD648	Extension file register	Use block No.
D9036	SD1036		Extension file register for designation of device number	Device number when individual devices from extension file register are directly accessed
D9037	SD1037			
D9038	SD1038	SD207	LED display priority ranking	Priorities 1 to 4
D9039	SD1039	SD208		Priorities 5 to 7
D9044	SD1044	—	For sampling trace	Step or time during sampling trace
D9049	SD1049	—	Work area for SFC	Block number of extension file register
D9050	SD1050	—	SFC program error number	Error code generated by SFC program
D9051	SD1051	—	Error block	Block number where error occurred
D9052	SD1052	—	Error step	Step number where error occurred
D9053	SD1053	—	Error transition	Transition condition number where error occurred
D9054	SD1054	—	Error sequence step	Sequence step number where error occurred

Special Register List (Continued)

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9055	SD1055	SD812	Status latch	Status latch step
D9072	SD1072	—	PC communications check	Computer link data check
D9081	SD1081	SD714	Number of empty blocks in communications request registrtion area	Number of empty blocks in communications request registration area
D9085	SD1085	—	Register for setting time check value	Default value 10s
D9090	SD1090	—	Number of special functions modules over	Number of special functions modules over
D9091	SD1091	—	Detailed error code	Self-diagnosis detailed error code
D9094	SD9094	SD251	Head I/O number for replacement	Head I/O number for replacement
D9100	SD9100	SD1300	Fuse blown module	Bit pattern in units of 16 points, indicating the modules whose fuses have blown
D9101	SD9101	SD1301		
D9102	SD9102	SD1302		
D9103	SD9103	SD1303		
D9104	SD9104	SD1304		
D9105	SD9105	SD1305		
D9106	SD9106	SD1306		
D9107	SD9107	SD1307		
D9116	SD9116	SD1400	I/O module verification error	Bit pattern, in units of 16 points, indicating the modules with verification errors.
D9117	SD9117	SD1401		
D9118	SD9118	SD1402		
D9119	SD9119	SD1403		
D9120	SD9120	SD1404		
D9121	SD9121	SD1405		
D9122	SD9122	SD1406		
D9123	SD9123	SD1407		
D9124	SD9124	SD63	Annunciator detection quantity	Annunciator detection quantity
D9125	SD1125	SD64	Annunciator detection number	Annunciator detection number
D9126	SD1126	SD65		
D9127	SD1127	SD66		
D9128	SD1128	SD67		
D9129	SD1129	SD68		
D9130	SD1130	SD69		
D9131	SD1131	SD70		
D9132	SD1132	SD71		
D9200	SD1200	—	ZNRD (LRDP for ACPU) processing results	0 : Normal end 2 : ZNRD instruction setting fault 3 : Error at relevant station 4 : Relevant station ZNRD execution disabled
D9201	SD1201	—	ZNWR (LWTP for ACPU) processing results	0 : Normal end 2 : ZNWR instruction setting fault 3 : Error at relevant station 4 : Relevant station ZNWR execution disabled

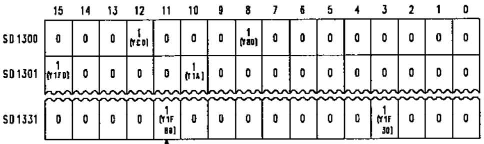
ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9202	SD1202	—	Local station link type	Stores conditions for up to numbers 1 to 16
D9203	SD1203	—		Stores conditions for up to numbers 17 to 32
D9241	SD1241	—		Stores conditions for up to numbers 33 to 48
D9242	SD1242	—		Stores conditions for up to numbers 49 to 64
D9204	SD1204	—	Link state	0 : Forward loop, during data link 1 : Reverse loop, during data link 2 : Loopback implemented in forward/reverse directions 3 : Loopback implemented only in forward direction 4 : Loopback implemented only inreverse direction 5 : Data link disabled
D9205	SD1205	—	Station implementing loopback	Station that implemented forward loopback
D9206	SD1206	—	Station implementing loopback	Station that implemented reverse loopback
D9207	SD1207	—	Link scan time	Maximum value
D9208	SD1208	—		Minimum value
D9209	SD1209	—		Present value
D9210	SD1210	—	Number of retries	Stored as cumulative value
D9211	SD1211	—	Number of times loop selected	Stored as cumulative value
D9212	SD1212	—	Local station operation state	Stores conditions for up to numbers 1 to 16
D9213	SD1213	—		Stores conditions for up to numbers 17 to 32
D9214	SD1214	—		Stores conditions for up to numbers 33 to 48
D9215	SD1215	—		Stores conditions for up to numbers 49 to 64
D9216	SD1216	—	Local station error detect state	Stores conditions for up to numbers 1 to 16
D9217	SD1217	—		Stores conditions for up to numbers 17 to 32
D9218	SD1218	—		Stores conditions for up to numbers 33 to 48
D9219	SD1219	—		Stores conditions for up to numbers 49 to 64
D9220	SD1220	—	Local station parameters non-conforming; remote I/O station I/O allocation error	Stores conditions for up to numbers 1 to 16
D9221	SD1221	—		Stores conditions for up to numbers 17 to 32
D9222	SD1222	—		Stores conditions for up to numbers 33 to 48
D9223	SD1223	—		Stores conditions for up to numbers 49 to 64

Special Register List (Continued)

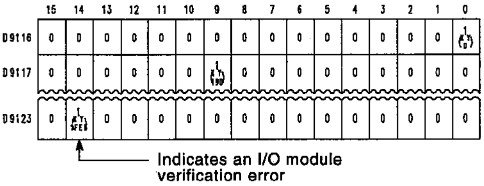
ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9224	SD1224	—	Local station and remote I/O station initial communications underway	Stores conditions for up to numbers 1 to 16
D9225	SD1225	—		Stores conditions for up to numbers 17 to 32
D9226	SD1226	—		Stores conditions for up to numbers 33 to 48
D9227	SD1227	—		Stores conditions for up to numbers 49 to 64
D9228	SD1228	—	Local station and remote I/O station error	Stores conditions for up to numbers 1 to 16
D9229	SD1229	—		Stores conditions for up to numbers 17 to 32
D9230	SD1230	—		Stores conditions for up to numbers 33 to 48
D9231	SD1231	—		Stores conditions for up to numbers 49 to 64
D9232	SD1232	—	Local station and remote I/O station loop error	Stores conditions for up to numbers 1 to 8
D9233	SD1233	—		Stores conditions for up to numbers 9 to 16
D9234	SD1234	—		Stores conditions for up to numbers 17 to 24
D9235	SD1235	—		Stores conditions for up to numbers 25 to 32
D9236	SD1236	—		Stores conditions for up to numbers 33 to 40
D9237	SD1237	—		Stores conditions for up to numbers 41 to 48
D9238	SD1238	—		Stores conditions for up to numbers 49 to 56
D9239	SD1239	—		Stores conditions for up to numbers 57 to 64

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9240	SD1240	—	Number of times communications errors detected	Stores cumulative total of receive errors
D9243	SD1243	—	Station number information for host station	Stores station number (0 to 64)
D9244	SD1244	—	Number of link device stations	Stores number of slave stations
D9245	SD1245	—	Number of times communications errors detected	Stores cumulative total of receive errors
D9248	SD1248	—	Local station operation state	Stores conditions for up to numbers 1 to 16
D9249	SD1249	—		Stores conditions for up to numbers 17 to 32
D9250	SD1250	—		Stores conditions for up to numbers 33 to 48
D9251	SD1251	—		Stores conditions for up to numbers 49 to 64
D9252	SD1252	—	Local station error conditions	Stores conditions for up to numbers 1 to 16
D9253	SD1253	—		Stores conditions for up to numbers 17 to 32
D9254	SD1254	—		Stores conditions for up to numbers 33 to 48
D9255	SD1255	—		Stores conditions for up to numbers 49 to 64

(10) Fuse blown module

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD1300	Fuse blown module	Bit pattern in units of 16 points, indicating the modules whose fuses have blown 0 : No blown fuse 1 : Blown fuse present	• The numbers of output modules whose fuses have blown are input as a bit pattern (in units of 16 points). (If the module numbers are set by parameter, the parameter-set numbers are stored.) • Also detects blown fuse condition at remote station output modules  15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 SD1300: 0 0 0 1 (YC0) 0 0 0 1 (Y80) 0 0 0 0 0 0 0 0 SD1301: 1 (Y1F0) 0 0 0 0 1 (Y1A) 0 0 0 0 0 0 0 0 0 0 SD1331: 0 0 0 0 1 (Y1FB0) 0 0 0 0 0 0 0 1 (Y1F30) 0 0 0 Indicates a blown fuse • Not cleared even if the blown fuse is replaced with a new one. This flag is cleared by error resetting operation (refer to 9.3.).	S (Error)	D9100
SD1301					D9101
SD1302					D9102
SD1303					D9103
SD1304					D9104
SD1305					D9105
SD1306					D9106
SD1307					D9107
SD1308					New
SD1309 to SD1330					New to New
SD1331					New

(11) I/O module verification

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD1400	I/O module verification error	Bit pattern, in units of 16 points, indicating the modules with verification errors. 0 : No I/O verification errors 1 : I/O verification error present	• When the power is turned on, the module numbers of the I/O modules whose information differs from the registered I/O module information are set in this register (in units of 16 points). (If the I/O numbers are set by parameter, the parameter-set numbers are stored.) • Also detects I/O module information  Indicates an I/O module verification error	S (Error)	D9116
SD1401					D9117
SD1402					D9118
SD1403					D9119
SD1404					D9120
SD1405					D9121
SD1406					D9122
SD1407					D9123
SD1408					New
SD1409 to SD1430					New to New
SD1431					New

Special Register List

(12) For redundant systems (Host system CPU information *1) for Q4AR only SD1510 to SD1599 are only valid for redundant systems. They are all set to 0 for standalone systems.

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD1500 SD1501	Basic period	Basic period tome	• Set the basic period (1 second units) use for the process control instruction using floating point data. Floating points data = | SD1501 | SD1500 |	U	New
SD1502	Process control instruction detail error code	Process control instruction detail error code	• Shows the detailed error contents for the error that occurred in the process control instruction	S (Error occurrence)	New
SD1503	Process control instruction generated error location	Process control instruction generated error location	• Shows the error process block that occurred in the process control instruction.	S (Error occurrence)	New
SD1512	Operation mode during CPU start up	Hot start switch power out time	• Shows the power out time (S) during the automatic switch from hot start to initial start in the operation mode when the CPU is started up.	S (Initial)	New
SD1590	Switch request network No.	Request origin network No.	• Stores the request origin at work No. when the SM1590 is turned on.	S (Error occurrence)	New

*1 Stores the host system CPU information.

Special Register List

(13) For redundant system (Other system CPU information *1) for Q4AR only SD1600 to SD1650 is only valid during the back up mode for redundant systems, and refresh cannot be done when in the separate mode.SD1651 to SD1699 is valid for both the backup mode and the separate mode. When a standalone system SD1600 to SD1699 are all 0.

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9[ ][ ][ ]
SD1600	Diagnosis error	Diagnosis error No.	• Stores as BIN code the error No. of the error that occurred during the other system CPU diagnosis. • Stores the latest error currently occurring.	S (Each END)	SD0
SD1601	Diagnosis error occurrence time	Diagnosis error occurrence time	• SD1600 stores the updated date and time. • Stores each of the BCD two digits. • Refer to SD1 to SD3 for the storage status. (SD1→SD1601, SD2→SD1602, SD3→SD1603)	S (Each END)	SD1 to SD3
SD1602					
SD1603					
SD1604	Error information classification	Error information classification	• Stores the error comment information/individual information classification code. • Refer to SD4 for the storage status.	S (Each END)	SD4

*1 Stores other system CPU self-diagnostic information and system information.

*2 Shows the special register (SD[ ][ ]) for the host system CPU.

Special Register List (Continue)

Number	Name	Meaning	Explanation	Set by (When Set)	For the host system SD[ ][ ][ ]*2
SD1605	Error common information	Error common information	• Stores the common information for the error code. • Refer to SD5 to SD15 for the storage status. (SD5→SD1605, SD6→SD1606, SD7→SD1607, SD8→SD1608, SD9→SD1609, SD10→SD1610, SD11→SD1611, SD12→SD1612, SD13→SD1613, SD14→SD1614, SD15→SD1615)	S (Each END)	SD5 to SD15
SD1606					
SD1607					
SD1608					
SD1609					
SD1610					
SD1611					
SD1612					
SD1613					
SD1614					
SD1615					
SD1616	Error individual information	Error individual information	• Stores the individual information for the error code • Refer to SD16 to SD26 for the storage status. (SD16→SD1616, SD17→SD1617, SD18→SD1618, SD19→SD1619, SD20→SD1620, SD21→SD1621, SD22→SD1622, SD23→SD1623, SD24→SD1624, SD25→SD1625, SD26→SD1626)	S (Each END)	SD16 to SD26
SD1617					
SD1618					
SD1619					
SD1620					
SD1621					
SD1622					
SD1623					
SD1624					
SD1625					
SD1626					
SD1650	Switch status	CPU switch status	• Stores the CPU switch status. • Refer to SD200 for the storage status. (SD1650→SD200)	S (Each END)	SD200
SD1651	LED status	CPU-LED status	• Stores the CPU's LED status. • Shows 0 when turned off, 1 when turned on, and 2 when flashing. • Refer to SD201 for the storage status. (SD1651→SD201)	S (Each END)	SD201
SD1653	CPU operation status	CPU operation status	• Stores the CPU operation status. • Refer to SD203 for the storage status. (SD1653→SD203)	S (Each END)	SD203

*2 Shows the special register (SD[ ][ ]) for the host system CPU.

Special Register List

(14) For redundant systems (Trucking) for Q4AR only
SD1700 to SD1799 is valid only for redundant systems. These are all 0 for standalone systems.

Number	Name	Meaning	Explanation	Set by (When Set)	For the host system SD [ ][ ][ ]*2
SD1700	Trucking error detection count	Trucking error detection count	• Make it the trucking error detection +1.	Error occurrence	New

*2 Shows the special register (SD[ ][ ]) for the host system CPU.

INDEX

QnACPU
PROGRAMMING MANUAL (Common Instructions)

MODEL	QNA-P(KYOUTU)-E
MODEL CODE	13JF47
IB(NA)66615-C(9902)MEE	

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE : MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100-8310 TELEX : J24532 CABLE MELCO TOKYO
NAGOYA WORKS : 1-14 , YADA-MINAMI 5 , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of International Trade and Industry for service transaction permission.