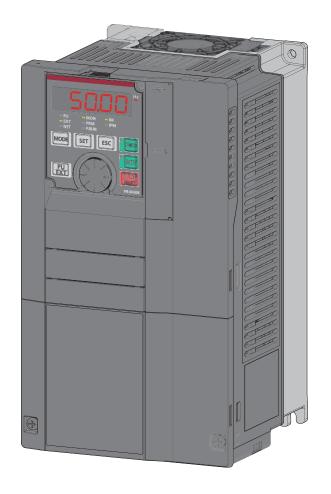




# PLC FUNCTION PROGRAMMING MANUAL



PLC Function	1
CC-Link communication	2
Sequence program	3
Error code list	4

7

# 1 PLC Function

1.1	Applicable inverter model	8
1.2	Function block diagram	8
1.3	Operation panel indication	9
1.4	PLC function specifications	10
1.5	System configuration	11
1.6	Operations of PLC function (Pr.414, Pr.415, Pr.498, Pr.1150 to Pr.1199)	
1.7	Prior to creating sequence programs	
1.7.		
1.7.		
1.7		
1.7.		
1.7.		
1.8	Device map	
	•	
1.8.	·	
1.8.		
1.8.		
1.8.	•	
1.8.		
1.9	Special registers to monitor and control inverter status	30
1.9	1 Always readable data	30
1.9	2 Data read by controlling read command (from OFF to ON)	34
1.9.	3 Data writing method by controlling write command (from OFF to ON)	36
1.9.		
1.9		
1.9		
1.9.	,	
1.9.	8 Monitor setting selection (SD1215 to SD1218)	43
1.10	Read/Write method of inverter parameters	44
1.10	0.1 Reading inverter parameters	44
1.10	0.2 Writing inverter parameters	45
1.11	User area reading/writing	46
1.12	Analog I/O function	
	2.1 Analog input	
	2.2 Analog output	
	Pulse train input function	

1.14 P	ID control	49
1.15 C	learing the flash memory of the PLC function	50
1.16 C	onstant scan	51
2 CC-	Link communication	53
2.1 S	ystem configuration	54
2.1.1	System configuration example	54
2.1.2	Function block diagram	55
2.2 C	C-Link parameter	56
2.2.1	CC-Link extended setting (Pr.544)	56
2.3 C	C-Link I/O specifications	57
2.3.1	I/O signals when one station in the CC-Link Ver.1 is occupied (Pr.544 = "100")	
2.3.2	I/O signals when the double setting is set in CC-Link Ver.2 (Pr.544 = "112")	
2.3.3	I/O signals when the quadruple setting is set in CC-Link Ver.2 (Pr. 544 = "114")	61
2.3.4	I/O signals when the octuple setting is set in CC-Link Ver.2 (Pr. 544 = "118")	62
2.4 B	uffer memory	64
2.4.1	Remote output signals (master station to inverter (FR-A8NC))	64
2.4.2	Remote input signals Pr.544 = "100" (inverter (FR-A8NC) to master station)	
2.4.3	Remote registers Pr.544 = "100" (master station to inverter (FR-A8NC))	66
2.4.4	Remote registers Pr.544 = "100" (inverter (FR-A8NC) to master station)	67
3 Seq	uence program	69
3.1 C	verview	70
3.1.1	Overview of operation	70
3.2 R	UN/STOP operation	72
3.3 P	rogram configuration	72
3.4 P	rogramming language	73
3.4.1	Relay symbolic language (ladder mode)	
3.4.2	Logic symbolic language (list mode)	
3.4.3	Function block (FB)	
3.5 C	peration processing method of the PLC function	76
3.6 I/	O processing method	77
3.6.1	Refresh method	77
3.6.2	Response delay in refresh mode	

3.7	Scan time	. 79
3.8	Values that can be used in sequence programs	. 80
3.8	.1 BIN (binary)	81
3.8	.2 HEX (hexadecimal)	82
3.9	Explanation of devices	. 83
3.9	.1 Device list	83
3.9	.2 I/O X and Y	84
3.9	.3 Internal relay M	85
3.9	.4 Timer T	86
3.9	.5 100 ms timer, 10 ms timer, and 100 ms retentive timer	86
3.9	.6 Processing and accuracy of timers	87
3.10	Counter C	. 89
3.1	0.1 Count process in refresh mode	90
3.1	0.2 Maximum counting speed of counter	90
3.11	Data register D	. 91
3.12	Special relays and special registers	. 92
3.13	Function list	. 93
3.14	RUN/STOP method of PLC function from an external source (remote RUN/	
	STOP)	. 94
3.15	Watchdog timer (watchdog error supervision timer)	. 96
3.16		. 97
3.1	Self-diagnostic function	97
3.1	Self-diagnostic function  6.1 Operation mode when there is an operation error	97 <b>. 98</b>
3.17	Self-diagnostic function  6.1 Operation mode when there is an operation error	97 . <b>98</b> . <b>99</b>
3.17 3.18	Self-diagnostic function  6.1 Operation mode when there is an operation error	97 . 98 . 99 100
3.17 3.18 3.19	Self-diagnostic function  6.1 Operation mode when there is an operation error.  Registering file password  Output (Y) status settings when STOP status → RUN status  Structure of instructions  Bit device processing method	97 . 98 . 99 100 101
3.17 3.18 3.19 3.20	Self-diagnostic function  6.1 Operation mode when there is an operation error.  Registering file password  Output (Y) status settings when STOP status → RUN status  Structure of instructions  Bit device processing method	97 . 98 . 99 100 101
3.17 3.18 3.19 3.20	Self-diagnostic function  6.1 Operation mode when there is an operation error.  Registering file password  Output (Y) status settings when STOP status → RUN status.  Structure of instructions  Bit device processing method.	97 . 98 . 99 100 101
3.17 3.18 3.19 3.20 3.2 3.2	Self-diagnostic function  6.1 Operation mode when there is an operation error	97 . 98 . 99 100 101
3.17 3.18 3.19 3.20 3.2 3.21	Self-diagnostic function  6.1 Operation mode when there is an operation error.  Registering file password  Output (Y) status settings when STOP status → RUN status.  Structure of instructions.  Bit device processing method.  0.1 1-bit processing.  0.2 Digit specification processing.  Handling of numerical values.  Operation error.	97 . 98 . 99 100 101 101 101 104 105
3.17 3.18 3.19 3.20 3.2 3.21 3.22	Self-diagnostic function  6.1 Operation mode when there is an operation error  Registering file password  Output (Y) status settings when STOP status → RUN status  Structure of instructions  Bit device processing method  0.1 1-bit processing  0.2 Digit specification processing  Handling of numerical values  Operation error  Sequence instructions list	97 . 98 . 99 100 101 101 104 105 106
3.17 3.18 3.19 3.20 3.2 3.21 3.22 3.23 3.23	Self-diagnostic function  6.1 Operation mode when there is an operation error  Registering file password  Output (Y) status settings when STOP status → RUN status  Structure of instructions  Bit device processing method  0.1 1-bit processing  0.2 Digit specification processing  Handling of numerical values  Operation error  Sequence instructions list	97 . 98 . 99 100 101 101 104 105 106
3.17 3.18 3.19 3.20 3.2 3.21 3.22 3.23 3.2 3.23	Self-diagnostic function  6.1 Operation mode when there is an operation error	97 . 98 . 99 100 101 101 104 105 106 106 108
3.17 3.18 3.19 3.20 3.2 3.21 3.22 3.23 3.2 3.2 3.2	Self-diagnostic function  6.1 Operation mode when there is an operation error  Registering file password  Output (Y) status settings when STOP status → RUN status  Structure of instructions  Bit device processing method  0.1 1-bit processing  0.2 Digit specification processing  Handling of numerical values  Operation error  Sequence instructions list  3.1 How to view the instruction list table  3.2 Sequence instructions list	97 . 98 . 99 100 101 101 104 105 106 108 110

3.24	Ho	ow to view instructions	116
3.25	Se	quence instructions	117
3.	25.1	Contact instruction: operation start, series connection, parallel connection LD, LDI, AND, ANI, OR, ORI	117
3.	25.2	Association instruction: ladder block series connection, parallel connection ANB, ORB	119
3.	25.3	Association instruction: operation results, push, read, pop MPS, MRD, MPP	121
3.	25.4	Output instruction: bit device, timer, counter OUT	124
3.	25.5	Output instruction: device set, reset SET, RST	126
3.	25.6	Output instruction: rising, falling differential output PLS, PLF	128
3.	25.7	Shift instruction: bit device shift SFT, SFTP	130
3.	25.8	Master control instruction: master control set, reset MC, MCR	132
3.	25.9	Termination instruction: termination of main program FEND	135
3.	25.10	Termination instruction: sequence program termination END	135
3.	25.11	Other instructions: non-processing NOP, NOPLF, PAGE n	136
3.26	Ва	sic instruction (16-bit)	139
3.	26.1	Comparison operation instruction	139
3.	26.2	Comparison operation instruction:  16-bit data comparison =, <>, >, <=, <, >=	
3.	26.3	Arithmetic operation instruction	
3.	26.4	Arithmetic operation instruction: BIN 16-bit addition/subtraction +, +P, -, -P	
3.	26.5	Arithmetic operation instruction: BIN 16-bit multiplication/division *, *P, /, /P	
3.	26.6	Arithmetic operation instruction: BIN 16-bit data increment/decrement INC, INCP, DEC, DECP	
3.	26.7	Data transfer instruction	
3.	26.8	Data transfer instruction:  16-bit data transfer MOV, MOVP	150
3.	26.9	Data conversion instruction: 2's complements of BIN 16-bit data NEG, NEGP	151
3.	26.10	Data conversion instructions: BIN 16-bit data BCD conversions BCD, BCDP	153
3.	26.11	Data conversion instructions: BIN 16-bit data BIN conversions BIN, BINP	154
3.27	Ва	sic instruction (32-bit)	155
3.	27.1	Comparison operation instruction	155
3.	27.2	Comparison operation instruction: 32-bit data comparison D=, D<>, D>, D<=, D<, D>=	156
3.	27.3	Arithmetic operation instruction	
3.		Arithmetic operation instruction: BIN 32-bit addition/subtraction D+, D+P, D-, D-P	
3.	27.5	Arithmetic operation instruction: BIN 32-bit multiplication/division D*, D*P, D/, D/P	161
3.	27.6	Arithmetic operation instruction:	

	BIN 32-bit data increment/decrement DINC, DINCP, DDEC, DDECP	164
3.27.7	Data transfer instruction	165
3.27.8	Data transfer instruction: 32-bit data transfer DMOV, DMOVP	165
3.27.9	Data conversion instruction: 2's complements of BIN 32-bit data DNEG, DNEGP	167
3.27.10	Data conversion instructions: BIN 32-bit data BCD conversions DBCD, DBCDP	168
3.27.11	Data conversion instructions: BIN 32-bit data BIN conversions DBIN, DBINP	169
3.28 Ap	oplication instructions (16-bit)	170
3.28.1	Logical operation instructions	170
3.28.2	Logical operation instruction: BIN 16-bit logical AND WAND, WANDP	171
3.28.3	Logical operation instruction: BIN 16-bit logical OR WOR, WORP	174
3.28.4	Logical operation instruction: BIN 16-bit data exclusive OR WXOR, WXORP	176
3.28.5	Logical operation instruction: BIN 16-bit data exclusive NOR WXNR, WXNRP	178
3.29 Ap	oplication instructions (32-bit)	180
3.29.1	Logical operation instructions	180
3.29.2	Logical operation instruction: BIN 32-bit logical AND DAND, DANDP	181
3.29.3	Logical operation instruction: BIN 32-bit logical OR DOR, DORP	184
3.29.4	Logical operation instruction: BIN 32-bit data exclusive OR DXOR, DXORP	187
3.29.5	Logical operation instruction: BIN 32-bit data exclusive NOR DXNR, DXNRP	190
3.30 Di	splay instruction	192
3.30.1	Character string data transfer \$MOV, \$MOVP	192
3.30.2	Character string output G. PRR, GP.PRR, UMSG	193
Erro	r code list	197
4.1 Ho	ow to read error codes	198
PPEN	IDIX	201
Appendi	ix 1 Instruction processing time	202

### **♦**About the related manuals

The manuals related to A800 PLC function are shown below. The download of the latest manuals is free at the Mitsubishi Electric FA Global Website.

#### Related manuals

Manual name	Manual number
FR-A800 Instruction Manual (Startup)	IB-0600493
FR-A800 Instruction Manual (Detailed)	IB-0600503ENG
FR Configurator2 Instruction Manual	IB-0600516ENG
GX Works2 Version 1 Operating Manual (Common)	SH-080779ENG

# 1 PLC Function

This instruction manual explains the functions and devices required for programming.

1.1	Applicable inverter model	8
1.2	Function block diagram	8
1.3	Operation panel indication	9
1.4	PLC function specifications	10
1.5	System configuration	11
1.6	Operations of PLC function (Pr.414, Pr.415, Pr.498, Pr.1150 to Pr.1199)	12
1.7	Prior to creating sequence programs	13
1.8	Device map	17
1.9	Special registers to monitor and control inverter status	30
1.10	Read/Write method of inverter parameters	44
1.11	User area reading/writing	46
1.12	Analog I/O function	47
	Pulse train input function	48
	PID control	49
	Clearing the flash memory of the PLC function	50
	Constant scan	51

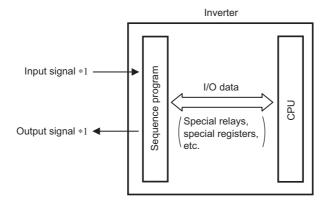
### **1.1** Applicable inverter model

This manual explains the PLC function of the FR-A800 series.

### 1.2 Function block diagram

The following function blocks explain I/O information flow to and from the inverter in the PLC function.

- I/O information can be read or written by accessing the inverter with special relays, special registers, etc. using predetermined methods.
- It is possible to operate the inverter and read and write parameters using input information from control input terminals according to created sequence programs (saved in the inverter).
  - Output signals are also be output from control output terminals, not only as status signals of the inverter but as signals to turn the pilot lamp ON and OFF, interlock signals, and other control signals set by users as desired.



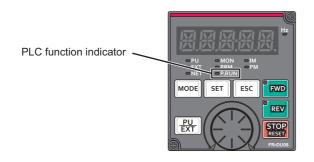
\*1 Set "9999" in any of Pr.178 to Pr.189 (input terminal function assignment) and Pr.190 to Pr.196 (output terminal function selection) to use terminals as general-purpose I/O terminals. For details on Pr.178 to Pr.189 and Pr.190 to Pr.196, refer to the Instruction Manual of the inverter.

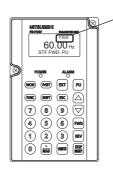
### Ц

## **1.3** Operation panel indication

When a sequence program is running, the following indicators are displayed on the operation panel (FR-DU08) and parameter unit (FR-PU07):

FR-DU08 FR-PU07





P.RUN is displayed on the LCD when PLC function is operating.

#### P.RUN state

LED (LCD) action	Status
Off (normal indication)	Sequence program stopped
On (inverted indication)	Sequence program running
Flickering	Sequence error occurring



• While the offline auto tuning is being activated by FR-PU07, P.RUN is not displayed even during PLC function operation.

# **1.4** PLC function specifications

The following table shows the program capacity and devices of the PLC function.

Item			A800 PLC function specifications
Control method			Repeated operation (by stored program)
I/O control mode			Refresh
Programming language			Relay symbolic language (ladder) Logic symbolic language (list) Function block
No. of	Sequence inst	ructions	25
No. of instructions	Basic instruction	ons	84
instructions	Application ins	tructions	37
Processing s	peed		Sequence instructions 1.9 μs to 12 μs/step*1
Number of I/0	O points		128 (input: 64 points, output: 64 points) 19 points built-in (input: 12 points, output: 7 points)*2 FR-A8AX (input: 16 points) FR-A8AY (output: 6 points) FR-A8AR (output: 3 points)
Number of ar	nalog I/O points		19 points built-in (input: 12 points, output: 7 points) FR-A8AX (input: 16 points) FR-A8AY (output: 6 points) FR-A8AR (output: 3 points)
Watchdog tim	ner		10 to 2000 (ms)
Memory capa	acity		6k bytes for sequence programs and parameters.
Program capa	acity		6K steps (0 to 6144 steps can be set)
	Internal relay (	M)	128 (M0 to M127)
	Latch relay (L)	)	Not used (Can be set with parameters but will not latch)*3
		Number of points	16
	Timer (T)	Specifications	100 ms timer: 0.1 to 3276.7 s (T0 to T15) can be set 10 ms timer: 0.01 to 327.67 s can be set 100 ms retentive timer: 0.1 to 3276.7 s can be set
Device		Number of points	16
	Counter (C)	Specifications	Normal counter: Setting range 1 to 32767 (C0 to C15) Interrupt program counter: Not used
	Data register (	D)	256 (D0 to D255)
	Special relay (SM)		2048 (SM0 to SM2047) with limited functions
	Special register (SD)		2048 (SD0 to SD2047) with limited functions

- \*1 The scan time is approximately 40 ms for 1K steps as inverter control is also performed in actual operations.
- \*2 The signals same as the ones assigned to the inverter I/O terminals are used. One point is always required for a sequence start (RUN/STOP).
- \*3 There is no device latch function for power failures.

Use the Pr.1150 to Pr.1199 PLC function user parameters 1 to 50 (D206 to D255) to store device values in the EEPROM. (Refer to page 46.)



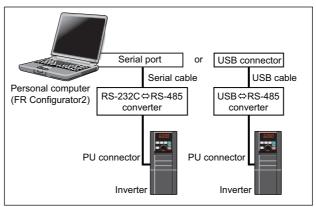
• There is no buffer memory.

## 1.5 System configuration

To write to the PLC program, use the FR Configurator2 (Developer) on a personal computer that is connected to the inverter via RS-485 communication.

The following figure shows the system configuration for use of the PLC function.

<System configuration example>



Commercially available products (as of February 2012)

• RS-232C ⇔ RS-485 conversion cable

Model	Manufacturer
Interface embedded cable dedicated for inverter DINV-CABV •1	Diatrend Corp.

- \*1 The conversion cable cannot connect multiple inverters. (The computer and inverted are connected in a 1:1 pair.) This product is a RS-232C  $\Leftrightarrow$  RS-485 conversion cable that has a built-in converter. No additional cable or connector is required. For the product details, contact the manufacturer.
- USB ⇔ RS-485 conversion cable

Model	Manufacturer
DINV-U4	Diatrend Corp.



• FR Configurator2 (Developer) is a programming tool for designing and debugging of the sequence programs on the Windows personal computer. To use the PLC function of the inverter, write sequence programs to the inverter using FR Configurator2 (Developer). (For the details, refer to the Instruction Manual of FR Configurator2.)

### NOTE

- For details on wiring, refer to the Instruction Manual of the inverter.
- For the specifications of the FR Configurator2 (Developer) and personal computers on which FR Configurator2 (Developer) runs, refer to the Instruction Manual of FR Configurator2.
- Programming tools other than FR Configurator2 (Developer) cannot be used. (Tools such as GX Developer and GX Works cannot be used.)

# **1.6** Operations of PLC function (Pr.414, Pr.415, Pr.498, Pr.1150 to Pr.1199)

I/O information can be read or written by accessing the inverter with special relays, special registers, etc. using predetermined methods.

It is possible to operate the inverter and read and write parameters using input information from control input terminals according to created sequence programs (saved in the inverter).

Output signals are also be output from control output terminals, not only as status signals of the inverter but as signals to turn the pilot lamp ON and OFF, interlock signals, and other control signals set by users as desired.

Parameter number	Name	Initial value	Setting range	Refer to page
414	PLC function operation selection	0	0 to 2	14
415	Inverter operation lock mode setting	0	0, 1	14
498	PLC function flash memory clear	0	0 to 9999	50
1150 to 1199	PLC function user parameters 1 to 50	0	0 to 65535	46



<sup>•</sup> The FR-PU07 cannot read or write parameters Pr.1000 or later.

### 1

# 1.7 Prior to creating sequence programs

# 1.7.1 Precautions for creating sequence programs

### POINT)

- Writing and access to other stations are not possible while a sequence program is running.
   Also, reading and writing the program from and to other stations cannot be performed.
- To avoid accidentally erasing the created sequence programs, use FR Configurator2 (Developer) to regularly save the data in a personal computer, etc.

If a sequence program contains instructions (refer to page 106) or devices (refer to page 10) that cannot be used with the PLC function, an instruction code error occurs at the execution of that instruction.

Error code SD0 = 4000
Error information SD5 to SD26
Error flag SM0: ON



• For the error codes, refer to page 33.

# 1.7.2 The main functions of FR Configurator2 (Developer)

- · Reading and writing parameter and sequence programs
- · Ladder monitor
- · Device batch monitor
- · Present value change
- Remote RUN/STOP

### NOTE

Present values can be changed using FR Configurator2 (Developer) ([Debug] -> [Present value change]). Even if changes
are made to the devices corresponding to the control terminal signals (STF, STR, etc.), such changes are not applied to the
inverter operation. (Device present values of the sequence programs are changed.)

## 1.7.3 Sequence program execution

Parameter number	Name	Initial value	Setting range	Description	
			0	PLC functi	on disabled
414 *1	PLC function operation selection	0	1	PLC function	The SQ signal is enabled by input from a command source (external input terminal/communication).
			2	enabled	The SQ signal is enabled by input from an external input terminal
			The start signal of the inverter is enabled regardless of execution instruction of the sequence program.		
415	Inverter operation lock mode setting	0	1	The start signal of the inverter is enabled only wher the sequence program is in RUN state. (The SQ sign is ON.)  When the sequence program is in STOP state (the Signal is OFF), the inverter will not start even if the start signal STF or STR is turned ON. (If the state is changed from RUN to STOP during operation, the inverter decelerates and stops.)	

<sup>\*1</sup> Reset the inverter after configuring settings to enable the PLC function.

### **♦PLC** function operation selection (Pr.414)

- To enable the PLC function, set "1" or "2" in **Pr.414 PLC function operation**. When **Pr.414** = "1", the command source for the sequence start (SQ) signal is determined by **Pr.338 Communication operation command source**. When "2" is set in **Pr.414**, SQ signal from the external input terminal is valid regardless of the setting of the **Pr.338**.
- Switch the execution key (RUN/STOP) of the sequence program by turning the SQ signal ON/OFF. The sequence program can be executed by turning the SQ signal ON. To input the SQ signal, set "50" in any of **Pr.178 to Pr.189 (input terminal function selection)** to assign the function to a terminal.
- Remote RUN/STOP of the PLC function can be executed in any of the following methods:
  - · Using the PLC function parameters (contact)
  - Using FR Configurator2 (Developer)
  - Via CC-Link communication (refer to page 56.)

### NOTE

- Turn OFF (STOP) the SQ signal when creating a sequence program.
- The RUN state output (Y) is internally stored and all outputs (Y) are turned off by turning the SQ signal OFF (STOP) after sequence program execution (SQ signal ON).

The other devices retain the data prior to STOP. To clear the remaining device data, power OFF or reset (RES signal ON  $(0.1 \text{ s}) \rightarrow \text{OFF}$ ) the inverter.

### ♦Inverter operation lock mode setting (Pr.415)

• When "1" is set in **Pr.415 Inverter operation lock mode setting**, the inverter can be operated only when the sequence program is set to the "RUN" status. By changing the PLC program status from RUN to STOP during inverter operation, the motor decelerates to stop.



• To run the inverter without using the PLC function, set **Pr.415** = "0" (initial value) (the start signal of the inverter is enabled).

- This parameter cannot be written during inverter operation, regardless of the Pr.77 setting.
- During automatic operation using SD1148 (or SM1200 to SM1211) in the sequence program, the inverter stops operation if the sequence program is in the STOP state while **Pr.415** = "1". However, when **Pr.415** = "0", the device data is retained and the operating status does not change (the inverter continues operation) even if the sequence program is in the STOP state.
- Pr.415 setting is also enabled for the start signals of the operation panel and the FR-PU07.

## 1.7.4 Setting the communication parameter

### POINT)

- While the inverter communication parameters (Pr.118 to Pr.120, and Pr.122) and the [Detailed setting] of the [System setting] window of FR Configurator2 are set to initial values, communication with FR Configurator2 (Developer) cannot be established.
- While the inverter communication parameters (**Pr.118 to Pr.120**, and **Pr.122**) are initial values, communication with FR Configurator2 (Developer) cannot be established. Set the inverter communication parameters as shown below.

Parameter number	Parameter name	Initial value	Setting value
118	PU communication speed	192 (19200 bps)	1152 (115200 bps)
119	PU communication stop bit length / data length	1 (data length: 8 bits, stop bit: 2 bits)	0 (data length: 8 bits, stop bit: 1 bit)
120	PU communication parity check	2 (with even parity check)	1 (with odd parity check)
122	PU communication check time interval	9999 (without communication cl	neck)

• To use the Developer and other functions of FR Configurator2, open the "System setting" window in FR Configurator2, and set the [Detailed setting] as follows.

Item	Initial value	Setting value
Communication speed	19200(bps)	115200(bps)
Data length	8(bit)	
Stop bit	2(bit)	1(bit)
Parity	Even	Odd
Delimiter	CR	

- Use the operation panel (FR-DU08) or parameter unit (FR-PU07) to change the settings of the inverter parameters.

  The operation unit and FR Configurator2 (personal computer) cannot be connected to the PU connector simultaneously.
- For details on each inverter communication parameter, refer to the Instruction Manual (Detailed) of the inverter. For the setting method of the [Detailed setting] of the [System setting] window of FR Configurator2, refer to the Instruction Manual of FR Configurator2.
- When parameter clear or all clear is performed, the communication parameter setting values are also cleared, and communication with FR Configurator2 (Developer) is therefore no longer possible.

## 1.7.5 Writing sequences programs

### POINT)

 Sequence programs can be written in any operation mode (External operation mode, PU operation mode and Network operation mode). For the operation modes, refer to the Instruction Manual (Detailed) of the inverter.

Check the following points when rewriting the PLC function parameters and sequence programs using FR Configurator2 (Developer):

- The sequence program execution status is in the STOP (SQ signal OFF) (refer to page 14).
- · The inverter is stopped.
- The communication specifications setting parameters (Pr.117 to Pr.124) are correctly set. Communication with FR Configurator2 (Developer) is not possible if these parameter setting values are incorrect.
- The PLC series setting and sequence program capacity are correctly set with FR Configurator2 (Developer) parameters. (Refer to page 11.)

Refer to the FR Configurator2 manual and write sequence programs.

- A sequence program cannot be written with its steps specified. The sequence program does not run if written in this way. (The program outside the specified range is initialized.)
- To read the PLC function parameters and sequence programs from the inverter, such information must be written to the inverter using FR Configurator2 (Developer) in advance. Always write the PLC function parameters and sequence program at least once as the inverter does not have valid data.
- There is a limitation on the number of times writing can be done (approximately 100,000 times) as the PLC function parameters and sequence program are written to the Flash ROM.
- · A program with more than 6K steps cannot be written.

# 1.8 Device map

# 1.8.1 I/O device map

	Device number	Name	Remarks	Device number	Name	Remarks
	X00	Terminal STF		Y00	Terminal RUN	
	X01	Terminal STR		Y01	Terminal SU	
	X02	Terminal RH		Y02	Terminal OL	
	X03	Terminal RM		Y03	Terminal IPF	External terminal
	X04	Terminal RL		Y04	Terminal FU	
Ħ	X05	Terminal JOG	External terminal	Y05	Terminal ABC1	
External input/output	X06	Terminal RT	External terminal	Y06	Terminal ABC2	
),Ind	X07	Terminal AU		Y07		
al in	X08	Terminal CS		Y08		
erna	X09	Terminal MRS		Y09		
Ext	X0A	Terminal STOP		Y0A		
	X0B	Terminal RES		Y0B	Empty (for temporary retention)	
	X0C	Vacant (for temporary retention)		Y0C		
	X0D			Y0D		
	X0E			Y0E		
	X0F			Y0F	1	
	X10	Terminal X0		Y10	Terminal DO0	
	X11	Terminal X1		Y11	Terminal DO1	
	X12	Terminal X2		Y12	Terminal DO2	
	X13	Terminal X3		Y13	Terminal DO3	Digital output FR-A8AY
	X14	Terminal X4		Y14	Terminal DO4	I IN-AOAT
	X15	Terminal X5		Y15	Terminal DO5	
0	X16	Terminal X6		Y16	Terminal DO6	
Plug-in option I/O	X17	Terminal X7	16 bits	Y17	Terminal RA1	
opti	X18	Terminal X8	digital input FR-A8AX	Y18	Terminal RA2	Relay output FR-A8AR
g-in	X19	Terminal X9		Y19	Terminal RA3	I IN-AOAIN
Plu	X1A	Terminal X10		Y1A		1
	X1B	Terminal X11		Y1B		
	X1C	Terminal X12		Y1C	Frank (for to see	4
	X1D	Terminal X13		Y1D	Empty (for temporary re	tention)
	X1E	Terminal X14		Y1E	-	
	X1F	Terminal X15		Y1F	-	

	Device number	Name	Remarks	Device number	Name	Remarks
	X20	Operation mode setting read complete	SD1140	Y20	Operation mode setting read command	SD1140
	X21	Set frequency read complete (RAM)	SD1141	Y21	Set frequency read command (RAM)	SD1141
	X22	Set frequency read complete (EEPROM)	SD1142	Y22	Set frequency read command (EEPROM)	SD1142
	X23	Operation mode setting write complete	SD1143	Y23	Operation mode setting write command	SD1143
	X24	Set frequency write complete (RAM)	SD1144	Y24	Set frequency write command (RAM)	SD1144
	X25	Set frequency write complete (EEPROM)	SD1145	Y25	Set frequency write command (EEPROM)	SD1145
	X26	Fault record batch clear complete	SD1146	Y26	Fault record batch clear command	SD1146
System I/O	X27	Parameter clear complete	SD1147	Y27	Parameter clear command	SD1147
Syste	X28	Parameter read complete (RAM)	SD1241, SD1242,	Y28	Parameter read request (RAM)	SD1241, SD1242,
	X29	Parameter write complete (RAM)	SD1234 SD1243, SD1244, SD1235	Y29	Parameter write request (RAM)	SD1234
	X2A	Parameter read complete (EEPROM)		Y2A	Parameter read request (EEPROM)	SD1243, SD1244,
	X2B	Parameter write complete (EEPROM)		Y2B	Parameter write request (EEPROM)	SD1235
	X2C	User parameter read complete (RAM)		Y2C	User parameter read (RAM)	
	X2D	User parameter write complete (RAM)	D206 to D255	Y2D	User parameter write (RAM)	D206 to D255
	X2E	User parameter read complete (EEPROM)	(Pr.1150 to Pr.1199)	Y2E	User parameter read (EEPROM)	(Pr.1150 to Pr.1199)
	X2F	User parameter write complete (EEPROM)		Y2F	User parameter write (EEPROM)	
	X30	RY0		Y30	RX0	
	X31	RY1		Y31	RX1	
	X32	RY2		Y32	RX2	
	X33	RY3		Y33	RX3	
	X34	RY4		Y34	RX4	
	X35	RY5		Y35	RX5	
)/( e	X36	RY6		Y36	RX6	
mot	X37	RY7		Y37	RX7	
Re	X38	RY8	FR-A8NC	Y38	RX8	FR-A8NC
CC-Link I/O Remote I/O	X39	RY9		Y39	RX9	
-Lin	ХЗА	RYA		Y3A	RXA	
S	ХЗВ	RYB		Y3B	RXB	
	X3C	RYC		Y3C	RXC	
	X3D	RYD		Y3D	RXD	
	X3E RYE	RYE		Y3E	RXE	
	X3F	RYF		Y3F	RXF	

# 1.8.2 Internal relay (M) device map

Device numb	er	Description
M0 to M127	Available for users.	

# 1.8.3 Data register (D) device map

Device number	Description	
D0 to D205	Available for users.	-
D206 to D255	Pr.1150 to Pr.1199 Parameters for user setting. Available for users.	46

## 1.8.4 Special relay

Special relay is an internal relay whose application is already determined in the PLC function. Therefore, do not turn ON/OFF the special relay in the program.

Device number	Name	Description	
SM0	Diagnostic error	Turned ON when a diagnostic error has been detected. ON status is retained even after the condition becomes normal.	
SM1	Self-diagnostic error	Turned ON when an error is detected by self diagnosis. ON status is retained even after the condition becomes normal.	
SM5	Common error information	When SM5 is turned ON, common error information (SD5 to SD15) is stored.	
SM16	Individual error information	When SM16 is turned ON, individual error information (SD16 to SD26) is stored.	
SM56	Operation error flag	Turned ON when an operation error is detected during instruction execution. ON status is retained even after the condition becomes normal.	
SM210	Clock data setting request	At OFF to ON of SM210, the clock data stored in SD210 to SD213 are written to the inverter.	
SM211	Clock data error	Turned ON when the clock data (SD210 to SD213) has an error or is not a BCD value.	
SM213	Clock data read request	Clock data is read and stored as BCD value in SD210 to SD213 when this relay is turned ON.  The process is not executed when the relay is OFF.	
SM400	Always ON	SM400 and SM401 are respectively turned ON and OFF regardless of the	
SM401	Always OFF	STOP and RUN statuses.	
SM402	ON only for one scan after RUN	SM402 and SM403 change depending on the STOP and RUN statuses. In the cases other than STOP:	
SM403	OFF only for one scan after RUN	SM402 is ON only for one scan. SM403 is OFF only for one scan.	
SM1200	Inverter operating status control flag (STF)	Flag used for controlling the inverter terminal STF.	
SM1201	Inverter operating status control flag (STR)	Flag used for controlling the inverter terminal STR.	
SM1202	Inverter operating status control flag (RH)	Flag used for controlling the inverter terminal RH.	
SM1203	Inverter operating status control flag (RM)	Flag used for controlling the inverter terminal RM.	
SM1204	Inverter operating status control flag (RL)	Flag used for controlling the inverter terminal RL.	

Device number	Name	Description
SM1205	Inverter operating status control flag (JOG)	Flag used for controlling the inverter terminal JOG.
SM1206	Inverter operating status control flag (RT)	Flag used for controlling the inverter terminal RT.
SM1207	Inverter operating status control flag (AU)	Flag used for controlling the inverter terminal AU.
SM1208	Inverter operating status control flag (CS)	Flag used for controlling the inverter terminal CS.
SM1209	Inverter operating status control flag (MRS)	Flag used for controlling the inverter terminal MRS.
SM1210	Inverter operating status control flag (STOP)	Flag used for controlling the inverter terminal STOP.
SM1211	Inverter operating status control flag (RES)	Flag used for controlling the inverter terminal RES.
SM1216	Inverter status (RUN)	Inverter running
SM1217	Inverter status (FWD)	Forward rotation
SM1218	Inverter status (REV)	Reverse rotation
SM1219	Inverter status (SU)	Up to frequency
SM1220	Inverter status (OL)	Overload alarm
SM1221	Inverter status (IPF)	Instantaneous power failure/undervoltage
SM1222	Inverter status (FU)	Output frequency detection
SM1223	Inverter status (ALM)	Fault output
SM1224	Inverter status (LF)	Warning output
SM1225	Inverter status (DO0)	Stores the operating status of an output terminal function assigned by <b>Pr.313</b> .*1
SM1226	Inverter statusDO1)	Stores the operating status of an output terminal function assigned by <b>Pr.314</b> .*1
SM1227	Inverter status (DO2)	Stores the operating status of an output terminal function assigned by <b>Pr.315</b> .*1
SM1228	Inverter status (DO3)	Stores the operating status of an output terminal function assigned by <b>Pr.316</b> .*1
SM1229	Inverter status (DO4)	Stores the operating status of an output terminal function assigned by <b>Pr.317</b> .*1
SM1230	Inverter status (DO5)	Stores the operating status of an output terminal function assigned by <b>Pr.318</b> .*!
SM1231	Inverter status (DO6)	Stores the operating status of an output terminal function assigned by <b>Pr.319</b> .*!
SM1232	Inverter status (RA1)	Stores the operating status of an output terminal function assigned by <b>Pr.320</b> .*1
SM1233	Inverter status (RA2)	Stores the operating status of an output terminal function assigned by <b>Pr.321</b> .*!
SM1234	Inverter status (RA3)	Stores the operating status of an output terminal function assigned by <b>Pr.322</b> .*!
SM1235	Safety monitor (terminal So)	The safety monitor (terminal So) status is stored. This is turned ON when no internal safety circuit failure has occurred.
SM1255	Inverter operating status control selection	Select the inverter operating status of SM1200 to SM1211 and SD1148.  OFF: Special relay selection  ON: Special register selection
SM1256	2-word parameter access notice	Turned ON when parameter read for the inverter is executed using X28 to X2B, Y28 to Y2B, and the object was 2-word parameter.  OFF: Read is completed as 1-word parameter.  ON: Read is completed as 2-word parameter.

Device number	Name	Description
SM1257	2-word parameter access setting	Set the object as 2 word parameter when inverter parameter write is executed using X28 to X2B, Y28 to Y2B.  OFF: Carries out writing as 1-word parameter.  ON: Carries out writing as 2-word parameter.
SM1258	Clock data setting completion flag	Turned ON when the clock data is set using an external peripheral device, parameter, or SM210. This is turned OFF when SM213 is turned ON.

<sup>\*1</sup> Even if the FR-A8AY or FR-A8AR is not installed, Pr.313 to Pr.322 are accessible during PLC function operation, and the operating status of the output terminal (virtual output terminal) function is stored in each device.

## 1.8.5 Special register

Special register is a data register in which the application is already determined in the PLC function. Therefore, do not write data into the special register on the program.

[	Device number	Name			Description Pag
	SD0	Self-diagnostic error	Error cod	de is sto	red when an error is detected by self diagnosis. 33
			stored in	the con	ory codes that show the type of error information nmon information (SD5 to SD15) and the ation (SD16 to SD26).
					al error information Common error information
			The follo	wing co	des are stored in the common information
			Value		Code
			0	Not u	sed
	SD4	Error information category	2	File n	ame/drive name
			3	Time	(setting value)
			4	Progr	am error location
			The follo	•	des are stored in the individual information
ter			Value		Code
egis			0	Not u	sed
ial			3	Time	(actual measurement value)
Special register			5	Parar	neter number
					on information category code is 2, the file name e stored in SD5 to SD11.
				Numbe	er Description
				SD5	Drive name.
				SD6	
				SD7	File name
	SD5 to SD15	Common error information		SD8	(ASCII code: 8 characters)
				SD9	
				SD10	Extension 2EH(.)
				SD11	(ASCII code: 3 characters)
				SD12 t SD15	Empty
			The step SD15.	numbe	rs where errors occurred are stored in SD14 and
					on information category code is 3, the time stored in SD5 and SD6.
	SD5 to SD15		Nur	nber	Description
		Common error information	SD5	5 Т	ime: in 1 μs increments (0 to 999 μs)
			SD6	5 T	ime: in 1 ms increments (0 to 65535 ms)
			SD7 SD1		Empty
				L	

	Device number	Name				Des	cription			Page
						mmon information	categor	y code is 4, the program		
			error location is stored in SD5 to SD15.							
				Numb	er		Descri	ption		
				SD5						
				SD6		File name MAIN				
				SD7		(ASCII code: 8 ch	naracters	S)		
				SD8						
	SD5 to SD15	Common error information		SD9		Extension		2Ен(.)		
	303 to 3013	Common error information		SD10		QPG	(AS	CII code: 3 characters)		-
				SD11		Always 0				
				SD12		Always 0				
				SD13		Always 0				
				SD14 SD15		Sequence step N Sequence step N				
				3015	<sub>*</sub> 1			four null characters are stored.		
			Т	he sten		-		ed are stored in SD14 an	d	
				D15.	iiuii	inders where errors	3 Occurre		u	
								ry code is 3, the time		
	SD16 to SD26	Individual error information	(8	(actual measured data) is stored in SD16 and SD17.						
				Num	ber		Descri	ption		
ster				SD1	SD16 Time: in 1 µs increm		crement	s (0 to 999 µs)		-
regi				SD1	7	Time: in 1 ms ir	ncremen	ts (0 to 65535 ms)		
Special register				SD1		Empty				
Spe				302						
			When the individual information category code is 5, the parameter number is stored in SD16.						er	
			n	umber is	SIC	ored in SD16.				
						umber	Descri			
						Paramete	er numbe	er		
						Empty Empty				
						·				
					Г	b15 ~ b12 b11 ~	h8 h7 ~	b4 b3 ∼ b0		
					L					
				bit		Item	Value	Description		
				b3 to b	0	CPU module	0	RUN		
				00 10 1	,0	operating status				
	SD203	CPU module operating				STOP/RUN	1	STOP		_
		status		b7 to b	)4	factor	0	Switch		
							1	Remote contact		
								Remote operation from the peripheral		
							2	S/W for the		
								programmable		
							4	controller Error		
				<u> </u>			<u> </u>	LIIOI		
		-								

	evice number	Name	Description	Page
			Year (the last two digits of the year) is stored in BCD code.	
	SD210	Clock data	b15 ~ b12 b11 ~ b8 b7 ~ b4 b3 ~ b0  Last two digits of the year Month	-
			Date and hour are stored in BCD code.	
	SD211	Clock data	b15 ~ b12 b11 ~ b8 b7 ~ b4 b3 ~ b0  Day of the week Hour	-
			Minute and second are stored in BCD code.	
	SD212	Clock data	b15 ~ b12 b11 ~ b8 b7 ~ b4 b3 ~ b0  Minute Second	-
			Year (the first two digits of the year) and day of the week are	
Special register	SD213	Clock data	stored in BCD code.  b15 ~ b12 b11 ~ b8 b7 ~ b4 b3 ~ b0    Day	-
	SD520	Current scan time	Scan time is stored at every END, and is constantly updated. (Measured in 100 µs increments)	-
	SD521		SD520: Stores the millisecond places (Stored range: 0 to 65535) SD521: Stores the microsecond places (Stored range: 0 to 900)	-
	SD524	Minimum scan time	Minimum scan time is stored at every END. (Measured in 100 μs increments)	-
	SD525	William Godin anno	SD524: Stores the millisecond places (Stored range: 0 to 65535) SD525: Stores the microsecond places (Stored range: 0 to 900)	-
	SD526	Maximum scan time	Maximum scan time is stored at every END. (Measured in 100 μs increments)	-
	SD527	waxiiiuiii scali tiilie	SD526: Stores the millisecond places (Stored range: 0 to 65535) SD527: Stores the microsecond places (Stored range: 0 to 900)	
	SD1062 to SD1093	Remote register	Special register for communicating with the master station in CC-Link.	57
	SD1133	Output frequency monitor	Present output frequency is stored. The increment is 0.01 Hz.	
	SD1134	Output current monitor	Present output current is stored. The increment is 0.01 A.	30
	SD1135	Output voltage monitor	Present output voltage is stored. The increment is 0.1 V.	

	Device number	Name	Description	Page
	SD1136	Faults history 1, 2		
	SD1137	Faults history 3, 4	Faults in the inverter are stored in the order of occurrence.	31
	SD1138	Faults history 5, 6	— Faults in the inverter are stored in the order of occurrence.	
	SD1139	Faults history 7, 8		
	SD1140	Operation mode setting read	Stores the present operation mode.	34
	SD1141	Set frequency read (RAM)	Reads and stores the set frequency (RAM).	35
	SD1142	Set frequency read (EEPROM)	Reads and stores the set frequency (EEPROM).	35
	SD1143	Operation mode setting write	Sets the operation mode to be changed.	37
	SD1144	Set frequency write (RAM)	Sets the running frequency (RAM).	37
힏	SD1145	Set frequency write (EEPROM)	Sets the running frequency (EEPROM).	38
cont	SD1146	Fault record batch clear	Clears the faults history when H9696 is written.	39
Special register for control	SD1147	Parameter clear	H9696 write: parameter clear H9966 write: all clear H5A5A write: parameter clear except for communication parameters H55AA write: all clear except for communication parameters While communicating with FR Configurator2, clear the parameters using H5A5A or H55AA.	39
	SD1148	Inverter operating status control	Turn ON/OFF the corresponding bit to control the inverter operating status.  All initial values are set as "0". When SM1255 is OFF, this function is disabled.    Distribution of the inverter operating status.   Distribution of the inverter operating status.	40
	SD1149	Inverter operating status control enable/disable setting	The inverter operating status control by SD1148 and SM1200 to SM1211 can be enabled/disabled by turning ON/OFF the corresponding bit.  The bit table is the same as SD1148.  All initial values are set to "0" (invalid).  Input through external terminals is always enabled for the SQ signal. (The SQ signal cannot be controlled with SD1149 bits.)	41
	SD1150	Inverter parameter access error	Stores the error number generated when parameter or data stored in the special register was not successfully reflected to the inverter.	42

C	evice number	Name	Description	Page
	SD1151	Inverter status	Operating status of the inverter is stored.    Description	42
	SD1152	Frequency setting value	0.01 Hz increments	-
	SD1153	Running speed	1 (0.1) r/min increments  Pr.811 Set resolution switchover enables 0.1 r/min unit indication	-
	SD1154	Motor torque	0.1% increments	-
	SD1155	Converter output voltage	0.1 V increments	-
	SD1156	Regenerative brake duty	0.1% increments	-
	SD1157	Electric thermal relay load factor	0.1% increments	-
	SD1158	Output power peak value	0.01 A increments	-
	SD1159	Converter output voltage peak value	0.1 V increments	-
	SD1160	Input power	0.01 kW increments	-
ntro	SD1161	Output power	0.01 kW increments	-
Special register for control	SD1162	Input terminal status	Details of input terminal status    b15 b12 b11 b8 b7 b4 b3 b0   0:OFF   1:ON   STF   STR   AU   RT   RL   RM   RH   JOG   MRS   STOP   RES   CS   CS	-
	SD1163	Output terminal status	Details of output terminal status  b15 ····· b7 ····b4 b3 ····b0 0:OFF 1:ON  RUN SU IPF OL FU ABC1 ABC2	-
	SD1164	Load meter	0.1% increments	-
	SD1165	Motor excitation current	0.01 A increments	-
	SD1166	Position pulse	Stores the encoder position pulse	-
	SD1167	Cumulative energization time	1 hour increments	-
	SD1169	Orientation status	Stores the orientation status.	-
	SD1170	Actual operation time	1 hour increments	-
	SD1171	Motor load factor	0.1% increments	-
	SD1172	Cumulative power	1 kWh increments	-
	SD1173	Position command (lower 16 bits)	Stores the position command setting value (lower 16 bits).	-

[	evice number	Name	Description	Page
	SD1174	Position command (upper 16 bits)	Stores the position command setting value (upper 16 bits).	-
	SD1175	Feedback pulse (lower 16 bits)	Stores the feedback pulse value (lower 16 bits).	-
	SD1176	Feedback pulse (upper 16 bits)	Stores the feedback pulse value (upper 16 bits).	-
	SD1177	Droop pulse (lower 16 bits)	Stores the droop pulse value (lower 16 bits).	-
	SD1178	Droop pulse (upper 16 bits)	Stores the droop pulse value (upper 16 bits).	-
	SD1179	Torque command	0.1% increments	-
	SD1180	Torque current command	0.1% increments	-
	SD1181	Motor output	0.1 kW increments	-
	SD1182	Feedback pulse	Stores the feedback pulse quantity.	-
	SD1184	Station number (PU port)	Stores the station number (PU port).	-
	SD1185	Station number (RS-485 terminal block)	Stores the station number (RS-485 terminal block).	-
	SD1186	Communication station number (CC-Link)	Stores the communication station number (CC-Link).	-
	SD1187	Remote output 1	Stores the analog remote output setting value 1.	-
	SD1188	Remote output 2	Stores the analog remote output setting value 2.	-
	SD1189	Remote output3	Stores the analog remote output setting value 3.	-
trol	SD1190	Remote output 4	Stores the analog remote output setting value 4.	-
Special register for control	SD1191	PTC thermistor resistance value	Stores the PTC thermistor resistance value.	-
ster	SD1197	Energy saving effect	According to the parameter cettings	-
egis	SD1198	Cumulative energy saving	According to the parameter settings.	-
ialr	SD1199	PID set point	Stores the PID set point. The increment is 0.1%.	-
Spec	SD1200	PID measured value	Stores the PID measured value. The increment is 0.1%.	-
0,	SD1201	PID deviation	Stores the PID deviation. The increment is 0.1%.	-
	SD1202	Second PID set point	Stores the second PID set point. The increment is 0.1%.	-
	SD1203	Second PID measured value	Stores the second PID measured value. The increment is 0.1%.	-
	SD1204	Second PID deviation	Stores the second PID deviation. The increment is 0.1%.	-
	SD1205	Option input terminal status 1	Stores the input status of the FR-A8AX. All are OFF (0) when there are no options installed.    b15:b12 b11:b8 b7:b4 b3b0   0:OFF   1:ON   x0   x1   x2   x2   x3   x4   x5   x6   x7   x7   x7	-
	SD1206	Option input terminal status 2	X8 X9 X10 X10 X11 X12 X13 X14 X15  b15·····b12 b11·····b8 b7 ·····b4 b3 ····b0 SD1206  SD1206  DY	-

C	evice number	Name	Description	Page
	SD1207	Option output terminal status	Stores the output status of the FR-A8AY or FR-A8AR.  All are OFF (0) when there are no options installed.    b15 \cdots b12 b11 \cdots b8 b7 \cdots b4 b3 \cdots b0 \\   Y0 \\   Y1 \\   Y2 \\   Y3 \\   Y4 \\   Y5 \\   Y6 \\   RA1 \\   RA2 \\   RA3	-
	SD1211	Dancer main speed setting	Stores the dancer main speed setting.	-
	SD1212	PID manipulated amount	Stores the PID manipulated amount. The increment is 0.1%.	-
	SD1213	PID measured value 2	Stores the PID measured value. The increment is 0.1%. (Monitoring can be performed even when PID control is disabled.)	-
	SD1214	User definition error	An inverter fault is initiated by setting the values of "16 to 20" in SD1214.	43
_	SD1215	Monitor setting selection	Set SD1215 to display the monitor set in SD1216 to SD1218.	43
ontr	SD1216	Monitor 1 setting value	The first to the third monitors on the operation panel (FR-DU08) or	
or c	SD1217	Monitor 2 setting value	the parameter unit (FR-PU 07) can be changed by setting the	43
ter f	SD1218	Monitor 3 setting value	value "40 to 42" in <b>Pr.774 to Pr.776.</b>	
egis	SD1220	Motor thermal load factor	Stores the motor thermal load factor.	-
ialr	SD1221	Inverter thermal load factor	Stores the inverter thermal load factor.	-
Special register for control	SD1222	Second PID manipulated amount	Stores the second PID manipulated amount. The increment is 0.1%.	-
	SD1223	Second PID measured value 2	Stores the second PID measured value. The increment is 0.1%. (Monitoring can be performed even when PID control is disabled.)	-
	SD1224	32-bit cumulative power (Lower 16 bits)	1 kWh	-
	SD1225	32-bit cumulative power (Upper 16 bits)	1 kWh	-
	SD1226	32-bit cumulative power (Lower 16 bits)	0.01 kWh/0.1 kWh *2	-
	SD1227	32-bit cumulative power (Upper 16 bits)	0.01 kWh/0.1 kWh *2	-
	SD1229	Trace status	Stores the trace status.	-
	SD1234	Second parameter change (RAM)	When setting a calibration parameter (bias/gain) H00: Frequency (torque)	44,
	SD1235	Second parameter change (EEPROM)	H01: Analog value set by parameters H02: Analog value input from terminals	45

<sup>\*2</sup> Settings differ according to the inverter capacity. (FR-A820-03160(55K) or lower and FR-A840-01800(55K) or lower, FR-A820-03800(75K) or higher and FR-A840-02160(75K) or higher)

	evice number	Name	Description	Page	
	SD1236	Pulse train input sampling pulse	The number of pulses counted in count cycle is stored. (0 to 32767)		
	SD1237	Pulse train input cumulative count value L	The cumulative value of the number of sampling pulses is		
	SD1238	Pulse train input cumulative count value H	stored. (0 to 99999999)	48	
	SD1239	Reset request of pulse train input count	The sampling pulses and cumulative count value are cleared. Automatically changes to "0" after reset. (1: count clear)		
	SD1240	Count start of the pulse train input	Start counting the sampling pulses and cumulative count value. (0: count stop, 1: count start)		
	SD1241	Parameter number (RAM)	Set the parameter number to read from/write to the inverter.		
	SD1242	Parameter detail (RAM)	Stores the inverter parameter content (RAM value) specified by SD1241. Input the parameter setting value when writing the parameters.		
	SD1243	Parameter number (EEPROM)	Set the parameter number to read from/write to the inverter.	44, 45	
	SD1244	Parameter content (EEPROM)	Stores the inverter parameter content (EEPROM value) specified by SD1243. Input the parameter setting value when writing the parameters.		
	SD1245	Terminal 1 input	Stores the analog input value (0.1% increments) to terminal 1.		
_	SD1246	Terminal 2 input	Stores the analog input value (0.1% increments) to terminal 2.	47	
ntro	SD1247	Terminal 4 input	Stores the analog input value (0.1% increments) to terminal 4.		
for control	SD1248	PID control set point/ deviation	Set the PID set point or PID deviation. (0.01% increments)		
iste	SD1249	PID control measured value	ID control measured value Set the PID measured value. (0.01% increments)	49	
Special register	SD1250	PID control manipulated amount	Stores the PID manipulated amount. (0.01% increments)		
Spec	SD1251	Terminal FM/CA output	When <b>Pr.54</b> ="70", analog output can be performed from the terminal FM/CA. (0.1% increments)		
	SD1252	Terminal AM output	When <b>Pr.158</b> = "70", analog output can be performed from terminal AM. (0.1% increments)	47	
	SD1253	AM0 output	Analog output can be performed from terminals AM0 and AM1		
	SD1254	AM1 output	of the FR-A8AY. (0.1% increments)		
	SD1255	PID operation control	Sets the start/stop of PID operation. Set "1" to start PID control.		
	SD1308	Second PID control set point/deviation	Set the second PID set point or the second PID deviation. (0.01% increments)	49	
	SD1309	Second PID control measured value	Set the second PID measured value. (0.01% increments)	43	
	SD1310	Second PID control manipulated amount	Stores the second PID manipulated amount. (0.01% increments)		
	SD1320	Read parameter content (2-word parameter: L)	When the parameter to be written is 2 words, storage locations are as follows:	-	
	SD1321	Read parameter content (2-word parameter: H)	SD1320: Lower 1 word SD1321: Upper 1 word	-	
	SD1322	Write parameter content (2-word parameter: L)	When the parameter to be read is 2 words, storage locations are as follows:	-	
	SD1323	Write parameter content (2-word parameter: H)	SD1322: Lower 1 word SD1323: Upper 1 word	-	

# 1.9 Special registers to monitor and control inverter status

Read from and write to user sequences are possible by assigning the data to monitor and change the operating status of the inverter from SD1133 to SD1147. (For the list of registers, refer to page 22.)

### 1.9.1 Always readable data

The following data is always readable. They are refreshed on every END instruction automatically.

### **♦**Operation monitor

These are the data devices by which the output frequency, output current, and output voltage of the inverter can be monitored, and which are always readable (write is disabled). Pay attention to the setting increments.

Device number	Name	Setting increments	Data example	Data access condition
SD1133	Output frequency monitor	0.01 Hz	Device content 6000→60.00 Hz	
SD1134	Output current monitor	0.01 A	Device content 200→2.00 A	Always accessible
SD1135	Output voltage monitor	0.01 V	Device content 1000→10.0 V	



• Setting frequency can be set in increments of 0.01 Hz, but operates at 0.1 Hz increments in actual operation.

#### 1

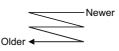
### **♦**Faults history (error code and error definition)

When the inverter fails, it stores error codes.

A maximum of eight faults are stored with error codes in the order shown below, and they are always readable (write is disabled).

#### <Error code storing method details>

	b15 to b8	b7 to b0
	Faults history 2	Faults history 1
	,	Faults history 3
		Faults history 5
SD1139	Faults history 8	Faults history 7



Error code	Fault record
H00	No failure
H10	E.OC1
H11	E.OC2
H12	E.OC3
H20	E.OV1
H21	E.OV2
H22	E.OV3
H30	E.THT
H31	E.THM
H40	E.FIN
H50	E.IPF
H51	E.UVT
H52	E.ILF
H60	E.OLT
H61	E.SOT

Error code	Fault record
H70	E.BE
H80	E.GF
H81	E.LF
H90	E.OHT
H91	E.PTC
HA0	E.OPT
HA1	E.OP1
HA4	E.16
HA5	E.17
HA6	E.18
HA7	E.19
HA8	E.20
НВ0	E.PE
HB1	E.PUE
HB2	E.RET
HB3	E.PE2

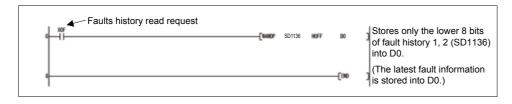
Error code	Fault record
HC0	E.CPU
HC1	E.CTE
HC2	E.P24
HC4	E.CDO
HC5	E.IOH
HC6	E.SER
HC7	E.AIE
HC8	E.USB
HC9	E.SAF
HD0	E.OS
HD1	E.OSD
HD2	E.ECT
HD3	E.OD
HD5	E.MB1
HD6	E.MB2
HD7	E.MB3

Error	Fault record	
code		
HD8	E.MB4	
HD9	E.MB5	
HDA	E.MB6	
HDB	E.MB7	
HDC	E.EP	
HE4	E.LCI	
HE5	E.PCH	
HE6	E.PID	
HF1	E.1	
HF2	E.2	
HF3	E.3	
HF5	E.5	
HF6	E.6	
HF7	E.7	
HFB	E.11	
HFD	E.13	

For details of fault records, refer to the Instruction Manual of the inverter.

### <Program example for reading a fault record>

The following description is of a program that reads out the latest fault record of the inverter and stores to D0.

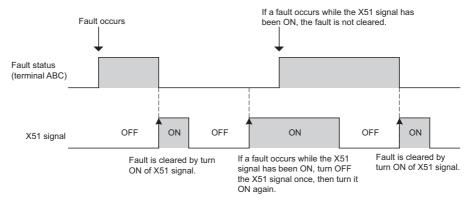


### **♦**Fault clear signal (X51 signal)

When an inverter fault occurs, X51 signal clears the fault without resetting the inverter.

By using the X51 signal, the inverter fault can be cleared with the PLC function still operating.

Set "51" in Pr.178 to Pr.189 (Input terminal function selection) in advance to enable the X51 signal.



- The fault clear function using the X51 signal does not function while the retry function is operating (including during the retry waiting time).
  - (The X51 signal operates when the retry count is exceeded (E.RET).
- When the fault is cleared, start commands from PU and communication are also cleared. (All commands except the start command from communication are maintained in the status before the fault is cleared.)

- E.CPU, E.P24, E.CTE, E.1 to E.3, E.6, E.7, E.13, E.14, E.16 to E.20, E.SAF and E.PE2 are not cleared by X51 signal.
- When a fault is cleared, the accumulated heat values of the electronic thermal O/L relay and regenerative brake are not cleared. Faults including E.THM, E.THT, and E.BE may occur again.
- The inverter remains in the stop status immediately after a fault is cleared. However, the inverter starts operating again after the fault is cleared if the start command is ON.
- When an error occurs, the normal reset function operates when the STOP/RESET key on the operation panel or parameter unit is pushed. Be careful not to push the key by mistake.
- Changing terminal assignment by using **Pr.178 to Pr.189 (input terminal function selection)** may affect the other functions. Set parameters after confirming the function of each terminal.

### ◆ Self-diagnostic error, error number and details

When an operation error occurs during execution of sequence programs, the following error numbers are stored in SD0. When a self-diagnostic error occurs, the "P.RUN" indicator (LED) blinks.

Error No.	Error name	Details		
1010	END NOT EXECUTE	END command was not executed.		
2200	MISSING PARA.	No parameter file.		
2502	CAN'T EXE.PRG	The program file is not correct. Or the content of the file is not a program.		
3000	PARAMETER ERR	The content of the parameter indicated by the individual error information (SD16) is incorrect.		
3003	PARAMETER ERR	The number of device points set in the PLC parameter device setting is not within the usable range.		
4000	INSTRCT. CODE ERR	An undecodable instruction code is included in the program. An unusable instruction is included in the program.		
4010	MISSING END INS.	No END (FEND) instruction in the program.		
4100	OPERATION ERROR	Data that cannot be used in instructions is included.		
5001	WDT ERROR	The program scan time exceeded the watchdog timer value set in the PLC RAS setting of the PLC parameter.		
5010	PRG.TIME OVER	The program scan time exceeded the constant scan time set in the PLC RAS setting of the PLC parameter.		



· Operation at error stop

Output (Y) is cleared.

Other devices remain in the status before the error stop occurred.

To clear the statuses, turn OFF or reset (turn RES signal ON (0.1 s) and then OFF) the inverter.

# 1.9.2 Data read by controlling read command (from OFF to ON)

The operation mode and set frequency of the inverter can be read.

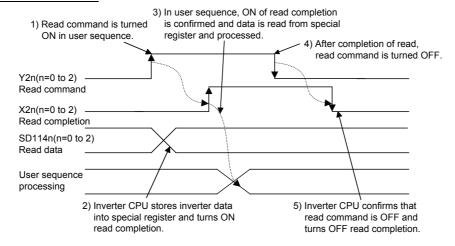
Device number	Name	Read instruction	Read completion	Data access condition
SD1140	Operation mode setting read	Y20	X20	
SD1141	Set frequency read (RAM)	Y21	X21	Always accessible
SD1142	Set frequency read (EEPROM)	Y22	X22	

For the data devices above, data is stored when the read command switches from OFF to ON and read completion switches from OFF to ON.

Data is not refreshed while the read command is ON. (Data is not updated.)

Data is refreshed by turning the read command OFF and ON again.

### Time chart of each data read



### ♦ Operation mode setting read (SD1140)

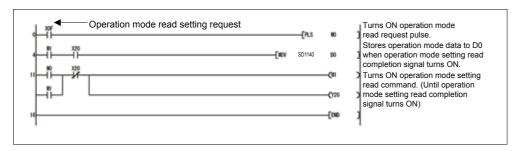
Data content	Operation mode
H0000	NET operation mode
H0001	External operation mode
H0002	PU operation mode



• When **Pr.79 Operation mode selection** ≠ "0", the setting is applied. However, if **Pr.79** = "3 or 4", "H0002" (PU operation mode) is applied.

### <Program example of reading operation mode settings>

The following description is of a program that reads out the operation mode data to D0.



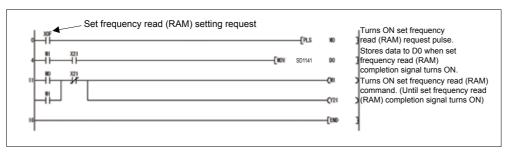
### ◆Set frequency read (RAM) (SD1141)

The frequency set in RAM can be read out from SD1141. Its increment is 0.01 Hz. (For example, "6000" means 60.00 Hz.)

If the rotation rate is set, the increment is 1 r/min or 0.1 r/min. (According to the setting value of Pr.811)

### <Program example of reading the set frequency (RAM)>

The following description is of a program that reads out the set frequency (RAM) to D0.



### • NOTE

• The frequency to be read is not the command value of an external signal.

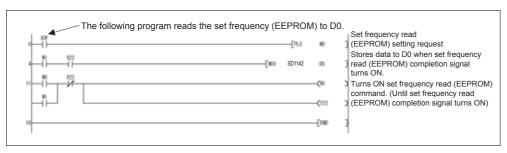
### ◆Set frequency read (EEPROM) (SD1142)

The frequency set in EEPROM can be read from SD1142. Its increment is 0.01 Hz. (For example, "6000" means 60.00 Hz.)

If the rotation rate is set, the increment is 1 r/min or 0.1 r/min. (According to the setting value of Pr.811)

### <Program example of reading the set frequency (EEPROM)>

The following description is of a program that reads out the set frequency (EEPROM) to D0.



### NOTE:

• The frequency to be read is not the command value of an external signal.

## 1.9.3 Data writing method by controlling write command (from OFF to ON)

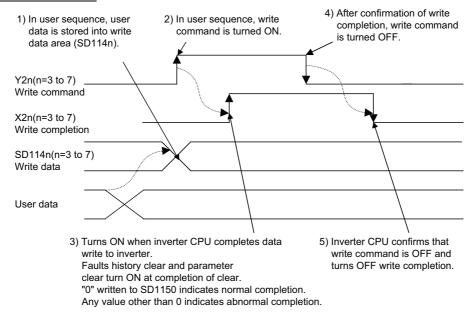
Writing of an operation mode and setting frequency to the inverter, a batch clear of fault records, and all parameter clear can be executed.

Device number	Name	Write instruction	Write completion	Data access condition
SD1143	Writing the operation mode setting	Y23	X23	<b>Pr.79</b> =0, 2
SD1144	Writing the setting frequency (RAM)	Y24	X24	PU operation mode (PU LED ON) or
SD1145	Writing the setting frequency (EEPROM)	Y25	X25	CC-Link operation mode (PU and EXT LEDs blink slowly.)
SD1146	Batch clearing of fault records	Y26	X26	Always accessible
SD1147	All parameter clear	Y27	X27	According to the setting of Pr.77

The data above is written at the point when write completion turns ON after the write command turns from OFF to ON. (Written at clear completion of SD1146 executed for fault record batch clear and SD1147 for all parameter clear.)

To write again, the write command must be turned OFF and then ON again.

#### Time charts of a data write

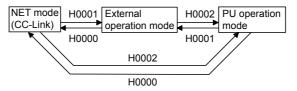


## **♦**Writing operation mode settings (SD1143)

Data content:

Data content	Operation mode
H0000	NET operation mode
H0001	External operation mode
H0002	PU operation mode

The method for changing operation modes is shown in the following chart when Pr.79 Operation mode selection = "0".



When **Pr.79** = "2", the chart is as follows.





When Pr.79 ≠ "0", the mode is fixed.

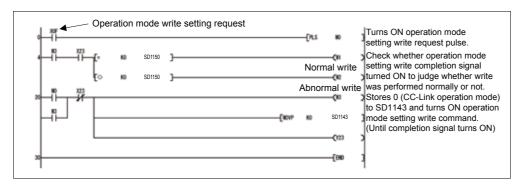
There are no limiting conditions for operation mode transitions.

When the operation mode settings are completed normally, write completion signal (X23) turns ON at the same time as SD1150 is set to "0".

Writing any setting value other than H0000 to H0002 or executing writing while the inverter is operating causes the write completion signal (X23) to turn ON at the same time as setting HFFFF in SD1150, resulting in error completion. If an error completion occurs, the operation mode is not changed.

#### <Program example of writing operation mode settings>

The following description is of the program that changes the operation mode to NET mode.



## ◆Setting frequency write (RAM) (SD1144)

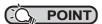
The content of SD1144 is written to the RAM as setting frequency. Its increment is 0.01 Hz. (For example, 60.00 Hz is written as "6000".)

If the rotation speed is set, the increment is 1 r/min or 0.1 r/min. (According to the setting value of Pr.811)

The frequency can be set within the range of 0 to 12000 (0 to 120.00 Hz).

When frequency settings are written normally, the write completion signal (X24) turns ON at the same time as SD1150 is set to "0".

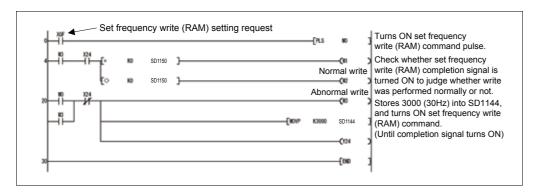
Executing writing with any value that is outside the specified range causes the write completion signal (X24) to turn ON at the same time as setting HFFFF in SD1150, resulting in error completion. If an error completion occurs, the setting frequency is not changed.



• This can be set in PU operation mode and NET operation mode. Refer to the Instruction Manual of the inverter.

#### <Program example of writing the setting frequency (RAM)>

The following description is of a program that changes the setting frequency (RAM) to 30 Hz.



## ◆ Setting frequency write (EEPROM) (SD1145)

The content of SD1145 is written to EEPROM as setting frequency. Its increment is 0.01 Hz. (For example, 60.00 Hz is written as "6000".)

If the rotation rate is set, the increment is 1 r/min or 0.1 r/min. (According to the setting value of Pr.811)

The frequency can be set within the range of 0 to 12000 (0 to 120.00 Hz).

When frequency settings are written normally, the write completion signal (X25) turns ON at the same time as SD1150 is set to "0".

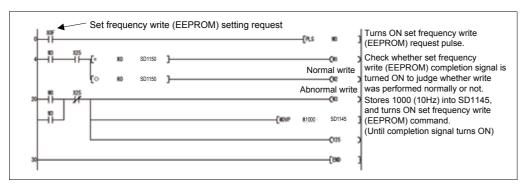
Executing writing with any value that is outside the specified range causes the write completion signal (X25) to turn ON at the same time as setting HFFFF in SD1150, resulting in error completion. If an error completion occurs, the setting frequency is not changed.



• This can be set in PU operation mode and NET operation mode. Refer to the Instruction Manual of the inverter.

#### <Program example of writing setting frequency (EEPROM)>

The following description is of a program that changes the setting frequency (EEPROM) to 10 Hz.





• When set frequencies need to be changed often, use the device SD1144 "set frequency (RAM)". There is a limitation on the number of times EEPROM can be written to. (Approximately 100,000 times)

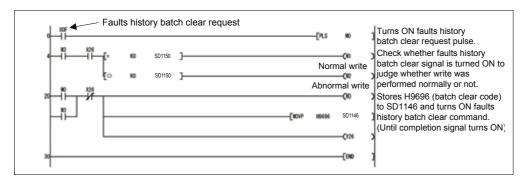
## **♦**Batch clear of fault records (SD1146)

When H9696 is written to SD1146, fault records are cleared in a batch.

When clearing is completed, the write completion signal (X26) turns ON at the same time as SD1150 is set to "0". Executing writing with any value that is outside the specified range or writing while the inverter is running causes the write completion signal (X26) to turn ON at the same time as setting HFFFF in SD1150, resulting in error completion. In the case of error completion, the fault records are not cleared.

#### <Program example of batch clearing of fault records>

The following description is of the program that batch clears fault records.



## ◆Parameter clear (SD1147)

When H9696 or H9966 is written to SD1147, all parameters are cleared. When H5A5A or H55AA is written to SD1147, parameters except the communication parameters are cleared. (Refer to the Instruction Manual of the inverter.)

Device No.	Setting value	Communication parameters	Other parameters	Details	
	H9696	0	0	Terminal functions are not cleared.	
SD1147	H9966	0	0	Terminal functions are cleared.	
301147	H5A5A	×	0	Terminal functions are not cleared.	
	H55AA	×	0	Terminal functions are cleared.	

O: Clear, x: Not clear

When clearing is complete, the write completion signal (X27) turns ON at the same time as SD1150 is set to "0". Executing writing with any value that is outside the specified range or writing while the inverter is running causes the write completion signal (X27) to turn ON at the same time as setting HFFFF in SD1150, resulting in error completion. In the case of an error completion, the parameters are not cleared.



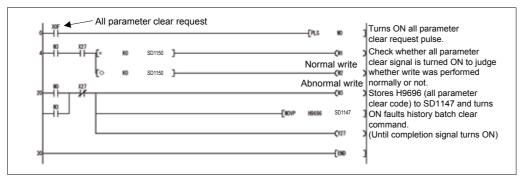
• Identify the terminal function parameters and communication parameters in the parameter list. (Refer to the Instruction Manual of the inverter.)



• This can be set in the PU operation mode and NET operation mode. Refer to the Instruction Manual of the inverter.

#### <Program example of clearing all parameters>

The following description is of the program that clears all parameters.



## • NOTE

• When parameter clear/all clear is executed, communication parameters are also cleared. Thus communication with FR Configurator2 (Developer) becomes impossible.

#### ♦ Related device ♦

Device SD1150: inverter parameter access error (refer to page 42)

## 1.9.4 Inverter operating status control

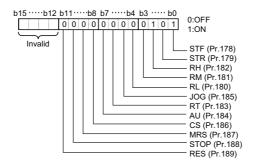
Device number	Name	Data access condition
SD1148	Inverter operating status control	Always accessible.
SD1149	Inverter operating status control enable/disable	However, this is only active in External/NET operation mode. (Disabled in PU operation mode)

## ♦Inverter operating status control (SD1148)

SD1148 is a device for controlling the operating status of the inverter. The inverter operation can be controlled by turning the bits b0 to b11 of SD1148 ON/OFF (1,0).

All initial values are "0".

Example) When SD1148 is set to "5", the bits b0 and b2 are turned to 1(ON) and thus STF and RH are turned ON, creating the fast forward rotation command.



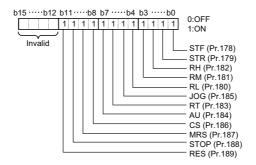
## NOTE

- The functions of SD1148 bits are assigned by **Pr.178 to Pr.189** in the same way as for the external input terminal. Note that SQ signal (setting value: 50) cannot be assigned.
- Turn ON SM1255 (special register selection) before using SD1148. (Refer to page 20.)

## ♦Inverter operating status control enable/disable setting (SD1149)

The inverter operating status control by SD1148 and SM1200 to SM1211 can be enabled/disabled. Control of the bits of SD1148 and SM1200 to SM1211 are enabled by turning the corresponding bits b0 to b11 of SD1149 ON/OFF (1, 0). All initial values are "0".

Example) When SD1149 is set to H1F, the bits b0 to b11 change to 1 (ON) disabling all inputs through external terminals, thus enabling operation control by the inverter operation control (SD1148) and inverter operating status control flag (SM1200 to SM1211).



## • NOTE

- When SD1148 and SM1200 to SM1211 are enabled by SD1149, control by external terminal input or CC-Link remote input are disabled for the bits that are enabled. (Same as setting **Pr.178 to Pr.189** to "no function".)
- · When terminals are enabled by PLC function, the control through external terminals is disabled.
- Input through external terminals is always enabled for SQ signal.
   (Control of SQ signal cannot be performed from each SD1149 bit.)

#### <Program example of operation command settings>

Program example of operating the inverter in fast forward rotation.



## 1.9.5 Inverter parameter access error (SD1150)

Device Name number		Data access condition
SD1150	Inverter parameter access error	Always accessible

When an out-of-range setting value is written to a parameter or an out-of-range frequency is set using sequence programs, or when writing is attempted while writing is disabled, a write error occurs and an error code is stored in SD1150.

#### <Parameters>

Parameter number + H8000 is stored to SD1150.

Example) If an error occurs when Pr.0 Torque boost is written, H8000 (H0 + H8000) is stored to SD1150.

If an error occurs when Pr.10 DC injection brake operation frequency is written, H800A is stored to SD1150.

<Operation mode, setting frequency, batch clear of fault records, and all parameter clear> HFFFF is stored to SD1150. (The initial values is "0".)



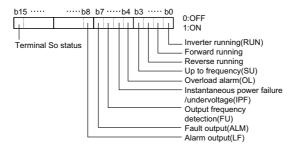
• The error code stored after the error occurrence will be cleared by writing "0" in SD1150. (The error code is retained until "0" is written in SD1150.)

## 1.9.6 Inverter status (SD1151)

Device number	Name	Data access condition	
SD1151	Inverter status	Always accessible	

Operating status of the inverter is stored.

Each bit is set according to the status of the inverter.



## 4

## 1.9.7 User-defined error (SD1214)

By setting the values "16 to 20" in SD1214, alarms for the inverter can be created. When an alarm occurs, the inverter stops. Values other than "16 to 20" are invalid. Furthermore, this function is disabled when **Pr.414** = "0".

The created alarms are recognized as E.16 to E.20 by the inverter.

## 1.9.8 Monitor setting selection (SD1215 to SD1218)

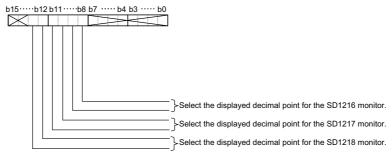
Monitor names and units need to be set in SD1216, SD1217, and SD1218 by using G.PRR, GP.PRR, UMSG instruction (refer to page 193).

To display the monitor names and units set in SD1216 to SD1218 on the parameter unit (FR-PU07), set "40 to 42" in **Pr.774 to Pr.776**. (For details of **Pr.774 to Pr.776**, refer to **the Instruction Manual (Detailed)** of the inverter.)

Device	Monitor description	Pr.774 to Pr.776 setting
SD1216	User monitor 1	40
SD1217	User monitor 2	41
SD1218	User monitor 3	42

To display the monitors set in SD1216 to SD1218 to decimal places, set SD1215 as follows.

#### <Setting of SD1215>



\*1 Ignore bits other than those above.

#### <Decimal point display setting>

bn + 1	bn	Increment
0	0	1 increment (no decimals)
0	1	0.1 increment
1	0	0.01 increment
1	1	0.001 increment

#### Setting example:

- 1. To display SD1216 without decimals
  - Set H0000 in SD1215.
  - Set "40" in Pr.774.
- 2. To display SD1216 in 0.1 increment, and SD1218 in 0.001 increment.
  - Set H3100 in SD1215.
  - Set "40" in Pr.774 and "42" in Pr.775.

## **1.10** Read/Write method of inverter parameters

## 1.10.1 Reading inverter parameters

Device number	Name	Command	Completion	Data access condition (Operation mode)
SD1241	Parameter number (RAM)			
SD1242	Parameter content (RAM)	Y28	X28	
SD1234	Second parameter change (RAM)			Alwaya allawad
SD1243	Parameter number (EEPROM)			Always allowed
SD1244	Parameter content (EEPROM)	Y2A	X2A	
SD1235	Second parameter change (EEPROM)			

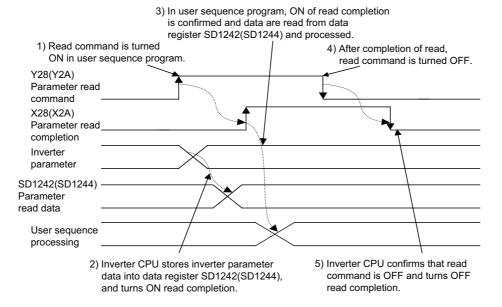
To read a parameter, store the parameter number in SD1241 (SD1243) and turn ON Y28 (Y2A). The parameter content will be stored in SD1242 (SD1244). After the reading is complete, X28 (X2A) turns ON to notify of the completion. (Use the device numbers in parentheses to read a parameter setting value from EEPROM.)

When reading a calibration parameter (**Pr.902 to Pr.935**), set a value listed below in SD1234 (SD1235) to read each calibration value.

- 0: Setting value (frequency)
- 1: Analog value by parameter setting
- 2: Analog value input from terminal

If an access error such as non-existent parameter occurs, the value of the specified parameter number plus 8000H is stored in SD1150. (Refer to page 42.)

#### Inverter parameter data read timing chart



## 1.10.2 Writing inverter parameters

Device number	Name	Command	Completion	Data access condition (Operation mode)
SD1241	Parameter number (RAM)			
SD1242	Parameter content (RAM)	Y29	X29	
SD1234	Second parameter change (RAM)			PU or NET operation mode
SD1243	Parameter number (EEPROM)			(depending on Pr.77)
SD1244	Parameter content (EEPROM)	Y2B	X2B	
SD1235	Second parameter change (EEPROM)			

To write to a parameter, store the parameter number in SD1241 (SD1243) and the value to be written in SD1242 (SD1244), and turn ON Y29 (Y2B) to execute writing. After the writing completes, X29 (X2B) turns ON to notify of the completion. (Use the device numbers in parentheses to write a parameter setting value to EEPROM.)

When writing to a calibration parameter (Pr.902 to Pr.935), set a value listed below in SD1234 (SD1235) to write each calibration value.

- 0: Setting value (frequency)
- 1: Analog value by parameter setting
- 2: Analog value input from terminal

When the inverter parameter write completion signal (X29 (RAM) or X2B (EEPROM)) turns ON with normal completion, SD1150 is set to 0.

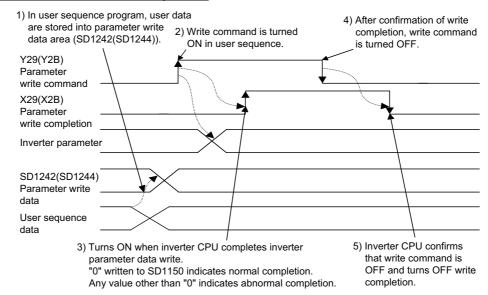
If an error in access to a parameter such as setting an out-of-range value and writing during inverter operation occurs, the write completion signal (X29 (RAM) or X2B (EEPROM)) turns ON and the value of the parameter number plus H8000 is set in SD1150 as soon as the write completion signal (X29 (RAM) or X2B (EEPROM)) is turned ON, resulting in error completion. When an error completion occurs, the parameter is not written. (For example, if an error occurs in **Pr.0 Torque boost**, H8000 is written to SD1150.)

For details on whether inverter parameter writing is allowed, refer to Pr.77 Parameter write selection.



• Inverter parameter writing must be performed in PU operation mode or NET operation mode. (Refer to the Instruction Manual of the inverter.)

#### Inverter parameter data write timing chart



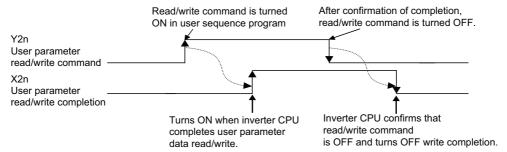
## 1.11 User area reading/writing

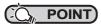
Inverter parameters Pr.1150 to Pr.1199 can be used as user parameters.

Since this parameter area and the devices D206 to D255 used in PLC function are accessible to each other, values set in **Pr.1150 to Pr.1199** are available in sequence programs. In addition, operation results in sequence programs can be monitored with **Pr.1150 to Pr.1199**.

Device number	Parameter number	Name	Command	Completion	Data access
	Pr.1150 to Pr.1199	User parameter read (RAM)	Y2C	X2C	
D206 to D255		User parameter write (RAM)	Y2D	X2D	Always allowed
		User parameter read (EEPROM/RAM)	Y2E	X2E	
		User parameter write (EEPROM/RAM)	Y2F	X2F	

Turn the read/write instruction from OFF to ON and then turn the read/write completion ON to read/write the user parameters from RAM and EEPROM.





· Application example of user parameter

Operation timings can be adjusted by assigning a timer in D206 and varying the value set in the timer. Values can be set to the timer with the **Pr.1150** setting without changing the program.

The user parameters (**Pr.1150 to Pr.1199**) and the devices (D206 to D255) can be freely read and written. Data transfer between **Pr.1150 to Pr.1199** and D206 to D255 is automatically executed.

1) Writing of user parameter and devices

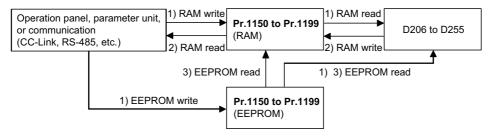
When values are written to **Pr.1150 to Pr.1199** via the operation panel, a parameter unit, or communication, they are written to the RAM area and the EEPROM area for parameter storage and also to D206 to D255 at the same time.

2) Reading user parameters and devices

When values are written to D206 to D255 using the PLC function, they are written to the RAM area for storing parameters (**Pr.1150 to Pr.1199**) and read via the operation panel, a parameter unit, or communication. (Since they are not written to EEPROM, resetting the power restores the previous values.)

3) Process at inverter reset or power restoration

When the inverter is reset, the values of **Pr.1150 to Pr.1199** stored in EEPROM are transferred to the RAM area and to D206 to D255.



## NOTE :

- If EEPROM is read after writing a parameter directly (to RAM), the values in RAM are replaced by the values in EEPROM.
- When Pr.342 Communication EEPROM write selection = "1", RAM values are read or written.

#### 4

## 1.12 Analog I/O function

## 1.12.1 Analog input

Analog input values from terminals 1, 2, and 4 can be read from SD1245 to SD1247.

Device number	Terminal name	Setting increments	Data access condition
SD1245	Terminal 1 input	0.1%	
SD1246	Terminal 2 input	0.1%	Always allowed
SD1247	Terminal 4 input	0.1%	

The actual reading is performed at END processing of the sequence program.



• The full scale value of analog input depends on the setting values of **Pr.73 Analog input selection** and **Pr.267 Terminal 4 input selection**. Refer to the Instruction Manual of the inverter.

## 1.12.2 Analog output

Write values from SD1251 to SD1254 to enable analog output from respective terminals.

Write "70" to the output signal selection parameter of each terminal (Terminal FM/CA: **Pr.54**, Terminal AM: **Pr.158**, FR-A8AY Terminal AM0, AM1: **Pr.306**, **Pr.310**) to enable output from PLC function.

Device number	Terminal name	Setting increments	Data access condition
SD1251	Terminal FM/CA	ninal FM/CA 0.1%	
SD1252	Terminal AM	0.1%	Always allowed
SD1253	Terminal AM0 (FR-A8AY)	0.1%	Always allowed
SD1254	Terminal AM1 (FR-A8AY)	0.1%	

The actual reading is performed at END processing of the sequence program.

## **1.13** Pulse train input function

Pulse train input values from terminal JOG can be read with the PLC function.

To read pulse train input values with the PLC function, set the terminal JOG by setting **Pr.291 Pulse train I/O selection** = "1, 11, 21, or 100" and **Pr.384 Input pulse division scaling factor** = "0 (initial value)".

After setting terminal JOG, set SD1240 to "1" to start counting the number of sampling pulses and accumulated count values. Pulse train (the number of sampling pulses) from terminal JOG is stored to SD1236.

When the sampling pulses overflow, make adjustment with the setting of Pr. 416 and Pr. 417.

The number of sampling pulses

= the number of input pulses per count cycle × pre-scale setting value (Pr. 417) × increments scaling factor (Pr. 416)

Parameter	Name	Initial value	Setting range	Description
291	Pulse train I/O selection	0	0, 10, 20	Terminal JOG
291	Fuise trail 1/O selection	0	1, 11, 21, 100	Pulse train input
416	Pre-scale function selection	0	0 to 5	Pre-scale function selection (increments scaling factor) 0: No function 1: ×1 2: ×0.1 3: ×0.01 4: ×0.001 5: ×0.0001
417	Pre-scale setting value	1	0 to 32767	Set the pre-scale value to calculate the number of sampling pulse when inputting the pulse train.

Device number	Name	Setting range	Description	
SD1236	Pulse train input sampling pulse	0 to 32767	The number of pulses counted in count cycle is stored.	
SD1237	Pulse train input cumulative count value L		The cumulative value of the number of	
SD1238	Pulse train input cumulative count value H	0 to 99999999	sampling pulses is stored.	
SD1239	Reset request of pulse train input count	0	Not clear	
301239	Reset request of pulse train input count	1	Count clear	
SD1240	Count start of the pulse train input	0	Stop counting	
30 1240	Count start or the pulse train input	1	Start counting	

## 1.14 PID control

Set Pr.128 (Pr.753) to enable setting of set point/ deviation and measured value for PID control with PLC function.

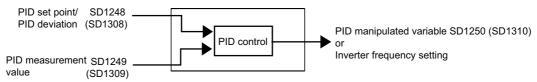
PID operation is performed using the value of SD1248 (SD1308) as the set point/deviation, and the value of SD1249 (SD1309) as the measured value. The manipulated amount will be stored in SD1250 (SD1310).

To perform PID control using the PLC function, set "1" in SD1255 instead of X14 signal.

When **Pr.128** = "70, 71, 80, or 81", PID operation is performed after the start. Thus, even if "1" is set in SD1255, the manipulated amount of SD1250 remains the same and the value of SD1250 is unchanged from "0".

When **Pr.128** = "90, 91, 100, or 101" and "1" is set in SD1255, PID operation is performed and the operation result is applied to the manipulated amount SD1250.

Set Pr.753 to enable the second PID control (SD1308 to SD1310).



Parameter	Name	Initial value	Setting range	Description		
			0	PID control disabled		
	10, 11, 20, 21, 40 to 43, 50, 51, 60, 61, 1000, 1001, 1010, 1011, 2000, 2001, 2010, 2011	e Instruction Manual of the inverter.				
128	DID action		70	PID reverse action	Deviation signal input (PLC function)	
753	PID action selection	0	71	PID forward action	- Deviation signal input (i Le function)	
			80	PID reverse action	Measured value and set point input (PLC	
			81	PID forward action	function)	
			90	PID reverse action	Deviation signal input (PLC function)	
			91	PID forward action	Not applied to inverter output frequency.	
			100	PID reverse action	Measured value and set point input (PLC	
			101	PID forward action	function)  Not applied to inverter output frequency.	

Device number	Name	Setting range	Description
SD1248	PID control set point/ Set point: 0 to 100% • Deviation: -100 to 100%		Set the PID set point or the PID deviation (in 0.01% increments.).
SD1249	PID control measured value	0 to 100% -1	Set the PID measured value (in 0.01% increments.).
SD1250	PID control manipulated amount	-100 to 100%	Stores the PID manipulated amount (in 0.01% increments).
SD1255	PID operation control	0	Stops PID control.
3D1233	T ID operation control	1	Starts PID control.
SD1308	Second PID control set point/deviation	Set point: 0 to 100% *1 Deviation: -100 to 100%	Set the second PID set point or the second PID deviation (in 0.01% increments-1).
SD1309	Second PID control measured value	0 to 100% *1	Set the second PID measured value (in 0.01% increments <sub>*1</sub> ).
SD1310	Second PID control manipulated amount	-100 to 100%	Stores the second PID manipulated amount (in 0.01% increments).

<sup>\*1</sup> When both **Pr.934** and **Pr.935** are set to values other than "9999", the set point of SD1248 (SD1308) and the measured value of SD1249 (SD1309) are set as coefficients. The setting range is from the smaller coefficient to the larger one set in **Pr.934** and **Pr.935**. (For details of **Pr.934** and **Pr.935**, refer to the Instruction Manual of the inverter.)

## NOTE:

- Depending on the setting value of Pr.128 (Pr.753), SD1248 (SD1308) automatically switches between set point and deviation.
- When **Pr.128** (**Pr.753**) is set to deviation input "70, 71, 90, or 91", the value set as the measured value in (SD1249 (SD1309)) becomes invalid.
- If an out-of-range value is set, the maximum value (or the minimum value) of the setting range is used for operation.

# **1.15** Clearing the flash memory of the PLC function

Set Pr.498 to "9696" to clear the flash memory used for the PLC function.

Parameter	Name	Initial value	Setting range	Description
				9696: Clears the flash memory.
498	PLC function flash memory clear	0	0 to 9999	Other than 9696: Does not clear the flash memory.

- The read value of Pr.498 is always "0" regardless of the set value.
- If the file password (registered by FR Configurator2 (Developer)) of the PLC function has been forgotten, use **Pr.498** to clear the flash memory and unlock the file password.
- The flash memory can only be cleared when the PLC function is disabled (Pr.414 ="0").

## NOTE

- Executing this function clears the PLC function program and all parameters of the PLC function. Create the program and set the parameters of the PLC function again.
- It takes approximately 5 s to clear the flash memory. Do not reset the inverter or turn OFF the power for 5 s after setting **Pr.498** = "9696". If the inverter is reset or the power is turned OFF within 5 s, set **Pr.498** = "9696" again.

## 1.16 Constant scan

A constant scan time can be maintained for the sequence programs.

Configure the setting using FR Configurator2 (Developer).

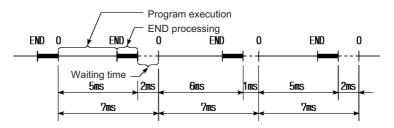
PLC parameter	Name	Initial value	Setting range
PLC RAS setting	Constant scan setting	None	0.5 ms to 2000 ms

- A waiting time is automatically set so that the sum of the execution time of the sequence programs and the waiting time is equal to the set constant scan time.
- PLC parameter: Although the setting range of PLC RAS setting is 0.5 ms to 2000 ms, the actual set constant scan time is a minimum of 10 ms. (Increments: 10 ms)

Setting value	Set constant scan time
0.5 ms to 10.0 ms	10 ms
10.5 ms to 20.0 ms	20 ms
1990.5 ms to 2000.0 ms	2000 ms

• I/O timing

After END processing is completed, a waiting time is automatically set as shown below.



Constant scan: operation when set to 7 ms.

## NOTE

- Processes other than the PLC function are executed even during the waiting time.
- The set constant scan time must meet the following condition: Set watchdog timer time > set constant scan time > maximum scan time of program.
- If a scan time is longer than the set constant scan time, "PRG.TIME OVER Error code: 5010" occurs. In this case, the constant scan setting is ignored in the operation.
- If a scan time is longer than the set watchdog timer time, a watchdog timer error is detected and execution of the sequence program stops. (Refer to page 33.)
- Only the "Execute it while waiting for constant scan setting" is enabled for the service processing setting.

  Settings other than "Execute it while waiting for constant scan setting" are ignored and no error occurs even if they are set.
- Scan time monitoring in FR Configurator2 (Developer) monitors the actual scan time excluding the waiting time for constant scan.

## **MEMO**

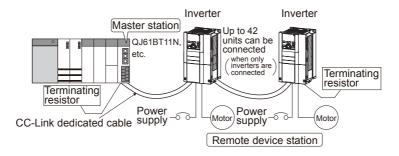
# 2 cc-Link communication

2.1	System configuration	54
2.2	CC-Link parameter	56
2.3	CC-Link I/O specifications	<b>57</b>
2.4	Buffer memory	64

## 2.1 System configuration

## 2.1.1 System configuration example

- Programmable controller side
   Mount the "CC-Link system master/local module" on the main base unit or extension base unit that uses the programmable controller CPU as the master station.
- Use the CC-Link dedicated cable to connect the CC-Link programmable controller module (master station) to the inverter.





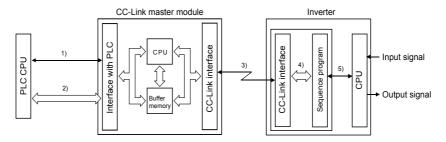
• For details on CC-Link communication wiring and the CC-Link cable, refer to the Instruction Manual of the FR-A8NC.

## 2

## 2.1.2 Function block diagram

The following function blocks explain the I/O information flow to and from the inverter in CC-Link.

- Link refresh between the master station of CC-Link system and the inverter is continuously performed at 3.5 to 18 ms (512 points).
- I/O refresh and the sequence program of the master station are executed asynchronously.
- Data read from the inverter is read from the buffer memory of the CC-Link system master/local module using FROM instruction
- Data to be written to the inverter is written to the buffer memory of the CC-Link system master/local module using TO instruction.



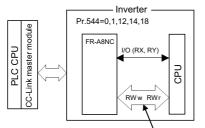
1)	CC-Link module I/O signals	I/O signals assigned to the CC-Link system master/local module. These signals are for communication between the programmable controller CPU and the CC-Link system master/local module.
2)	Reading from/writing to the buffer memory	Information that is input to the inverter can be read, and output information can be written. FROM/TO instruction of the sequence program enables reading from/writing to the buffer memory. For details on the buffer memory, refer to page 64.
3)	CC-Link dedicated cable	The PLC link start instruction is sent from the sequence program. After PLC link starts, link refresh is continuously performed asynchronously with execution of the sequence program.
4)	Sequence program	The sequence program exchanges I/O information between the CC-Link system master/local module and the inverter CPU.
5)	I/O information	The CC-Link system master/local module and the inverter CPU exchange I/O information. (When CC-Link is not used, only 5) is performed and 1) to 4) are not related to the operation.)

## NOTE

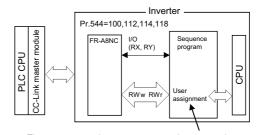
• Programs cannot be read or written via CC-Link communication.

## POINT)

• The differences between the normal CC-Link communication (**Pr.544** = "1, 2, 12, 14, or 18") and the CC-Link communication with the PLC function (**Pr.544** = "100, 112, 114, or 118") are shown below.



Parameter read/write, monitor, operation commands, etc. have been assigned in advance.



The user must assign parameters, monitors, etc. using sequence programs. Other data read/write, etc. can be assigned freely as user areas. \*Operation and speed commands have been assigned in advance.

## 2.2 CC-Link parameter

## 2.2.1 CC-Link extended setting (Pr.544)

The functions of the remote register can be extended.

Parameter	Name	Initial	Setting		Description	
number	Name	value	range	CC-Link version	Description	
			0	1	1 station occupied (FR-A5NC compatible) •1	
			1		1 station occupied	
			12 *2		1 station occupied, double	
	CC-Link extended setting	14 *2 18 *2 100 112 *2 114 *2	14 *2	2	1 station occupied, quadruple	
			18 *2		1 station occupied, octuple	
544			100	1	1 station occupied (PLC function)	
			112 *2		1 station occupied, double (PLC function)	
			114 *2	2	1 station occupied, quadruple (PLC function)	
			118 *2		1 station occupied, octuple (PLC function)	

<sup>\*1</sup> The program used for the conventional series inverter option (FR-A5NC) can be used.

<sup>\*2</sup> When using the double, quadruple, or octuple settings of the CC-Link Ver.2, station data of the master station must be set to double, quadruple, or octuple. (If the master station is CC-Link Ver.1, this setting is not available.)



The setting becomes valid after inverter reset.

## 2.3 CC-Link I/O specifications

# 2.3.1 I/O signals when one station in the CC-Link Ver.1 is occupied (Pr.544 = "100")

The number of device points available in CC-Link communication are 32 input (RX) points (16 points for the PLC function), 32 output (RY) points (16 points for the PLC function), 4 remote register (RWr) points, and 4 remote register (RWw) points.

#### **◆**Remote I/O

PLC function device number	· Signal name		PLC function device number	Remote input device number	Signal name	
X30	RYn0	Forward rotation command	Y30	RXn0	Forward rotating	
X31	RYn1	Reverse rotation command	Y31	RXn1	Reverse rotating	
X32	RYn2	High-speed operation command (Terminal RH function) *1	Y32	RXn2	Running (Terminal RUN function) *2	
X33	RYn3	Middle-speed operation command (Terminal RM function) *1	Y33	RXn3	Up to frequency (Terminal SU function) *2	
X34	RYn4	Low-speed operation command (Terminal RL function) *1	Y34	RXn4	Overload alarm (Terminal OL function) *2	
X35	RYn5	JOG operation command (Terminal JOG function)	Y35	RXn5	Instantaneous power failure (Terminal IPF function) *2	
X36	RYn6	Second function selection (Terminal RT function) *1	Y36	RXn6	Frequency detection (Terminal FU function) *2	
X37	RYn7	Current input selection (Terminal AU function) *1	Y37	RXn7	Fault (Terminal ABC1 function) *2	
X38	RYn8	Selection of automatic restart after instantaneous power failure (Terminal CS function) *1	Y38	RXn8	— (Terminal ABC2 function) *2	
X39	RYn9	Output stop	Y39	RXn9	Pr.313 assignment function (DO0) *3	
X3A	RYnA	Start self-holding selection (Terminal STOP function) *1	Y3A	RXnA	Pr.314 assignment function (DO1) *3	
X3B	RYnB	Reset (Terminal RES function)	Y3B	RXnB	Pr.315 assignment function (DO2) *3	
X3C	RYnC		Y3C	RXnC		
X3D	General-purpose remote		Y3D	RXnD	General-purpose remote	
X3E RYnE input for PLC function			Y3E	RXnE	input for PLC function	
X3F	RYnF		Y3F	RXnF		
_	RY (n + 1) 0 to RY (n + 1) 7	Reserved	_	RX (n + 1) 0 to RX (n + 1) 7	Reserved	

#### **CC-Link I/O specifications**

PLC function device number	Remote output device number	Signal name	PLC function device number	Remote input device number	Signal name	
_	RY (n + 1) 8	Unused (Initial data process completion flag)	_	RX (n + 1) 8	Unused (Initial data process request flag)	
_	RY (n + 1) 9	Unused (Initial data process request flag)	_	RX (n + 1) 9	Unused (Initial data process completion flag)	
_	RY (n + 1) A	Error reset request flag	_	RX (n + 1) A	Error status flag	
	RY (n + 1) B			RX (n + 1) B	Remote station ready	
	to RY (n + 1) F	Reserved	_	RX (n + 1) C to RX (n + 1) F	Reserved	

("n" indicates a value determined by the station number setting.)

- \*1 These signal names are initial values. The functions of input signals can be changed using **Pr.180** to **Pr.186**, **Pr.186**, and **Pr.189**. The signals of RYn0, RYn1, and RYn9 cannot be changed. Signals changed using **Pr.178**, **Pr.179**, and **Pr.187** are invalid. For details on **Pr.178** to **Pr.189**, refer to the Instruction Manual of the inverter.
- \*2 These signal names are initial values. The functions of output signals can be changed using **Pr.190 to Pr.196**. For details on **Pr.190 to Pr.196**, refer to the Instruction Manual of the inverter.
- \*3 Output signals can be assigned using **Pr.313 to Pr.315**.

  For details on the signals, refer to the description of **Pr.190 to Pr.196** in the Instruction Manual of the inverter.

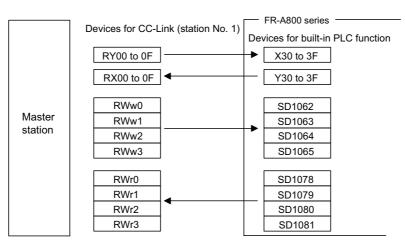
## **♦**Remote register

PLC function device number	Address	Description	
SD1062	RWwn		
SD1063	RWwn + 1	Registers for reading data received from the master	
SD1064	RWwn + 2	station.	
SD1065	RWwn + 3		

PLC function device number	Address	Description
SD1078	RWrn	
SD1079	RWrn + 1	Registers for writing data to be sent to the master
SD1080	RWrn + 2	station.
SD1081	RWrn + 3	

("n" indicates a value determined by the station number setting.)

## **♦**I/O figure



Automatically refreshed at every END.



· All remote registers are user areas and can be used freely.

# 2.3.2 I/O signals when the double setting is set in CC-Link Ver.2 (Pr.544 = "112")

The number of device points available in CC-Link communication are 32 input (RX) points (12 points for the PLC function), 32 output (RY) points (12 points for the PLC function), 4 remote register (RWr) points, and 4 remote register (RWw) points.

#### **♦**Remote I/O

PLC function device number	Remote output device number	Signal name	PLC function device number	Remote input device number	Signal name	
X30	RYn0	Forward rotation command	Y30	RXn0	Forward rotating	
X31	RYn1	Reverse rotation command	Y31	RXn1	Reverse rotating	
X32	RYn2	High-speed operation command (Terminal RH function) *1	Y32	RXn2	Running (Terminal RUN function)	
X33	RYn3	Middle-speed operation command (Terminal RM function) *1	Y33	RXn3	Up to frequency (Terminal SU function) *2	
X34	RYn4	Low-speed operation command (Terminal RL function) *1	Y34	RXn4	Overload alarm (Terminal OL function) *2	
X35	RYn5	JOG operation command (Terminal JOG function)	Y35	RXn5	Instantaneous power failure (Terminal IPF function) *2	
X36	RYn6	Second function selection (Terminal RT function) *1	Y36	RXn6	Frequency detection (Terminal FU function) *2	
X37	RYn7	Current input selection (Terminal AU function) *1	Y37	RXn7	Fault (Terminal ABC1 function) *2	
X38	RYn8	Selection of automatic restart after instantaneous power failure (Terminal CS function) *1	Y38	RXn8	— (Terminal ABC2 function) *2	
X39	RYn9	Output stop	Y39	RXn9	Pr.313 assignment function (DO0) *3	
X3A	RYnA	Start self-holding selection (Terminal STOP function) *1	Y3A	RXnA	Pr.314 assignment function (DO1) *3	
ХЗВ	RYnB	Reset (Terminal RES function)	Y3B	RXnB	Pr.315 assignment function (DO2) *3	
_	RYnC	Monitor command	_	RXnC	Monitoring	
_	RYnD	Frequency setting command (RAM)	_	RXnD	Frequency setting completion (RAM)	
_	RYnE	Frequency setting command (RAM, EEPROM)	_	RXnE	Frequency setting completion (RAM, EEPROM)	
_	RYnF	Instruction code execution request	_	RXnF	Instruction code execution completion	

#### **CC-Link I/O specifications**

PLC function device number	Remote output device number	Signal name	PLC function device number	Remote input device number	Signal name
_	RY (n + 1) 0 to RY (n + 1) 7	Reserved	_	RX (n + 1) 0 to RX (n + 1) 7	Reserved
_	RY (n + 1) 8	Unused (Initial data process completion flag)	_	_	Unused (Initial data process request flag)
_	RY (n + 1) 9	Unused (Initial data process request flag)	_	RX (n + 1) 9	Unused (Initial data process completion flag)
_	RY (n + 1) A	Error reset request flag	_	RX (n + 1) A	Error status flag
	RY (n + 1) B			RX (n + 1) B	Remote station ready
_	to RY (n + 1) F	Reserved	_	RX (n + 1) C to RX (n + 1) F	Reserved

("n" indicates a value determined by the station number setting.)

- \*1 These signal names are initial values. The functions of input signals can be changed using **Pr.180** to **Pr.186**, **Pr.188**, **and Pr.189**. The signals of RYn0, RYn1, and RYn9 cannot be changed. Signals changed using **Pr.178**, **Pr.179**, and **Pr.187** are invalid. For details on **Pr.178** to **Pr.189**, refer to the Instruction Manual of the inverter.
- \*2 These signal names are initial values. The functions of output signals can be changed using **Pr.190 to Pr.196**. For details on **Pr.190 to Pr.196**, refer to the Instruction Manual of the inverter.
- \*3 Output signals can be assigned using **Pr.313 to Pr.315**.

  For details on the signals, refer to the description of **Pr.190 to Pr.196** in the Instruction Manual of the inverter.

## **♦**Remote register

PLC function	Address	Descr	iption	PLC function	Address	Description		
device number	Address	Upper 8 bits	Lower 8 bits	device number	Address	Upper 8 bits	Lower 8 bits	
_	RWwn	Monitor code 2	Monitor code 1	_	RWrn	First monitor	value	
_	RWwn + 1	Set frequency/t command*1 (0.01 Hz incren	•	_	RWrn + 1	Second monitor value		
_	RWwn + 2	Link parameter extended setting	Instruction code		RWrn + 2	Reply code 2	Reply code 1	
_	RWwn + 3	Write data		_	RWrn + 3	Read data		
SD1062	RWwn + 4			SD1078	RWrn + 4			
SD1063	RWwn + 5	Registers for re	•	SD1079	RWrn + 5	Registers for	•	
SD1064	RWwn + 6	received from the master station.		SD1080	RWrn + 6	to be sent to the master station.		
SD1065	RWwn + 7			SD1081	RWrn + 7	1		

("n" indicates a value determined by the station number setting.)

<sup>\*1</sup> Under vector control (torque control) (**Pr.804 = "**3, 5")

# 2.3.3 I/O signals when the quadruple setting is set in CC-Link Ver.2 (Pr. 544 = "114")

The number of device points available in CC-Link communication are 32 input (RX) points (12 points for the PLC function), 32 output (RY) points (12 points for the PLC function), 8 remote register (RWr) points, and 8 remote register (RWw) points.

#### ◆Remote I/O

Same as when **Pr.544** = "112". (Refer to page 59.)

## **♦**Remote register

PLC function	Address	Descr	iption	PLC function	Address	Descr	ription	
device number	Address	Upper 8 bits	Lower 8 bits	device number	Address	Upper 8 bits	Lower 8 bits	
_	RWwn	Monitor code 2	Monitor code 1	_	RWrn	First monitor v	alue	
_	RWwn + 1	Set frequency/t command *1 (0.01 Hz incren		_	RWrn + 1	Second monitor value		
_	RWwn + 2	Link parameter extended setting	Instruction code	_	RWrn + 2	Reply code 2	Reply code 1	
_	RWwn + 3	Write data		_	RWrn + 3	Read data		
_	RWwn + 4	Monitor code 3		_	RWrn + 4	Third monitor value		
_	RWwn + 5	Monitor code 4		_	RWrn + 5	Fourth monitor value		
_	RWwn + 6	Monitor code 5		_	RWrn + 6	Fifth monitor value		
_	RWwn + 7	Monitor code 6		_	RWrn + 7	Sixth monitor value		
SD1062	RWwn + 8			SD1078	RWrn + 8			
SD1063	RWwn + 9			SD1079	RWrn + 9			
SD1064	RWwn + A			SD1080	RWrn + A			
SD1065	RWwn + B	Registers for re	•	SD1081	RWrn + B	Registers for v	vriting data to	
SD1066	RWwn + C	received from the master station.		SD1082	RWrn + C	be sent to the master station.		
SD1067	RWwn + D			SD1083	RWrn + D			
SD1068	RWwn + E			SD1084	RWrn + E			
SD1069	RWwn + F			SD1085	RWrn + F			

<sup>(&</sup>quot;n" indicates a value determined by the station number setting.)

<sup>\*1</sup> Under vector control (torque control) (**Pr.804** = "3, 5")

## 2.3.4 I/O signals when the octuple setting is set in CC-Link Ver.2 (Pr. 544 = "118")

The number of device points available in CC-Link communication are 32 input (RX) points (12 points for the PLC function), 32 output (RY) points (12 points for the PLC function), 16 remote register (RWr) points, and 16 remote register (RWw) points.

#### **◆Remote I/O**

Same as when **Pr.544** = 112. (Refer to **page 59**.)

## **♦**Remote register

PLC function	Address	Descr	ription	PLC function	Address	Desci	ription
device number	Address	Upper 8 bits	Lower 8 bits	device number	Address	Upper 8 bits	Lower 8 bits
_	RWwn	Monitor code 2	Monitor code 1	_	RWrn	First monitor v	/alue
_	RWwn + 1	Set frequency (0.01 Hz incren	nents)	_	RWrn + 1	Second monit	or value
_	RWwn + 2	Link parameter extended setting	Instruction code	_	RWrn + 2	Reply code 2	Reply code 1
_	RWwn + 3	Write data		_	RWrn + 3	Read data	
_	RWwn + 4	Monitor code 3		_	RWrn + 4	Third monitor	value
_	RWwn + 5	Monitor code 4		_	RWrn + 5	Fourth monitor value	
_	RWwn + 6	Monitor code 5		_	RWrn + 6	Fifth monitor value	
_	RWwn + 7	Monitor code 6	Monitor code 6		RWrn + 7	Sixth monitor value	
_	RWwn + 8	Fault record number	H00	_	RWrn + 8	Fault record number	Fault record data
_	RWwn + 9	PID set point (0.01% increme	ents) *1	_	RWrn + 9	Fault record (Output freque	ency)
_	RWwn + A	PID measured (0.01% increme		_	RWrn + A	Fault record (Output current)	
_	RWwn + B	PID deviation (0.01% increme	ents) *3	_	RWrn + B	Fault record (Output voltage)	
_	RWwn + C	Reserved		_	RWrn + C	Fault record (Energization time)	
_	RWwn + D			_	RWrn + D		
_	RWwn + E	H00 (Empty)		_	RWrn + E	H00 (Empty)	
_	RWwn + F			_	RWrn + F		

<sup>\*1</sup> Valid when **Pr.128** = "40, 41, 60, 61, 140, or 141".

<sup>\*2</sup> Valid when **Pr.128** = "60 or 61".

<sup>\*3</sup> Valid when **Pr.128** = "50 or 51".

PLC function		Descr	iption	PLC function		Descr	ription	
device number	Address	Upper 8 bits	Lower 8 bits	device number	Address	Upper 8 bits	Lower 8 bits	
SD1062	RWwn + 10			SD1078	RWrn + 10			
SD1063	RWwn + 11			SD1079	RWrn + 11			
SD1064	RWwn + 12			SD1080	RWrn + 12			
SD1065	RWwn + 13			SD1081	RWrn + 13			
SD1066	RWwn + 14			SD1082	RWrn + 14	1		
SD1067	RWwn + 15			SD1083	RWrn + 15			
SD1068	RWwn + 16			SD1084	RWrn + 16	]		
SD1069	RWwn + 17	Registers for received from	•	SD1085	RWrn + 17	Registers for writing data to		
SD1070	RWwn + 18	station.	i the master	SD1086	RWrn + 18	be sent to the master station.		
SD1071	RWwn + 19			SD1087	RWrn + 19			
SD1072	RWwn + 1A			SD1088	RWrn + 1A			
SD1073	RWwn + 1B			SD1089	RWrn + 1B			
SD1074	RWwn + 1C				RWrn + 1C			
SD1075	RWwn + 1D			SD1091	RWrn + 1D			
SD1076	RWwn + 1E			SD1092	RWrn + 1E			
SD1077	RWwn + 1F			SD1093	RWrn + 1F			

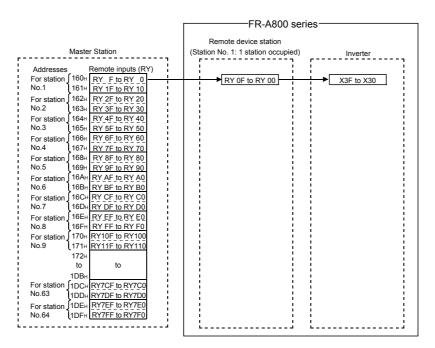
<sup>(&</sup>quot;n" indicates a value determined by the station number setting.)

## **2.4** Buffer memory

# 2.4.1 Remote output signals (master station to inverter (FR-A8NC))

- · Input statuses to the remote device station are stored.
- Each station uses two words.

(Do not use address 16n (n = 2 (X - 1) + 1, X = station number).)

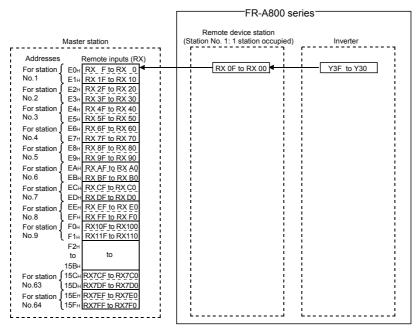


Station number	Buffer memory address						
1	160н	17	180н	33	1А0н	49	1С0н
2	162н	18	182н	34	1A2H	50	1C2H
3	164н	19	184н	35	1А4н	51	1С4н
4	166н	20	186н	36	1А6н	52	1С6н
5	168н	21	188н	37	1А8н	53	1С8н
6	16Ан	22	18Ан	38	1ААн	54	1САн
7	16Сн	23	18Сн	39	1АСн	55	1ССн
8	16Ен	24	18Ен	40	1АЕн	56	1СЕн
9	170н	25	190н	41	1В0н	57	1D0н
10	172н	26	192н	42	1В2н	58	1D2н
11	174н	27	194н	43	1В4н	59	1D4н
12	176н	28	196н	44	1В6н	60	1D6н
13	178н	29	198н	45	1В8н	61	1D8н
14	17Ан	30	19Ан	46	1ВАн	62	1DA <sub>H</sub>
15	17Сн	31	19Сн	47	1ВСн	63	1DC <sub>H</sub>
16	17Ен	32	19Ен	48	1ВЕн	64	1DE <sub>H</sub>

# 2.4.2 Remote input signals Pr.544 = "100" (inverter (FR-A8NC) to master station)

- · Input statuses from the remote device station are stored.
- · Each station uses two words.

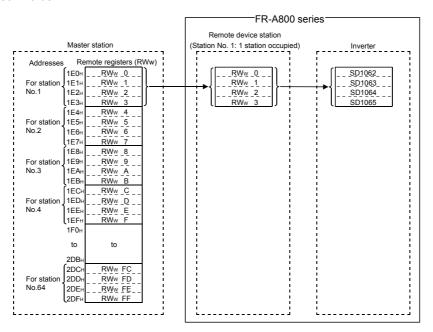
(Do not use address En (n = 2(X - 1) + 1, X = station number).)



Station number	Buffer memory address						
1	Е0н	17	100н	33	120н	49	140н
2	Е2н	18	102н	34	122н	50	142н
3	Е4н	19	104н	35	124н	51	144н
4	Е6н	20	106н	36	126н	52	146н
5	Е8н	21	108н	37	128н	53	148н
6	ЕАн	22	10Ан	38	12Ан	54	14Ан
7	ЕСн	23	10Сн	39	12Сн	55	14Сн
8	ЕЕн	24	10Ен	40	12Ен	56	14Ен
9	F0н	25	110н	41	130н	57	150н
10	F2н	26	112н	42	132н	58	152н
11	F4н	27	114н	43	134н	59	154н
12	<b>F</b> 6н	28	116н	44	136н	60	156н
13	F8 <sub>H</sub>	29	118н	45	138н	61	158н
14	FAH	30	11Ан	46	13Ан	62	15Ан
15	FСн	31	11Сн	47	13Сн	63	15Сн
16	FEH	32	11Ен	48	13Ен	64	15Ен

# 2.4.3 Remote registers Pr.544 = "100" (master station to inverter (FR-A8NC))

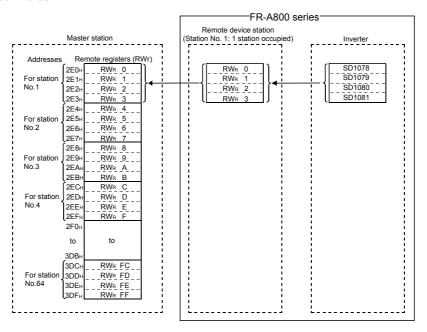
- Data to be sent to remote registers (RWw) of the remote device station are stored.
- · Each station uses four words.



Station number	Buffer memory address						
1	1E0н to 1E3н	17	220н to 223н	33	260н to 263н	49	2А0н to 2А3н
2	1Е4н to 1Е7н	18	224н to 227н	34	264н to 267н	50	2А4н to 2А7н
3	1Е8н to 1ЕВн	19	228н to 22Вн	35	268н to 26Вн	51	2A8н to 2AВн
4	1ECн to 1EFн	20	22Cн to 22Fн	36	26Сн to 26Гн	52	2ACн to 2AFн
5	1F0н to 1F3н	21	230н to 233н	37	270н to 273н	53	2В0н to 2В3н
6	1F4н to 1F7н	22	234н to 237н	38	274н to 277н	54	2В4н to 2В7н
7	1F8н to 1FBн	23	238н to 23Вн	39	278н to 27Вн	55	2B8н to 2BВн
8	1FCн to 1FFн	24	23Сн to 23Гн	40	27Сн to 27Гн	56	2BCн to 2BFн
9	200н to 203н	25	240н to 243н	41	280н to 283н	57	2C0н to 2C3н
10	204н to 207н	26	244н to 247н	42	284н to 287н	58	2С4н to 2С7н
11	208н to 20Вн	27	248н to 24Вн	43	288н to 28Вн	59	2С8н to 2СВн
12	20Cн to 20Fн	28	24Cн to 24Fн	44	28Cн to 28Fн	60	2ССн to 2СFн
13	210н to 213н	29	250н to 253н	45	290н to 293н	61	2D0н to 2D3н
14	214н to 217н	30	254н to 257н	46	294н to 297н	62	2D4н to 2D7н
15	218н to 21Вн	31	258н to 25Вн	47	298н to 29Вн	63	2D8н to 2DBн
16	21Сн to 21Fн	32	25Сн to 25Гн	48	29Сн to 29Гн	64	2DCн to 2DFн

# 2.4.4 Remote registers Pr.544 = "100" (inverter (FR-A8NC) to master station)

- Data sent from the remote registers (RWr) of the remote device station are stored.
- · Each station uses four words.



Station number	Buffer memory address						
1	2E0н to 2E3н	17	320н to 323н	33	360н to 363н	49	3A0н to 3A3н
2	2E4н to 2E7н	18	324н to 327н	34	364н to 367н	50	3А4н to 3А7н
3	2E8н to 2EВн	19	328н to 32Вн	35	368н to 36Вн	51	3А8н to 3АВн
4	2ECн to 2EFн	20	32Сн to 32Fн	36	36Сн to 36Гн	52	3ACн to 3AFн
5	2F0н to 2F3н	21	330н to 333н	37	370н to 373н	53	3В0н to 3В3н
6	2F4н to 2F7н	22	334н to 337н	38	374н to 377н	54	3В4н to 3В7н
7	2F8н to 2FBн	23	338н to 33Вн	39	378н to 37Вн	55	3В8н to 3ВВн
8	2FCн to 2FFн	24	33Сн to 33Гн	40	37Cн to 37Fн	56	3BCн to 3BFн
9	300н to 303н	25	340н to 343н	41	380н to 383н	57	3C0н to 3C3н
10	304н to 307н	26	344н to 347н	42	384н to 387н	58	3С4н to 3С7н
11	308н to 30Вн	27	348н to 34Вн	43	388н to 38Вн	59	3С8н to 3СВн
12	30Сн to 30Гн	28	34Сн to 34Гн	44	38Cн to 38Fн	60	3CCн to 3CFн
13	310н to 313н	29	350н to 353н	45	390н to 393н	61	3D0н to 3D3н
14	314н to 317н	30	354н to 357н	46	394н to 397н	62	3D4н to 3D7н
15	318н to 31Вн	31	358н to 35Вн	47	398н to 39Вн	63	3D8н to 3DBн
16	31Сн to 31Fн	32	35Сн to 35Гн	48	39Сн to 39Fн	64	3DCн to 3DFн

## **MEMO**

## 

# Sequence program

3.1	Overview	<b>70</b>
3.2	RUN/STOP operation	<b>72</b>
3.3	Program configuration	<b>72</b>
3.4	Programming language	<b>73</b>
3.5	Operation processing method of the PLC function	<b>76</b>
3.6	I/O processing method	
3.7	Scan time	<b>79</b>
3.8	Values that can be used in sequence programs	80
3.9	Explanation of devices	83
3.10	Counter C	89
3.11	Data register D	91
3.12	Special relays and special registers	92
3.13	Function list	93
3.14	RUN/STOP method of PLC function from an external source (remote RUN/	
	STOP)	
3.15	Watchdog timer (watchdog error supervision timer)	96
3.16	Self-diagnostic function	
3.17	Registering file password	98
	Output (Y) status settings when STOP status → RUN status	99
	Structure of instructions	100
3.20	Bit device processing method	101
	Handling of numerical values	
3.22	Operation error	105
3.23	Sequence instructions list	106
3.24	How to view instructions	116
3.25	Sequence instructions	117
3.26	Basic instruction (16-bit)	139
3.27	Basic instruction (32-bit)	155
	Application instructions (16-bit)	
	Application instructions (32-bit)	
	Display instruction	<b>192</b>
	e e	

## 3.1 Overview

## 3.1.1 Overview of operation

The following description is of the overview of processing performed after the inverter is turned ON and through to execution of a sequence program.

PLC function processing can roughly be classified into the following three types.

## Initial processing

Initial processing is pre-processing to execute sequence operations and is performed only once when the inverter is turned ON or reset.

- · Resets to initialize the input and output.
- Initializes the data memory (bit devices are turned OFF and word devices are set to 0).
- Performs self-diagnostic checks on such items as PLC function parameter settings and operation circuits.



• PLC function can be checked via FR Configurator2 (Developer). (Refer to the Instruction Manual of FR Configurator2.)

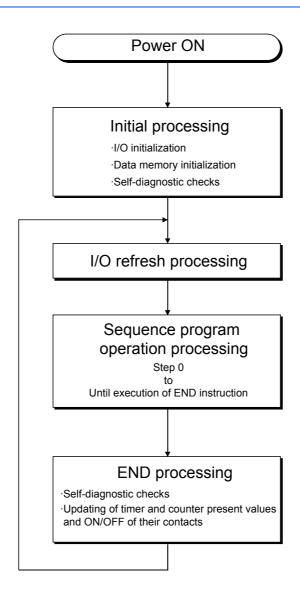
## **♦**Sequence program operation

Executes a sequence program written in the PLC function from step 0 through the END instruction.

## **♦END** processing

End processing is post-processing to end operation processing of a sequence program once and return execution to step 0 in the sequence program.

- · Performs a self-diagnostic check.
- · Updates timers and counters to the present values and turns contacts on/off.



## 3.2 RUN/STOP operation

The PLC function has two operating statuses, a RUN state and a STOP state.

The following description is of operation processing of the PLC function in each state.

### **♦RUN** state operation

In the RUN state, the PLC function operates a sequence program in the sequence of step  $0 \rightarrow END$  (FEND) instruction $\rightarrow$ step 0 repeatedly when the SQ signal is turned ON. (P.RUN is ON)

The output suspended in the STOP state is executed according to the PLC function parameter output mode set for STOP $\rightarrow$ RUN switching (refer to page 99) when the PLC function enters the RUN state.

### **♦STOP** state operation

In the STOP state, the PLC function stops operating a sequence program when the SQ signal is turned OFF or the inverter is stopped remotely. (P.RUN is OFF)

The PLC function saves the output status and turns off all output points when it enters the STOP state. Data other than the output (Y) is retained in the memory.



• The PLC function performs I/O refresh processing in both RUN and STOP states. Therefore, input/output can be monitored and tested from peripheral devices even in the STOP state.

## 3.3 Program configuration

### **♦**Program classification

Programs that can be used in the PLC function are main sequence programs only. Microcomputer programs, interrupt programs, and SFC programs cannot be used.

### **♦**Program capacity

The program capacity is the memory capacity that stores programs and is 6K steps. Set the program capacity using PLC function parameters (PLC parameters).

# 3.4 Programming language

Programming the PLC function can be performed through two methods. One uses figures and the other uses dedicated instructions.

- Programming with figures is performed by using the relay symbolic language. (Programming in FR Configurator2 (Developer) is performed in "ladder mode.")
- Programming with dedicated instructions is performed by using the logic symbolic language. (Programming that uses FR Configurator2 (Developer) is performed in "list mode.")

The same program is created with both the relay symbolic language and the logic symbolic language.

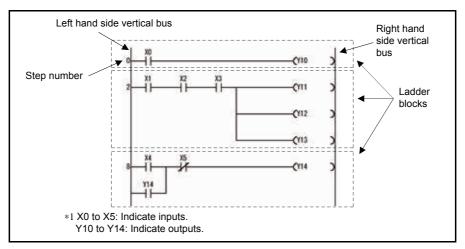
### 3.4.1 Relay symbolic language (ladder mode)

The relay symbolic language is based on the concept of relay control circuits.

This language allows programming through expressions similar to sequential circuits in relay control.

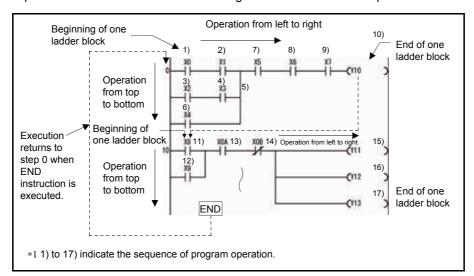
#### **♦**Ladder block

A ladder block is the smallest elements for the sequence program operation, and starts at a vertical rail on the left side and ends at the one on the right side.



### **♦** Sequence program operation method

The sequence program is operated repeatedly from the ladder block at step 0 through the END instruction. A ladder block is operated from the left vertical rail to the right vertical rail and from top down.

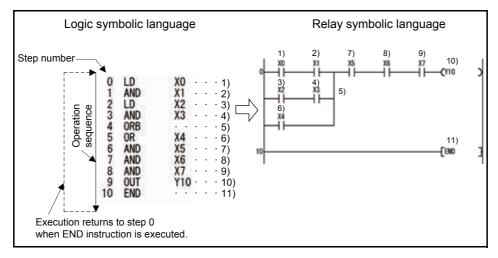


# 3.4.2 Logic symbolic language (list mode)

The logic symbolic language uses dedicated instructions for programming contacts, coils, and other devices that are expressed with symbols in the relay symbolic language.

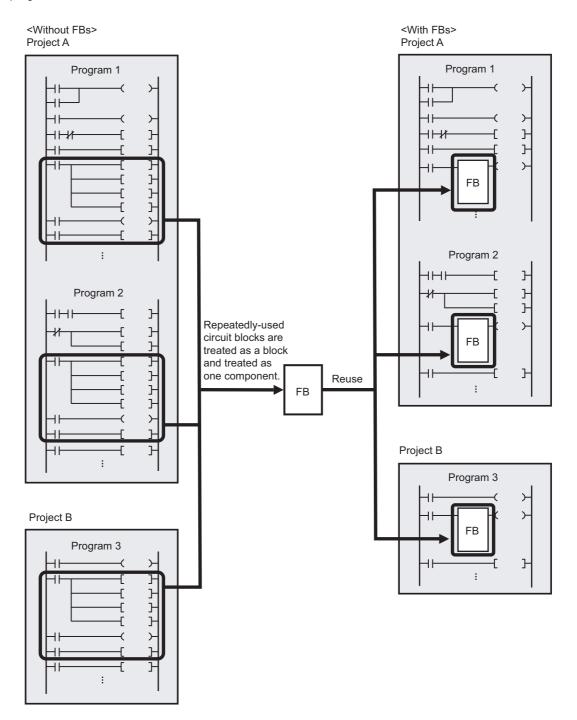
### Program operation method

The sequence program is operated in sequence from the instruction at step 0 through the END instruction. The operation is executed from step 0 again after the END instruction is executed.



# 3.4.3 Function block (FB)

A function block (FB) is a repeatedly used ladder blocks that is treated as a component so that it can also be used in other sequence programs.



# 3.5 Operation processing method of the PLC function

The PLC function uses the repetitive operation method of stored programs.

### **♦**Stored program method

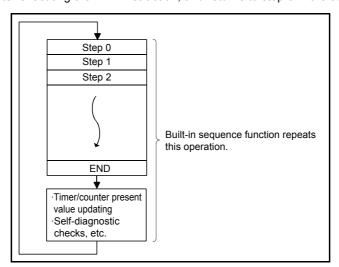
- The stored program method stores a sequence program to be operated in the internal memory in advance.
- At execution of the sequence program, the sequence program stored in the PLC function is read to the CPU by each instruction in order to execute operations, and that result is used to control each device status.

### **♦**Repetitive operation method

The repetitive operation method executes a sequence of operations repeatedly.

The PLC function executes the following process repeatedly.

- The PLC function executes a sequence program stored in the internal memory in sequence from step 0.
- The PLC function performs internal processing such as updating timers/counters to the present values and performing self-diagnostic checks after executing the END instruction, and returns to step 0 in the sequence program.





Processing from step 0 to the next step 0 or END to the next END is called a scan.

Therefore, a single scan time is the total time to process a user-created program (step 0 through END) and the time for internal processing of the PLC function.

### 3

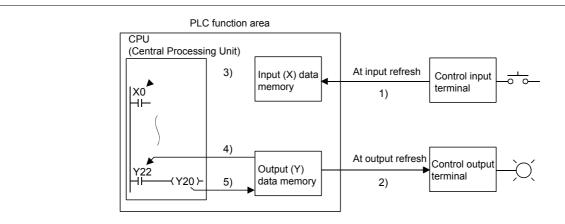
# 3.6 I/O processing method

The control method is a refresh method.

### 3.6.1 Refresh method

The refresh method stores changes in the control input terminals to the CPU input data memory in batch before every scan is executed and uses the data stored in this input data memory for operation execution.

Program operation results of the output (Y) are output to the output data memory every time, and contents stored in the output data memory are output in batch from the control output terminals after the END instruction is executed.

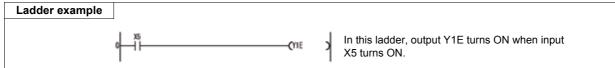


- · Input refresh
  - Input information is read 1) in batch from the PLC function area and stored in the input data memory (X) before executing step 0.
- Output refresh
  - Output information 2) stored in the output data memory (Y) is output in batch to the PLC function area before executing step 0.
- When executing a contact instruction for the input Input information is read 3) from the input data memory (X) to execute a sequence program.
- When executing a contact instruction for the output
   Output information is read 4) from the output data memory (Y) to execute a sequence program.
- When executing an OUT instruction for the output
  - Operation result of the sequence program 5) is stored in the output data memory (Y).

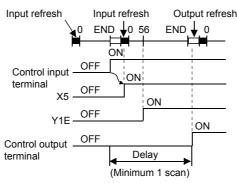
### 3.6.2 Response delay in refresh mode

The following description is about the output delay for the varied inputs.

As shown in the figures below, change in the output lags behind that in the input by up to two scans.



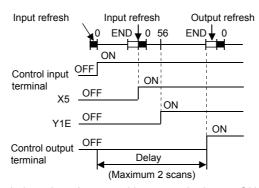
### ◆ When Y1E turns ON at the earliest timing



The Y1E output turns ON at the earliest timing when the control input terminal turns ON immediately before the refresh. In this case, X5 turns ON at input refresh, Y1E turns ON at step 0, and then the control output terminal turns ON at output refresh after END instruction execution.

Therefore, change in the control output terminal occurs one scan behind that in the control input terminal.

#### When Y1E turns ON at the latest timing



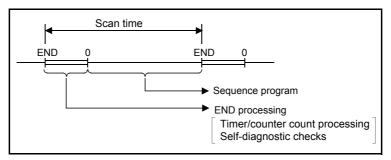
The Y1E output turns ON at the latest timing when the control input terminal turns ON immediately after the refresh. In this case, X5 turns ON at the next input refresh, Y1E turns ON at step 0, and then the control output terminal turns ON at the output refresh after END instruction execution.

Therefore, change in the control output terminal occurs two scans behind that in the control input terminal.

### 3.7 Scan time

### **♦**Scan time

Scan time is the time period from operation execution of a sequence program from step 0 until executing the next step 0. The scan time is not constant in every scan but differs according to whether instructions used are executed or not.



### **◆**Scan time check

The scan time from the END instruction to the next END instruction is measured inside the programmable controller and stored in special registers SD520 (SD521), SD524 (SD525), and SD526 (SD527).

Device number	Name	Description	
SD520		The scan time is stored at every END and is constantly updated.	
SD521	Current scan time	SD520: Stores the ms places (Stored range: 0 to 65535) SD521: Stores the µs places (Stored range: 0 to 900)	
SD524		The minimum scan time is stored at every END.	
SD525	Minimum scan time	SD524: Stores the ms places (Stored range: 0 to 65535) SD525: Stores the µs places (Stored range: 0 to 900)	
SD526		The maximum scan time is stored at every END.	
SD527	Maximum scan time	SD526: Stores the ms places (Stored range: 0 to 65535) SD527: Stores the µs places (Stored range: 0 to 900)	

<sup>·</sup> Scan time accuracy

The accuracy of the scan time observed inside the programmable controller is ±2 ms.

For example, the actual scan time is within the range of 3 ms to 7 ms when data stored in SD520 is 5.

# 3.8 Values that can be used in sequence programs

For the PLC function, data such as values and alphabetical characters are represented in two statuses, 0 (OFF) and 1 (ON). Data represented with 0 and 1 is called BIN (binary).

The PLC function can also use HEX (hexadecimal), which represents four bits of BIN data together.

The following table shows the numeric representations in BIN (binary), HEX (hexadecimal), and DEC (decimal).

DEC (decimal)	HEX (hexadecimal)	BIN (binary)
0	0	0
1	1	1
2	2	10
3	3	11
	-	
	•	
	•	
	-	
9	9	1001
10 11	A B	1010
12	С	1011 1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000
17	11	10001
l."		
1.		
1.		
1.		
47	2F	101111

## 3.8.1 **BIN** (binary)

### **♦**Binary

BIN represents a value with 0 (OFF) and 1 (ON).

In decimal notation, when a value increases from 0 to 9, a carry occurs and the next value becomes 10.

In BIN (binary) notation, a carry is generated after 0 and 1, and the next value becomes 10 (2 in decimal notation).

The following table shows the binary and decimal numeric representations.

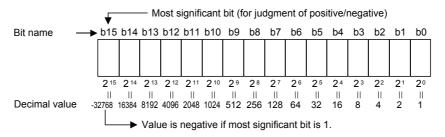
DEC (decimal)	BIN (binary)	
0	0000	
1	0001	Carry
2	0010	<b> </b> ←
3	0011	Carry
4	0100	<b> </b>
5	0101	
6	0110	
7	0111	Carry
8	1000	<b> </b>
9	1001	
10	1010	
11	1011	

### **♦**Numeric representation in BIN (binary)

Each register (such as a data register) used for the PLC function consists of 16 bits.

- · Most significant bit is 0 ... Positive
- Most significant bit is 1 ... Negative

The following figure shows the numeric representation of each register used for the PLC function.



Numerical data that can be used for the PLC function

In the numeric representation shown in the figure above, values can be represented in the range from -32768 to 32767. Therefore, each register used for the PLC function can store a value between -32768 and 32767.

# 3.8.2 HEX (hexadecimal)

### **♦**HEX

HEX represents four bits of binary data with one digit.

BIN uses 4 bits to represent the 16 values from 0 to 15.

HEX uses the letter A to represent the bit next to 9 (10) and B to represent 11, and then a carry occurs after F (15).

For the numeric representations of BIN, HEX, and DEC, refer to page 80.

### **♦**Numeric representation in HEX

Each register (such as a data register) used for the PLC function consists of 16 bits.

Therefore, the value that can be stored in each register can be represented in the range from 0 to HFFFF in HEX.

### 3

# 3.9 Explanation of devices

## 3.9.1 Device list

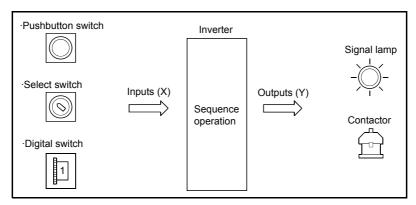
The following list shows device names and ranges that can be used for the PLC function.

Classification	Category	Device name	Number of points	Range of use	
		Input (X)	64 points	X0 to X3F <12 points installed>	HEX
		Output (Y)	64 points	Y0 to Y3F <12 points installed>	HEX
	Bit device	Internal relay (M)	128 points	M0 to M127	DEC
		Latch relay (L)	-	- (Can be set with PLC function parameters but will not latch)	-
Internal user device	•Bit device (contact/coil)	Timer (T)	16 points	T0 to T15 100 ms timer: 0.1 to 3276.7 s can be set 10 ms timer: 0.01 to 327.67 s can be set 100 ms retentive timer: 0.1 to 3276.7 s can be set	DEC
	•Word device (present value)	Counter (C)	16 points	C0 to C15 Normal counter: Setting range 1 to 32767 Interrupt program counter: Not used	DEC
	Word device	Data register (D)	256 points	D0 to D255	DEC
Internal system	Bit device	Special relay (SM)	2048 points	SM0 to SM2047 (with limited functions)	DEC
Internal system device	Word device	Special register (SD)	2048 points	SD0 to SD2047 (with limited functions)	DEC

### 3.9.2 I/O X and Y

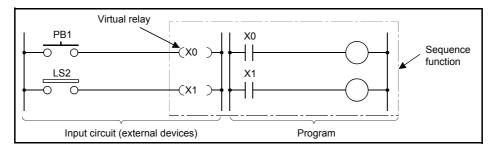
The input and output are devices that are used for communication between the inverter and external devices.

The input is given ON/OFF information externally to the control input terminals. Information is used as contacts (NO contact and NC contact) and source data of basic instructions in programs. On the other hand, the output is used to output program operation results from the control output terminals.

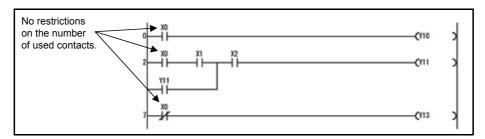


### ♦Input X

- The input allows external devices, such as push-button switches, selection switches, limit switches, and digital switches, to give commands and data to the inverter (PLC function).
- Assuming that the PLC function has internal virtual relays (Xn), the NO contacts and NC contacts of those Xn are used in programs.



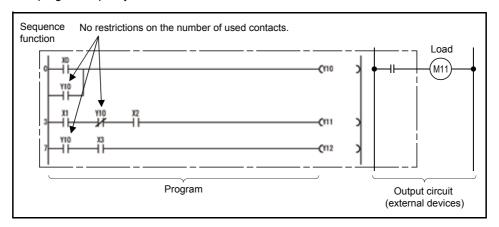
• There is no limit on the number of NO contacts and NC contacts of Xn used in a program.



When the inverter is used without connecting any external device to the control input terminal, "X" can be substituted for internal relay "M".

### **♦**Output Y

- The output performs output of program control results to external devices (signal lights, digital Human Machine Interfaces (HMI), electromagnetic switches (such as contactors and solenoids)).
- Output information can be output through a single NO contact or an equivalent device.
- There is no limit on the number of NO contacts and NC contacts of output Yn used in a program as long as it is within the range of the program capacity.



When the inverter is used without connecting the control input terminals to external devices, "Y" can be substituted for internal relay "M".

### 3.9.3 Internal relay M

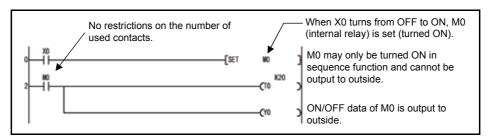
Internal relays are auxiliary relays that are used in the PLC function internally. These relays cannot latch (power failure retention).

Performing any of the following turns all internal relays OFF.

- When the power supply was turned ON
- · When resetting

There is no limit on the number of contacts (NO contact and NC contact) to be used in a program.

Use the output (Y) to output sequence program operation results.



### 3.9.4 Timer T

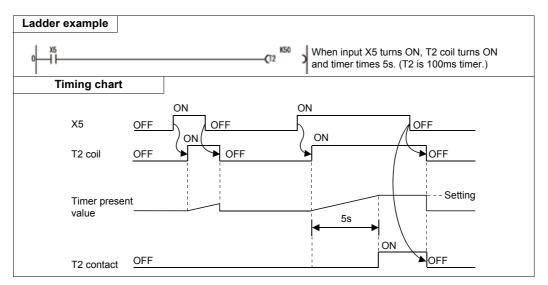
The PLC function uses up-timing timers.

The up-timing timer starts measuring a present value when the timer's coil turns ON, and then the timer's contact turns ON when the present value reaches a setting value.

# 3.9.5 100 ms timer, 10 ms timer, and 100 ms retentive timer

### ♦100 ms timer and 10 ms timer

The timer starts measuring a present value when the timer's coil turns on, and then the present value returns to 0 and the timer's contact turns OFF when the coil turns OFF.



### • NOTE

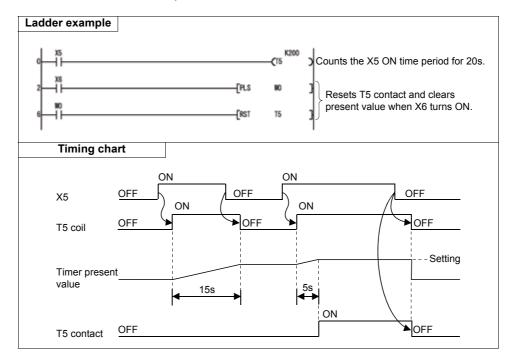
• Changing between the 100 ms timer, 10 ms timer, and 100 ms retentive timer can be done using PLC function parameters. (The initial setting is the 100 ms timer.)

#### ◆100 ms retentive timer

The 100 ms retentive timer is a timer that measures the time period during which its coil is ON.
 It starts measuring a present value when its coil turns ON and retains the present value and its contact ON/OFF status even when the coil turns OFF.

It resumes measurement from the retained present value when the coil turns ON again.

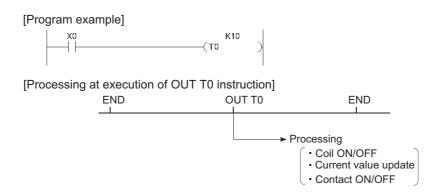
• Use the RST T instruction to clear the present value and turn OFF the contact.



## 3.9.6 Processing and accuracy of timers

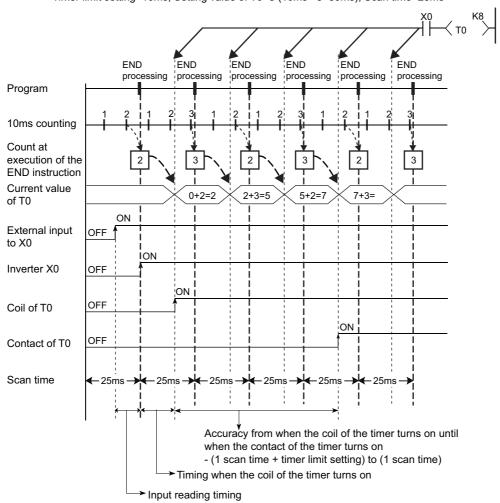
### **♦**Processing

When the OUT TD instruction is executed, the on/off switching of the timer coil, current value update, and on/off switching of the contact are performed. In the END processing, the current timer value is not updated and the contact is not turned on/off.



### **◆**Accuracy

The value obtained by the END instruction is added to the current value when the OUT TD instruction is executed. The current value is not updated while the timer coil is off even if the OUT TD instruction is executed.



Timer limit setting=10ms, Setting value of T0=8 (10ms×8=80ms), Scan time=25ms

Accuracy of the timer response that is from reading input (X) to output the data are up to "2-scan time + timer limit setting".

# 3.10 Counter C

The PLC function uses up-timing counters.

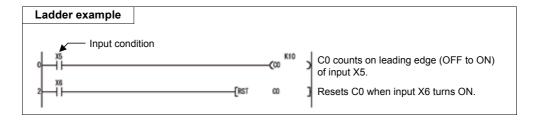
The up-timing counter turns its contact ON when the count value reaches a setting value.

### **◆**Count processing

- The counter's coil turns on/off at execution of the OUT C instruction, and then the counter's present value is updated and its contact turns ON after the END instruction is executed.
- The counter detects the coil's rise (OFF→ON) and then starts counting. Therefore, it will not start counting if the coil
  remains ON.

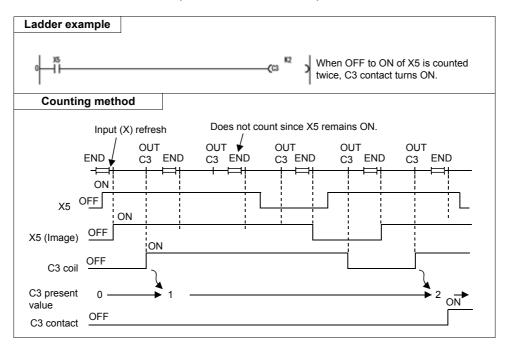
#### **◆**Counter reset

- The count value is not cleared even when the coil turns OFF. Use the RST C□ instruction to clear the count value and turn the contact OFF.
- If the counter is reset with the RST instruction, the counter's present value and contact are cleared at execution of the RST instruction.



## 3.10.1 Count process in refresh mode

The counter counts at the rise of the counter's input conditions stored at input refresh.





• For the maximum counting speed of the counter, refer to page 90.

# 3.10.2 Maximum counting speed of counter

The maximum counting speed of the counter is determined by scan time, and counting is possible only when the ON/OFF time specified in the input conditions is longer than the scan time.

Maximum counting speed Cmax = 
$$\frac{n}{100} \times \frac{1}{ts}$$
 [times/s]  $\frac{n : Duty (\%)}{ts : Scan time [s]}$ 



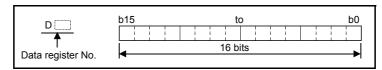
• Duty n is a ratio between the ON and OFF time of count input signals and is represented as a percentage (%).

When T1
$$\leq$$
T2  $n = \frac{T1}{T1 + T2} \times 100[\%]$   
When T1>T2  $n = \frac{T2}{T1 + T2} \times 100[\%]$   
T1 T2  
ON OFF

# 3.11 Data register D

Data registers are memories that can store numerical data (from -32768 to 32767 or from H0000 to HFFFF) within the PLC function

Each data register consists of 16 bits; therefore, data can be read and written in 16-bit increments.



- Data stored during execution of a sequence program is retained until overwritten with new data.
- Unused timers (T) and counters (C) can be substituted for data registers if the number of data registers is insufficient.

# **3.12** Special relays and special registers

Special relays and special registers are internal relays and data registers, respectively, whose applications are already determined in the PLC function.

The following are the main applications of special relays and special registers.

### **♦**Sequence operation check

The following special relays and special registers can be used for checking sequence program operations.

- · Operating status (RUN/STOP) check
- · Error detection through self-diagnostic function
- · Operation error detection
- · Scan time check

### **◆Timing contact**

The following items are special relays with different operating statuses that can be used for sequence programs.

- · Always ON/OFF flag
- RUN flag (OFF for one scan)
- · Initial processing flag (ON for one scan)



• For special relays and special registers that can be used in the PLC function, refer to page 17.

ltem	Special relay number	Application/description
Diagnostic error	SM0	Turned ON when a diagnostic error has been detected. ON status is retained even after the condition becomes normal.
Self-diagnostic error	SM1	Turned ON when an error is detected by self-diagnosis. ON status is retained even after the condition becomes normal
Common error information	SM5	When SM5 is turned ON, common error information (SD5 to SD15) is stored.
Individual error information	SM16	When SM16 is turned ON, error individual information (SD16 to SD26) is stored.
Operation error flag	SM56	Turned ON when an operation error is detected during instruction execution. ON status is retained even after the condition becomes normal.
Clock data read request	SM213	Clock data is read in BCD value to SD210 to SD213 when this relay is turned ON. The process is not executed when the relay is OFF.
Always ON	SM400	SM400 and SM401 are respectively turned ON and OFF regardless of the
Always OFF	SM401	STOP and RUN states.
ON only for one scan after RUN	SM402	SM402 and SM403 change depending on the STOP and RUN states. In the cases other than STOP:
OFF only for one scan after RUN	SM403	SM402 is ON only for one scan. SM403 is OFF only for one scan.

# 3.13 Function list

Function	Description
Remote RUN/STOP	This function executes remote RUN/STOP from an external source when the SQ signal is ON (PLC function RUN state (P.RUN is ON)).
Watchdog timer variable (10 to 2000 ms)	This is a PLC function internal timer for detecting errors in the hardware or programs. Its setting value can be changed.
Self-diagnostic function	This function diagnoses the presence of an error within the PLC function itself, and performs error detection, display and stoppage of the PLC function.
Output settings for STOP→RUN	The output (Y) status when the state changes from the STOP state to the RUN state.
Keyword registration	This setting prevents reading/writing of programs (parameter and main/sub programs) and comments.

### NOTE

• The following functions cannot be used.

Constant scan, latch (retention at power failure), PAUSE, status latch, sampling trace, step operation, clock, interrupt processing, comment, microcomputer mode, print title entry, annunciator display mode, ERROR LED priority settings.

# **3.14** RUN/STOP method of PLC function from an external source (remote RUN/STOP)

The PLC function RUN/STOP is executed by ON/OFF of the SQ signal.

Remote RUN/STOP is performed by RUN/STOP of the PLC function from an external source when the SQ signal remains in the ON state (RUN state).

### **◆**Application of remote RUN/STOP

Remote RUN/STOP can be performed by remote control using remote RUN/STOP in the following types of cases.

- · When the inverter is out of reach
- · When executing RUN/STOP for the inverter in an enclosure from an external source

### **♦**Operation at remote RUN/STOP

The operations of the sequence program that performs remote RUN/STOP are as follows.

· Remote STOP:

The sequence program is executed up to the END instruction, and enters the STOP state.

· Remote RUN:

If remote RUN is executed when the inverter has been switched to the "STOP state" by remote STOP, the state changes to RUN state again, and the sequence program is executed from step 0.

### **♦ Remote RUN/STOP method**

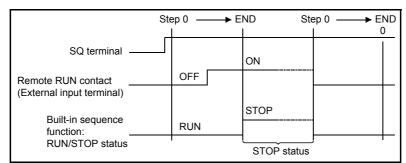
The following methods can be used for remote RUN/STOP.

• Setting using the PLC function parameters (by contact)

Remote RUN/STOP can be executed by turning the remote RUN contact OFF/ON.

For example, this can be used to STOP the PLC function at the emergency stop contact.

(The state is "RUN" when the remote RUN contact is OFF, and "STOP" when the remote RUN contact is ON.)

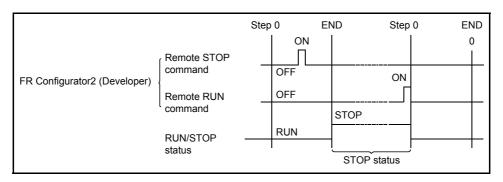




 PLC function parameter settings for the remote RUN contact X0 to X1F can be set for the remote RUN contact.
 (For details, refer to the Instruction Manual of FR Configurator2)

#### RUN/STOP method of PLC function from an external source (remote RUN/STOP)

Using FR Configurator2 (Developer)
 RUN/STOP can be performed by operating remote RUN/STOP from FR Configurator2 (Developer).
 For example, this can be used to STOP the inverter in order to rewrite the sequence program when it is installed in an out of reach location.



#### **♦** Note

Note the following points because the PLC function has priority on STOP.

- The PLC function switches to the STOP state when remote STOP is executed from any source such as the remote RUN contact or FR Configurator2 (Developer), etc.
- After switching the PLC function to the STOP state with remote STOP, all external factors (remote RUN contact, FR
  Configurator2 (Developer), etc.) that executed the remote STOP must be RUN in order to switch the PLC function back
  to the RUN state.

### • NOTE

In the RUN state, the sequence program step 0 to the END instruction are executed repeatedly.
 In the STOP state, all sequence program operations are stopped, and all outputs (Y) are OFF.

# 3.15 Watchdog timer (watchdog error supervision timer)

### ♦ Watchdog timer

The watchdog timer is a PLC function internal timer for detecting errors in the hardware or sequence program. Use FR Configurator2 (Developer). Select the [PC parameter] window, [PC RAS setting] tab, and set the watchdog timer.

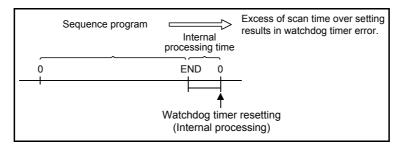
PLC function parameter	Name	Initial value	Setting range	Minimum setting increments
PLC RAS setting	WDT (Watchdog timer) setting	200 ms	10 to 2000 ms	10 ms

### **◆**Watchdog timer reset

The PLC function resets the watchdog timer before step 0 is executed (after the END processing is executed).

When the PLC function operates normally and the END instruction is executed by the sequence program within the setting value, the watchdog timer does not output the signal.

When a PLC function hardware failure occurs or if the END instruction of the scan time could not be executed well within the setting value, the watchdog timer outputs the signal.



### ◆Process when the watchdog timer reaches the setting value

If the scan time exceeds the setting value of the watchdog timer, a watchdog timer error occurs and the PLC function is as follows.

- · All outputs of the PLC function turn OFF.
- The P.RUN LED blinks.
- SM1 turns on, and an error code is stored in SD0. (Refer to page 33.)

# 3.16 Self-diagnostic function

The self-diagnostic function diagnoses the presence of an error within the PLC function itself.

### **♦** Self-diagnostic timing

The self-diagnostic is executed at power-on, at reset, when each instruction is executed and when the END instruction is executed.

- · At power-on, at reset.
  - Diagnoses whether the operation can be executed.
- · When each instruction is executed
  - An error occurs if the operation of each instruction of the sequence program could not be executed.
- When the END instruction is executed Watchdog error supervision is performed.

### Operation mode when an error is detected

There are two types of PLC function operations for when an error is detected by self-diagnosis; the operation stops or operation continues.

Even if the operation is set to continue, some errors can cause operation stop with the PLC function settings. (Refer to page 97.)

- If an operation-stop error is detected by the self diagnosis, the operation is stopped as soon as the error is detected, and all the outputs (Y) turn OFF.
- If an operation-continued error is detected, the faulty program area is skipped, and the operation continues from the next step.

### **◆**Confirmation of fault record

When an operation error occurs, SM0 (self-diagnosis error) turns ON and an error code is stored in SD0 (self-diagnosis error). Particularly when the operation is set to be continued, use in the program and to prevent a malfunction in the machine system.

For details on the fault record detected by self-diagnosis, refer to the error code list on page 198.

### 3.16.1 Operation mode when there is an operation error

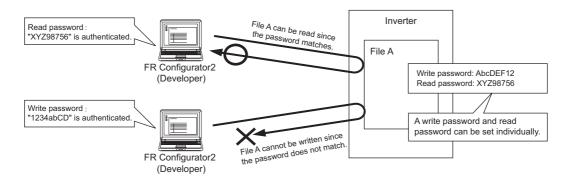
The PLC function can be set to either stop or continue operation of the sequence program when an operation error occurs. The setting of whether to stop or continue operation is set in the PLC function parameters.

• The initial status of the operation mode and PLC function status when there is an operation error are as shown in the following table.

Error definition		PLC function status				
		Operation	Operation Special relay		Self-diagnostic	P.RUN LED
		Initial status	turned ON	register for data storage	error number (SD0)	
Operation error	An error such as an attempting BCD conversion on a value that exceeds 0 to 9999 (or 0 to 9999999) occurred in the sequence program.	Continue	SM0	SD0	50	ON

# 3.17 Registering file password

This function sets write password and read password for each file stored in the inverter so that files are protected against tampering and theft by unauthorized persons. To set up a file password, select [Online] of FR Configurator2 (Developer), [Password/key word], and then [Registration/change].



### **♦**File protection timing

File protection is enabled immediately after the passwords are registered, and it is disabled immediately after the passwords are deleted.

### ◆Password target files

A password can be set to the following files.

- Program
- · Device comment
- · Initial device value
- Parameter
- · Source information

### ♦ Operations that are controlled and the number of characters

A password can be set to the following operations. A password can be 4 to 32 alphanumeric characters (capital or lowercase letters).

- · Reading files
- · Writing files
- · Reading/writing files

### ♦Online operations that require password authentication

Authentication is required to execute the following operations to password-protected files.

- · Write to PLC (data writing)
- Read from PLC (data reading)
- Online change (data writing)
- · Change TC setting value (data writing)
- · Verify with PLC (data reading)
- · Create/Change or Delete of a password (data reading and writing)
- Delete PLC data (data writing)

### • NOTE

- For the procedure and precaution on the password change, cancellation, and unlock, refer to the GX Works2 Version1
   Operating Manual (Common).
- Even when the password function (Pr.296, Pr.297) is enabled, FR Configurator2 can be used to read/write the inverter parameter settings. To use the password function and PLC function at the same time, register keywords and lock reading/ writing of the ladder program.
- The purpose of the file password is different from that of the inverter password function (Pr.296, Pr.297).

### 3

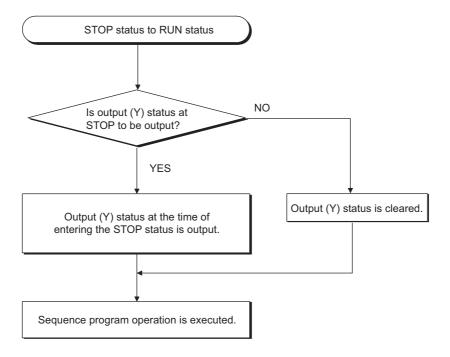
# **3.18** Output (Y) status settings when STOP status $\rightarrow$ RUN status

When changing from a state such as the RUN state to the STOP state, the RUN state output (Y) is stored in the PLC function. When changing from the STOP state to the RUN state, the PLC function parameter settings can be configured to either reoutput the output (Y) or to output after operation execution.

"Output the output (Y) status before STOP"

After outputting the output (Y) status directly before the STOP state, the sequence program operation is executed. "Clear the output (Y) (output after 1 scan)"

After clearing all outputs (Y) and executing the sequence program operation, the output (Y) is output.



# 3.19 Structure of instructions

Most of the instructions can be divided between the instruction section and device, and the applications are as follows.

- Instruction section
   Indicates the functions of the instruction.
- Device Indicates the data used by the instruction.

The structure of instructions based on a combination of the instruction section and device can be broadly divided into the following categories.

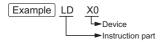
Instruction section

Instructions that do not change the device status, and mainly perform program controls.

Example END

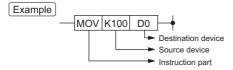
Instruction section + Device

Controls device ON/OFF, controls the execution condition according to the device ON/OFF status, and performs program branching, etc.



Instruction section
 + Source device + Destination device

Performs operation on the destination data and source data, and stores the operation result in the destination.



· Others

Combinations other than those above.

### ◆Source (S)

A source is the data used in the operation.

It is as shown below depending on the specified device.

Constant

Specifies the value used in the operation. It is a fixed value that cannot be changed when the program is being executed because it is set when the program is created.

· Bit device, word device

Specifies the device in which the data used in the operation is stored. Therefore, the data needs to be stored in the specified device before the operation is executed. The data used in the instruction can be changed by changing the data stored in the specified device while the program is being executed.

### **♦** Destination (D)

The data after operation is stored in the destination.

However, when instructions are constructed of a combination of Instruction section + Source device

Destination device , the data to be used in the operation needs to be stored in the destination before the operation.

The device in which data is to be stored must be specified in the destination.

### • NOTE

In this manual, the source and destination shall be indicated as the following abbreviations.

# 3.20 Bit device processing method

Processing methods available for when a bit device (X, Y, M) is specified are 1-bit processing, and 16-bit and 32-bit processing which are accompanied by digit specification.

# 3.20.1 1-bit processing

When sequence instructions are used, the device that is the operation processing target is 1 bit (1 point) of the bit device, and multiple bits cannot be specified.

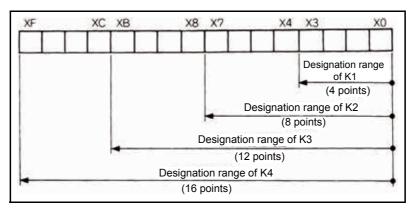
Example LD X0, OUT Y20

## 3.20.2 Digit specification processing

When basic instructions or application instructions are used, there are situations in which the bit device that is the operation processing target needs to be specified with digit specification. When the processing increment of this digit specification is a 16-bit instruction, up to 16 points can be specified in 4-point increments.

### ◆16-bit instruction: K1 to 4 (4 to 16 points)

Example Setting range from 16-bit data digit specification of X0 to F



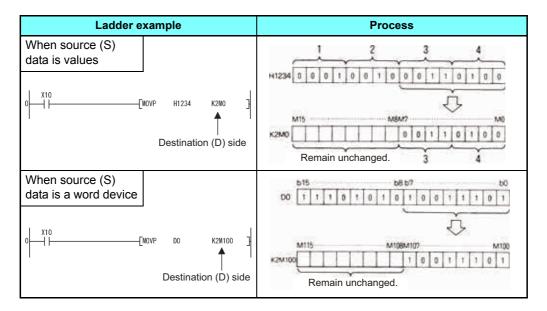
• When there is a digit specification on the source (S) side, the values that can be handled as source data are indicated in the table below.

Specified number of digits	16-bit instruction
K1 (4-point)	0 to 15
K2 (8-point)	0 to 255
K3 (12-point)	0 to 4095
K4 (16-point)	-32768 to 32767

Ladder example		Process	
16-bit instruction		×1×0 ×3 ×2 ×1 ×0	
0 X10	[MOVP K1X0 D0 ]	Change to 0.	

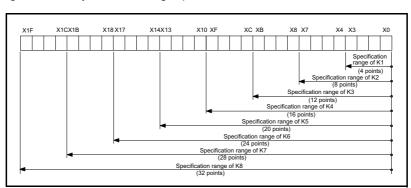
3

• When there is a digit specification on the destination (D) side, the number of points from the digit specification is applied to the destination side.



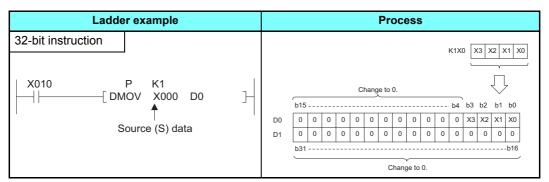
### ◆32-bit instruction: K1 to 8 (4 to 32 points)

Example Setting range X0 to 1F by 32-bit data digit specification

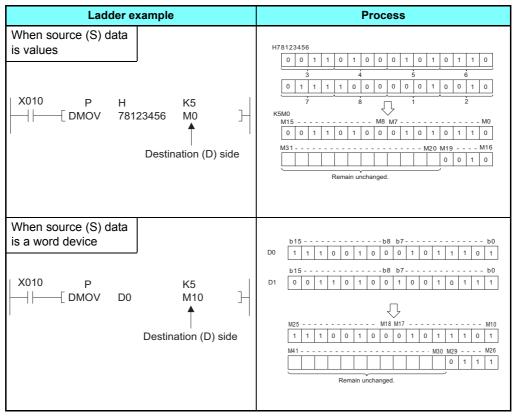


• When there is a digit specification on the source (S) side, the values that can be handled as source data are indicated in the table below.

Specified number of digits 32-bit instruction		Specified number of digits	32-bit instruction
K1 (4-point)	0 to 15	K5 (20-point)	0 to 1048575
K2 (8-point)	0 to 255	K6 (24-point)	0 to 16777215
K3 (12-point)	0 to 4095	K7 (28-point)	0 to 268435455
K4 (16-point)	0 to 65535	K8 (32-point)	-2147483648 to 2147483647



• When there is a digit specification on the destination (D) side, the number of points from the digit specification is applied to the destination side.



### • NOTE

• When 32-bit data is stored in word devices, it is stored in two consecutive word devices. An error does not occur even if the stored data exceeds the range of the corresponding device, but the content of devices outside of the range is changed. When storing data, check beforehand that the amount of devices required for storage can be allocated.

# **3.21** Handling of numerical values

The PLC function has an instruction for handling values indicated in 16 bits and 32 bits.

The most significant bit of the 16 bits and 32 bits is used to judge positive or negative. Therefore, the values that can be handled are as follows.

16-bit: -32768 to 32767

32-bit: -2147483648 to 2147483647



· Value setting method

1) Decimal

2) Hexadecimal

Decimal and hexadecimal displays are supported as shown below.

• 16 bits

Decimal display	Hexadecimal display
32767	H7FFF
to	to
5	H0005
4	H0004
3	H0003
2	H0002
1	H0001
0	H0000
-1	HFFFF
-2	HFFFE
-3	HFFFD
-4	HFFFC
-5	HFFFB
to	to
-32768	H8000

• 32 bits

Decimal display	play Hexadecimal display	
2147483647	H7FFFFFF	
to	to	
5	H00000005	
4	H00000004	
3	H00000003	
2	H00000002	
1	H00000001	
0	H0000000	
-1	HFFFFFFF	
-2	HFFFFFFE	
-3	HFFFFFFD	
-4	HFFFFFFC	
-5	HFFFFFFB	
to	to	
-2147483648	H80000000	

# 3.22 Operation error

An operation error occurs in the basic instruction in the following cases.

When an error listed in the description of each instruction occurs.



• Note that an operation error does not occur when the device specified range exceeds the corresponding device range, and data is written to devices other than the specified device.

### **♦**Error processing

If an operation error occurs when a basic instruction is executed, SM0 turns ON and an error code is stored in SD0. (SM0 remains ON even if subsequent processing is normal.)

- The error code of the instruction in which the operation error occurred when SM0 changed from OFF to ON is stored in SD0. Therefore, the content of SD0 does not change when SM0 remains ON.
- The following type of program resets SD0 and SM0.



• When an operation error occurs, the choice of whether to stop or continue sequence processing can be made in the PLC function parameter settings. For details, refer to page 97.

# **3.23** Sequence instructions list

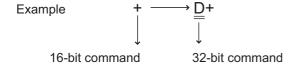
### 3.23.1 How to view the instruction list table

Classification	Instruction symbol	Symbol	Process description	Execution condition	Number of steps
Transfer	MOV	MOV S D	- (S)→(D)		- 5
Hansiei	MOVP	MOVP S D			
↑ 1)	↑ 2)	↑ 3)	↑ 4)	↑ 5)	↑ 6)

- 1): Instructions are classified according to application.
- 2): Indicates the instruction symbol used in the program.

Instruction symbols are based on 16-bit instructions.

• Change the instruction symbols for 32-bit instructions as shown below. 32-bit instructions: Add a D to the start of the instruction.

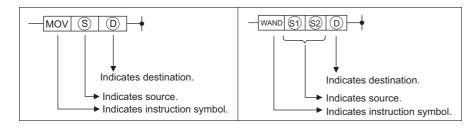


• Change the instruction symbol so that the start execution command is output when the signal is ON, as shown below. Add a P to the end of the instruction that is executed only at the leading edge of OFF to ON.



Instruction executed during ON Instruction executed only at the leading edge of OFF to ON

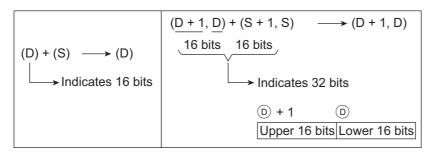
3): Indicates the symbol diagram on the ladder.



Destination: Indicates the destination of the data after operation.

Source: Stores the data before operation.

4): Indicates the process description of each instruction.



5): Details of the execution conditions for each instruction are as follows.

Symbol	Execution condition		
Blank	An always executed instruction. It is always executed irrespective of whether the instruction prior condition is ON/OFF. When the prior condition is OFF, the instruction executes the OFF process.		
	The instruction is executed only while the prior condition of the instruction is ON. When the prior condition is OFF, the instruction is not executed and is not processed.		
T	The instruction is executed one time only at the leading edge of the prior condition (OFF to ON), and is not executed or processed even when any subsequent conditions are ON.		
	The instruction is executed only while the prior condition of the instruction is OFF. When the prior condition is ON, the instruction is not executed and is not processed.		
Ţ	The instruction is executed one time only at the trailing edge of the prior condition (ON to OFF), and is not executed or processed even when any subsequent conditions are OFF.		

<sup>6):</sup> Indicates the number of program steps required for each instruction.

### 3.23.2 Sequence instructions list

Classification	Instruction symbol	Symbol	Process description	Execution Condition	Refer to page			
	LD	<del> </del>	Logical operation start (NO contact operation start)					
	LDI	<del>                                     </del>	Logical NOT operation start (NC contact operation start)					
Contact	AND	<del></del>	Logical AND (NO contact series connection)		117			
Contact	ANI	——————————————————————————————————————	Logical AND NOT (NC contact series connection)		1117			
	OR		Logical OR (NO contact parallel connection)					
	ORI	L	Logical OR NOT (NC contact parallel connection)					
	ANB		AND between logical blocks (Series connection between blocks)					
	ORB		OR between logical blocks (Parallel connection between blocks)		<b>- 119</b>			
Link	MPS	<del></del>	Stores operation results					
	MRD	MRD MRD	Reads operation results stored in MPS		121			
	MPP	MPP MPP	Reads and resets operation results stored in MPS					
	OUT	<del></del>	Device output		124			
	SET	— SET D	Device set		126			
Output	RST	RST D	Device reset		126			
	PLS	PLS D	Pulses of 1 program cycle are generated at the leading edge of input signal	<u></u>	420			
	PLF	PLF D	Pulses of 1 program cycle are generated at the trailing edge of input signal	Ę.	<b>- 128</b>			
Shift	SFT	SFT D	Davise 1 bit shift	Л	- 130			
Shiit	SFTP	SFTP D	Device 1-bit shift	<u></u>	130			
Master	МС	MC n D	Master control start		132			
control	MCR	MCR n	Master control release		132			
Program end	FEND	FEND	Termination of main program		135			
i iografii end	END	END	Always placed at the end of a sequence program to return to step 0.		135			

~	

Classification	Instruction symbol	Symbol	Process description	Execution Condition	Refer to page
	NOP	-	Non-processing. For program erasure or spacing.		
Non- processing	NOPLF	NOPLF	Non-processing.  New line instruction for printer output.		136
	PAGE	PAGE n	Ignored (Subsequent programs will be controlled from step 0 of page n)		

### 3.23.3 Basic instructions

Classification	Instruction symbol	Symbol	Process description	Execution condition	Refer to page
	LD=	= \$1 \$2			
	AND=	= (§1) (§2)	When (S1) = (S2), conductive. When (S1) $\neq$ (S2), non-conductive.		
	OR=	= \$1 \$2		П	
	LD<>	<b>├</b>		П	
	AND<>		When (S1) ≠ (S2), conductive. When (S1) = (S2), non-conductive.		
	OR<>			П	
	LD>	> (\$1) (\$2)—		Л	
	AND>	- > \$1 \$2 -	When (S1) > (S2), conductive. When (S1) $\leq$ (S2), non-conductive.	П	140
16-bit data	OR>	- 81 82		П	
comparison	LD<=	<= S1 S2 —		Л	
	AND<=	- < = S1 S2 -	When $(S1) \le (S2)$ , conductive. When $(S1) > (S2)$ , non-conductive.	П	
	OR<=	<= \(\si\) \(\si\)2		П	
	LD<	< (\$1) (\$2)		П	
	AND<	- < S1 S2 -	When (S1) < (S2), conductive. When (S1) $\geq$ (S2), non-conductive.		
	OR<	- < S1 S2 -			
	LD>=	>= (\$1) (\$2)—		П	
	AND>=	- > = S1 S2 -	When (S1) ≥ (S2), conductive. When (S1) < (S2), non-conductive.	П	
	OR>=	>= (31) (52)		П	
	LDD=	LDD= \$1 \$2	When (S1+1, S1) = (S2+1, S2),	П	
32-bit data comparison	ANDD=		conductive. When (S1+1, S1) ≠ (S2+1, S2),	П	156
Companion	ORD=	ORD= \$1 \$2	non-conductive.	П	

Classification	Instruction symbol	Symbol	Process description	Execution condition	Refer to page
	LDD<>	LDD<> §1 §2	When (S1+1, S1) ≠ (S2+1, S2),		
	ANDD<>	—ANDD (\$1) (\$2) —	conductive. When (S1+1, S1) = (S2+1, S2),	П	
	ORD<>	ORD<> (\$1) (\$2)	non-conductive.	П	
	LDD>	LDD> S1 S2	When (S1+1, S1) > (S2+1, S2),	Л	
	ANDD>	—ANDD> (\$1) (\$2) —	conductive. When $(S1+1, S1) \le (S2+1, S2)$ ,	П	
	ORD>	ORD> \$1 \$2	non-conductive.	П	
	LDD<=	LDD<= \$1 \$2	When $(S1+1, S1) \le (S2+1, S2)$ ,	Л	
32-bit data comparison	ANDD<=		conductive. When (S1+1, S1) > (S2+1, S2),	П	156
Companson	ORD<=	ORD<= \$1 \$2	non-conductive.	П	_
	LDD<	LDD< \$1 \$2	When (S1+1, S1) < (S2+1, S2),	Л	
	ANDD<	—ANDD< (\$1) (\$2) —	conductive. When $(S1+1, S1) \ge (S2+1, S2)$ ,	П	
	ORD<	ORD< (\$1) (\$2)	non-conductive.	Л	
	LDD>=	LDD>= \$1 \$2	When (S1+1, S1) ≥ (S2+1, S2),	Л	
	ANDD>=	—ANDD>= (S1) (S2) —	conductive. When (S1+1, S1) < (S2+1, S2),	П	
	ORD>=	ORD>= \$1 \$2	non-conductive.	П	
	+	+ S D	(C) + (D) + (D)	Л	
	+P		(S) + (D)→(D)	<u></u>	
	+	- + S1 S2 D	(04) + (02) + (D)	Л	
BIN 16-bit	+P		(S1) + (S2)→(D)	<u></u>	442
addition/ subtraction	-	- SD	(C) (D) (D)	Л	143
	-P		- (S) - (D)→(D)	<u></u>	-
	-	- S1 S2 D	(64) (62) (7)	П	
	-P	-P (§1) (\$2) (D)	- (S1) - (S2)→(D)	<u></u>	1

Classification	Instruction symbol	Symbol	Process description	Execution condition	Refer to page
	D+	D+ \s\_D	(D+1, D) + (S+1, S)	Л	
	D+P	D+P S D	→(D+1, D)	<u></u>	
	D+	D+ \$1 \$2 D	(S1+1, S1) + (S2+1, S2)	Л	
BIN 32-bit	D+P	D+P (51) (52) (D)	→(D+1, D)	<u></u>	450
addition/ subtraction	D-		(D+1, D) - (S+1, S)	Л	158
	D-P		→(D+1, D)	<u></u>	
	D-	D- S1 S2 D	(S1+1, S1) - (S2+1, S2)	Л	
	D-P	D-P \$1 \$2 D	→(D+1, D)	<u></u>	
	*	* \$1 \$2 D	(S1) × (S2)→(D+1, D)		
BIN 16-bit	*P	- *P \$1 \$2 D	(61) × (62) × (8 1, 8)	<u></u>	- 146
multiplication/ division	/	- / (S) (S) (D)	(S1)/(S2)→quotient(D),	Л	
	/P	/P (51) (52) (D)	remainder (D+1)	<u></u>	
	D*	D* \$1 \$2 D	(S1+1, S1) × (S2+1, S2)	Л	
BIN 32-bit	D*P	D*P \$1 \$2 D	→(D+3, D+2, D+1, D)	<u></u>	161
multiplication/ division	D/	D/ \$1 \$2 D	(S1+1, S1)/(S2+1, S2) →quotient (D+1, D),	Л	
	D/P	D/P \$1 \$2 D	remainder (D+3, D+2)	<u></u>	
	INC	INC D	(D)+1→(D)	Л	
BIN 16-bit data	INCP	INCP D		<u></u>	149
increment/ decrement	DEC	DEC D	(D) - 1→(D)	Л	
	DECP	DECP D		<u></u>	
	DINC	DINC D	- (D+1, D)+1→(D+1, D)	Л	
BIN 32-bit data	DINCP	DINCP D	(- ', -, ' ', -', -')	<u></u>	164
increment/ decrement	DDEC	DDEC D	(D+1, D) - 1→(D+1, D)	Л	164
	DDECP	DDECP D	, , , , , , , , , , , , , , , , , , , ,		

,	-	
۲	r	

Classification	Instruction symbol	Symbol	Process description	Execution condition	Refer to page	
16-bit	MOV	MOV S D	(S)→(D)	П	- 150	
transfer	MOVP	MOVP S D	(3)->(D)	<u></u>	130	
32-bit	DMOV	— DMOV S D	(C.1.4. C) . (D.1.4. D.)	П	165	
transfer	DMOVP	— DMOVP S D	(S+1, S)→(D+1, D)	<u> </u>	105	
BIN 16-bit 2's	NEG	NEG D	0 (D) (D)	П	454	
complement	NEGP	NEGP D	0 - (D)→(D)	<u></u>	<b>— 151</b>	
BIN 32-bit 2's	DNEG	DNEG D	0 (D.11 D) (D.11 D)	П	167	
complement	DNEGP	DNEGP D	0 - (D+1, D)→(D+1, D)	<u></u>		
BIN 16-bit BCD	BCD	BCD S D	BCD conversions (S) → (D)	П	153	
conversions	BCDP	BCDP S D	(S) → (D) ↑ BIN (0 to 9999)	<u></u>	153	
BIN 32-bit BCD	DBCD	— DBCD S D	BCD conversions (S+1, S) (D+1, D)	П	168	
conversions	DBCDP	—DBCDP S D	BIN (0 to 99999999)	<u></u>	100	
BIN 16-bit	BIN	BIN S D	BIN conversions (S) (D)	П	154	
conversions	BINP	BINP S D	(S) → (D)	<u></u>	154	
BIN 32-bit BIN	DBIN	— DBIN S D BIN conversions		П	160	
conversions	DBINP	— DBINP S D	(S+1, S) ——→(D+1, D) BCD (0 to 99999999)	<u></u>	<b>– 169</b>	

### 3.23.4 Application instructions

Classification	Instruction symbol	Symbol	Process description	Execution condition	Refer to page
	WAND	WAND S D	(D) AND (C) - (D)	П	
BIN 16-bit	WANDP	WANDP S D	(D) AND (S)→(D)	<u> </u>	
logical AND	WAND		(04) AND (00) (D)	Л	171
	WANDP		(S1) AND (S2)→(D)	<u></u>	
	DAND	— DAND S D	(D.11, D) AND (C.11, C) ((D.11, D)	Л	
BIN 32-bit	DANDP	— DANDP S D	(D+1, D) AND (S+1, S)→(D+1, D)	<u></u>	181
logical AND	DAND	DAND S1 S2 D	(S1+1, S1) AND (S2+1, S2)	Л	101
	DANDP		→(D+1, D)	<u> </u>	
	WOR	-WOR S D	(D) OR (S)→(D)	П	
BIN 16-bit logical OR	WORP	WORP S D		<u></u>	- 174
	WOR		(S1) OR (S2)→(D)	П	
	WORP		(31) OK (32)→(D)	<u></u>	
	DOR	—DOR S D	(D+1, D) OR (S+1, S)→(D+1, D)	П	184
BIN 32-bit	DORP	DORP S D	(D+1, D) OR (G+1, G)—(D+1, D)	<u></u>	
logical OR	DOR	DOR \$1 \$2 D	(S1+1, S1) OR (S2+1, S2)	П	
	DORP		→(D+1, D)		
	WXOR	-wxor s D	(D) XOR (S)→(D)	П	
BIN 16-bit	WXORP	-WXORP S D	(b) XOI((s)(b)	<u></u>	176
exclusive OR	WXOR		(S1) XOR (S2)→(D)		170
	WXORP		(01) XOK (02)-7(D)	<u></u>	
	DXOR	—DXOR S D	(D+1, D) XOR (S+1, S)→(D+1, D)		
BIN 32-bit	DXORP	-DXORP S D	(5·1, b) NON (6·1, 6)-(b·1, b)	<u></u>	187
exclusive OR	DXOR	DXOR \$1 \$2 D	(S1+1, S1) XOR (S2+1, S2)		
	DXORP		→(D+1, D)		

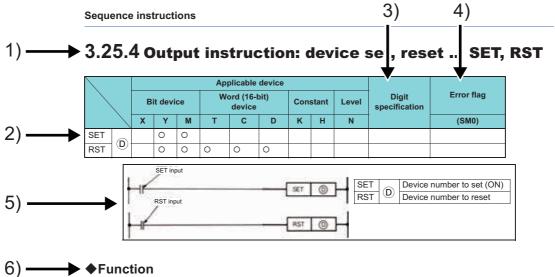
Classification	Instruction symbol	Symbol	Process description	Execution condition	Refer to page	
	WXNR	WXNR S D	$\overline{(D) \text{ XOR (S)}} \rightarrow (D)$		178	
BIN 16-bit exclusive	WXNRP	-WXNRP S D		<u></u>		
NOR	WXNR		(S1) XOR (S2)→(D)	П		
	WXNRP		(31) NON (32)→(D)	<u></u>		
	DXNR	DXNR S D	(D+1, D) XOR (S+1, S)	П		
BIN 32-bit exclusive	DXNRP	DXNRP S D	→(D+1, D)	<u></u>	190	
NOR	DXNR		(S1+1, S1) XOR (S2+1, S2)	П	190	
	DXNRP		→(D+1, D)	<u></u>		

### 3.23.5 Display instruction

Classification	Instruction symbol	Symbol	Process description	Execution condition	Refer to page
Character string data			Character string specified with (S) is transferred to devices after that specified		192
transfer	\$MOVP		with (D).	<u></u>	132
	G.PRR	G.PRR n S D	Data stored in the device specified with (S) is sent to PU.		
Character string output	GP.PRR	GP.PRR n S D	(n, (D) are dummies)	<u></u>	193
	UMSG	UMSG S	Data stored in the device specified with (S) is sent to PU.	Л	

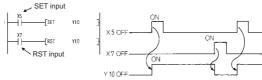
### 3.24 How to view instructions

The subsequent descriptions shall be in the following format.



SET

- · When SET input is turned ON, the specified device turns ON.
- · A device that was turned ON remains ON even if the SET input is turned OFF. It can be turned OFF with the RST instruction.



• When the SET input is OFF, the device state does not change

### ◆Execution condition

The SET and RST instructions are executed for each scan.

#### • NOTE

The number of steps is three when the following devices are used SET instruction ... Special relay (M) RST instruction ... Special relay (M), all word devices

### → Program example

SET RST

• Program that sets Y8 (ON) when X8 turns ON, and resets Y8 (OFF) when X9 turns ON.



### Description

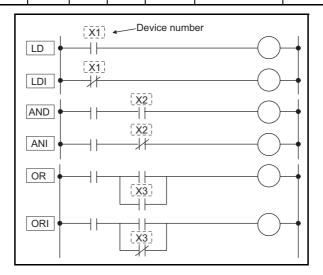
- 1) Indicates the item number, instruction overview and instruction symbol.
- 2) Devices that can be used by instructions are marked with O.
- 3) When a bit device is used, this indicates digit specifications that can be set in the instruction required by the digit specification.
- 4) When an operation error occurs, instructions whose error flags turn ON are marked with O.
- 5) Indicates the format in ladder mode.
- 6) Describes the instruction.
- 7) Indicates the execution conditions of the instruction.
- 8) Indicates the program example in ladder mode and list mode.

### 3.25 Sequence instructions

Sequence instructions are used in relay control circuits, etc.

# 3.25.1 Contact instruction: operation start, series connection, parallel connection ... LD, LDI, AND, ANI, OR, ORI

Applicable device											
Е	Bit devi	ce	W	ord (16-l device	oit)	Constant Lev		Constant Level		Digit specification	Error flag
Х	Υ	M	Т	С	D	K	Н	N		(SM0)	
0	0	0	0	0							



### **◆**Function

LD, LDI

• LD is the NO contact operation start instruction and LDI is the NC contact operation start instruction. They import the ON/ OFF information of the specified device as the operation result.

AND. ANI

- AND is the NO contact series connection instruction, and ANI is the NC contact series connection. They import the ON/OFF
  information of the specified device to perform the AND operation with the operation result up to this point and take the
  resulting value as the operation result.
- There are no usage limitations on AND and ANI, but the following limitations exist in ladder mode.
  - · Write: When AND or ANI are connected in a series, a ladder with a maximum of 21 steps can be created.
  - Read: When AND or ANI are connected in a series, a ladder with a maximum of 24 steps can be displayed. If the ladder exceeds 24 steps, steps up to the 24th step are displayed.

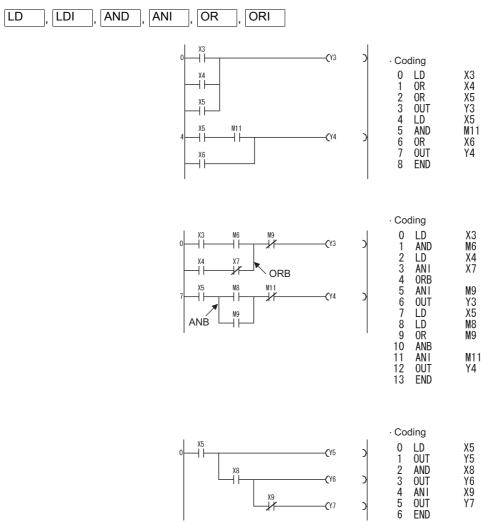
OR, ORI

- OR is the parallel connection instruction with a single NO contact, and ORI is the parallel connection instruction with a single NC contact. They import the ON/OFF information of the specified device to perform the OR operation with the operation result up to this point and take the resulting value as the operation result.
- There are no usage limitations on OR and ORI, but the following limitations exist in ladder mode.
  - Write: A ladder that contains 23 OR or ORI continually linked can be created.
  - Read: A ladder that contains 23 OR or ORI continually linked can be displayed. Ladders that contains more than 23 OR
    or ORI cannot be displayed correctly.

### **◆**Execution condition

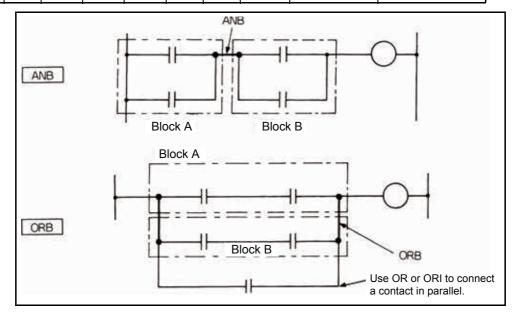
The instructions explained in this section are executed every scan irrespective of the device ON/OFF status or the operation result directly prior to the execution.

### **♦**Program example



## 3.25.2 Association instruction: ladder block series connection, parallel connection ... ANB, ORB

			App								
Bit device			Wo	ord (16-l device	bit)	Constant		Level	Digit specification	Error flag	
Х	Υ	M	T	С	D	КН		N		(SM0)	



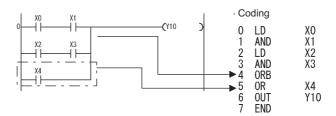
### **♦**Function

#### ANB

- The AND operation is executed for the A block and B block to produce the operation result.
- The AND symbol is a connection symbol, not a contact symbol.
- A maximum of 7 instructions (8 blocks) can be written continuously for ANB.
   If more ANB instructions than indicated above are written continuously, the PLC function cannot operate correctly.

#### ORB

- The OR operation is executed for the A block and B block to produce the operation result.
- ORB creates parallel connections with ladder blocks that have two or more contacts. Use OR or ORI to create parallel connections with ladder blocks that only have one contact, there is no need to use ORB.

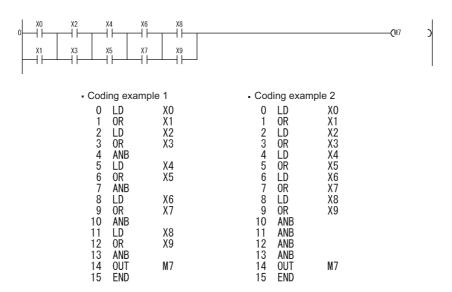


- The ORB symbol is a connection symbol, not a contact symbol.
- A maximum of 7 instructions (8 blocks) can be written continuously for ORB.
   If more ORB than indicated above are written continuously, the PLC function cannot operate correctly.

### **♦**Program example

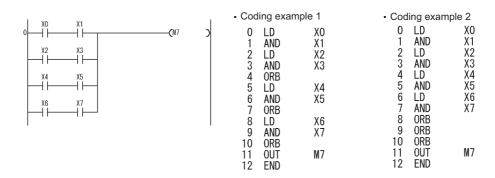
ANB

The following two types of program coding are available for continuous series connection of ladder blocks. However, the coding example 1 should be applied.



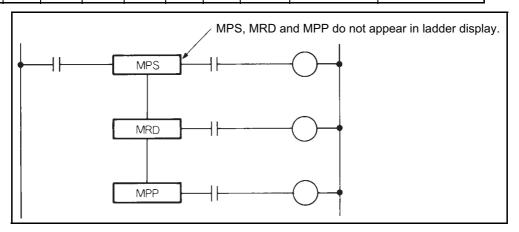
ORB

The following two types of program coding are available for continuous parallel connection of ladder blocks. However, the coding example 1 should be applied.



# 3.25.3 Association instruction: operation results, push, read, pop ... MPS, MRD, MPP

			App	Digit specification						
Bit device			Word (16-bit) device			Constant		Level	Error flag	
Х	Υ	M	Т	С	D	КН		N		(SM0)



### **♦**Function

#### MPS

- The operation result (ON/OFF) directly prior to the MPS instruction is stored.
- The MPS instruction can be used continuously up to 12 times.
   However, it can only be used up to 11 times when created with ladder mode.
   When the MPP instruction is used during the program, the number of the MPS instructions to be used is reduced by 1.

#### MRD

• The operation result stored by the MPS instruction is read, and the operation of the following step is continued by this operation result.

#### MPP

- The operation result stored by the MPS instruction is read, and the operation of the following step is continued by this operation result.
- The operation result stored by the MPS instruction is cleared.

3

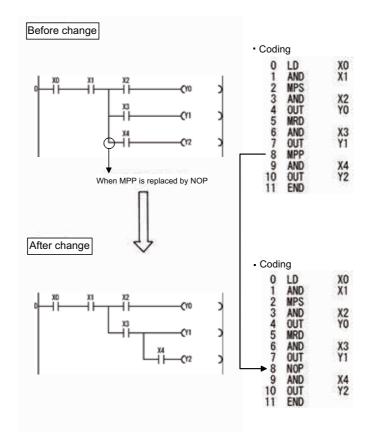


• The ladder is as shown below when MPS, MRD and MPP are used and when they are not used.

Ladder when MPS, MRD and MPP are used	Ladder when MPS, MRD and MPP are not used					
0 X0 X1 X2 (Y10 )  X3 X4 (Y11 )  X5 (Y12 )	0					

• Be sure to match the number of MPS and MPP instructions. The following will occur if the number used is different.

If the number of MPS instructions is greater, the ladder is changed and the PLC function executes operations on the changed ladder.

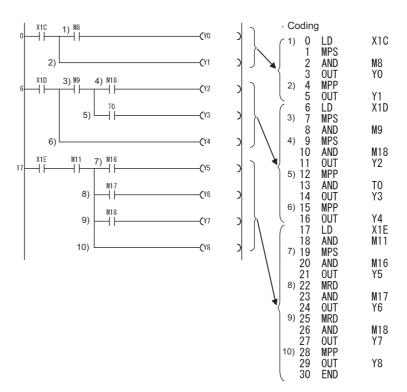


If the number of MPP instructions is greater, a ladder creation failure occurs for that ladder block and the PLC function cannot execute operations normally.

### **♦**Program example

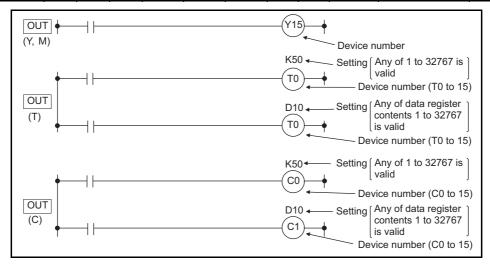
MPS MRD MPP

• Program using MPS, MRD and MPP



## 3.25.4 Output instruction: bit device, timer, counter ... OUT

					App	licable	device					
		Bit device			W	Word (16-bit) device			stant	Level	Digit specification	Error flag
		Х	Υ	M	Т	С	D	K	Н	N		(SM0)
Bit device			0	0								
	Device				0							
Timer	Setting value						0	0				
	Device					0						
Counter	Setting value						0	0				



### **♦**Function

OUT (Y, M)

• Operation results up to the OUT instruction are output to the specified device.

		OUT instruction									
Operation result	Coil	Contact									
	Coll	NO contact	NC contact								
OFF	OFF	Non-conduction	Conduction								
ON	ON	Conduction	Non-conduction								



<sup>•</sup> The OUT instruction repuires 3 steps when a special relay (M) is used.

• When the operation result up to the OUT instruction is ON, the timer coil turns ON and counts up the value up to the setting value. When the coil's timer reaches the setting value (counted value ≥ setting value), the contacts enter the following states:

NO contact	
NC contact	Non-conduction

• When the operation result up to the OUT instruction changes from ON to OFF, the following occurs.

Type of timer	Timer	Present value	Before th	e time up	After the time up		
Type of times	coil	of timer	NO contact	NC contact	NO contact	NC contact	
100 ms timer	OFF	0	Non-	Conduction	Non-	Conduction	
10 ms timer	OFF	0	conduction	Conduction	conduction	Conduction	
100 ms retentive timer	entive timer OFF Retention of present value Non-conduction		Non- conduction	Conduction	Conduction	Non- conduction	

- After the timer reaches the setting value, the state of the retentive timer contact does not change until the RST instruction is executed.
- Negative values (-32768 to -1) cannot be set for the setting value.
- The timer's setting value becomes infinity, if its setting value is set to "0". For this reason, the timer does not time up.
- For details on the counting method of the timer, refer to page 87.

OUT (C)

• If the operation result changes from OFF to ON up to the OUT instruction, +1 is added to the present value (count value), and when counting reaches the setting value (present value = setting value), the contact is as follows.

NO contact	
NC contact	Non-conduction

- The timer will not start counting if the operation result remains ON. (Count input does not need to be converted into pulse form.)
- After a count-up, the count value and state of the contact do not change until the RST instruction is executed.
- Negative values (-32768 to -1) cannot be set for the setting value. Furthermore, if the setting value is 0, the same process as for a setting value of 1 is performed.
- For details on the counting method of the counter, refer to page 89.

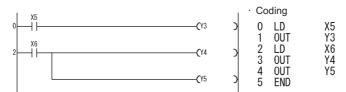
### **◆**Execution condition

The OUT instruction is executed for each scan irrespective of the operation result up to the instruction.

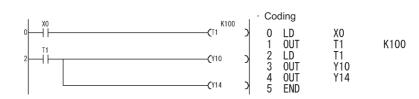
### **♦**Program example

OUT

• Program that outputs to the output module.

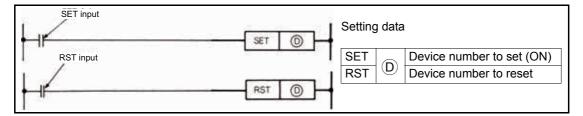


• Program that turns X0 ON, and then 10 seconds later turns Y10 and Y14 ON.



### 3.25.5 Output instruction: device set, reset ... SET, RST

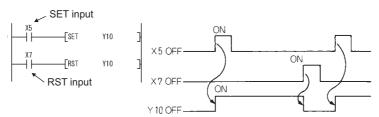
				Applicable device								
		В	Bit device Word (16-bit) Constant Leve		Level	Digit specification	Error flag					
		Х	Υ	M	Т	С	D	K	Н	N		(SM0)
SET			0	0								
RST	(D)		0	0	0	0	0					



### **♦**Function

SET

- When SET input is turned ON, the specified device turns ON.
- A device that was turned ON remains ON even if the SET input is turned OFF. It can be turned OFF with the RST instruction.



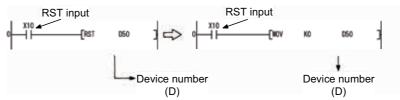
· When the SET input is OFF, the device state does not change.

RST

· When the RST input is turned ON, the specified device is as follows.

Device	Status
Y, M	The coil and contact are turned OFF.
T, C	The present value is set to 0, and the coil and contact are turned OFF.
D	The content is set to 0.

- When the RST input is OFF, the device state does not change.
- The RST (D) function is identical to the following ladder.



### **◆**Execution condition

The SET and RST instructions are executed for each scan.

### NOTE

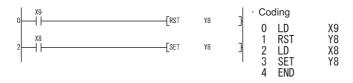
• The number of steps is three when the following devices are used.

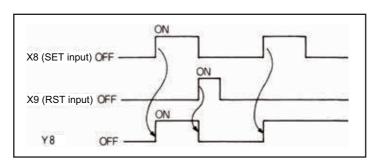
SET instruction ... Special relay (M)

RST instruction ... Special relay (M), all word devices

### **♦**Program example

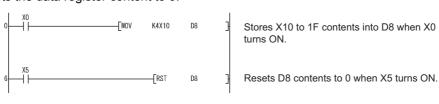
• Program that sets Y8 (ON) when X8 turns ON, and resets Y8 (OFF) when X9 turns ON.





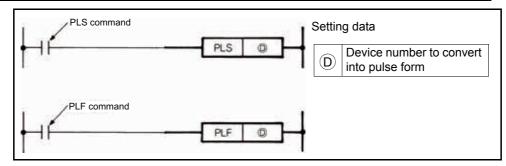
Operations of SET and RST instructions

• Program that sets the data register content to 0.



## 3.25.6 Output instruction: rising, falling differential output ... PLS, PLF

				Арр	licable	device					
	В	it devi	се		ord (16-l device	oit)	Cons	Constant Level		Digit specification	Error flag
	Х	Υ	M	T	С	D	K	Н	N		(SM0)
(D)		0	0								



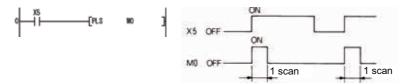
### **◆**Function

PLS

• The specified device is turned ON when the PLS command changes from OFF to ON, and OFF at all times other than when the PLS command is turned OFF to ON (OFF to OFF, ON to ON, ON to OFF).

If there is one PLS instruction for the device specified by  $\bigcirc$  during one scan, the specified device is turned ON for one scan.

Do not execute the PLS instruction of the same device multiple times during one scan.



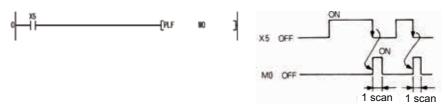
· After the PLS instruction is executed, it will not be executed again even if STOP is executed and RUN is executed again.

PLF

• The specified device is turned ON when the PLF command changes from ON to OFF, and OFF at all times other than when the PLS command is turned ON to OFF (OFF to OFF, OFF to ON, ON to ON).

If there is one PLF instruction for the device specified by  $\bigcirc$  during one scan, the specified device is turned ON for one scan.

Do not execute the PLF instruction of the same device multiple times during one scan.

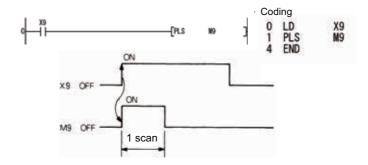


• After the PLF instruction is executed, it will not be executed again even if STOP is executed and RUN is executed again.

### **♦**Program example

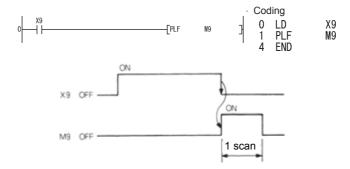
PLS

Program that executes the PLS instruction when X9 is turned ON.



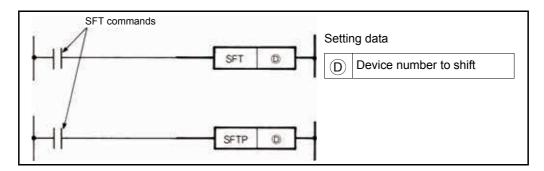
PLF

Program that executes the PLF instruction when X9 is turned OFF.



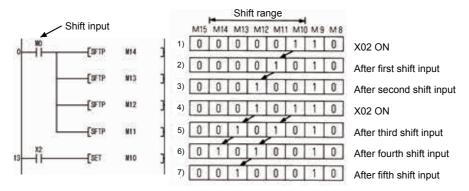
### 3.25.7 Shift instruction: bit device shift ... SFT, SFTP

				App	licable	device							
	В	it devi	се		ord (16-k device	oit)	Cons	Constant L		Constant Level		Level Digit specification	Error flag
	Х	Υ	M	T	С	D	K	Н	N		(SM0)		
(D)		0	0										



### **♦**Function

- The ON/OFF status is shifted to the device immediately prior to the device specified with ①, and the prior device is set to OFF.
- Use the SET instruction to turn ON the start device to be shifted.
- When SFT or SFTP are used continuously, start programming from larger device numbers.



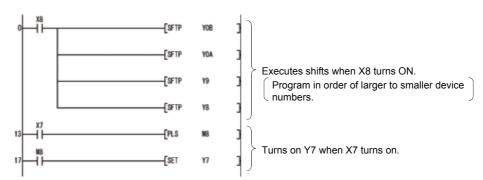
\*1 At M8 to 15, 1 indicates ON and 0 indicates OFF.

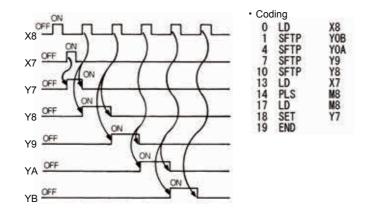
#### 3

### **♦**Program example

SFT

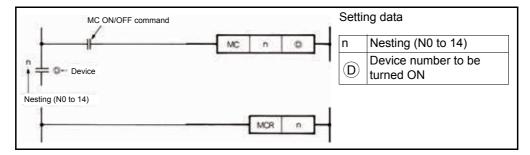
• Program that shifts Y7 to YB when X8 turns ON.





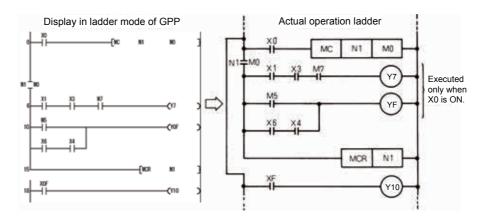
## 3.25.8 Master control instruction: master control set, reset ... MC, MCR

		Applicable device												
	Ві	Word (16-bit) Const		stant	Level	Digit Error flag specification								
	Х	Υ	M	T	С	D	K	Н	N		(SM0)			
n									0					
(D)		0	0											



#### Function

• This instruction is for creating an efficient ladder switching sequence program by opening and closing the common bus of the ladder using the master control instruction. The following ladders use the master control.



#### MC

- When the MC ON/OFF instruction is turned ON at the start of master control, the operation results from MC to MCR are as per the instruction (ladder).
- Even when the MC instruction is OFF, the scan time is not shortened because the scan is executed between the MC instruction and MCR instruction.

Also, when the MC instruction is OFF, the operation results of MC and MCR are as follows.

100 ms, 10 ms timer	The count value becomes 0, and both coil and contact turn OFF.				
100 ms retentive timer, counter	The coil turns OFF, but the count value and contact both retain the current state.				
Device currently under OUT instruction	All turn OFF.				
SET, RST or SFT, or devices currently executing basic or application instructions	Retain the current state.				

- The MC instruction can use the same nesting (N) number any number of times by changing the  $\bigcirc$  device.
- When the MC instruction is ON, the coil of the device specified by  $\bigcirc$  is ON. Furthermore, as a double coil occurs when the same device is used by instructions such as the OUT instruction, do not use a device specified by  $\bigcirc$  when it is executing a different instruction.

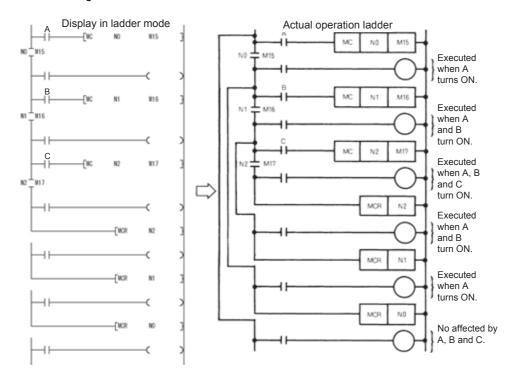
#### MCR

- The master control release instruction that indicates the end of the master control range.
- Do not add a contact instruction to the front of the MCR instruction.

### ◆Program example

• The master control instruction can be used as a nesting structure. Each master control area is separated by nesting (N). NO to N14 can be used for nesting.

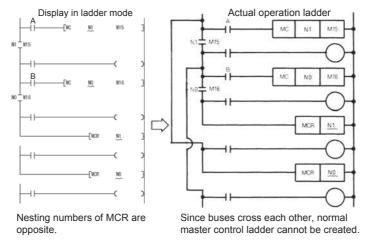
A ladder in which the program execution conditions are limited successively can be created by using the nesting structure. A ladder that uses a nesting structure is as follows.



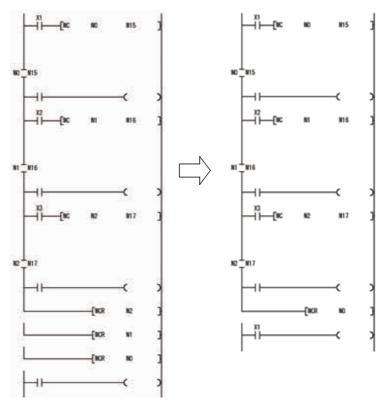
### **Sequence instructions**

Take note of the following points when using a nesting structure.

• A maximum of 15 nests (N0 to 14) can be used. When applying nesting, MC uses numbers from the lower nesting (N) numbers upwards, whereas MCR uses numbers from the higher numbers downwards. If the order is reversed, the structure is not a nesting structure and the PLC function cannot operate normally.



• If the nesting structure consists of MCR instructions collected in one location, the single lowest nesting (N) number can be used to terminate all master controls.



### 3

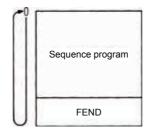
## 3.25.9 Termination instruction: termination of main program ... FEND

	Applicable device									
В	it devi	се	Word (16-bit) device		Constant		Level Digit specification		Error flag	
Х	Υ	M	T	С	D	K	Н	N		(SM0)

1	 	
ľ	[ LEIND ]	T

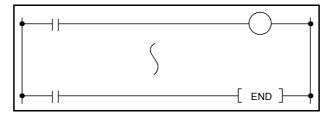
### **♦**Function

- Execution of the FEND instruction will cause the CPU module to terminate the program it was executing.
- Even sequence programs following the FEND instruction can be displayed in ladder display at a peripheral device. (Peripheral devices continue to display ladders until encountering the END instruction.)



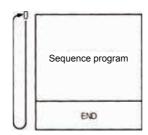
## 3.25.10Termination instruction: sequence program termination ... END

	Applicable device									
В	it devi	ce	Wo	ord (16-l device	oit)	Constant		Level	Digit specification	Error flag
Х	Υ	M	T	С	D	K	Н	N		(SM0)



### **♦**Function

• Indicates the end of a program. Scanning terminates with this step, and returns to step 0.



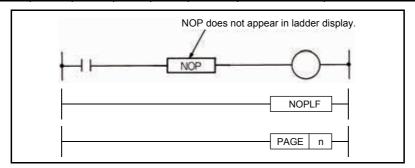
• The END instruction cannot be used during the sequence program.



• An operation error occurs if there is no END instruction during the program, and the PLC function does not operate.

## 3.25.110ther instructions: non-processing ... NOP, NOPLF, PAGE n

			Арр							
В	Bit device Word (16-bit) device		Constant		Level	Digit specification	Error flag			
Х	Υ	M	Т	С	D	K	Н	N		(SM0)
										0



### **◆**Function

NOP

Non-processing instructions do not have an effect on previously executed operations.

NOP are used in the following situations.

- Creating space for sequence program debugging.
- Deleting instructions without changing the number of steps. (Replaced by NOP)
- · Temporarily deleting instructions.

NOPLF

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

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

- · When printing ladders
  - A page break will be inserted between ladder blocks with the presence of the NOPLF instruction.
  - 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.
- · When printing instruction lists
  - The page will be changed after the printing of the NOPLF instruction.

Refer to the Operating Manual for the peripheral device in use for details of printouts from peripheral devices.

PAGE n

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

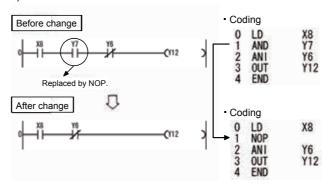
No processing is performed at peripheral devices with this instruction.

### 3

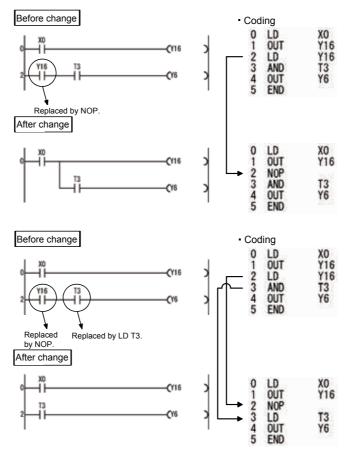
### **♦**Program example

NOP

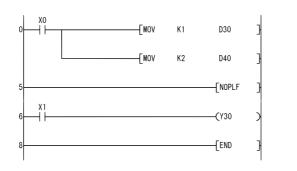
· Contact short-circuit (AND, ANI)



Contact short-circuit (LD, LDI)
 Caution is required because the ladder completely changes if LD or LDI are replaced by NOP.



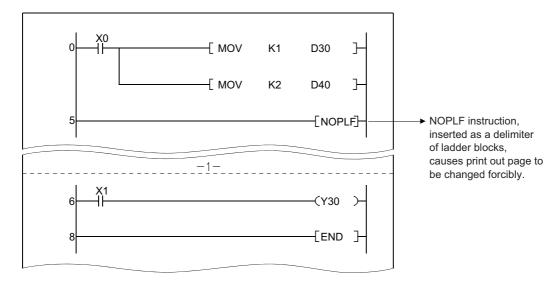
NOPLF



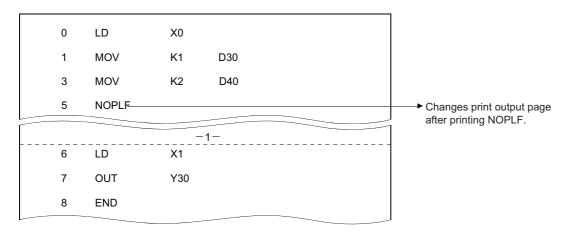


### **Sequence instructions**

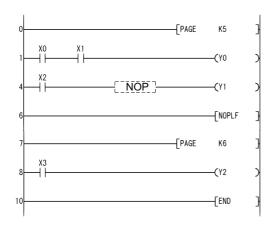
• Printing the ladder will result in the following:



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



PAGE n



Coding 0 1 2 3 4 5 6 7 7 8 9 10	G PAGE LD AND OUT LD OUT OUT NOP OUT NOPLF PAGE LD OUT	K5 X0 X1 Y0 Y2 Y1 K6 X3 Y2
11		12

### 3.26 Basic instruction (16-bit)

The basic instruction (16-bit) can handle 16 bits of numeric data.

### 3.26.1 Comparison operation instruction

- Comparison operation instructions execute size comparisons (=, >, < etc.) of two data, and instructs a contact to turn ON when the conditions are met.
- The usage method of comparison operation instructions is the same as the contact instructions of sequence instructions, and is as follows.

• LD, LDI: LD= • AND, ANI: AND=

• OR, ORI: OR=

• There are 18 types of comparison operation instruction as shown below. For details, refer to page 140.

Classification	Instruction symbol	Classification	Instruction symbol	Classification	Instruction symbol
	LD=		LD>		LD<
=	AND=	>	AND>	<	AND<
	OR=		OR>		OR<
	LD<>		LD<=		LD>=
<b>≠</b>	AND<>	≤	AND<=	≥	AND>=
	OR<>		OR<=		OR>=

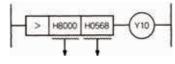
• The conditions for turning comparison operation instructions ON are as follows.

	98	99	100	101	102	
Dn = K100	OI	FF	ON	OFF		
Dn≠K100	C	N	OFF	ON		
Dn>K100		OFF		(	NC	
Dn≤K100		ON			OFF	
Dn < K100	0	N		OFF		
Dn≥K100	0	FF	ON			

The comparison instruction assumes the specified data as a BIN value for comparison. For this reason, when performing hexadecimal comparison, when the value whose most significant bit (b15) is 1 (8 to F), the BIN value is assumed as a negative number for the comparison.

Example

Comparison of HEX / 4-digit value

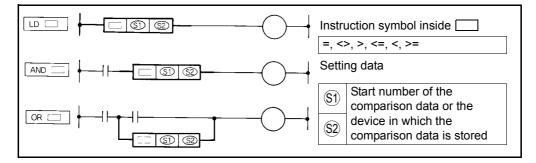


Regarded Regarded as -32767 as 1384 in BIN. in BIN.

Therefore, -32767 < 1384, and Y10 does not turn ON.

## 3.26.2 Comparison operation instruction: 16-bit data comparison ... =, <>, >, <=, <, >=

				App	olicable		<b>-</b>	Error flag			
	В	it devi	се	Word	(16-bit)	device	Con	stant	Level	Digit specification	Lifor nag
	Х	Υ	M	Т	С	D	K	Н	N	opoomounom	(SM0)
<b>§</b> 1	0	0	0	0	0	0	0	0		K1 to K4	0
<u>\$2</u>	0	0	0	0	0	0	0	0		1010	



### **◆**Function

- 16-bit comparison operation is handled as NO contact.
- · Comparison operation results are as follows.

Instruction symbol inside	Condition	Comparison operation results
=	<u>\$1</u> ) = <u>\$2</u>	
<>	<u>\$1</u> ) ≠ <u>\$2</u>	
>	<u>\$1</u> ) > <u>\$2</u>	Conducted
<=	<u>\$1</u> ) ≤ <u>\$2</u>	Conducted
<	<u>\$1</u> < <u>\$2</u>	
>=	§1) ≥ §2)	

Instruction symbol inside	Condition	Comparison operation results
=	§1) ≠ \$2	
<>	<u>\$1</u> ) = <u>\$2</u>	
>	<u>§1</u> ) ≤ <u>\$2</u> )	Not conducted
<=	<u>\$1</u> ) > <u>\$2</u>	Not conducted
<	<u>\$1</u> ) ≥ <u>\$2</u>	
>=	<u>\$1</u> < <u>\$2</u>	

### **◆**Execution condition

The execution conditions for LD \_\_\_\_, AND \_\_\_\_ and OR \_\_\_\_ are as follows.

Instruction	Execution condition						
LD 🗀	Each scan execution						
AND 🔲	Executed when the previous contact instruction is ON						
OR 🔲	Each scan execution						

### • NOTE

• When the digit setting of a bit device is other than K4 and a start bit device value is other than a multiple of 8, the number of steps is 7.

### **♦**Program example

=

Program that compares the X0 to F data with D3 data.

<>

Program that compares the BCD value 100 with D3 data.

>

Program that compares the BIN value 100 with D3 data.

<=

Program that compares the D0 with D3 data.

### 3.26.3 Arithmetic operation instruction

An arithmetic operation instruction instructs the addition, subtraction, multiplication or division for two BIN data, or operation of increment or decrement.

### **♦BIN** arithmetic operation (binary)

- If the operation result of an addition instruction exceeds 32767, the value is negative.
- If the operation result of a subtraction instruction is smaller than -32768, the value is positive.
- Operations of positive values and negative values are as follows.

```
5 + 8 \rightarrow 13
```

$$5 - 8 \rightarrow -3$$

$$5 \times 3 \rightarrow 15$$

$$\text{-}5\times3\rightarrow\text{-}15$$

$$-5 \times (-3) \rightarrow 15$$

-5 
$$\div$$
 3  $\rightarrow$  -1 remainder -2

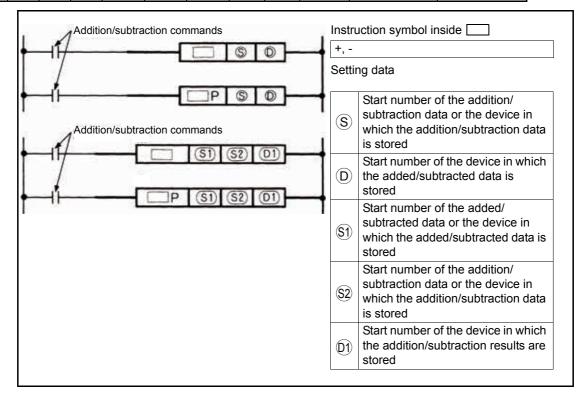
$$5 \div (-3) \rightarrow -1$$
 remainder 2

$$-5 \div (-3) \rightarrow 1 \text{ remainder } -2$$

### 3

## 3.26.4 Arithmetic operation instruction: BIN 16-bit addition/subtraction ... +, +P, -, -P

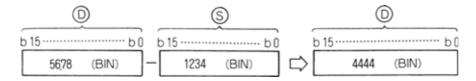
	Applicable device								5	Error flag	
	Bit device			Word (16-bit) device			Constant		Level	Digit specification	Lifoi nag
	Х	Υ	M	T	С	D	K	Н	N	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	(SM0)
S	0	0	0	0	0	0	0	0		K1 to K4	1 to K4 O
(D)		0	0	0	0	0					
<b>§1</b>	0	0	0	0	0	0	0	0			
<u>\$2</u>	0	0	0	0	0	0	0	0			
<b>(D1)</b>		0	0	0	0	0					



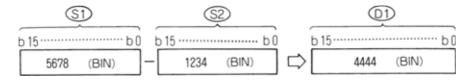
#### **♦**Function



• Executes addition of the BIN data specified with  $\bigcirc$  and BIN data specified with  $\bigcirc$ , and stores the addition results in the device specified with  $\bigcirc$ .



• Executes addition of the BIN data specified with (S1) and BIN data specified with (S2), and stores it in the device specified with (D1).

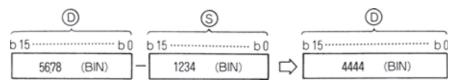


- -32768 to 32767 (BIN 16-bit) can be specified in (S), (S1), (S2) or (D).
- Positive/negative judgment of the S, S1, S2 and D data is performed by the most significant bit (b15). (0: positive, 1: negative)
- The carry flag does not turn ON for the 0-bit underflow. The carry flag does not turn ON for the 15th bit overflow. (There is no carry flag)

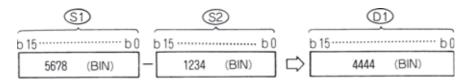
## **♦**Function



• Executes subtraction of the BIN data specified with  $\bigcirc$  and BIN data specified with  $\bigcirc$ , and stores the subtraction results in the device specified with  $\bigcirc$ .



• Executes subtraction of the BIN data specified with (3) and BIN data specified with (52), and stores it in the device specified with (D1).

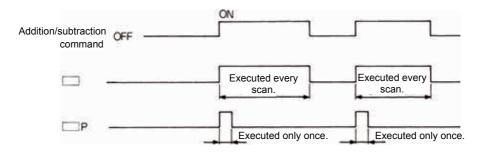


- -32768 to 32767 (BIN 16-bit) can be specified in (S), (S1), (S2) or (D).
- Positive/negative judgment of the S, S1, S2 and D data is performed by the most significant bit (b15). (0: Positive, 1: negative)
- The carry flag does not turn ON for the 0-bit underflow. The carry flag does not turn on for the 15th bit overflow. (There is no carry flag)

#### J

## **◆**Execution condition

Addition/subtraction instruction

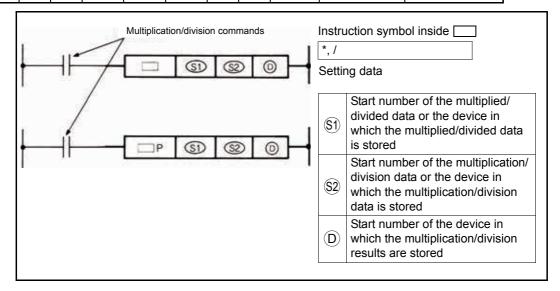


# **♦**Program example

Program that adds the content of D3 to the content of D0 when X5 turns ON, and outputs the results to Y38 to 3F.

# 3.26.5 Arithmetic operation instruction: BIN 16-bit multiplication/division ... \*, \*P, /, /P

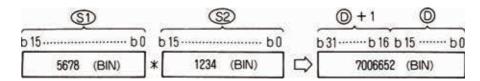
				Арј	D: '4	Error flag					
	Bi	t devi	се	Word (16-bit) device			Constant		Level	Digit specification	Lifor mag
	Х	Υ	M	T	С	D	K	Н	N	Оросиновно	(SM0)
<b>§</b> 1)	0	0	0	0	0	0	0	0			
<b>S2</b>	0	0	0	0	0	0	0	0		K1 to K4	0
(D)		0	0	0	0	0					



#### **♦**Function

\*

• Executes multiplication of the BIN data specified with 🗐 and BIN data specified with 🗐, and stores the multiplication results in the device specified with 🔘.



 $\bullet$  When  ${\Large \bigcirc}$  is a bit device, specify from the lower bits.

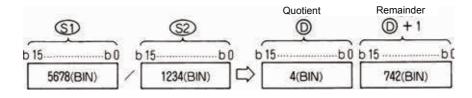
Example

K1: Lower 4 bits (b0 to 3) K4: Lower 16 bits (b0 to 15)

- -32768 to 32767 (BIN 16-bit) can be specified in \$1 or \$2.
- Positive/negative judgment of the data is performed by the most significant bit (b15) for 🐒 and 🐒, and (b31) for ①. (0: Positive, 1: negative)

1

• Executes division of the BIN data specified with 🐒 and BIN data specified with 🗐, and stores the division results in the device specified with 🔘.



• The division results are stored as the quotient and remainder using 32 bits for a word device, or stored as only the quotient using 16 bits for a bit device.

Quotient: Stored in the lower 16 bits.

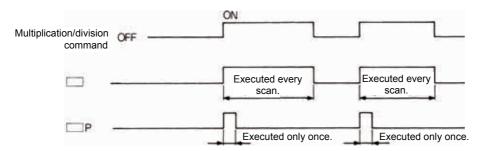
Remainder: Stored in the upper 16 bits. (Only stored for word devices.)

- -32768 to 32767 (BIN 16-bit) can be specified in (\$1) or (\$2).
- Positive/negative judgment of the \$1, \$2, \$\overline{D}\$ and \$\overline{D}\$+1 data is performed by the most significant bit (b15). (Both quotient and remainder have a sign.)

(0: positive, 1: negative)

#### **◆**Execution condition

The execution conditions for multiplication/division instructions are as follows.



# **♦**Operation error

An operation error occurs in the following cases, and the error flag turns on.

- When A1 or V are specified for  $\bigcirc$ .
- When the divisor \$2 is 0.

# **♦**Program example

\*

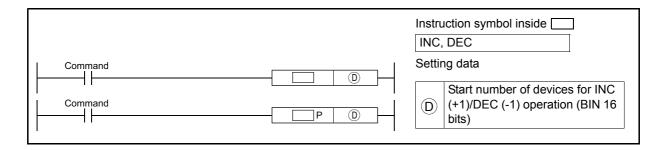
• Program that stores the multiplication results of BIN 5678 and 1234 in D3 and 4 when X5 turns on.

/

Program that outputs the results of dividing the X8 to F data by 3.14 to Y30 to 3F when X3 turns on.

# 3.26.6 Arithmetic operation instruction: BIN 16-bit data increment/decrement ... INC, INCP, DEC, DECP

					App	olicable	device				<b>-</b>	Error flag	
		Bit device			Word (16-bit) device			Cons	stant	Level	Digit specification	Littor mag	
		Х	Υ	M	Т	С	D	K	Н	N	, op	(SM0)	
INC, INCP, DEC, DECP	(D)	0	0	0	0	0	0				K1 to K4	0	



#### **♦**Function

INC

• Adds 1 to the device designated by ① (16-bit data).

• When INC/INCP operation is executed for the device designated by ①, whose content is 32767, the value -32768 is stored at the device designated by ①.

DEC

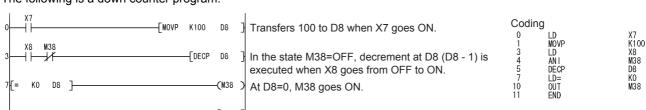
• Subtracts 1 from the device designated by  $\bigcirc$  (16-bit data).



• When DEC/DECP operation is executed for the device designated by , whose content is -32768, the value 32767 is stored at the device designated by ①.

# **♦**Program Example

The following is a down counter program.



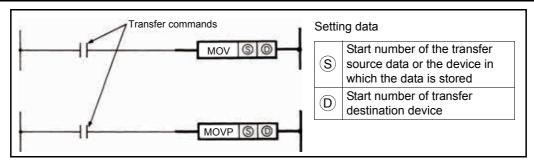
# 3.26.7 Data transfer instruction

The data transfer instruction is an instruction that executes the transfer of data.

Data transferred by the data transfer instruction is retained until new data is transferred.

# 3.26.8 Data transfer instruction: 16-bit data transfer ... MOV, MOVP

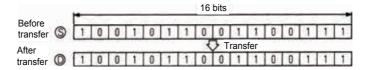
					App	licable	device				<b>5</b>	Error flag	
		Bit device			Word (16-bit) device			Cons	stant	Level Digit specification		Lifornag	
		Х	Υ	M	T	С	D	K	Н	N	Оросински	(SM0)	
MOV,	S	0	0	0	0	0	0	0	0		K1 to K4		
MOVP	<b>(D)</b>		0	0	0	0	0		·		1010		



#### **♦**Function

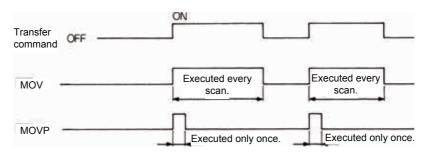
MOV

16-bit data from the device specified with  $\bigcirc$  is transferred to the device specified with  $\bigcirc$ .



### **◆**Execution condition

The execution conditions for transfer instructions are as follows.



# **♦**Program example

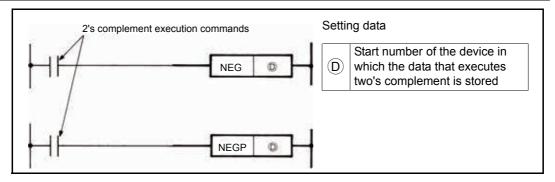
MOV

• Program that stores 155 as a binary value in D8 when X8 turns ON.



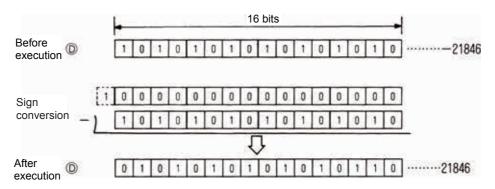
# 3.26.9 Data conversion instruction: 2's complements of BIN 16-bit data ... NEG, NEGP

	\					Арр	olicable	device				<b>5</b>	Error flag
	Bit device Word (16-b				(16-bit)	) device   Constant   Level				Digit specification	Efformag		
			Х	Υ	M	Т	С	D	K	Н	N	, , , , , , , , , , , , , , , , , , , ,	(SM0)
Ν	EG	(D)		0	0	0	0	0	0	0		K1 to K4	0



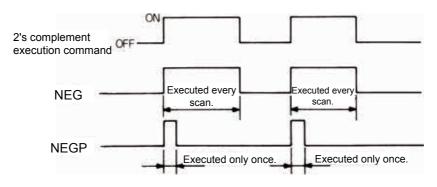
#### **♦**Function

• Invert the sign of the 16-bit device specified with  $\bigcirc$ , and store the device specified with  $\bigcirc$ .



• This is used when inverting the positive/negative signs.

#### **◆**Execution condition

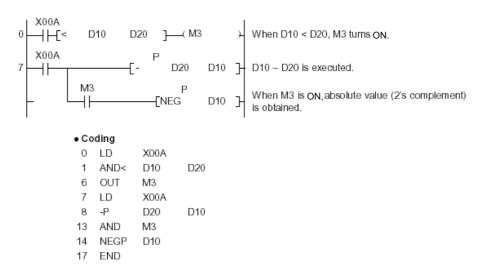


3

# **♦**Program example

NEG

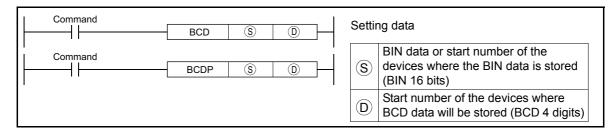
• Program that calculates D10 to D20 when XA is turned ON, and calculates the absolute value when this result is negative.



# 3

# 3.26.10Data conversion instructions: BIN 16-bit data BCD conversions ... BCD, BCDP

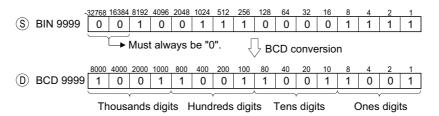
				Арј	olicable	device						
		Bit device		Word (16-bit) device		Constant		Level	Digit specification	Error flag		
		Х	Υ	М	T	С	D	K	Н	N		(SM0)
BCD	S	0	0	0	0	0	0	0	0		K1 to K4	0
BCDP	(D)	0	0	0	0	0	0				1014	



## **♦**Function

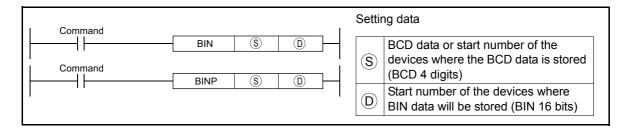
BCD

• Converts BIN data (0 to 9999) at the device designated by S to BCD data, and stores it at the device designated by D.



# 3.26.11Data conversion instructions: BIN 16-bit data BIN conversions ... BIN, BINP

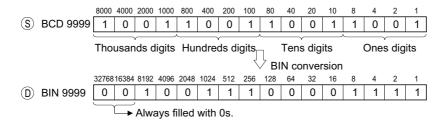
					Арр	olicable	device					
		Bit device		Word (16-bit) device		Constant		Level	Digit specification	Error flag		
		Х	Υ	M	T	С	D	K	Н	N		(SM0)
BIN	S	0	0	0	0	0	0	0	0		K1 to K4	0
BINP	(D)	0	0	0	0	0	0				1014	



#### **♦**Function

BIN

• Converts BCD data (0 to 9999) at device designated by S to BIN data, and stores at the device designated by D.



# **3.27** Basic instruction (32-bit)

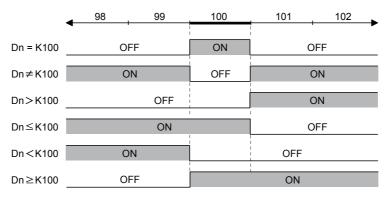
The basic instruction (32-bit) is an instruction that can handle numeric data that is expressed in 32 bits.

# 3.27.1 Comparison operation instruction

- Comparison operation instructions execute size comparisons (D=, D>, D< etc.) of two data, and instructs the contact to turn ON when the conditions are met.
- The usage method of comparison operation instructions is the same as the contact instructions of sequence instructions, and is as follows.
  - LD, LDI.....LDD=
  - AND, ANI...ANDD=
  - OR, ORI....ORD=
- There are 18 types of comparison operation instruction as shown below.
   For details, refer to page 156.

Classification	Instruction symbol	Classification	Instruction symbol	Classification	Instruction symbol
	LDD=		LDD>		LDD<
=	ANDD=	>	ANDD>	<	ANDD<
	ORD=		ORD>		ORD<
	LDD<>		LDD<=		LDD>=
≠	ANDD<>	≤	ANDD<=	≥	ANDD>=
	ORD<>		ORD<=		ORD>=

• The conditions for turning comparison operation instructions ON are as follows.

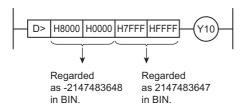


# NOTE

The comparison instruction assumes the specified data as a BIN value for comparison. For this reason, when performing
hexadecimal comparison, when the value whose most significant bit (b31) is 1 (8 to F), the BIN value is assumed as a
negative number for the comparison.

Example

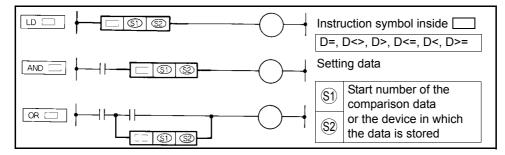
Comparison of HEX/8 digit value



Therefore, -2147483648 < -2147483647, and Y10 does not turn ON.

# 3.27.2 Comparison operation instruction: 32-bit data comparison ... D=, D<>, D>, D<=, D<, D>=

					Error flag						
	В	it devi	се	Word (16-bit) device			Constant		Level	Digit specification	Litoring
	Х	Υ	M	T	С	D	K	Н	N		(SM0)
<b>§</b> 1)	0	0	0	0	0	0	0	0		K1 to K8	0
<u>\$2</u>	0	0	0	0	0	0	0	0		I KI TO NO	



#### **♦**Function

- · 32-bit comparison operation handled as NO contact.
- · Comparison operation results are as follows.

Instruction symbol inside	Condition	Comparison operation results
D=	<u>S1</u> = <u>S2</u>	
D<>	§1) ≠ §2	
D>	§1) > <u>\$2</u>	Conducted
D<=	<u>§1</u> ) ≤ <u>\$2</u>	Conducted
D<	§1) < <u>\$2</u>	
D>=	§1) ≥ <b>§2</b>	

Instruction symbol inside	Condition	Comparison operation results
D=	§1) ≠ <b>§2</b>	
D<>	<u>§1</u> = <u>\$2</u>	
D>	§1) ≤ <b>§2</b> )	Not conducted
D<=	<u>§1</u> > <u>\$2</u>	Not conducted
D<	§1) ≥ <b>§2</b> )	
D>=	§1 < <u>\$2</u>	

### **◆**Execution condition

The execution conditions for LD \_\_\_\_\_, AND \_\_\_\_ and OR \_\_\_\_ are as follows.

Instruction Execution condition								
LD 🗀	Each scan execution							
AND 🔲	Executed when the previous contact instruction turns ON.							
OR	Each scan execution							

# • NOTE

 When the digit setting of a bit device is other than K8 and a start bit device value is other than a multiple of 8, the number of steps is 7.

# **♦**Program example

D=

Program that compares the M0 to 31 data with D3 and D4 data.

D<>

Program that compares the BCD value 18000 with D3 and D4 data.

D>

Program that compares the BIN value -80000 with D3 and D4 data.

D<=

Program that compares D0 and D1 with D3 and D4 data.

# 3.27.3 Arithmetic operation instruction

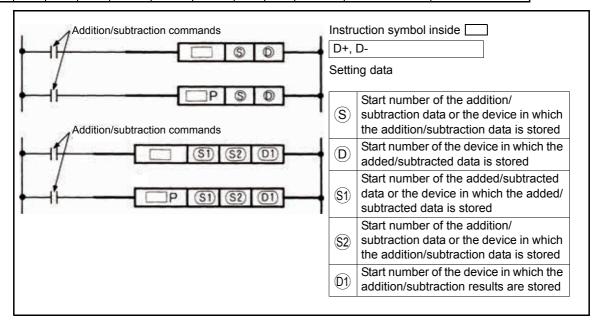
An arithmetic operation instructs the addition, subtraction, multiplication or division for two BIN data, or operation of increment or decrement.

## **♦BIN** arithmetic operation (binary)

- If the operation result of an addition instruction exceeds 2147483647, the value is negative.
- If the operation result of a subtraction instruction is smaller than -2147483648, the value is positive.
- · Operations of positive values and negative values are as follows.
- $5 + 8 \rightarrow 13$
- $5 8 \rightarrow -3$
- $5 \times 3 \rightarrow 15$
- $-5 \times 3 \rightarrow -15$
- $-5 \times (-3) \rightarrow 15$
- $-5 \div 3 \rightarrow -1$  remainder -2
- $5 \div (-3) \rightarrow -1$  remainder 2
- $-5 \div (-3) \rightarrow 1 \text{ remainder } -2$

# 3.27.4 Arithmetic operation instruction: BIN 32-bit addition/subtraction ... D+, D+P, D-, D-P

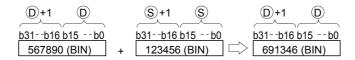
				App	<b>5</b>	Error flag					
	Ві	it devi	се	Word	(16-bit)	device	Cons	stant	Level	Digit specification	Lifornag
	Х	Υ	M	Т	С	D	K	Н	N	<b>-</b>	(SM0)
S	0	0	0	0	0	0	0	0			
(D)		0	0	0	0	0					0
S	0	0	0	0	0	0	0	0		K1 to K8	
<b>S2</b>	0	0	0	0	0	0	0	0			
<u>\$2</u>		0	0	0	0	0					



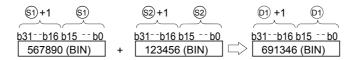
#### **♦**Function

D+

• Executes addition of the BIN data specified with  $\bigcirc$  and BIN data specified with  $\bigcirc$ , and stores the addition results in the device specified with  $\bigcirc$ .



• Executes addition of the BIN data specified with 🐒 and BIN data specified with 🕲, and stores the addition results in the device specified with 🕦.

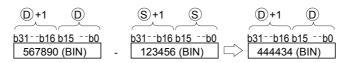


- -2147483648 to 2147483647 (BIN 32-bit) can be specified in S, S1, S2 or D.
- Positive/negative judgment of the (S), (S1), (S2) and (D) data is performed by the most significant bit (b31). (0: positive, 1: negative)
- The carry flag does not turn ON for the 0-bit underflow. The carry flag does not turn ON for the 31st bit overflow. (There is no carry flag)

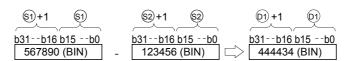
## **♦**Function

D-

• Executes subtraction of the BIN data specified with  $\bigcirc$  and BIN data specified with  $\bigcirc$ , and stores the subtraction results in the device specified with  $\bigcirc$ .



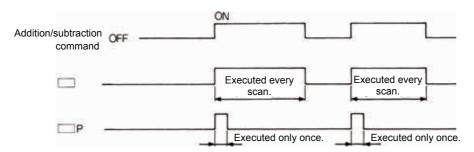
• Executes subtraction of the BIN data specified with (\$\sigma)\$ and BIN data specified with (\$\sigma)\$, and stores the subtraction results in the device specified with (\$\sigma)\$).



- -2147483648 to 2147483647 (BIN 32-bit) can be specified in  $\bigcirc$  ,  $\bigcirc$  ,  $\bigcirc$  or  $\bigcirc$  .
- Positive/negative judgment of the S, S1, S2 and D data is performed by the most significant bit (b31). (0: positive, 1: negative)
- The carry flag does not turn ON for the 0-bit underflow. The carry flag does not turn ON for the 31st bit overflow. (There is no carry flag)

#### **◆**Execution condition

Addition/subtraction instruction



# **♦**Program example



Program that adds the X10 to X2B 28-bit data to the D9 and D10 data when X0 turns ON, and outputs the results to M0 to M27.

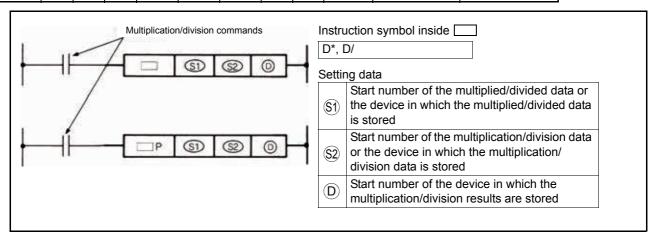
D-

Program that subtracts the M0 to M23 data from the D0 and D1 data when X0 turns ON, and stores the results in D10 and D11.

#### 3

# 3.27.5 Arithmetic operation instruction: BIN 32-bit multiplication/division ... D\*, D\*P, D/, D/P

				Арр	olicable	device				<b>-</b>	Error flag	
	Bi	it devi	се	Word	(16-bit)	device	Cons	stant	Level	Digit specification	Lifoi nag	
	Х	Υ	M	Т	С	D	K	Н	N	,	(SM0)	
<u>\$1</u>	0	0	0	0	0	0	0	0			0	
<u>\$2</u>	0	0	0	0	0	0	0	0		K1 to K8		
(D)		0	0	0	0	0						



#### **◆**Function

D\*

• Executes multiplication of the BIN data specified with 🗐 and BIN data specified with 🗐, and stores the results in the device specified with 🔘



- When  ${\color{red} \overline{\mathbb{O}}}$  is a bit device, specify from the lower bits.

Example

K1: Lower 4 bits (b0 to 3)

K4: Lower 16 bits (b0 to 15)

K8: Lower 32 bits (b0 to 31)

- -2147483648 to 2147483647 (BIN 32-bit) can be specified in (\$1) or (\$2).
- Positive/negative judgment of the data is performed by the most significant bit (b31) for \$\ext{S1}\$ and \$\ext{S2}\$, and (b63) for \$\extstyle{\D}\$. (0: positive, 1: negative)

#### **Basic instruction (32-bit)**

D/

• Executes division of the BIN data specified with §1) and BIN data specified with §2, and stores the results in the device specified with ①.



• The division results are stored as the quotient and remainder using 64 bits for a word device, or stored as only the quotient using 32 bits for a bit device.

Quotient: Stored in the lower 32 bits.

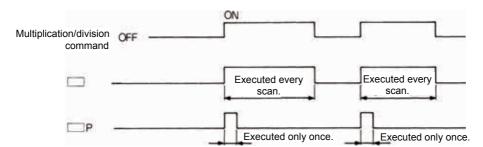
Remainder: Stored in the upper 32 bits. (Only stored for word devices.)

- -2147483648 to 2147483647 (BIN 32-bit) can be specified in 🕥 or 👀.
- Positive/negative judgment of the \$1, \$2, \$\overline{D}\$ and \$\overline{D}\$+2 data is performed by the most significant bit (b31). (Both quotient and remainder have a sign.)

(0: positive, 1: negative)

## **◆**Execution condition

The execution conditions for multiplication/division instructions are as follows.



## **♦**Operation error

An operation error occurs in the following cases, and the error flag turns ON.

- When A1 or V is specified in (S1) or (S2), or A0, A1, Z or V are specified in (D).
- When the divisor S2 is 0.

# **♦**Program example



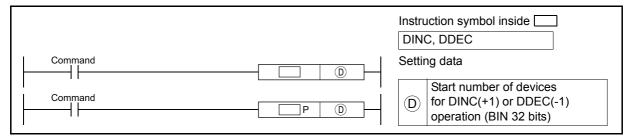
Program that stores the results of multiplying D7 and D8 BIN data with D18 and D19 BIN data in D1 to D4 when X5 turns ON.

D/

Program that outputs the result of multiplying M0 to 7 by 3.14 to D3 when X3 turns ON.

# 3.27.6 Arithmetic operation instruction: BIN 32-bit data increment/decrement ... DINC, DINCP, DDEC, DDECP

					Арј	olicable	device				D: 14	Error flag	
		Bi	t devi	се	Word (16-bit) device			Cons	stant Level		Digit specification	Lifor mag	
		Х	Υ	M	Т	С	D	K	Н	N		(SM0)	
DINC, DINCP, DDEC, DDECP	(D)	0	0	0	0	0	0				K1 to K4	0	



#### **◆**Function

DINC

• Adds 1 to the device designated by (D) (32-bit data).

• When DINC/DINCP operation is executed for the device designated by ①, whose content is 2147483647, the value -2147483648 is stored at the device designated by ②.

DDEC

• Subtracts -1 from the device designated by (D) (32-bit data).

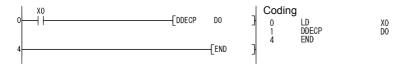
• When DDEC/DDECP operation is executed for the device designated by (D), whose content is 0, the value -1 is stored at the device designated by (D).

# **♦**Program example

• The following program adds 1 to the data at D0 and D1 when X0 is ON.



• The following program subtracts 1 from the data at D0 and D1 when X0 goes ON.



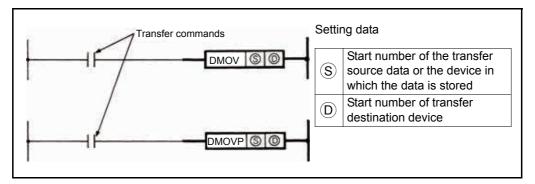
# 3.27.7 Data transfer instruction

The data transfer instruction is an instruction that executes the transfer of data.

Data transferred by the data transfer instruction is retained until new data is transferred.

# 3.27.8 Data transfer instruction: 32-bit data transfer ... DMOV, DMOVP

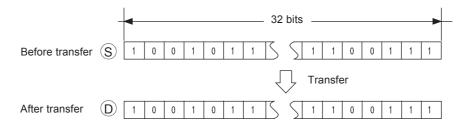
					Арј	olicable	device				D: 11	Error flag	
		Bi	it devi	се	Word (16-bit) device			Cons	stant	Level Digit specification		Lifor hag	
		Х	Υ	M	Т	С	D	K	Н	N	-	(SM0)	
DMOV,	S	0	0	0	0	0	0	0	0		K1 to K8	0	
DMOVP D			0	0	0	0	0				ICI to ICO	Ö	



### **♦**Function

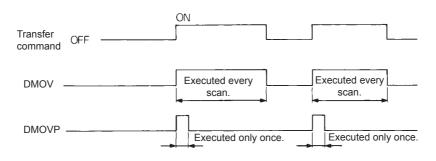
DMOV

32-bit data from the device specified with  $\bigcirc$  is transferred to the device specified with  $\bigcirc$ .



# **◆**Execution condition

The execution conditions for transfer instructions are as follows.



# **♦**Program example

#### DMOV

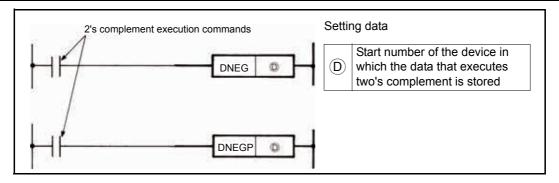
• Program that stores the input D2 to D3 data in D0 and D1.

• Program that stores M0 to M31 data in D0 and D1.

# 3

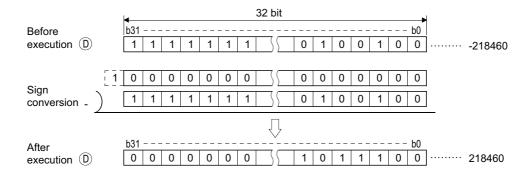
# 3.27.9 Data conversion instruction: 2's complements of BIN 32-bit data ... DNEG, DNEGP

						App	olicable	device	Applicable device										
			Bit device			Word (16-bit) device			Constant		Level	Digit specification	Error flag						
			Х	Υ	M	T	С	D	K	Н	N		(SM0)						
	DNEG(P)	(D)		0	0	0	0	0	0	0		K1 to K8	0						



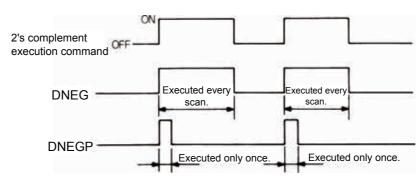
#### **◆**Function

• Invert the sign of the 32-bit device specified with  $\bigcirc$ , and store the device specified with  $\bigcirc$ .



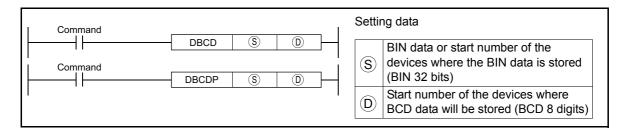
• This is used when inverting the positive/negative signs.

#### **◆**Execution condition



# 3.27.10Data conversion instructions: BIN 32-bit data BCD conversions ... DBCD, DBCDP

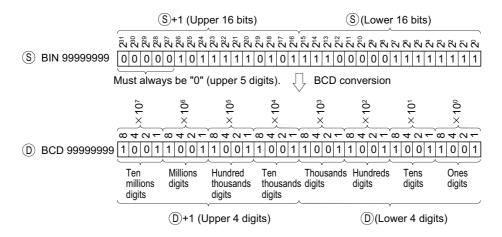
					Арр	olicable	device						
		Ві	it devi	се	Word (16-bit) device			Constant		Level	Digit specification	Error flag	
		Х	Υ	M	T	С	D	K	Н	N		(SM0)	
DBCD	S	0	0	0	0	0	0	0	0		K1 to K8	0	
DBCDP	(D)	0	0	0	0	0	0				INT TO INO		



#### **♦**Function

DBCD

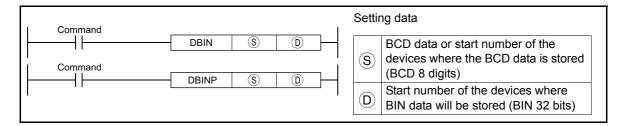
• Converts BIN data (0 to 99999999) at the device designated by S to BCD data, and stores it at the device designated by D.



# 3

# 3.27.11 Data conversion instructions: BIN 32-bit data BIN conversions ... DBIN, DBINP

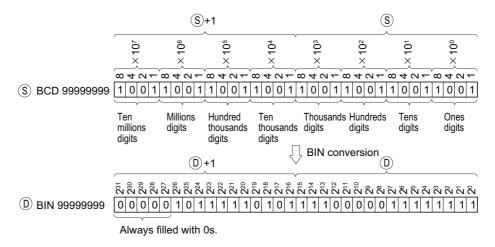
					Арј	olicable	device						
		Bit device			Word (16-bit) device			Constant		Level	Digit specification	Error flag	
		Х	Υ	М	T	С	D	K	Н	N		(SM0)	
DBIN S		0	0	0	0	0	0	0	0		K1 to K8	0	
DBINP	(D)	0	0	0	0	0	0				I I IU IVO		



#### **♦**Function

DBIN

• Converts BCD data (0 to 99999999) at the device designated by (S) to BIN data, and stores at the device designated by (D).



# **3.28** Application instructions (16-bit)

Application instructions (16-bit) are used when special processing is required.

# 3.28.1 Logical operation instructions

- Logical operation instructions are instructions that execute logical operations such as logical OR or logical AND.
- There are 10 types of logical operation instruction as shown below.

Classification	Instruction symbol
Logical AND	WAND
Logical AND	WANDP
Logical OR	WOR
Logical OR	WORP

Classification	Instruction symbol
Exclusive OR	WXOR
Exclusive OR	WXORP
Exclusive NOR	WXNR
Exclusive NON	WXNRP

Classification	Instruction symbol
2's complement	NEG
(code reversed)	NEGP

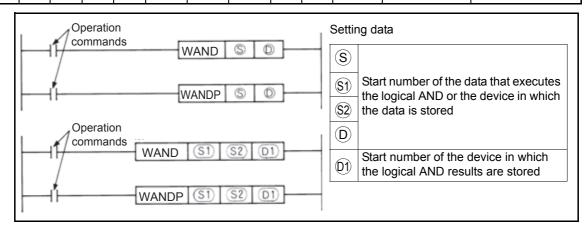
## • NOTE

· Logical operation instructions execute the following processes in increments of bits.

Classification	Process	Formula	E	xamp	le
Classification	FTOCESS	Formula	Α	В	Υ
			0	0	0
Logical AND	Set 1 when the inputs of both A and B are 1.	Y = A · B	0	1	0
Logical AND	Set 0 in all other cases.	I - A B	1	0	0
			1	1	1
			0	0	0
Logical OR	Set 0 when the inputs of both A and B are 0.	Y = A + B	0	1	1
Logical OIX	Set 1 in all other cases.	I - A · B	1	0	1
			1	1	1
			0	0	0
Exclusive OR	Set 0 when the inputs of both A and B are equal.	Y = A · B + A · B	0	1	1
LACIUSIVE OIL	Set 1 when they are different.		1	0	1
			1	1	0
			0	0	1
Exclusive NOR	Set 1 when the inputs of both A and B are equal.	$Y = (\overline{A} + B) (A + \overline{B})$	0	1	0
LACIUSIVE NOR	Set 0 when they are different.	1 - (A + B) (A + B)	1	0	0
			1	1	1

# 3.28.2 Logical operation instruction: BIN 16-bit logical AND ... WAND, WANDP

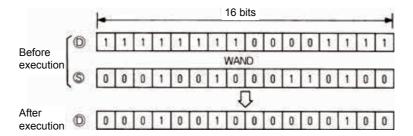
					Арр	olicable	device				Divit	Error flag	
		Bit device			Word (16-bit) device			Constant Level			Digit specification	Litorinag	
		Х	Υ	М	T	С	D	K H		N	оростоиноп	(SM0)	
	S	0	0	0	0	0	0	0	0				
Ī	(D)		0	0	0	0	0					0	
WAND	<b>§</b> 1	0	0	0	0	0	0	0	0		K1 to K4		
(	<u>\$2</u>	0	0	0	0	0	0	0	0				
	<b>(D1)</b>		0	0	0	0	0						



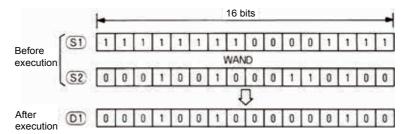
#### **♦**Function

#### WAND

• Logical AND is executed for each bit of the 16-bit data of the device specified with  $\bigcirc$  and 16-bit data of the device specified with  $\bigcirc$ , and the results are stored in the device specified with  $\bigcirc$ .



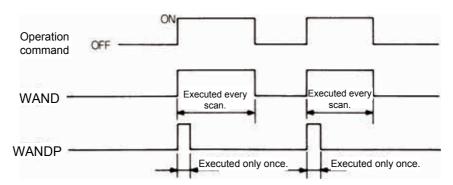
• Logical AND is executed for each bit of the 16-bit data of the device specified with (\$\sigma\$) and 16-bit data of the device specified with (\$\sigma\$), and the results are stored in the device specified with (\$\sigma\$).



• Bits of the bit device higher than the specified digit are operated as 0.

#### **◆**Execution condition

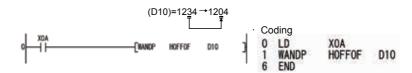
The execution conditions for logical AND instructions are as follows.



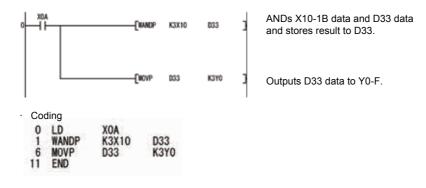
# ◆Program example

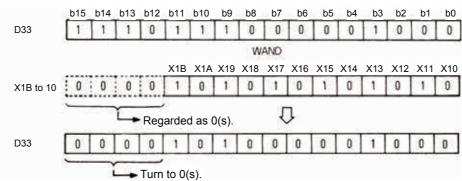
#### WAND

• Program that masks the tenth digit (the second digit from the bottom) of the BCD 4 digits when XA turns ON.



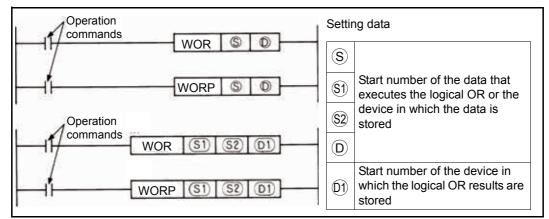
• Program that executes logical AND for X10 to 1B data and D33 data when XA turns ON, and outputs the results to Y0 to B.





# 3.28.3 Logical operation instruction: BIN 16-bit logical OR ... WOR, WORP

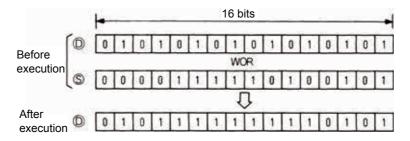
					App	olicable	device				<b>5</b>	Error flag		
			Bit device			Word (16-bit) device			stant	Level	Digit specification			
		Х	Υ	M	Т	С	D	K	Н	N	С	(SM0)		
	S	0	0	0	0	0	0	0	0					
	D		0	0	0	0	0					0		
WOR	<b>S</b> 1)	0	0	0	0	0	0	0	0		K1 to K4			
	<u>\$2</u>	0	0	0	0	0	0	0	0					
	<b>(D1)</b>		0	0	0	0	0							



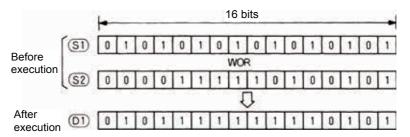
## **♦**Function

#### WOR

• Logical OR is executed for each bit of the 16-bit data of the device specified with  $\bigcirc$  and 16-bit data of the device specified with  $\bigcirc$  and the results are stored in the device specified with  $\bigcirc$ .



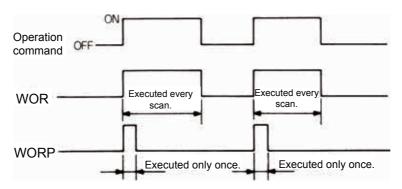
• Logical OR is executed for each bit of the 16-bit data of the device specified with (\$1) and 16-bit data of the device specified with (\$2), and the results are stored in the device specified with (\$1).



• Bits of the bit device higher than the specified digit are operated as 0.

#### **◆**Execution condition

• The execution conditions for logical OR instructions are as follows.



# **♦**Program example

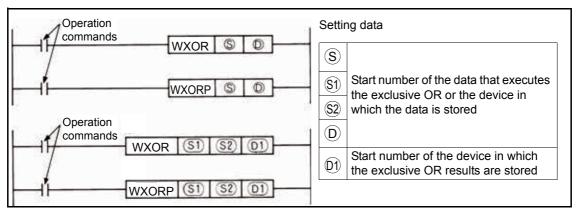
WOR

• Program that executes logical OR for D10 and D20 data when XA turns ON, and stores the results in D10.

• Program that executes logical OR for X10 to 1B data and D33 data when XA turns ON, and outputs the results to Y0 to B.

# 3.28.4 Logical operation instruction: BIN 16-bit data exclusive OR ... WXOR, WXORP

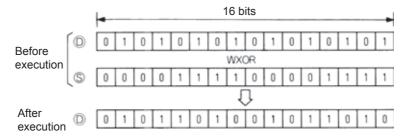
					Арр	<b>5</b> 1 11	Error flag					
		Bit device			Word (16-bit) device			Constant		Level	Digit specification	
		Х	Υ	М	T	С	D	K	Н	N	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	(SM0)
WXOR	S	0	0	0	0	0	0	0	0		K1 to K4	0
	(D)		0	0	0	0	0					
	<b>§</b> 1)	0	0	0	0	0	0	0	0			
	S2	0	0	0	0	0	0	0	0			
	<b>(D1)</b>		0	0	0	0	0					



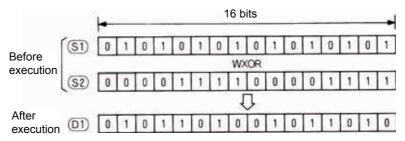
#### **◆**Function

#### WXOR

• Exclusive OR is executed for each bit of the 16-bit data of the device specified with  $\bigcirc$  and 16-bit data of the device specified with  $\bigcirc$  and the results are stored in the device specified with  $\bigcirc$ .



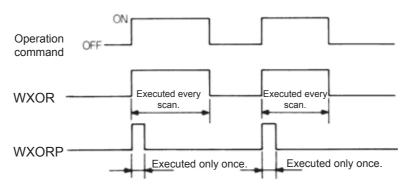
• Exclusive OR is executed for each bit of the 16-bit data of the device specified with (\$\sqrt{9}\$) and 16-bit data of the device specified with (\$\sqrt{9}\$), and the results are stored in the device specified with (\$\sqrt{9}\$).



• Bits of the bit device higher than the specified digit are operated as 0.

#### **♦**Execution condition

The execution conditions for exclusive OR instructions are as follows.



## **♦**Program example

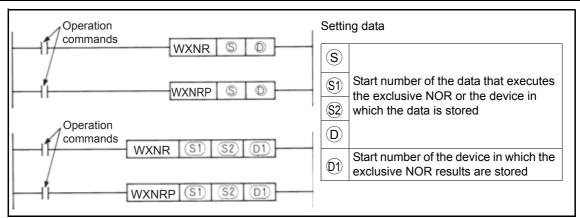
WXOR

• Program that executes exclusive OR for D10 and D20 data when XA turns ON, and stores the results in D10.

• Program that executes exclusive OR for X10 to 1B data and D33 data when XA turns ON, and outputs the results to Y30 to 3B.

# 3.28.5 Logical operation instruction: BIN 16-bit data exclusive NOR ... WXNR, WXNRP

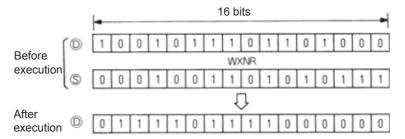
					Арј	olicable	<b></b>	Error flag				
		Bit device			Word (16-bit) device			Constant		Level	Digit specification	Lifornag
		Х	Υ	М	Т	С	D	K	Н	N	<b> </b>	(SM0)
WXNR	S	0	0	0	0	0	0	0	0		K1 to K4	0
	D		0	0	0	0	0					
	<b>§</b> 1)	0	0	0	0	0	0	0	0			
	<u>\$2</u>	0	0	0	0	0	0	0	0			
	<b>(D1)</b>		0	0	0	0	0					



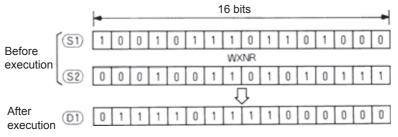
# **♦**Function

#### WXNR

• Exclusive NOR is executed for the 16-bit data of the device specified with  $\bigcirc$  and 16-bit data of the device specified with  $\bigcirc$ , and the results are stored in the device specified with  $\bigcirc$ .



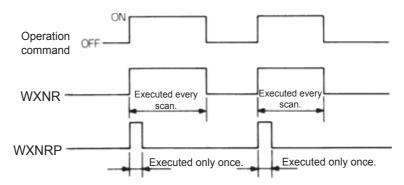
• Exclusive NOR is executed for the 16-bit data of the device specified with (31) and 16-bit data of the device specified with (32), and the results are stored in the device specified with (101).



• Bits of the bit device higher than the specified digit are operated as 0.

#### **◆**Execution condition

The execution conditions for exclusive NOR instructions are as follows.



# **♦**Program example

#### WXNR

• Program that executes exclusive NOR for the 16-bit data of X30 to 3F and D99 data when X0 turns ON, and stores the results in D7.

# **3.29** Application instructions (32-bit)

Application instructions (32-bit) are used when special processing is required.

## 3.29.1 Logical operation instructions

- Logical operation instructions are instructions that execute logical operations such as logical OR or logical AND.
- There are 10 types of logical operation instruction as shown below.

Classification	Instruction symbol
Logical AND	DAND
Logical AND	DANDP
Logical OR	DOR
Logical OK	DORP

Classification	Instruction symbol
Exclusive OR	DXOR
Exclusive OR	DXORP
Exclusive NOR	DXNR
EXCIUSIVE NOR	DXNRP

Classification	Instruction symbol
2's complement	DNEG
(code reversed)	DNEGP

#### NOTE

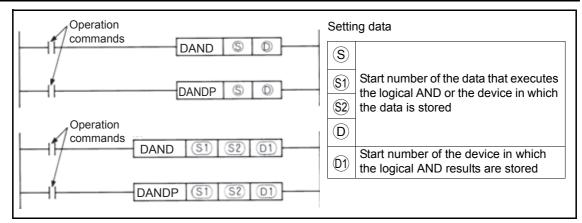
• Logical operation instructions execute the following processes in increments of bits.

Classification	Process	Formula	E	xamp	le
Classification	FTOCESS	Formula	Α	В	Υ
			0	0	0
Logical AND	Set 1 when the inputs of both A and B are 1.	Y = A · B	0	1	0
Logical AND	Set 0 in all other cases.	I - A B	1	0	0
			1	1	1
			0	0	0
Logical OR	Set 0 when the inputs of both A and B are 0.	Y = A + B	0	1	1
Logical OIX	Set 1 in all other cases.	I - A · B	1	0	1
			1	1	1
			0	0	0
Exclusive OR	Set 0 when the inputs of both A and B are equal.	Y = A · B + A · B	0	1	1
LACIUSIVE OIL	Set 1 when they are different.		1	0	1
			1	1	0
			0	0	1
Exclusive NOR	Set 1 when the inputs of both A and B are equal.	$Y = (\overline{A} + B) (A + \overline{B})$	0	1	0
LACIUSIVE NOR	Set 0 when they are different.	1 - (A + B) (A + B)	1	0	0
			1	1	1

#### 3

# 3.29.2 Logical operation instruction: BIN 32-bit logical AND ... DAND, DANDP

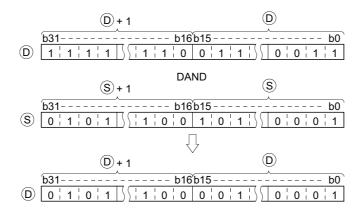
					Арр	olicable	device					
		Bit device			Word (16-bit) device			Con	stant	Level	Digit specification	Error flag
		Х	Υ	М	T	С	D	K	Н	N		(SM0)
	S	0	0	0	0	0	0	0	0			
	(D)		0	0	0	0	0					
DAND (P)	<b>§1</b> )	0	0	0	0	0	0	0	0		K1 to K8	0
	<u>\$2</u>	0	0	0	0	0	0	0	0			
	<b>(D1)</b>		0	0	0	0	0					



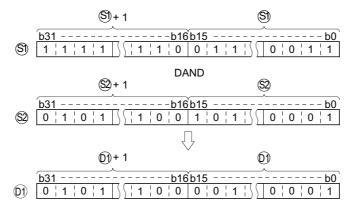
#### **♦**Function

DAND

• Logical AND is executed for each bit of the 32-bit data of the device specified with ① and 32-bit data of the device specified with ③, and the results are stored in the device specified with ①.



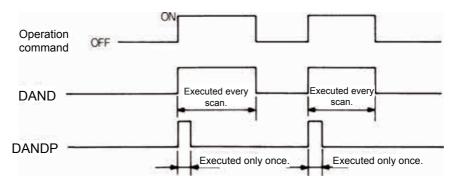
• Logical AND is executed for each bit of the 32-bit data of the device specified with (\$\sigma\$) and 32-bit data of the device specified with (\$\sigma\$), and the results are stored in the device specified with (\$\sigma\$).



• Bits of the bit device higher than the specified digit are operated as 0.

#### **◆**Execution condition

The execution conditions for logical AND instructions are as follows.

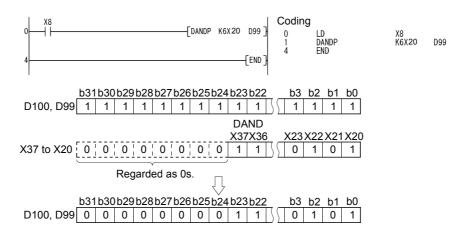


#### 3

#### **♦**Program example

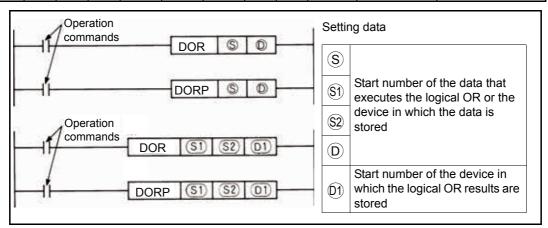
DAND

• The following program performs a logical product operation on the data at D99 and D100, and the 24-bit data between X20 and X37 when X8 is ON, and stores the results at D99 and D100.



# 3.29.3 Logical operation instruction: BIN 32-bit logical OR ... DOR, DORP

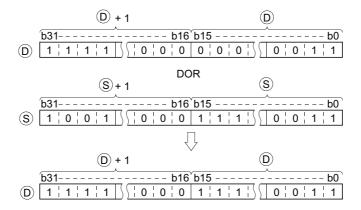
		Applicable device											
		Ві	it devi	се	Word (16-bit) device			Cons	stant	Level	Digit specification	Error flag	
		Х	Υ	M	T	С	D	K	Н	N		(SM0)	
	S	0	0	0	0	0	0	0	0				
	D		0	0	0	0	0						
DOR (P)	<u>\$1</u>	0	0	0	0	0	0	0	0		K1 to K8	0	
	<u>\$2</u>	0	0	0	0	0	0	0	0				
	<b>(D1)</b>		0	0	0	0	0						



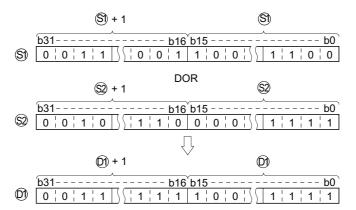
#### **♦**Function

#### DOR

• Logical OR is executed for each bit of the 32-bit data of the device specified with  $\bigcirc$  and 32-bit data of the device specified with  $\bigcirc$ , and the results are stored in the device specified with  $\bigcirc$ .



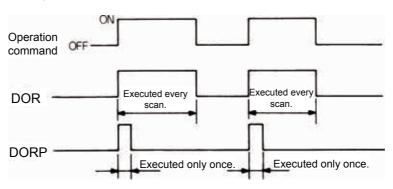
• Logical OR is executed for each bit of the 32-bit data of the device specified with (\$1) and 32-bit data of the device specified with (\$2), and the results are stored in the device specified with (\$1).



• Bits of the bit device higher than the specified digit are operated as 0.

#### **◆**Execution condition

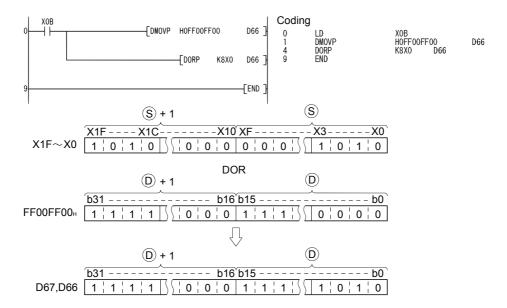
· The execution conditions for logical OR instructions are as follows.



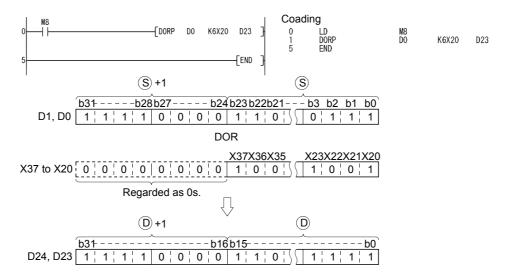
#### **♦**Program example

DOR

• 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 is turned ON, and stores the results at D66 and D67.



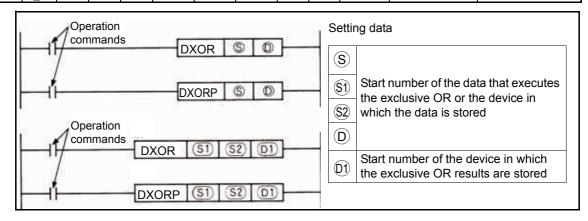
• The following program performs a logical sum operation on the 32-bit data at D0 and D1, and the 24-bit data from X20 to X37, and stores the results at D23 and D24 when M8 is ON.



#### 3

# 3.29.4 Logical operation instruction: BIN 32-bit data exclusive OR ... DXOR, DXORP

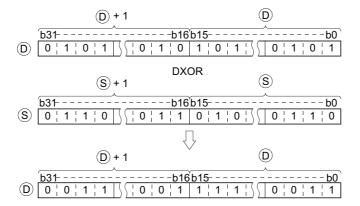
					Арј	olicable	device						
		В	it devi	се	Word (16-bit) device			Constant		Level	Digit specification	Error flag	
		Х	Υ	M	T	С	D	K	Н	N		(SM0)	
	S	0	0	0	0	0	0	0	0				
	(D)		0	0	0	0	0						
DXOR (P)	<b>(\$1)</b>	0	0	0	0	0	0	0	0		K1 to K8	0	
(- /	<u>\$2</u>	0	0	0	0	0	0	0	0				
	(D1)		0	0	0	0	0						



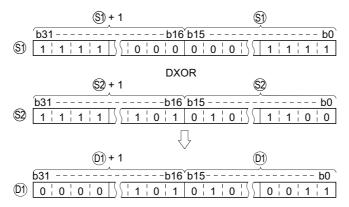
#### **♦**Function

#### DXOR

• Exclusive OR is executed for each bit of the 32-bit data of the device specified with  $\bigcirc$  and 32-bit data of the device specified with  $\bigcirc$  and 32-bit data of the device specified with  $\bigcirc$ .



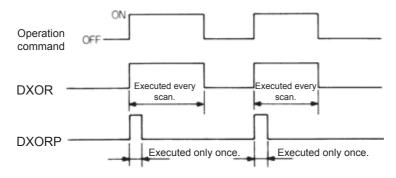
• Exclusive OR is executed for each bit of the 32-bit data of the device specified with (31) and 32-bit data of the device specified with (32), and the results are stored in the device specified with (31).



• Bits of the bit device higher than the specified digit are operated as 0.

#### Execution condition

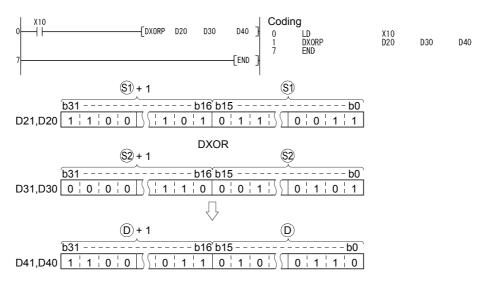
The execution conditions for exclusive OR instructions are as follows.



#### **♦**Program example

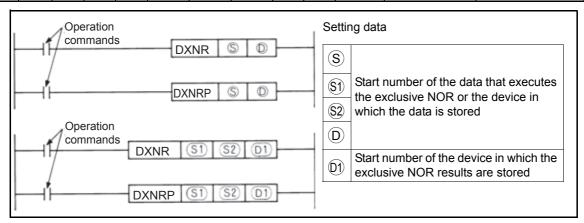
DXOR

• The following program conducts an exclusive OR operation on the data at D20 and D21, and the data at D30 and D31 when X10 is turned ON, and stores the results at D40 and D41.



# 3.29.5 Logical operation instruction: BIN 32-bit data exclusive NOR ... DXNR, DXNRP

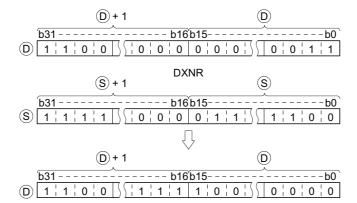
	Applicable device												
		В	it devi	се	Word (16-bit) device			Constant Level			Digit specification	Error flag	
		Х	Υ	М	T	С	D	K	Н	N		(SM0)	
	S	0	0	0	0	0	0	0	0				
	D		0	0	0	0	0						
DXNR (P)	<b>S1</b>	0	0	0	0	0	0	0	0		K1 to K8	0	
	<u>\$2</u>	0	0	0	0	0	0	0	0				
	(D1)		0	0	0	0	0						

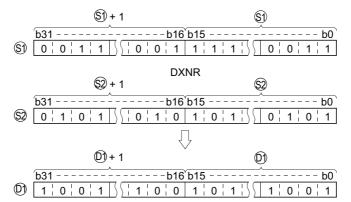


#### **♦**Function

DXNR

• Exclusive NOR is executed for the 32-bit data of the device specified with  $\bigcirc$  and 32-bit data of the device specified with  $\bigcirc$ , and the results are stored in the device specified with  $\bigcirc$ .

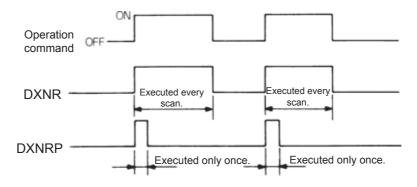




• Bits of the bit device higher than the specified digit are operated as 0.

#### **◆**Execution condition

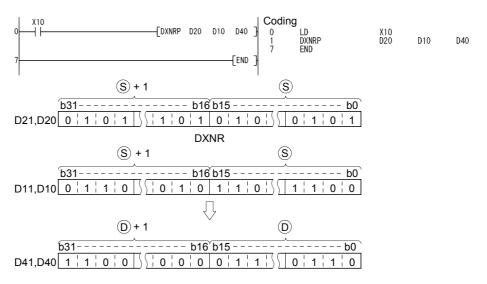
The execution conditions for exclusive NOR instructions are as follows.



#### **♦**Program example

DXNR

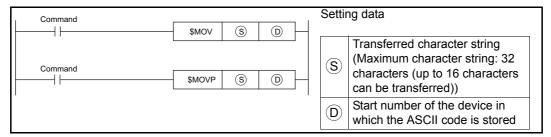
• The following program performs an exclusive NOR operation on the 32-bit data at D20 and D21 and the data at D10 and D11 when X10 is turned ON, and stores the result to D40 and D41.



## 3.30 Display instruction

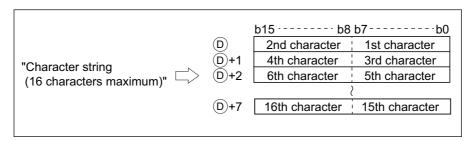
## 3.30.1 Character string data transfer ... \$MOV, \$MOVP

				Арр		Error flag																					
	В	it devi	се	Word	Word (16-bit) device		Constant		Constant		Constant		Constant		Constant		vice Constant		vice Consta		e Constan		evice Consta		Level	Digit specification	Error nag
	Х	Υ	M	T	С	D	K	Н	N		(SM0)																
(D)				0	0	0					0																

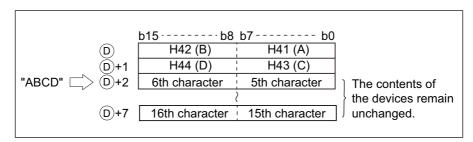


#### **♦**Function

- The character string data enclosed in double-quotation marks (" ") specified with (S) are transferred after the device number specified in (D).
- Only character string data that is enclosed in double-quotation marks (" ") can be set in S. A device can also be input, but doing so results in an error at execution (INSTRCT. CODE ERR: 4000).
- Only word devices can be set in ①. Bit devices can also be input, but doing so results in an error (INSTRCT. CODE ERR: 4000) at execution.
- A maximum of 32 characters can be set in  $\bigcirc$  . (Up to 16 characters can be transmitted from  $\bigcirc$  to  $\bigcirc$  . The 17th character or later are invalid.)



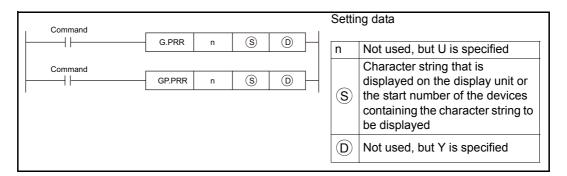
• If the set number of characters is less than 16, the content of  $\bigcirc$  in the sections without corresponding characters remains unchanged.



## 3.30.2 Character string output ... G. PRR, GP.PRR, UMSG

#### G.PRR, GP.PRR

				App	olicable	device					Error flag	
	Bi	it devi	се	Word	(16-bit)	device	Con	stant	Level	Others U	Digit specification	Error nag
	Х	Υ	M	Т	С	D	K	Н	N		opcomouno	(SM0)
n										0		
S				0	0	0						0
(D)	0	0	0			0						



#### UMSG

				App	olicable	D: '	Error flag							
	Bi	it devi	се	Word	(16-bit) device		evice Constant			device Cons		Level	Digit specification	Lifoi ilag
	X	Υ	M	T	С	D	K	Η	N	•	(SM0)			
S				0	0	0					0			

Command		Settir	ng data
	UMSG	<u>s</u>	Character string that is displayed on the display unit or the start number of the devices containing the character string to be displayed

#### **♦**Function

G.PRR, GP.PRR

n is not used, but specify U.

Register the data to be displayed on the parameter unit (FR-PU07) in the device specified with (S).

D is not used, but specify Y.

UMSG

Register the data to be displayed on the parameter unit (FR-PU07) in the device specified with S.

#### Setting data: For user monitor name

To monitor SD1216 to 1218 using the parameter unit (FR-PU07), set the monitor name as shown below.

	Upper 8 bits	Lower 8 bits	
S)+0	****	01	←Set the corresponding monitor number to the upper 8 bits, and 01 to the lower 8 bits.
	2nd character	1st character	←Monitor name: Character start
	4th character	3rd character	
	6th character	5th character	
	8th character	7th character	
	10th character	9th character	
	12th character	11th character	
	14th character	13th character	
S+8	16th character	15th character	
		_	

- Set 01 to the lower 8 bits of (S)+0.
- Set the monitor description number to replace the user monitor name display in the upper 8 bits of S+0. The monitor description number is as shown below.

Monitor description number	Monitor description
40	User monitor 1: SD1216 description
41	User monitor 2: SD1217 description
42	User monitor 3: SD1218 description

- When the upper 8 bits of S+0 are values other than those in the table above, an operation error (OPERATION ERROR: 4100) is generated.
- When the 7th bit of S+0 is executed as 1 (lower 8 bits as H81), the registered data is cleared and the monitor display returns to normal.
- Settings can be made up to the 16th character. (Data from S)+1 to S)+8 are effective as characters.)

#### NOTE

• To display the monitor above on the parameter unit (FR-PU07), set "40, 41, 42" in **Pr.774 to Pr.776** in advance. (For details on **Pr.774 to Pr.776**, refer to **the Instruction Manual (Detailed)** of the inverter.)

#### Setting data: For user-defined error name

To display the user-defined error names on the parameter unit (FR-PU07), set those names as shown below. (A user defined error occurs when values 16 to 20 are set in SD1214.)

	Upper 8 bits	Lower 8 bits	
S)+0	***	02	$\leftarrow$ Set the corresponding error number to the upper 8 bits, and 02 to the lower 8 bits
	2nd character	1st character	←Error name: Character start
	4th character	3rd character	
	6th character	5th character	
	8th character	7th character	
	10th character	9th character	
	12th character	11th character	
	14th character	13th character	
S)+8	16th character	15th character	

- Set an error number that corresponds to the upper 8 bits of (S)+0. The setting range is from 16 to 20.
- When the upper 8 bits of S+0 are values other than 16 to 20, an operation error (OPERATION ERROR: 4100) is generated.
- When the 7th bit of S+0 is executed as 1 (lower 8 bits as H82), the registered data is cleared.
- Settings can be made up to the 16th character. (Data from (S)+1 to (S)+8 are effective as characters.)

#### Setting data: For unit names

This function sets up the increment for the user monitors SD1216 to SD1218 using the parameter unit (FR-PU07).

	Upper 8 bits	Lower 8 bits	_
<b>S</b> +0	***	04	•
	2nd character	1st character	4
S+2		3rd character	

- ← Set the corresponding user monitor number to the upper 8 bits, and 04 to the lower 8 bits.
- ← Unit name: Character start
- Set 04 to the lower 8 bits of S+0.
- Set the monitor numbers to replace the unit displayed in the upper 8 bits of S+0. Numbers that can be specified are shown in the following table.

Number of unit name	Item to add unit
40	User monitor 1: Unit corresponding to the description of SD1216
41	User monitor 2: Unit corresponding to the description of SD1217
42	User monitor 3: Unit corresponding to the description of SD1218

- When the upper 8 bits of S+0 are values other than those in the table above, an operation error (OPERATION ERROR: 4100) is generated.
- When the 7th bit of S+0 is executed as 1 (lower 8 bits as H84), the registered data is cleared.
- Settings can be made up to the 3rd character. (S)+1 to S)+2: Data up to the lower 8 bits are effective as characters.)

#### Quantity that can be set

Setting data	Quantity
User monitor (Refer to page 194)	Up to 3*1
User definition error (Refer to page 194)	Up to 5*1
Unit name (Refer to page 195)	Up to 3*1

\*1 If any of the setting quantities is exceeded, an operation error (OPERATION ERROR: 4100) occurs at the point when the exceeded setting is executed. Furthermore, an operation error (OPERATION ERROR: 4100) also occurs if the lower 8 bits of S+0 are set to values other than 01 to 04, or H81 to H84.

#### • NOTE

- An error does not occur even if the stored data exceeds the range of the corresponding device, but the content of devices
  outside of the range is changed. When storing data, check beforehand that the amount of devices required for storage can be
  allocated.
- Do not change any of the device data while it is being displayed because the data stored in the device is used in actual communication. Changing such data will change the transmitted data.
- If the ASCII data is other than 0×20 to 0×7A that can be displayed in the FR-PU07, it can be replaced with 0×20 (space).
- [ ^ ] (H5E), [ \_ ] (H5F) and [ ' ] (H60) cannot be displayed by the FR-PU07.
- If the ASCII data is other than 0×20 to 0×7A that can be displayed in PU, it will be replaced with 0×20 (space).

# **MEMO**

# 4

# **Error code list**

4.1	How to read error codes	19	36
<b></b> .			_

#### How to read error codes

If an error occurs when PLC function is RUN or during RUN, an error indication is generated by the self-diagnostic function, and an error code and error step are stored in the special register. The error description and corrective action are as follows:

## 4.1 How to read error codes

When an error occurs, the error code can be read using a peripheral device.

For details on the operation method, refer to the operating manual of the peripheral device.

The error code, error name, error description possible cause, and corrective actions are as follows.

Error codes are stored in SD0, and the error steps are stored in SD4 to SD26.

Error code (SD0)	Error name	Error description possible cause	Corrective action
1010	END NOT EXECUTE	All programs in the program capacity are executed without executing the END instruction.  • After the END instruction was executed, a different instruction code was read due to noise, etc.  • The END instruction has changed to a different instruction code for some reason.	Take measures against noise. Perform an inverter reset and go to the RUN state again. If the same error is displayed again, a CPU hardware error has occurred. Please contact your sales representative and explain the failure symptom.
2200	MISSING PARA.	There are no parameter files.	Set a parameter file.
2502	CAN'T EXE. PRG.	The program file is not correct, or the content of the file is not a program.	Check that the program file type is ***.QPG, and check that the file content is a program.
3000	PARAMETER ERROR	The content of the parameter indicated by the error individual information (SD16) is incorrect.	Read the error individual information with a programming tool, check the parameter items that correspond to the values (parameter number) and correct them. Re-write the corrected parameters, reset the power or reset the inverter. If the same error is displayed again, a CPU hardware error has occurred. Please contact your sales representative and explain the failure symptom.
3003	PARAMETER ERROR	The number of device points set in the PLC parameter device setting is not within the usable range.	Read the error individual information with a programming tool, check the parameter items that correspond to the values (parameter number) and correct them. If the error occurs again after correcting the parameter, a memory error has occurred in the program memory. Please contact your sales representative and explain the failure symptom.
4000	INSTRCT. CODE ERR	An undecodable instruction code is included in the program.     An unusable instruction is included in the program.	Read the common error information with a programming tool, check the error step that corresponds to the value (program error location) and correct it.
4010	MISSING END INS.	No END (FEND) instruction in the program.	Read the common error information with a programming tool, check the error step that corresponds to the value (program error location) and correct it.
4100	OPERATION ERROR	Data that cannot be used in instructions is included.	Read the common error information with a programming tool, check the error step that corresponds to the value (program error location) and correct it.

Error code (SD0)	Error name	Error description possible cause	Corrective action
5001	WDT ERROR	The program scan time exceeded the watchdog timer value set in the PLC RAS setting of the PLC function parameter.	<ul> <li>Read the error individual information with the programming tool, check the values (time) and shorten the scan time.</li> <li>Change the initial execution monitoring time or watchdog setting values with the PLC RAS settings of the PLC function parameter.</li> <li>Delete the infinite loop from the jump transition.</li> <li>Check the number of executions of the interrupt program with a programming tool, and reduce the number of interrupt occurrences.</li> </ul>
5010	PRG. TIME OVER	The program scan time exceeded the constant scan time set in the PLC RAS setting of the PLC function parameter.	Review the constant scan time setting. Review the PLC parameter constant scan time and low-speed program execution time in order to secure sufficient remaining time for the constant scan.

# **MEMO**

# **APPENDIX**

Annendix 1	Instruction	nrocessino	ı time	20	12
Threllary I	III SU UCUOII	piocessiiid	UIIIC		, _

# **Appendix 1** Instruction processing time

#### **♦**Basic instructions

#### • Sequence instructions

Classification	Instruction	Condition (device)	Processing time (µs)
	LD		1.9
	LDI		1.9
Contact	AND		1.9
Contact	ANI		1.9
	OR		2.0
	ORI		2.0
	ORB		1.3
	ANB		1.3
Link	MPS		1.4
	MRD		1.4
	MPP		1.4
	OUT	Y,M,SM	2.4
	OUT	Т	7.6
	OUT	С	7.9
	SET	Y,M,SM	2.6
Output	RST	Y,M,SM	2.6
Output	RST	Т	9.2
	RST	С	10.0
	RST	D	4.3
	PLS		3.4
	PLF		3.4
Shift	SFT		3.4
Siliit	SFTP		4.2
Master control	MC		3.8
iviasiei control	MCR		1.3
Program end	END,FEND		0.8
Non- processing	NOP		1.3

#### Comparison operation instructions

		Condition	Processing time (µs)		
Classification	Instruction	(device)	Subset *1	Other than subset	
	LD=	S1 S2	8.0	9.9	
	LD<>	S1 S2	8.0	9.9	
	LD<	S1 S2	8.0	9.9	
	LD>	S1 S2	8.0	9.9	
	LD<=	S1 S2	8.1	10.0	
	LD>=	S1 S2	8.1	10.0	
	AND=	S1 S2	8.2	9.6	
BIN	AND<>	S1 S2	8.2	9.7	
16-bit	AND<	S1 S2	8.4	9.8	
data	AND>	S1 S2	8.3	9.7	
comparison	AND<=	S1 S2	8.3	9.7	
	AND>=	S1 S2	8.3	9.7	
	OR=	S1 S2	8.7	9.8	
	OR<>	S1 S2	8.6	9.8	
	OR<	S1 S2	8.6	9.8	
	OR>	S1 S2	8.6	9.8	
	OR<=	S1 S2	8.8	9.9	
	OR>=	S1 S2	8.8	9.9	
	LDD=	S1 S2	8.8	11.1	
	LDD<>	S1 S2	8.8	11.1	
	LDD<	S1 S2	8.8	11.1	
	LDD>	S1 S2	8.8	11.1	
	LDD<=	S1 S2	8.8	11.1	
	LDD>=	S1 S2	8.8	11.1	
	ANDD=	S1 S2	9.2	11.0	
BIN	ANDD<>	S1 S2	9.2	11.0	
32-bit	ANDD<	S1 S2	9.2	11.0	
data	ANDD>	S1 S2	9.2	11.0	
comparison	ANDD<=	S1 S2	9.2	11.0	
	ANDD>=	S1 S2	9.2	11.0	
	ORD=	S1 S2	9.4	9.9	
	ORD<>	S1 S2	9.4	9.9	
	ORD<	S1 S2	9.4	9.9	
	ORD>	S1 S2	9.4	9.9	
	ORD<=	S1 S2	9.4	9.9	
	ORD>=	S1 S2	9.4	9.9	

#### Arithmetic operation instructions

	Condition	Processing time (μs)		
Classification	Instruction	(device)	Subset *1	Other than subset
	+	SD	8.4	13.5
	+P	SD	11.0	14.3
BIN	-	SD	8.4	13.5
16-bit	-P	SD	11.0	14.3
addition/	+	S1S2 D	8.4	13.6
subtraction	+P	S1S2 D	11.0	14.5
	-	S1S2 D	8.4	13.6
	-P	S1S2 D	11.0	14.5
	D+	SD	9.6	15.6
	D+P	SD	12.0	16.4
BIN	D-	SD	9.6	15.6
32-bit	D-P	SD	12.0	16.4
addition/	D+	S1S2 D	9.6	15.7
subtraction	D+P	S1S2 D	12.0	16.5
	D-	S1S2 D	9.5	15.7
	D-P	S1S2 D	12.0	16.5
BIN	*	S1S2 D	8.7	13.9
16-bit	*P	S1S2 D	11.2	14.7
multiplication	1	S1S2 D	8.9	14.2
/division	/P	S1S2 D	11.5	15.0
BIN	D*	S1S2 D	9.7	15.7
32-bit	D*P	S1S2 D	12.2	16.5
multiplication	D/	S1S2 D	10.1	15.9
/division	D/P	S1S2 D	12.6	16.7
	INC	D	5.8	11.1
Data increment/ decrement	INCP	D	7.6	11.8
	DEC	D	5.8	11.1
	DECP	D	7.6	11.9
	DINC	D	6.5	12.6
	DINCP	D	8.2	13.4
	DDEC	D	6.5	12.6
	DDECP	D	8.2	13.4

#### Data transfer instructions

Classification	Instruction	Condition (device)	Processing time (µs)	
			Subset	Other than subset
16-bit transfer	MOV		5.8	13.5
	MOVP		7.6	14.2
32-bit transfer	DMOV		6.4	15.6
	DMOVP		8.3	16.4

#### ◆ Data conversion instructions

Classification	Instruction	Condition (device)	Processing time (µs)	
			Subset	Other than subset
	NEG	D	8.8	11.1
2's complement	NEGP	D	9.5	11.8
	DNEG	D	8.8	12.6
	DNEGP	D	9.6	13.4
BCD conversion	BCD	SD	5.8	13.7
	BCDP	SD	8.3	14.5
	DBCD	SD	7.8	16.9
	DBCDP	SD	9.6	17.7
BIN conversion	BIN	SD	5.9	13.7
	BINP	SD	7.6	14.6
	DBIN	SD	6.8	16.1
	DBINP	SD	8.5	16.9

- \*1 When all the devices used in the instruction satisfy one of the following conditions, they are regarded as a subset.
  - All the devices used are word devices.
  - When the devices used are bit devices, the specified number of bits is the multiple of 16, or the digits are specified as K4 (word data) or K8 (double word data).
  - The devices used are constants.

#### NOTE

 During inverter control, reading 1000 steps takes a scan time of about 40 ms.

#### **♦**Application instructions

#### ◆ Logical operation instructions

	Instruction (device)	Processing time (μs)		
Classification			Subset *1	Other than subset
	WOR	SD	8.6	13.6
	WORP	SD	11.2	14.4
	WOR	S1 S2 D	8.6	13.6
Logical OR	WORP	S1 S2 D	11.2	14.4
Logical OK	DOR	SD	9.7	15.7
	DORP	SD	12.2	16.5
	DOR	S1 S2 D	9.7	15.7
	DORP	S1 S2 D	12.2	16.5
	WAND	SD	8.6	13.6
	WANDP	SD	11.2	14.4
	WAND	S1 S2 D	8.6	13.6
Logical	WANDP	S1 S2 D	11.2	14.4
AND	DAND	SD	9.7	15.7
	DANDP	SD	12.2	16.5
	DAND	S1 S2 D	9.7	15.7
	DANDP	S1 S2 D	12.2	16.5
	WXOR	SD	8.6	13.6
	WXORP	SD	11.2	14.4
	WXOR	S1 S2 D	8.6	13.6
Exclusive	WXORP	S1 S2 D	11.2	14.4
OR	DXOR	SD	9.7	15.7
	DXORP	SD	12.2	16.5
	DXOR	S1 S2 D	9.7	15.7
	DXORP	S1 S2 D	12.2	16.5
	WXNR	SD	8.6	13.6
Exclusive NOR	WXNRP	SD	11.2	14.4
	WXNR	S1 S2 D	8.6	13.6
	WXNRP	S1 S2 D	11.2	14.4
	DXNR	SD	9.7	15.7
	DXNRP	SD	12.2	16.5
	DXNR	S1 S2 D	9.7	15.7
	DXNRP	S1 S2 D	12.2	16.5

#### Instructions regarding the message display on PU

Classification	Instruction	Condition (device)	Processing time (µs)
Character string data transfer	\$MOV		12.6
	\$MOVP		13.3
	G.PRR		8.7
Character string output	GP.PRR		9.6
	UMSG		5.3

- \*1 When all the devices used in the instruction satisfy one of the following conditions, they are regarded as a subset.
  - · All the devices used are word devices.
  - When the devices used are bit devices, the specified number of bits is the multiple of 16, or the digits are specified as K4 (word data) or K8 (double word data).
  - The devices used are constants.



 During inverter control, reading 1000 steps takes a scan time of about 40 ms.

# **MEMO**

#### **REVISIONS**

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

Print date	*Manual number	Revision
Aug. 2013	IB(NA)-0600492ENG-A	First edition
	<u> </u>	

